IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 April 2025, accepted 5 June 2025, date of publication 17 June 2025, date of current version 27 June 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3580588

RESEARCH ARTICLE

Long Short-Term Memory Neural Network Combined With a Hybrid-Modular Clockwork Structure for Transient Modeling of Nonlinear Circuits

RAZIEH MORADI CHALESHTORI[1], AMIN SABOOHI[1], AMIN FARAJI[2], SAYED ALIREZA SADROSSADAT[3], (Senior Member, IEEE), ALI MOFTAKHARZADEH[4], AND YVON SAVARIA[5], (Life Fellow, IEEE)

[1]Department of Electrical Engineering, Isfahan University, Isfahan 8174673441, Iran
[2]Department of Computer Engineering, Shahid Beheshti University, Tehran 1983969411, Iran
[3]Department of Computer Engineering, Yazd University, Yazd 8915818411, Iran
[4]Department of Electrical Engineering, Yazd University, Yazd 8915818411, Iran
[5]Department of Electrical Engineering, Polytechnique Montreal, Montreal, QC H3T lJ4, Canada

Corresponding author: Sayed Alireza Sadrossadat (alireza.sadr@yazd.ac.ir)

ABSTRACT This paper presents a novel macromodeling method and neural network structure called Clockwork Long Short-Term Memory (CWLSTM), tailored for high-speed nonlinear circuits. The proposed CWLSTM method is considered a more powerful yet simpler model than conventional LSTM due to its reduced parameter count and more efficient structure and training strategy. This structure promotes improved model generalization, resulting in better model accuracy and training time due to its unique modular gating connections. Additionally, the required training data is considerably reduced for generating a model with similar accuracy compared to the conventional LSTM. To further improve the proposed method, a hybrid version of CWLSTM, known as Hybrid-Modular CWLSTM, is introduced, utilizing various module types to enhance the model's accuracy further. The reported experimental results reveal the superior performance of the proposed methods compared to the conventional LSTM in modeling high-speed nonlinear circuits. On top of the above advantages, the proposed methods can produce models that execute much faster than those based on existing simulation tools (LTspice and NGspice). The performance of the proposed methods is validated by modeling two high-speed nonlinear circuits.

INDEX TERMS Computer-aided design (CAD), clockwork LSTM (CWLSTM), hybrid structure, macromodeling, nonlinear component, recurrent neural network (RNN).

I. INTRODUCTION

In the rapidly evolving field of circuits and systems, there has been a significant interest in research dedicated to high-accuracy macromodeling [1], [2], [3], [4], [5], [6], [7], [8]. A macromodel is a high-level behavioral abstraction of a complex circuit. It retains necessary information for calculating some desired output variables, while other details can be suppressed. It can be viewed as a reduced-order representation that preserves critical system-level properties (like stability, nonlinearity, frequency response) with minimal computational cost and evaluation time. A macromodel should have similar accuracy to the original circuit but with much faster simulation time [1]. Moreover, as new technologies emerge, ensuring adequate accuracy in current models based on equivalent circuits has become increasingly challenging. Consequently, there is a pressing need to develop enhanced models. However, extending a new equivalent circuit model is time-consuming due to the manual trial-and-error processes it involves.

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

One powerful tool in computer-aided design (CAD) is artificial neural networks (ANN), which has been employed successfully for modeling and designing components and circuits [9], [10], [11], [12], [13], [14]. ANNs can also efficiently be utilized in design optimization, enhancing quality [15].

ANNs are highly effective in modeling statistical design of active and passive devices, as well as circuit optimization. They represent a robust method for mapping any nonlinear relationship between given inputs and their corresponding outputs [13], [14], [15], [16].

Two main models are commonly used in nonlinear electronic circuits and component development: transistor-level and behavioral models. Transistor-level models, such as those used in SPICE simulations, provide more complex and accurate representations with longer computational times. By contrast, behavioral models focus on the input-output signal relationship without hunting through internal circuit details [17].

The purpose of nonlinear macromodeling is to represent accurately and with high computational efficiency the input-output relations of a nonlinear circuit. With the added complication of nonlinear circuits, there is a growing demand for less complex models that accurately capture the dynamic behaviors of these systems and components. Recurrent neural networks (RNNs) are presented as one of the solutions for nonlinear macromodeling [8]. Employing dynamic neural network structures such as RNNs enables the modeling of nonlinear circuits in the time domain. RNNs can learn straight from simulated or measured input and output data, bypassing the need for internal circuit details. Moreover, RNNs are acknowledged as global approximators, affirming their ability to characterize relations between any nonlinear input-output [18].

In [19], a hybrid approach was introduced, merging RNN and a polynomial regression method (HRPR) for nonlinear applications. This method reduces training time and provides a speed-up in simulation compared to conventional RNNs, all while maintaining accuracy.

In [20], a macromodeling approach called batch-normalized RNN (BN-RNN) was utilized for modeling high-speed nonlinear circuits. This approach significantly reduces training times and generates models with better precision than conventional RNNs.

In [21], an advanced method for modeling high-dimensional microwave circuits is introduced, integrating a refined version of the Rectified Linear Unit (ReLU) into deep neural networks. In this method, sigmoid and refined ReLU are utilized as activation functions. The deep neural network model is trained using a three-phase deep learning algorithm. This algorithm helps identify the appropriate number of hidden layers during training, effectively addressing the vanishing gradient issue in the training phase.

In [22], a deep neural network is presented that automatically obtains the optimal number of hidden layers and is suitable for parametric modeling of passive components. A three-stage training algorithm is proposed to attain model accuracy comparable to shallow neural networks using fewer training parameters.

In [17], for macromodeling nonlinear electronic circuits, a neural network-based method called Long Short-Term Memory (LSTM) is presented. Unlike conventional networks like RNNs, which encounter the issue of vanishing gradients during training, LSTM networks are specifically designed to address this issue effectively.

Traditional RNNs are defined by simple hidden states that use nonlinear activation functions and weighted summation of feedforward and recurrent data [23], [24], [25]. This inherent design causes traditional RNNs to transfer data from the current time step to the next directly, making them more susceptible to the vanishing gradient problem. In contrast, LSTM networks incorporate gates within the structure, effectively addressing the challenge of vanishing gradients by integrating a memory cell capable of retaining state information over extended periods. However, the inclusion of three gates along with a memory cell causes a considerable number of parameters for LSTMs, leading to increased evaluation times in macromodeling and a higher risk of model overfitting. Also, despite their capability to address the vanishing gradient problem, LSTMs still face challenges in generating accurate models that can be represented using very large time steps.

In [26], a Clockwork-RNN (CWRNN) was introduced for nonlinear circuits macromodeling. In this study, a modular approach was used on the recurrent connections to create non-fully connected recurrent connections, resulting in a more efficient structure and training strategy with fewer parameters.

In this paper, we propose new macromodeling methods called Clockwork-LSTM (CWLSTM) and its hybrid version called Hybrid-Modular CWLSTM (HM-CWLSTM) to address the overfitting issue associated with LSTM. As LSTM suffers from using large number of parameters, this approach reduces the LSTM parameter counts. That reduced number offers a more efficient implementation and training of the conventional LSTM architecture, effectively diminishing model complexity. In CWLSTM, the embedded modules functionalities are similar to the hidden layers of an LSTM. However, these modules operate at distinct clock periods and let the model learn diverse frequencies in the training signals. Moreover, the understanding and implementation of this method are straightforward, featuring a flexible architecture configured by modules with varying clock rates based on exponents of 2. Implementing this method improves model accuracy while reducing the training data needed for accurate modeling and also the simulation time. To further enhance the model accuracy, the HM-CWLSTM is proposed. This improved method employs various prime numbers and their exponents to represent module clock rates more effectively while decreasing the model test error. Consequently, CWLSTM and HM-CWLSTM are promising for modeling high-speed nonlinear circuits. The rest of the paper is

organized as follows. Section II describes a review of conventional LSTM. In Section III, the proposed CWLSTM and HM-CWLSTM macromodeling approaches are explained, their structure and training are discussed, and a comparison is made with traditional LSTM. Section IV presents the numerical results, including three nonlinear examples for validating the methods proposed to derive macromodels of nonlinear circuits. Finally, Section V presents the conclusion drawn from this research.

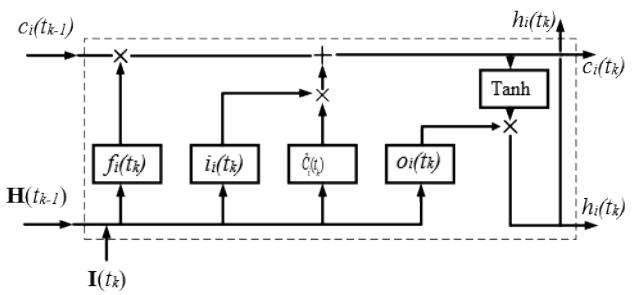

FIGURE 1. An LSTM cell.

II. BACKGROUND

A. CONVENTIONAL LONG-SHORT TERM MEMORY(LSTM)

Before introducing the presented method, an investigation of the structure of the conventional LSTM is required to clearly compare the traditional LSTM and the presented CWLSTM modeling methods. The structure of the traditional Long-Short Term Memory (LSTM) network is discussed in this section. In [17], for the first time, the LSTM network was utilized to build macromodels for nonlinear circuits. The structure of an LSTM cell at time step k, denoted as t_k, is illustrated in Fig. 1. In this figure, $\mathbf{H}(t_{k-1}) = [h_1(t_{k-1}) \ldots h_{N_h}(t_{k-1})]$ where $h_i(t_{k-1})$ is hidden state of the ith cell at time step $k-1$ and N_h represents the number of LSTM cells. Also, $\mathbf{I}(t_k) = [I_1(t_k) \ldots I_{N_I}(t_k)]$ where N_I denotes the number of inputs of the network and $I_i(t_k)$ is the ith input of the network at time step k. This network benefits from a gating structure that enables it to remember the dependency of the signal values over time. As depicted in Fig. 1, an LSTM cell comprises a hidden state ($h_i(t_k)$) and a cell state ($c_i(t_k)$), alongside other gates. The former state preserves the short-term, while the latter retains long-term information. The states and gates of the LSTM cells are detailed as follows.

1) THE INPUT GATE

This gate determines how the information from the current input should be passed to the next time step. Equation (1) presents the formulation of the input gate for the ith cell of the LSTM network at time step k.

$$i_i(t_k) = \sigma\Big(\sum_{s=1}^{N_I} w_{i,s}^{in} I_s(t_k) + \sum_{x=1}^{N_h} u_{i,x}^{in} h_x(t_{k-1})\Big) \tag{1}$$

In (1), σ represents the sigmoid activation function, $I_s(t_k)$, $w_{i,s}^{in}$, $h_x(t_{k-1})$, and $u_{i,x}^{in}$ are the sth input of the network at time step k, the weight connecting the sth input to the input gate of the ith cell, the hidden state of the xth cell at time step $k-1$, and the recurrent weight connecting the hidden state of the xth cell at time step $k-1$ to the ith input gate at time step k, respectively.

2) THE FORGET GATE

The forget gate determines which information should be retained or discarded. The formulation of this gate in cell i is provided in (2):

$$f_i(t_k) = \sigma\Big(\sum_{s=1}^{N_I} w_{i,s}^{f} I_s(t_k) + \sum_{x=1}^{N_h} u_{i,x}^{f} h_x(t_{k-1})\Big) \tag{2}$$

where, $w_{i,s}^{f}$ represents the weight connecting the sth input to the forget gate of the ith cell, while $u_{i,x}^{f}$ denotes the weight connecting the hidden state of the xth cell at time step $k-1$ to the forget gate of the ith cell at time step k.

3) THE OUTPUT GATE

The output gate's value plays a major role in determining the hidden state. The formulation of this gate for cell i at time step k is as below:

$$o_i(t_k) = \sigma\Big(\sum_{s=1}^{N_I} w_{i,s}^{o} I_s(t_k) + \sum_{x=1}^{N_h} u_{i,x}^{o} h_x(t_{k-1})\Big) \tag{3}$$

In (3), $w_{i,s}^{o}$ represents the weight connecting the sth input to the output gate of the ith cell, while $u_{i,x}^{o}$ denotes the weight connecting the hidden state of the xth cell at time step $k-1$ to the output gate of the ith cell at time step k.

4) THE NEW MEMORY CELL

The new memory cell is another part of the LSTM cell that specifies how the long-term dependency is affected by the current input. The formulation for this module in cell i at time step k is provided in (4):

$$\hat{c}_i(t_k) = \tanh\Big(\sum_{s=1}^{N_I} w_{i,s}^{c} I_s(t_k) + \sum_{x=1}^{N_h} u_{i,x}^{c} h_x(t_{k-1})\Big) \tag{4}$$

where $w_{i,s}^{c}$ represents the weight connecting the sth input to the new memory cell of the ith cell, and $u_{i,x}^{c}$ demonstrates the weight connecting the hidden state of the xth cell at time step $k-1$ to the new memory cell of the ith cell at time step k.

5) THE MEMORY CELL STATE

By utilizing the input, forget, and output gates alongside the new memory cell, the memory cell state acquires a new value to retain long-term information that is also passed on to the next time step. The formulation of this process for cell i at time step k is shown below:

$$c_i(t_k) = f_i(t_k) \times c_i(t_{k-1}) + i_i(t_k) \times \hat{c}_i(t_k) \tag{5}$$

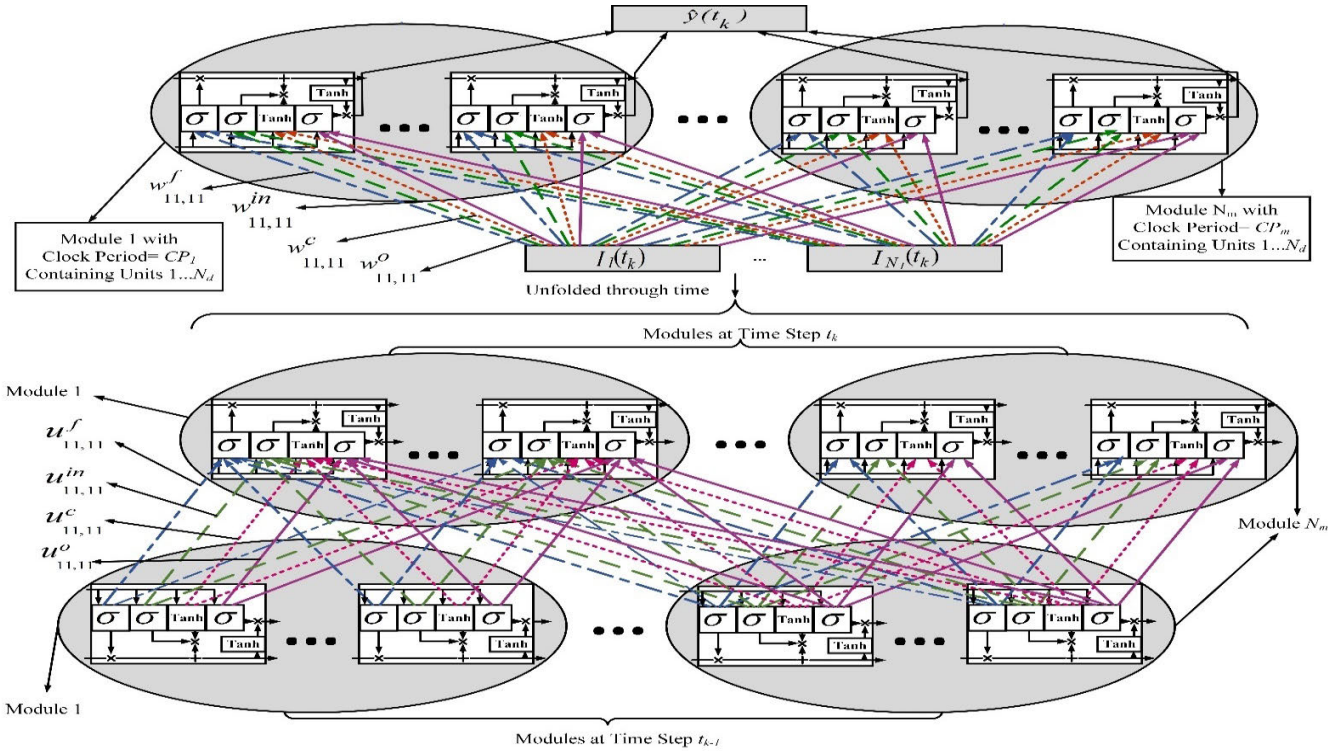

FIGURE 2. Structure of the proposed CWLSTM.

6) THE HIDDEN STATE

The hidden state incorporates the values generated by the output gate and the memory cell state. This aspect of the LSTM cell retains short-term information over time. The hidden state is transmitted both as the output of the LSTM cells to the output layer and as the recurrent state through time. The formulation of the hidden state for cell i at time step k is presented below:

$$h_i(t_k) = o_i(t_k) \times \tanh(c_i(t_k)) \tag{6}$$

The equations outlined in (1)-(6) are used to generate the hidden states for the cells. The weighted summation of these hidden states produces the LSTM output as (7) (assuming a single cell layer for simplicity):

$$\hat{y}(t_k) = \sum_{i=1}^{N_h} w_i^{out} h_i(t_k) \tag{7}$$

where $\hat{y}(t_k)$ represents the output of the LSTM at time step k, and w_i^{out} denotes the weight connecting the ith hidden state to the output layer.

III. THE PROPOSED LONG-SHORT TERM MEMORY MACROMODELING METHOD

In this section, we explore the detailed structure and formulations of the proposed macromodeling method, highlighting its differences from the traditional LSTM discussed in Section II.

A. STRUCTURE OF THE CLOCKWORK LONG-SHORT TERM MEMORY (CWLSTM)

The proposed Clockwork Long-Short Term Memory (CWLSTM) macromodeling method combines the gating advantages of the LSTM macromodeling method [17] with the clockwork approach [26], [27], [28].

Fig. 2 illustrates a schematic of the structure of the proposed CWLSTM macromodeling method. This structure contains input, hidden, and output layers. The feedforward connections in this structure are identical to those in conventional LSTM used for macromodeling circuits [17], [23], [29], meaning they are fully connected across layers. However, an essential distinction lies in the recurrent connections, setting apart the proposed method from conventional macromodeling methods used for nonlinear circuits. As illustrated in Fig. 2, the recurrent connections in the proposed CWLSTM follow a modular approach, avoiding fully connected recurrent connections throughout the network. The idea for this modular approach is to partition the LSTM cells into multiple modules, each containing an equal number of units. The partial connectivity of recurrent connections is structured such that units within each module at the current time step receive recurrent information solely from specific modules at the previous time step rather than from all modules. This process is governed by a criterion known as the *clock period*. If we consider m modules, each with an equal number of units labeled modules $1, 2, \ldots, m$, and associate each module with its respective clock period ($CP_1, CP_2, \ldots, CP_m$), where

$CP_i = 2^{i-1}$, then these clock periods shape the recurrent connections. Specifically, the units within module a at time step $k-1$ are linked to the units within module b at time step k only if $CP_a \geq CP_b$. It is important to note that in CWLSTM, if two modules are connected, the connections between their units are fully established.

In CWLSTM, when establishing recurrent connections, they are organized from left to right based on clock periods. This means the module with CP_1 is positioned on the left side, while the module with CP_m is on the right. Furthermore, modules with lower clock periods are denoted as fast modules, while those with higher ones are called slow modules. For instance, the 2nd module with $CP_2 = 2$ is considered a faster module in comparison to the 5th module with $CP_5 = 16$.

As depicted in Fig. 2, the recurrent connections in CWLSTM are designed so that each module at time step k receives recurrent information from time step $k-1$ solely from modules that are slower than itself. This results in faster modules having more complex recurrent connections than slower ones. Considering that high-frequency content in a signal carries more information, there should be a greater emphasis on these parts of the data. Consequently, when dealing with high (or low) frequency content, more (or less) sophisticated connections are necessary to capture rapid (or gradual) changes effectively over time. The means through which recurrent connections are implemented in the CWLSTM results in fast changes activating fast modules, while slow changes activate slow modules. The specific type of recurrent connections in CWLSTM allows the network to capture high-frequency content using faster modules with more recurrent connections. Conversely, slower modules assist in capturing low-frequency information with fewer connections. As a result, the proposed CWLSTM macro-modeling method exhibits a more thoughtful structure than traditional methods, leading to fewer parameters and more efficient training.

B. FORMULATION OF THE PROPOSED CWLSTM

Section II outlined various components of the traditional LSTM, with Equations (1)-(6) illustrating fully connected recurrent connections for these components. The structure of the proposed CWLSTM method results in different recurrent connections for these components. Therefore, the states and gates of the CWLSTM should be viewed as modified versions of Equations (1)-(6), as outlined below.

1) THE INPUT GATE

Considering the recurrent connections in the modular approach of CWLSTM, the formulation of the input gate for the ith unit of the module a at time step k can be expressed as follows:

$$i_{ia}(t_k) = \sigma\left(\sum_{s=1}^{N_I} w^{in}_{ia,s} I_s(t_k) + \sum_{m=a}^{N_m}\sum_{x=1}^{N_d} u^{in}_{ia,xm} h_{xm}(t_{k-1})\right) \quad (8)$$

where N_m, N_d, and $h_{xm}(t_{k-1})$ represent the number of modules, the number of units in each module, and the hidden state of the unit x in module m at time step $k-1$, respectively. Also, $w^{in}_{ia,s}$ and $u^{in}_{ia,xm}$ denote the weight connecting the sth input to the input gate of ith unit in module a and the weight connecting the hidden state of the xth unit in module m at time step $k-1$ to the input gate of the unit i in module a at time step k, respectively.

In (8), the feedforward connections remain the same as in (1), while the recurrent connections have been restructured according to the modular approach of CWLSTM. As can be seen in (8), the recurrent information from the previous time step (i.e., the hidden state of the previous time step) is directed to the focused unit at the current time step (unit i in module a), following the module order. In simpler terms, the focused unit receives recurrent information from modules slower than itself, i.e., modules with larger (or equal) clock periods.

2) THE FORGET GATE

The same concept, as described in (8), is also evident in (9), which represents the forget gate of unit i in module a within CWLSTM.

$$f_{ia}(t_k) = \sigma\left(\sum_{s=1}^{N_I} w^{f}_{ia,s} I_s(t_k) + \sum_{m=a}^{N_m}\sum_{x=1}^{N_d} u^{f}_{ia,xm} h_{xm}(t_{k-1})\right) \quad (9)$$

In (9), $w^{f}_{ia,s}$ and $u^{f}_{ia,xm}$ are the weight linking the sth input to the forget gate of ith unit in module a and the weight connecting the hidden state of the xth unit in module m at time step $k-1$ to the forget gate of unit i in module a at time step k, respectively.

3) THE OUTPUT GATE

Like (8) and (9), the modular approach of the CWLSTM affects the formulation of the output gate for the ith unit in module a at time step k:

$$o_{ia}(t_k) = \sigma\left(\sum_{s=1}^{N_I} w^{o}_{ia,s} I_s(t_k) + \sum_{m=a}^{N_m}\sum_{x=1}^{N_d} u^{o}_{ia,xm} h_{xm}(t_{k-1})\right) \quad (10)$$

in which, $w^{o}_{ia,s}$ and $u^{o}_{ia,xm}$ are the weight that connects the sth input to the output gate of ith unit in module a and the weight that links the hidden state of the xth unit in module m at time step $k-1$ to the output gate of the unit i in module a at time step k, respectively.

4) THE NEW MEMORY CELL

To bring the modular approach of the CWLSTM to the new memory cell, the following formulation is provided as a new memory cell in i^{th} unit in module a of the CWLSTM:

$$\hat{c}_{ia}(t_k) = \tanh\left(\sum_{s=1}^{N_I} w^{c}_{ia,s} I_s(t_k) + \sum_{m=a}^{N_m}\sum_{x=1}^{N_d} u^{c}_{ia,xm} h_{xm}(t_{k-1})\right) \quad (11)$$

where, $w^{c}_{ia,s}$ and $u^{o}_{ia,xm}$ are the weight linking the sth input to the new memory cell of the ith unit in module a and the

weight that links the hidden state of the xth unit in module m at time step $k-1$ to the new memory cell of the ith unit in module a at time step k, respectively.

5) THE MEMORY CELL STATE

Equations (8)-(11) illustrate the modular approach used to describe the gates and the new memory cell of the CWLSTM method. However, since the memory cell state of CWLSTM incorporates (8)-(11), this aspect differs from traditional LSTM. The equation (12) outlines the memory cell state of unit i in module a of CWLSTM.

$$c_{ia}(t_k) = f_{ia}(t_k) \times c_{ia}(t_{k-1}) + i_{ia}(t_k) \times \hat{c}_{ia}(t_k) \tag{12}$$

As shown in (12), the memory cell state of CWLSTM comprises input, forget, and output gates, as well as the new memory cell specific to the proposed CWLSTM, all of which have modular recurrent connections. Therefore, the memory cell state of CWLSTM differs from conventional LSTM's.

6) THE HIDDEN STATE

As demonstrated in (6), the hidden state of the conventional LSTM contains the output gate and memory cell state. In CWLSTM, the hidden state also includes the output gate and memory cell state, although restructured and modular. Therefore, the hidden state of cell i in module a of CWLSTM is expressed as follows:

$$h_{ia}(t_k) = o_{ia}(t_k) \times \tanh(c_{ia}(t_k)) \tag{13}$$

7) THE OUTPUT LAYER

Equation (13) represents the hidden state of the CWLSTM in a specified module unit. As previously mentioned, despite the modular connections in CWLSTM, the feedforward layers remain fully connected. Thus, the output of the CWLSTM at time step k is calculated as below (for convenience, a single neuron output layer is regarded in the equations):

$$\hat{y}(t_k) = \sum_{m=1}^{N_m} \sum_{x=1}^{N_d} w_{xm}^{out} h_{xm}(t_k) \tag{14}$$

where, w_{xm}^{out} is the weight connecting the xth unit of the mth module to the output layer. As demonstrated in (14), to generate the output at time step k, all units in all modules are interconnected.

Equations (8)-(11) introduce different terms (i.e., the doubled sigma) compared to those in (1)-(4) to represent the recurrent connections. These terms, following the modular approach in CWLSTM, prevent the fully connected recurrent structure seen in conventional LSTM formulations. The recurrent connections in CWLSTM can also be represented in matrix form. To illustrate this, consider the matrix formulation of a gate in CWLSTM (such as the input gate), as presented in (15):

$$\mathbf{i}(t_k) = \vec{\sigma}(\mathbf{W}^{in}\,\mathbf{I}(t_k) + \mathbf{U}^{in}\mathbf{H}(t_{k-1})) \tag{15}$$

where $\mathbf{i}(t_k) \in \mathbb{R}^{(N_m \times N_d)\times 1}$, $\vec{\sigma}$, $\mathbf{W}^{in} \in \mathbb{R}^{(N_m \times N_d)\times N_I}$, $\mathbf{I}(t_k) \in \mathbb{R}^{N_I \times 1}$, and $\mathbf{H}(t_{k-1}) \in \mathbb{R}^{(N_d \times N_m)\times 1}$ are the vector containing the values of the input gates of all units at time step k. This function maps a vector of values to a vector of sigmoid values, the matrix of weights connecting the input vector to the hidden layer, the input vector at time step k, and a vector containing hidden states of all units at time step $k-1$, respectively. Also, $\mathbf{U}^{in}$ is shown in (16), at the bottom of the next page.

The matrix presented in (16) consists of multiple matrixes. The ith matrix in the first row of (16) is specified for the weights used in the connections between module 1 at time step k and module i at time step $k-1$. The ith matrix in row 2 is reserved for the weights used in the connections of module 2 at time step k and module i at time step $k-1$, and so on. The matrices 1 to $j-1$ of row j in (16) are set to zero to implement the clockwork structure. A similar matrix as (16) is derived for each of the other gates of CWLSTM.

C. UPDATING MECHANISM OF CWLSTM VALUES

In conventional LSTM, like other traditional recurrent neural networks utilized in macromodeling nonlinear circuits, the hidden states of the recurrent network undergo updates to acquire new values. This means that at each time step, the hidden states are refreshed based on the new input provided to the hidden layer in a feedforward manner and the recurrent information received from the hidden states of the preceding time step.

In contrast to traditional LSTM used in circuit macromodeling, CWLSTM employs a selective mechanism for updating hidden states. This mechanism specifies that the values of hidden states are updated at each time step based on the input's time index and the modules' clock period, specifically, at time step k, units in module a receive new values only if k is a multiple of CP_a. Otherwise, the old value (i.e., the hidden state value of that module at the previous time step) remains unchanged. This selective mechanism encourages the network to respond to fast features of the input data with fast modules and to slow features with slow modules.

Based on the mechanism described earlier, the hidden state of unit i in module a in CWLSTM can be represented as follows over time:

$$h_{ia}(t_k) = \begin{cases} o_{ia}(t_k) \times \tanh(c_{ia}(t_k)) & \text{if} \begin{pmatrix} t_k \bmod \\ CP_a = 0 \end{pmatrix} \\ h_{ia}(t_{k-1}) & \text{otherwise} \end{cases} \tag{17}$$

As shown in (17), for a given unit in a specific module, if the index of the current time step is a multiple of the clock period of that module, the hidden state is updated with a new value. Otherwise, the previous value is retained as the current value. For instance, suppose there are 5 modules in CWLSTM and $t_k = 14$. In this scenario, only modules with $CP_1 = 1$ and $CP_2 = 2$ receive new values, while modules with $CP_3 = 4$, $CP_4 = 8$, and $CP_5 = 16$ retain their old values.

D. THE CWLSTM GRADIENTS

This section presents the gradients of the proposed CWLSTM macromodeling method to be used in training the proposed

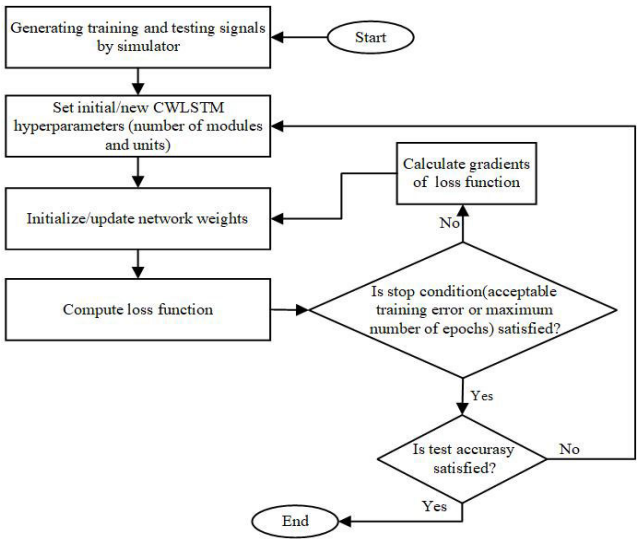

FIGURE 3. The flowchart of generating a macromodel of a nonlinear circuit using the proposed CWLSTM method.

network. To do this, there should be a loss function, which is shown below:

$$E_n(t_k) = \hat{y}_n(t_k) - y_n(t_k) \quad E_T = \frac{1}{2}\sum_{n=1}^{N_n}\sum_{k=1}^{N_T}(E_n(t_k))^2 \tag{18}$$

where $\hat{y}_n(t_k)$, $y_n(t_k)$, E_T and $E_n(t_k)$ stand for the predicted output for n^{th} training data at time step k using the proposed CWLSTM method, the desired output at time step k, the error for all time steps across all training data, and the error for one training data at time step k, respectively. Also, N_{n} and N_{T} denote the total number of training data and the time steps of training data, respectively.

For simplicity, the gradients are calculated for $E_n(t_k)$. To illustrate the gradient calculation procedure in the proposed CWLSTM, we specifically focus on a weight entangled in (10). To elaborate, we concentrate on $w^o_{ef,s}$ and determine the gradients with respect to this weight:

$$\frac{\partial E_n(t_k)}{\partial w^o_{ef,s}} = \frac{\partial E_n(t_k)}{\partial \hat{y}_n(t_k)}\sum_{m=1}^{N_m}\sum_{x=1}^{N_d}\frac{\partial \hat{y}_n(t_k)}{\partial h_{xm}(t_k)}\frac{\partial h_{xm}(t_k)}{\partial w^o_{ef,s}} \tag{19}$$

In (19), calculating $\partial E_n(t_k)\big/\partial w^o_{ef,s}$ involves computing three distinct terms. The terms $\partial E_n(t_k)\big/\partial \hat{y}_n(t_k)$ and $\partial \hat{y}_n(t_k)\big/\partial h_{xm}(t_k)$ follow a standard procedure and are determined using (18) and (14), respectively. However, the term $\partial h_{xm}(t_k)\big/\partial w^o_{ef,s}$ is specific to the CWLSTM network, and its calculation is outlined in (22):

$$\frac{\partial h_{xm}(t_k)}{\partial w^o_{ef,s}} = \begin{pmatrix} \frac{\partial o_{xm}(t_k)}{\partial w^o_{ef,s}}\tanh(c_{xm}(t_k)) \\ +\frac{\partial \tanh(c_{xm}(t_k))}{\partial w^o_{ef,s}} o_{xm}(t_k) \end{pmatrix} \tag{20}$$

The derivative term $\partial o_{xm}(t_k)\big/\partial w^o_{ef,s}$ in (20) can be computed in (21). As evident from (21), the modular approach is incorporated into the gradient computations when analyzing the gradients of the output gate.

$$\begin{cases} A = I_s(t_s) + \sum_{m'=m}^{N_m}\sum_{x'=1}^{N_d} u^o_{xm,x'm'} \dfrac{\partial h_{x'm'}(t_{k-1})}{\partial w^o_{ef,s}} & \text{if } e=x \text{ and } f=m \\ A = \sum_{m'=m}^{N_m}\sum_{x'=1}^{N_d} u^o_{xm,x'm'} \dfrac{\partial h_{x'm'}(t_{k-1})}{\partial w^o_{ef,s}} \\ \dfrac{\partial o_{xm}(t_k)}{\partial w^o_{ef,s}} = \sigma' \times A & \text{otherwise} \end{cases} \tag{21}$$

The subsequent derivative term, as shown in (20), is then calculated below:

$$\begin{aligned} B = \frac{\partial c_{xm}(t_k)}{\partial w^o_{ef,s}} &= \frac{\partial f_{xm}(t_k)}{\partial w^o_{ef,s}} c_{xm}(t_{k-1}) + \frac{\partial c_{xm}(t_{k-1})}{\partial w^o_{ef,s}} f_{xm}(t_k) \\ &+ \frac{\partial i_{xm}(t_k)}{\partial w^o_{ef,s}}\hat{c}_{xm}(t_k) + \frac{\partial \hat{c}(t_k)}{\partial w^o_{ef,s}} i_{xm}(t_k) \\ \partial \tanh(c_{xm}(t_k))\big/\partial w^o_{ef,s} &= \tan h' \times B \end{aligned} \tag{22}$$

The term $\partial c_{xm}(t_{k-1})\big/\partial w^o_{ef,s}$ in (22) is a recursive component of this equation (with respect to c_{xm}), while the computation of other derivatives follows the same method as in (21). Fig. 3 depicts the flowchart for generating a macromodel using the proposed CWLSTM.

$$\mathbf{U}^{in} = \begin{bmatrix} \begin{bmatrix} u^i_{11,11} & \cdot\cdot & u^i_{11,N_d1} \\ \vdots & \cdot\cdot & \vdots \\ u^i_{N_d1,11} & \cdot\cdot & u^i_{N_d1,N_d1} \end{bmatrix} & \begin{bmatrix} u^i_{11,12} & \cdot\cdot & u^i_{11,N_d2} \\ \vdots & \cdot\cdot & \vdots \\ u^i_{N_d1,12} & \cdot\cdot & u^i_{N_d1,N_d2} \end{bmatrix} & \cdot\cdot & \begin{bmatrix} u^i_{11,1N_m} & \cdot\cdot & u^i_{11,N_dN_m} \\ \vdots & \cdot\cdot & \vdots \\ u^i_{N_d1,1N_m} & \cdot\cdot & u^i_{N_d1,N_dN_m} \end{bmatrix} \\ 0 & \begin{bmatrix} u^i_{12,12} & \cdot\cdot & u^i_{12,N_d2} \\ \vdots & \cdot\cdot & \vdots \\ u^i_{N_d2,12} & \cdot\cdot & u^i_{N_d2,N_d2} \end{bmatrix} & \cdot\cdot & \begin{bmatrix} u^i_{12,1N_m} & \cdot\cdot & u^i_{12,N_dN_m} \\ \vdots & \cdot\cdot & \vdots \\ u^i_{N_d2,1N_m} & \cdot\cdot & u^i_{N_d2,N_dN_m} \end{bmatrix} \\ \vdots & \vdots & \cdot\cdot & \vdots \\ 0 & 0 & \cdot\cdot & \begin{bmatrix} u^i_{1N_m,1N_m} & \cdot\cdot & u^i_{1N_m,N_dN_m} \\ \vdots & \cdot\cdot & \vdots \\ u^i_{N_dN_m,1N_m} & \cdot\cdot & u^i_{N_dN_m,N_dN_m} \end{bmatrix} \end{bmatrix} \tag{16}$$

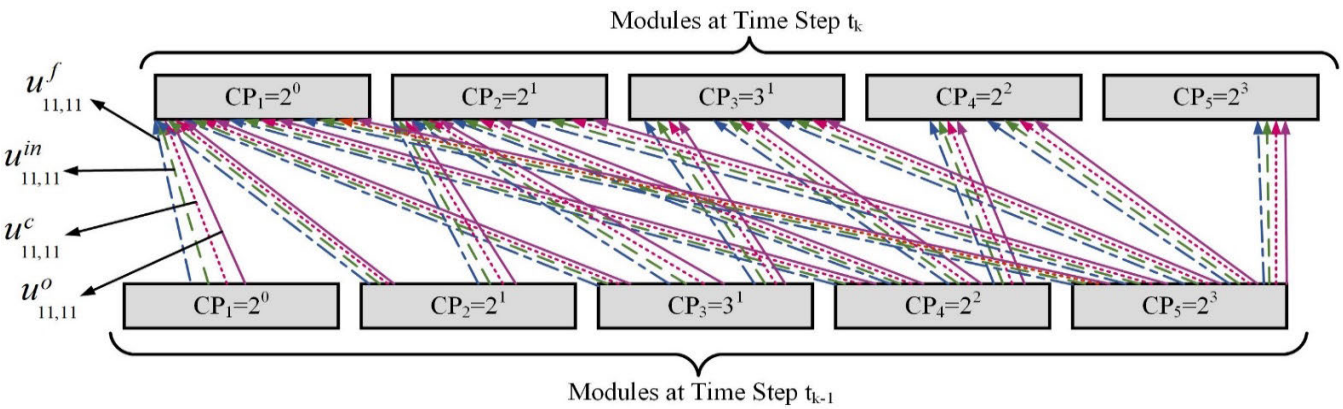

FIGURE 4. Recurrent connection in the proposed HM-CWLSTM with modules of 2^0, 2^1, 3^1, 2^2 and 2^3 clock periods.

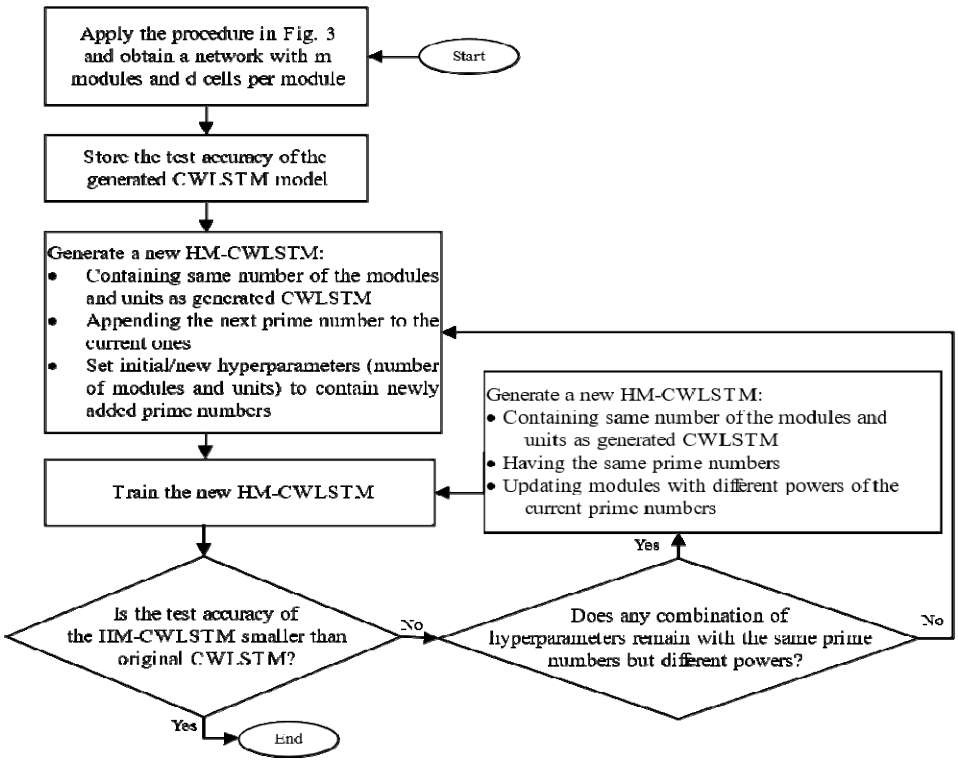

FIGURE 5. Flowchart of the method for generating a macromodel of a nonlinear circuit using the HM-CWLSTM.

E. HYBRID-MODULAR CWLSTM (HM-CWLSTM)

In section III-A, it was noted that the clock periods of the modules in the proposed CWLSTM are based on different powers of the number 2. To further enhance the performance of the CWLSTM method, we introduce the HM-CWLSTM here. This hybrid approach is inspired by the concept of using prime numbers of connection and their powers. In the HM-CWLSTM, we include the number 3 (and other prime numbers) alongside 2 to create additional entangled modules for capturing information across time steps. Fig. 4 provides an overview of the recurrent connections in the HM-CWLSTM, where the clock periods are derived from 2 and 3. The clock periods in the figure include 2^0, 2^1, 3^1, 2^2, and 2^3. In (17), it is shown that an LSTM unit in a module of CWLSTM is updated at the time step t_k only if k divides the clock period of that module. By incorporating multiple prime numbers in the HM-CWLSTM to generate clock periods, more modules are updated at the current time step. For example, if $k = 12$, three modules are updated in CWLSTM, whereas four modules are upgraded in HM-CWLSTM. This allows the latter model to capture the input-output relationship more effectively than the former, resulting in a more accurate model.

The process of generating the HM-CWLSTM is illustrated in Fig. 5.

F. CLOCKWORK CONCEPT IN GATING STRUCTURE

Gating mechanisms in network structures, such as Long Short-Term Memory (LSTM) networks, offer several advantages over non-gating structures (such as RNN). The primary benefit of gating is the ability to control the flow of information through the network. This control helps manage long-term dependencies and mitigate the vanishing gradient problem, which is a common issue in traditional recurrent neural networks (RNNs). Gating mechanisms, such as the input gate, forget gate, new memory cell, and output gate in LSTMs, allow the network to selectively retain or discard information, leading to more efficient learning and better performance on tasks involving sequential data.

Incorporating a clockwork approach into the gates of an LSTM (including input, forget, output gates, and memory cell state), changes the dynamics of information processing, storage, and propagation within the network. This modification causes the operations of different gates in an LSTM to occur at different clock periods. The integration of clockwork mechanisms with gating structures results in a more flexible network, capable of adapting to various temporal characteristics of the input data and improving computational efficiency and also test error. Detailed explanation of the effects is in below:

1) THE CLOCKWORK INPUT GATE

The input gate determines what new information is added to the next cell from the current input. It allows the LSTM to decide which parts of the new information are relevant for the current state, ensuring the network can focus on the most important aspects of the input. When input data has high-frequency noise and irregular changes, LSTM input couldn't filter them and causes decreasing LSTM performance. The clockwork approach can be applied to the input gate for modifying when and how the fast or slow changes in the input information is allowed to the next cell which results in only significant long or short-term patterns to be integrated into the next cell.

2) THE CLOCKWORK FORGATE GATE

The forget gate specifies what information to keep or to remove from the cell. Because of its formulation, if new information is not related to old information, it will let it be retained in the memory cell state and forget the old data. This ensures that the LSTM does not become overwhelmed by outdated information. Different modules process information at different specific frequencies when the clockwork approach is applied to the forget gate. In particular, fast modules can rapidly forget short-term information. Slower modules retain long-term information and update less frequently. This temporal differentiation helps the model optimizes the retention and forgetting of information within the cell more efficiently.

3) THE CLOCKWORK OUTPUT GATE

The output gate decides which part of the cell state should be chosen as cell output. It uses a combination of sigmoid and tanh functions to control the flow of information. It determines the next hidden state by filtering the current cell state. It ensures that only the most relevant information is passed on to the next time step, helping the LSTM to make accurate predictions based on the current context. When the clockwork approach is applied to the output gate, in fast signal segments, outputs primarily reflect immediate inputs and recent states. Fast modules can immediately pass short-term information to the output and enable rapid response to transient changes. On the other hand, in slow signal segments, outputs integrate information from larger temporal windows, aligning with patterns over extended durations because slow modules ensure that long-term information is gradually included in the output and prevent the model from being overly reactive to short-term noise. It reduces unnecessary high-frequency updates, saves computation, and determines when hidden state information is exposed to the rest of the network and output gates at clock-specific frequencies, controlling how often hidden state contributions are sent to the next time step. This selective output contribution optimizes temporal adaptation by balancing short-term and long-term influences. This structured approach enhances the LSTM's ability to adapt its outputs to varying temporal contexts dynamically.

4) THE CLOCKWORK NEW MEMORY CELL

The new memory cell represents the main part of the new information that is proposed to be added to the cell state. It provides a way for the LSTM cell to update its memory with new information while preserving the existing information stored in the cell state. When The clockwork approach is applied to a new memory cell, the main part of the cell state could be updated selectively based on the temporal relevance of data: Fast modules are updated frequently to capture short-term dynamics. Slow modules update less frequently, focusing on long-term patterns. This temporal segmentation improves the overall adaptability of the cell state. In addition, this approach segregates the cell state into short-term (fast modules) and long-term (slow modules) components. Short-term memory is stored and updated quickly, enabling the network to respond to sharp transient changes. Long-term memory is preserved and updated slowly, preventing overwriting by irrelevant short-term information.

With this structure and its functionality, the proposed method in this paper can be tailored to different datasets and circuits with varying characteristics by adjusting several hyperparameters and selecting which gates the clockwork approach should be applied to (In this paper, the clockwork structure is applied to all gates.). This approach allows for the design of a specialized network suitable for specific circuit response.

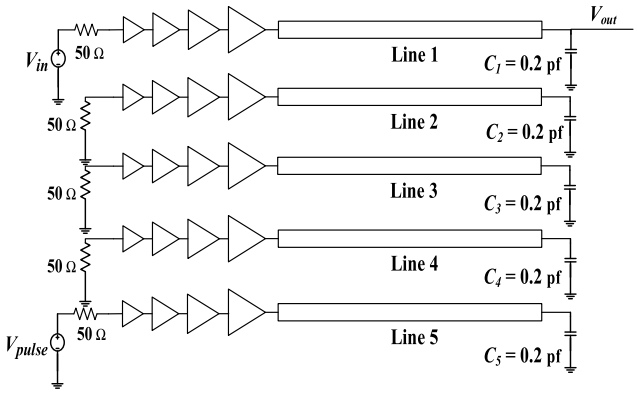

FIGURE 6. Schematic of a five-coupled line interconnect driven by 4-stage drivers.

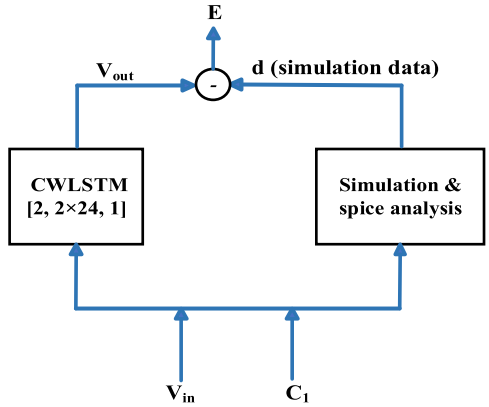

FIGURE 7. Structure of the proposed CWLSTM method for the circuit of figure. 6.

TABLE 1. Comparison among conventional RNN, conventional LSTM, and proposed CWLSTM methods for modeling the circuit of figure. 6.

Modeling method	Structure	Training error	Testing error	Training time(s)
Conventional RNN	(2,48,1)	33.6×10^{-4}	35×10^{-4}	3,694
Conventional LSTM	(2,48,1)	3.8×10^{-4}	10.5×10^{-4}	5,559
Proposed CWLSTM	(2,2×24,1)	3.1×10^{-4}	7.6×10^{-4}	3,821

TABLE 2. Comparison of run time speed-up among conventional LSTM-based, proposed CWLSTM-based, and transistor-level models for the circuit of figure. 6.

Modeling method	CPU Time (ms)	Speed-up
Conventional LSTM	37.17	46.49
Proposed CWLSTM	29.45	58.68
Transistor-level	1728	1

IV. NUMERICAL RESULTS

This section uses two nonlinear examples from high-frequency applications to demonstrate the proposed methods' performance and advantages.

A. FIVE-COUPLED TRANSMISSION LINE DRIVEN BY 4-STAGE DRIVER CIRCUIT

Fig. 6 illustrates the schematic of five-coupled transmission line driven by a driver circuit. Transmission lines, or T-lines, are conductive structures designed for efficiently transmitting electrical signals between locations. They are particularly relevant at high frequencies where distributed effects are prominent, and traditional lumped circuit analysis needs to be revised. The voltage and current of signals traveling along T-lines correspond to the line's characteristic impedance. Generating a model capturing the transient behavior in such transmission lines is advantageous. In this simulation, the length of T-lines is considered 12.4 cm. The driver circuit employs a cascade of four push-pull inverter stages implemented in a 130 nm CMOS process. Each stage is progressively upsized to achieve proper fan-out and to provide sufficient drive strength for the capacitive and inductive load presented by the coupled lines and load. This driver topology facilitates low output impedance and efficient signal transition. The transient time and transient step of SPICE simulation are considered 20 ns and 0.1 ns

The block diagram representing the CWLSTM used to model the circuit in Fig. 6 is illustrated in Fig. 7. In this figure, C_1 is a static parameter that signifies the load in the circuit of Fig. 6. The voltage across C_1 serves as the output voltage to be modeled. To model this circuit using the CWLSTM method, V_{in} and V_{pulse} generate square waveforms with a 2 ns period for training the model. V_{in} is the input signal whose rise/fall times were varied from 10 ps to 35 ps with steps of 5 ps, and V_{pulse} has an amplitude of 1.2 V. Additionally, the static parameter values ranged from 200 fF to 450 fF with steps of 50 fF. Additionally, some test signals were also generated that were not used in training, with rise/fall times ranging from 12.5 ps to 32.5 ps with 5 ps steps, and load capacitances for the test data varied from 225 fF to 425 fF with a step of 50 fF.

Table 1 is intended to compare the training time, as well as the absolute training and testing errors, among the conventional RNN, the traditional LSTM, and the proposed CWLSTM methods for modeling the circuit depicted in Fig. 6. Their network structure is also outlined in Table 1. For instance, (2, 48, 1) indicates 2 inputs, 48 hidden units, and 1 output, 2 × 24 in (2, 2 × 24, 1) comprises 2 modules, each containing 24 units. Additionally, the table reports the total number of model parameters. Reviewing Table 1, it becomes evident that the model generated using the proposed CWLSTM method yields significantly lower testing errors than the conventional RNN, the traditional LSTM. This makes the CWLSTM-based model a more accurate solution for addressing large-scale modeling challenges, like those in complex nonlinear circuits. Moreover, with its reduced number of parameters compared to the LSTM-based model, the proposed technique offers faster training.

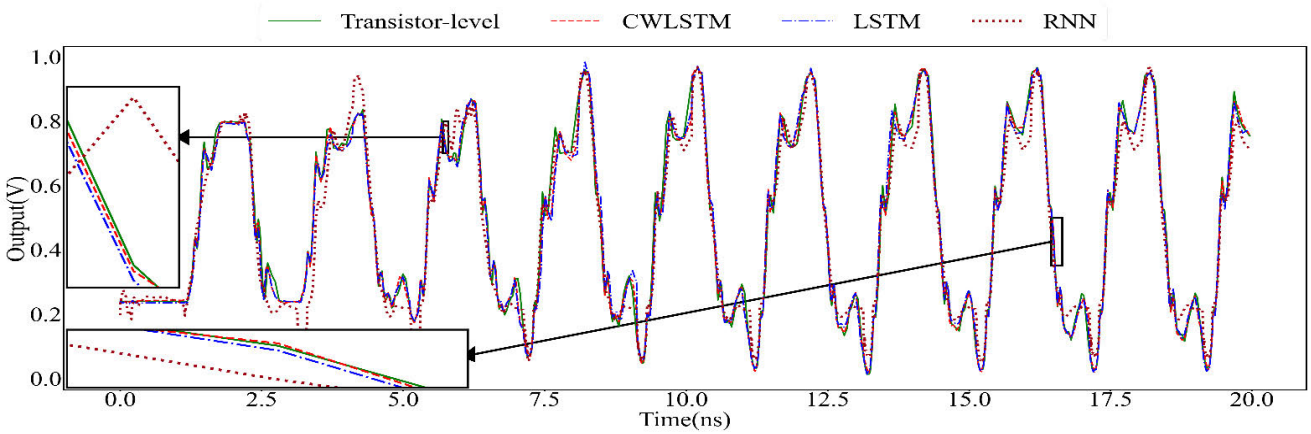

FIGURE 8. Comparison among the transistor-level, the proposed CWLSTM-based, the conventional LSTM-based, and the conventional RNN-based models when generating the output of a test signal for the circuit of figure. 6.

Table 2 compares the speed-up and CPU time expended among the conventional LSTM-based, the proposed CWLSTM-based, and the transistor-level models for the circuit depicted in Fig. 6. The results demonstrate that the CWLSTM-based model, while being the fastest, significantly reduces the time required to generate outputs compared to the transistor-level model. Consequently, the proposed method is a highly appropriate option for nonlinear applications.

TABLE 3. Comparison between the required amount of training data for conventional LSTM, and proposed CWLSTM methods for generating model with similar accuracy for the circuit of figure. 6.

Modeling method	No. of Training data	Training error	Testing error
Conventional LSTM	36	3.8×10^{-4}	10.5×10^{-4}
Proposed CWLSTM	22	3.6×10^{-4}	10.8×10^{-4}

Fig. 8 displays the simulated waveforms of the circuit depicted in Fig. 6, illustrating the test output signals generated by the proposed CWLSTM-based, the conventional RNN-based, the conventional LSTM-based, and the transistor-level models. The comparison in Fig. 8 demonstrates that the model produced by the proposed CWLSTM closely resembles the original output. In contrast, the conventional RNN modeling method cannot even train the data with suitable accuracy (due to vanishing gradient for large time steps) and models based on LSTM exhibit poorer performance in this context. Hence, the proposed CWLSTM method is superior in generating fast and accurate models.

Table 3 compares the number of data required for training, along with the training and testing error values, for the traditional LSTM approach and the proposed CWLSTM method applied to the circuit shown in Fig. 6. This comparison indicates that the CWLSTM method needs considerably less training data to attain a model with equivalent accuracy compared to the traditional LSTM method.

B. DC GENERATION CIRCUIT

Fig. 9.a shows a DC generator's schematic. This voltage generator is on-chip and suppresses defects such as the slow rise and fall times of the chip's input signal in the input buffers of a digital receiver. The output of this circuit is a clean digital signal for on-chip usage. This circuit slices the buffer's input data in the center by averaging the maximum and minimum values of the input waveform. The buffer, shown in Fig. 9.b, comprises NMOS and PMOS input buffers integrated in parallel. These input buffers adjust the rise and fall times of the original signal. The design of this nonlinear buffer is based on CMOS 50nm technology [30]. The transient time of the SPICE simulation is considered 30 ns.

TABLE 4. Comparison among conventional RNN, conventional LSTM, proposed CWLSTM, and proposed HM-CWLSTM methods for modeling the circuit of figure. 9.

Modeling method	Structure	Training error ($\times 10^{-4}$)	Testing error ($\times 10^{-4}$)
RNN	(2,48,1)	583	593
LSTM	(2,48,1)	70	71.7
CWLSTM	(2,3×16,1)	9.0	45.7
HM-CWLSTM	(2, [2^0, 2^1, 3^1],1)	8.3	34.3

Fig. 10 outlines the block diagram representing the CWLSTM approach employed for generating a model for the circuit in Fig. 9. In this diagram, the input voltage and C_{load} (load capacitance) are designated as inputs of the model. Similarly, Fig. 11 illustrates a block diagram for the HM-CWLSTM is used to model the same circuit. In this diagram, the structure (2, [2^0, 2^1, 3^1], 1) shows an HM-CWLSTM structure with two inputs, a hidden layer with 3 modules, each containing 16 neurons with clock rates of 2^0, 2^1, and 3^1 for modules 1 to 3, respectively, along with 1 output. Square waveforms with a period of 0.7 ns and an amplitude of 1 V

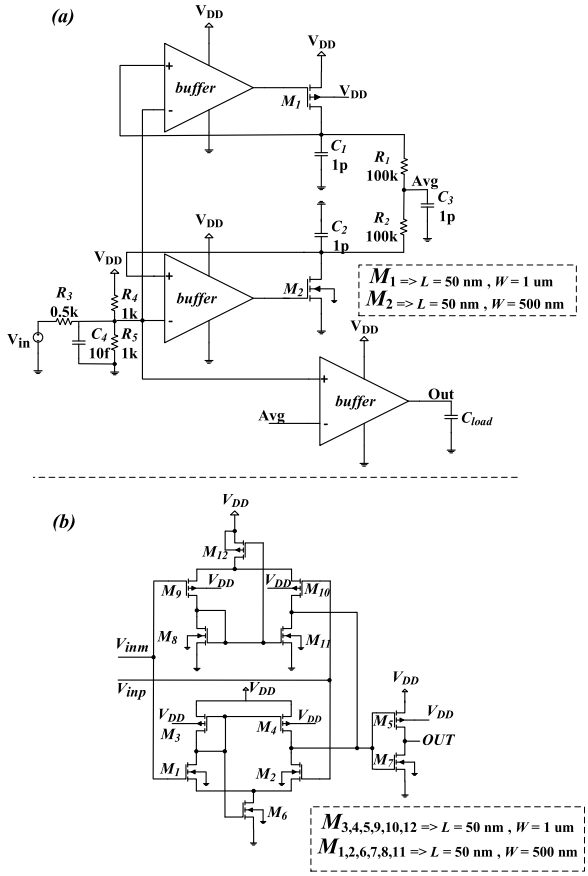

FIGURE 9. Schematic of DC voltage generator circuit (a) and buffer used in this circuit (b).

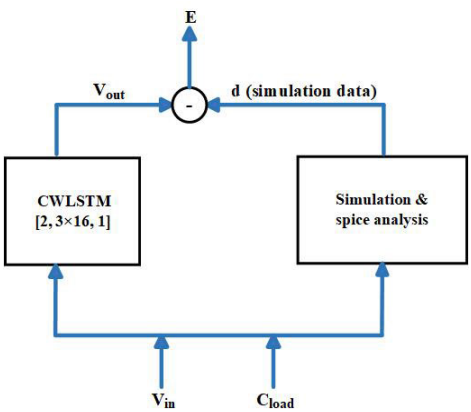

FIGURE 10. Structure of the proposed CWLSTM method for modeling the circuit of figure. 9.

were generated as training waveforms in modeling the circuit. The range of rise and fall times for these signals varied from 1 to 9 ps with steps of 2 ps. Additionally, the fixed parameter value (C_{load}) ranged from 1 to 4 fF with steps of 1 fF for generating training data.

Table 4 presents a comparison of the training and testing errors for the traditional RNN, the conventional LSTM, the presented CWLSTM, and HM-CWLSTM. Based on the results, it is evident that in this scenario, due to the extensive time steps needed for signal sampling (see Fig. 13) and the prolonged transient time, the conventional RNN faces the vanishing gradient problem and also the traditional LSTM method faces challenges in training and shows limited reduction in training errors. Conversely, the proposed CWLSTM and HM-CWLSTM methods not only achieve a significant reduction in training errors but also demonstrate satisfactory prediction accuracy for test data. Therefore, it can be inferred that the presented CWLSTM method is well-suited for modeling complex circuits with signals requiring a very large number of sampling time steps.

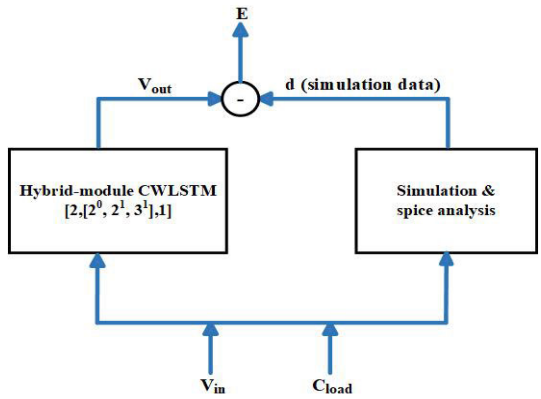

FIGURE 11. Structure of the proposed HM-CWLSTM method for modeling the circuit of figure. 9.

TABLE 5. Comparison of the evaluation time speed-up for conventional LSTM-based, proposed CWLSTM-based, proposed HM-CWLSTM-based, and transistor-level models for the circuit of figure. 9.

Modeling method	CPU time (ms)	Speed-up
LSTM	56.2	22.74
CWLSTM	41.4	30.87
HM-CWLSTM	42.2	30.28
Transistor-level	1278	1

TABLE 6. Comparison between the amount of training data for the conventional LSTM and the proposed HM-CWLSTM for generating model with similar accuracy for the circuit of figure. 9.

Modeling method	No. of Training data	Training error	Testing error
Conventional LSTM	20	70×10^{-4}	71×10^{-4}
HM-CWLSTM	5	59.3×10^{-4}	67.5×10^{-4}

Table 4 demonstrates the superior performance of the proposed CWLSTM method over other approaches and highlights the advantages of the presented HM-CWLSTM method. By combining numbers 2 and 3 for clock rates in the Hybrid-Modular structure, more modules are updated during training, allowing for a broader range of clock rates to be

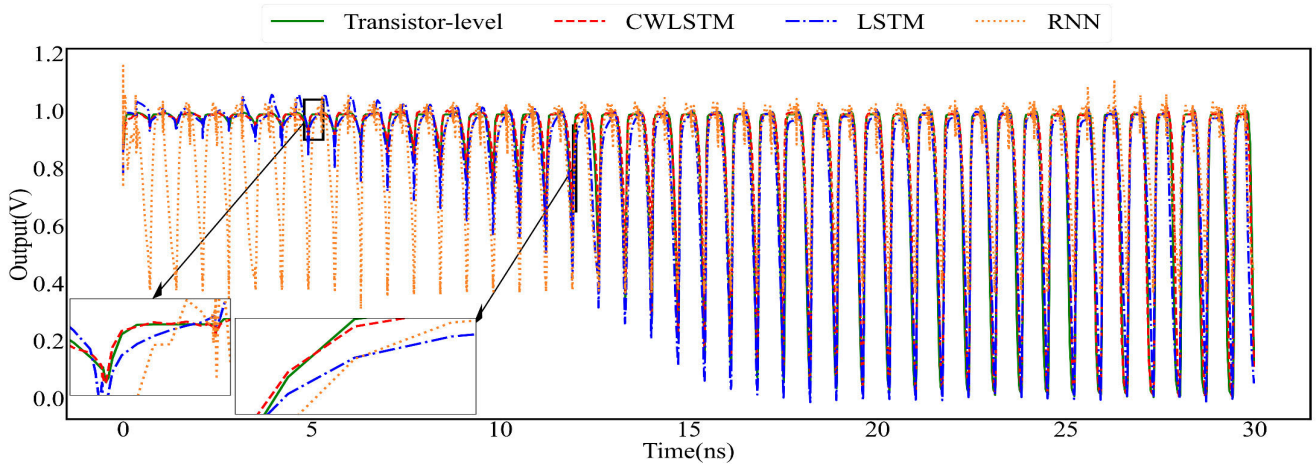

FIGURE 12. Comparison among the transistor-level, the proposed CWLSTM-based, the conventional LSTM-based, and the conventional RNN-based models when generating the output of a test signal for the circuit of figure. 9.

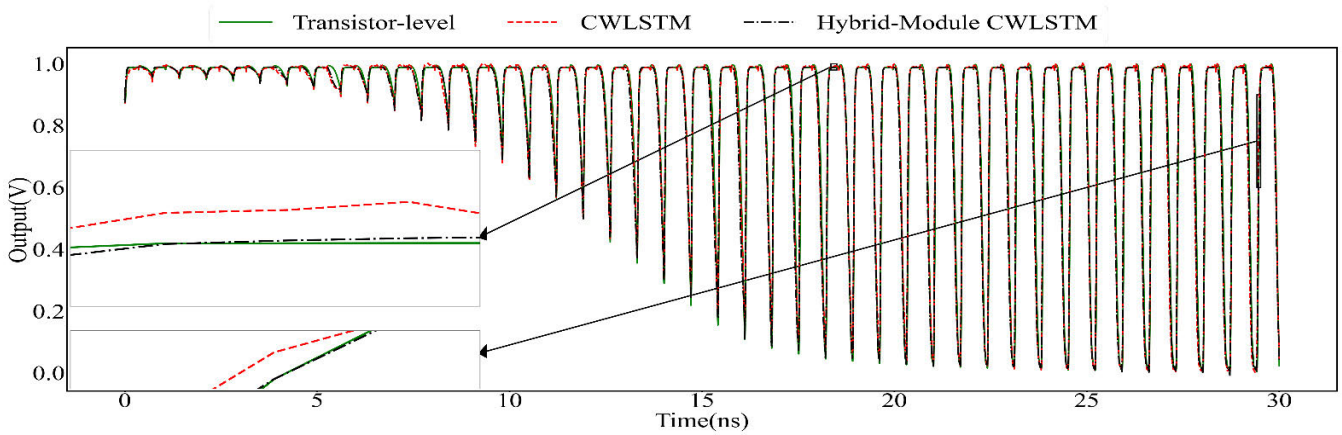

FIGURE 13. Comparison among the transistor-level, the proposed CWLSTM-based, and the HM-CWLSTM-based models when generating the output of a test signal for the circuit of figure. 9.

trained. As the table indicates, the proposed HM-CWLSTM method exhibits improved model accuracy compared to the original CWLSTM.

Table 5 compares the speed-up and CPU time for the proposed HM-CWLSTM-based, the proposed CWLSTM-based, the conventional LSTM-based, and the transistor-level models. The results demonstrate that the methods presented in this paper generate models that require significantly less time to compute than the transistor-level models. The advantages of the proposed CWLSTM and HM-CWLSTM reflect their suitability for modeling complex nonlinear circuits with large transient times.

Table 6 compares the number of data required for training, along with the training and testing error values, for the traditional LSTM approach and the proposed HM-CWLSTM method applied to the circuit shown in Fig. 9. This comparison indicates that the HM-CWLSTM method needs considerably less training data to attain a model with equivalent accuracy compared to the traditional LSTM method.

Fig. 12 depicts the test output waveforms produced by the proposed CWLSTM-based, conventional LSTM-based, conventional RNN-based, and transistor-level models for the circuit of Fig. 9. This figure demonstrates that the output signal obtained by the generated CWLSTM-based model closely aligns with the actual circuit output, surpassing the accuracy of the RNN and LSTM methods.

Fig. 13 compares the test output waveform generated by the CWLSTM method to the waveform derived from the HM-CWLSTM-based model for the circuit illustrated in Fig. 9. This comparison demonstrates that the HM-CWLSTM-based model exhibits a closer match with the target output, indicating its superior performance compared to the original proposed CWLSTM method.

V. CONCLUSION

In this study, we presented novel macromodeling techniques known as CWLSTM and its Hybrid version for modeling of high-speed nonlinear circuits. The proposed CWLSTM utilizes a unique type of recurrent connection different from those traditionally employed in circuit modeling with conventional RNN and LSTM. The modular recurrent connections in CWLSTM contribute to its reduced complexity yet enhanced effectiveness compared to the existing LSTM macromodeling approach. Within CWLSTM structure, modules are assigned clock periods based on powers of the number 2, determining their activation patterns. Additionally, we introduced a hybrid variant of CWLSTM aimed at further refining accuracy. The hybrid approach allows for other prime numbers with varying powers as clock rates. This modular design of CWLSTM is adaptable across different architectural setups.

Our experimental results indicated the superior performance of the proposed methods over conventional RNN, traditional LSTM, indicating significantly faster speeds than simulation tools. By successfully modeling two nonlinear high-speed circuits, the applicability of our methods has been validated, suggesting their potential as practical solutions for modeling intricate nonlinear circuits.

REFERENCES

[1] N. Mirzaie and R. Rohrer, "A macromodeling approach for analog behavior of digital integrated circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 12, pp. 5025–5031, Dec. 2020.

[2] W. Qi, G. Zong, and W. X. Zheng, "Adaptive event-triggered SMC for stochastic switching systems with semi-Markov process and application to boost converter circuit model," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 786–796, Feb. 2021.

[3] D. Lin, X. Liao, L. Dong, R. Yang, S. S. Yu, H. H. Iu, T. Fernando, and Z. Li, "Experimental study of fractional-order RC circuit model using the caputo and caputo-fabrizio derivatives," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 3, pp. 1034–1044, Mar. 2021.

[4] M. J. Shawon and V. Saxena, "Rapid simulation of photonic integrated circuits using verilog—A compact models," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3331–3341, Oct. 2020.

[5] R. Riaza, "Homogeneous models of nonlinear circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 6, pp. 2002–2015, Jun. 2020.

[6] Y. Li, L. Fang, T. Tao, D. Li, E.-X. Liu, N. Jin, M. Ahmed, and E.-P. Li, "Modeling and signal integrity analysis of RRAM-based neuromorphic chip crossbar array using partial equivalent element circuit (PEEC) method," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 9, pp. 3490–3500, Sep. 2022.

[7] J. K. Eshraghian, Q. Lin, X. Wang, H. H. C. Iu, Q. Hu, and H. Tong, "A behavioral model of digital resistive switching for systems level DNN acceleration," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 5, pp. 956–960, May 2020.

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

[9] 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.

[10] W. Na, W. Liu, L. Zhu, F. Feng, J. Ma, and Q.-J. Zhang, "Advanced extrapolation technique for neural-based microwave modeling and design," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 10, pp. 4397–4418, Oct. 2018.

[11] F. Charoosaei, M. Noohi, S. A. Sadrossadat, A. Mirvakili, W. Na, and F. Feng, "High-order deep recurrent neural network with hybrid layers for modeling dynamic behavior of nonlinear high-frequency circuits," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 12, pp. 5340–5358, Dec. 2022.

[12] M. A. Hemmat, O. Bilaniuk, S. Wagner, Y. Savaria, and J.-P. David, "RISC-V barrel processor for deep neural network acceleration," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, Dauegu, South Korea, Aug. 2021, pp. 1–5.

[13] P. Jaraut, S. K. Dhar, M. Helaoui, N. Boulejfen, M. Rawat, W. Chen, K. Rawat, N. Outaleb, and F. M. Ghannouchi, "Behavior modeling and digital predistortion of mismatched wireless transmitters using convolutional neural networks," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 1, pp. 336–340, Jan. 2023.

[14] M. Fayazi, Z. Colter, E. Afshari, and R. Dreslinski, "Applications of artificial intelligence on the modeling and optimization for analog and mixed-signal circuits: A review," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2418–2431, Jun. 2021.

[15] H. Zhang, Y. Jing, and P. Zhou, "Machine learning-based device modeling and performance optimization for FinFETs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 4, pp. 1585–1589, Apr. 2023.

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

[17] M. Moradi A., S. A. Sadrossadat, and V. Derhami, "Long short-term memory neural networks for modeling nonlinear electronic components," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 11, no. 5, pp. 840–847, May 2021.

[18] W. Liu, W. Na, F. Feng, L. Zhu, and Q. Lin, "A Wiener-type dynamic neural network approach to the modeling of nonlinear microwave devices and its applications," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Hangzhou, China, Dec. 2020, pp. 1–3.

[19] A. Faraji, S. A. Sadrossadat, M. Yazdian-Dehkordi, M. Nabavi, and Y. Savaria, "A hybrid approach based on recurrent neural network for macromodeling of nonlinear electronic circuits," *IEEE Access*, vol. 10, pp. 127996–128006, 2022.

[20] A. Faraji, M. Noohi, S. A. Sadrossadat, A. Mirvakili, W. Na, and F. Feng, "Batch-normalized deep recurrent neural network for high-speed nonlinear circuit macromodeling," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 11, pp. 4857–4868, Nov. 2022.

[21] J. Jin, C. Zhang, F. Feng, W. Na, J. Ma, and Q.-J. Zhang, "Deep neural network technique for high-dimensional microwave modeling and applications to parameter extraction of microwave filters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 4140–4155, Oct. 2019.

[22] J. Jin, F. Feng, J. Zhang, S. Yan, W. Na, and Q. Zhang, "A novel deep neural network topology for parametric modeling of passive microwave components," *IEEE Access*, vol. 8, pp. 82273–82285, 2020.

[23] A. Faraji, S. A. Sadrossadat, W. Na, F. Feng, and Q.-J. Zhang, "A new macromodeling method based on deep gated recurrent unit regularized with Gaussian dropout for nonlinear circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 7, pp. 2904–2915, Jul. 2023.

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

[25] P. Kumar, S. R. Arya, K. D. Mistry, and A. K. Giri, "Hybrid self-learning controller for restoration of voltage power quality using optimized multilayer neural network," *Int. J. Circuit Theory Appl.*, vol. 49, no. 12, pp. 4248–4273, Dec. 2021.

[26] F. Charoosaei, A. Faraji, S. A. Sadrossadat, A. Mirvakili, W. Na, F. Feng, and Q.-J. Zhang, "High-speed nonlinear circuit macromodeling using hybrid-module clockwork recurrent neural network," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 71, no. 2, pp. 767–780, Feb. 2024.

[27] D. Georgiadou, V. Diakoloukas, V. Tsiaras, and V. Digalakis, "ClockWork-RNN based architectures for slot filling," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 2481–2485.

[28] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber, "A clockwork RNN," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, Feb. 2014, pp. 1863–1871.

[29] M. Noohi, A. Faraji, S. A. Sadrossadat, A. Mirvakili, and A. Moftakharzadeh, "Modeling and implementation of a novel active voltage balancing circuit using deep recurrent neural network with dropout regularization," *Int. J. Circuit Theory Appl.*, vol. 51, no. 5, pp. 2351–2374, May 2023.

[30] R. Baker, *CMOS: Circuit Design Layout Simulation*. Hoboken, NJ, USA: Wiley, 2011.

RAZIEH MORADI CHALESHTORI received the B.S. degree in electrical engineering from the University of Shahrekord, Shahrekord, Iran, in 2014, and the master's degree in electronic engineering at Isfahan University, Isfahan, Iran, in 2018. Her research interests include digital system design, linear/nonlinear circuits macromodeling, and neural networks and their applications in modeling and simulation of circuits and components.

AMIN SABOOHI received the B.S. degree in electrical engineering from the University of Yazd, Yazd, Iran, in 2014, and the master's degree in electronic engineering from Isfahan University, Isfahan, Iran, in 2017. His research interests include digital system design, linear/nonlinear circuits macromodeling, and neural networks and their applications in modeling and simulation of circuits and components.

AMIN FARAJI received the B.S. degree in computer engineering from the University of Mohaghegh Ardabili, Ardabil, Iran, in 2019, and the master's degree in artificial intelligence from the Department of Computer Engineering, Yazd University, Yazd, Iran, in 2022. He is currently pursuing the Ph.D. degree in artificial intelligence with Shahid Beheshti University, Tehran, Iran. His research interests include computer-aided design, deep learning, and neural networks and their applications in modeling and simulation of circuits and components.

SAYED ALIREZA SADROSSADAT (Senior Member, IEEE) received the B.Sc. degree from the University of Tehran, Tehran, Iran, in 2007, the master's degree from the University of Waterloo, Waterloo, ON, Canada, in 2010, and the Ph.D. degree from Carleton University, Ottawa, ON, Canada, in 2015. He is currently a Faculty Member of Artificial Intelligence Group, Department of Computer Engineering, Yazd University, Yazd, Iran, and affiliated with the Department of Electronics Group, Electrical Engineering, Yazd University. 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. He 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*, and IEEE Canadian Journal of Electrical and Computer Engineering.

ALI MOFTAKHARZADEH received the B.Sc. degree in electrical engineering from K. N. Toosi University of Technology, Tehran, Iran, in 2002, and the M.Sc. and Ph.D. degrees in electrical engineering from Sharif University of Technology, Tehran, in 2004 and 2009, respectively. In 2010, he joined the Department of Electrical Engineering, Yazd University, Yazd, Iran. His current research interests include digital system design, DSP, image processing, and linear/nonlinear circuits macromodeling.

YVON SAVARIA (Life Fellow, IEEE) received the B.Ing. and M.Sc. degrees in electrical engineering from École Polytechnique Montreal, in 1980 and 1982, respectively, and the Ph.D. degree in electrical engineering from McGill University, in 1985.

Since 1985, he has been with Polytechnique Montréal, where he is currently a Professor with the Department of Electrical Engineering. He has carried out work in several areas related to microelectronic circuits and microsystems, such as testing, verification, validation, clocking methods, defect and fault tolerance, effects of radiation on electronics, high-speed interconnects, and circuit design techniques, CAD methods, reconfigurable computing and applications of microelectronics to telecommunications, aerospace, image processing, video processing, radar signal processing, and the acceleration of digital signal processing. He is currently involved in several projects related to embedded systems in aircraft, wireless sensor networks, virtual networks, software-defined networks, machine learning (ML), embedded ML, computational efficiency, and application-specific architecture design. He holds 16 patents, has published 211 journal papers and 495 conference papers, and was the thesis advisor of 190 graduate students who completed their studies. He is a Fellow of Canadian Academy of Engineering. He was the Program Co-Chairperson of NEWCAS'2018 and the General Chairperson of NEWCAS2020. He has been working as a Consultant or was sponsored for carrying out research with Bombardier, Buspass, CNRC, Design Workshop, Dolphin, DREO, Ericsson, Genesis, Gennum, Huawei, Hyperchip, Intel, ISR, Kaloom, LTRIM, Miranda, MiroTech, Nortel, Octasic, PMC-Sierra, Space Codesign, Technocap, Thales, Tundra, and Wavelite. He is the Co-Director of the Regroupement Stratégique en Microé lectronique du Qué bec (RESMIQ) and a member of the Ordre des Ingé nieurs du Qué bec (OIQ). In 2001, he received the Tier 1 Canada Research Chair (www.chairs.gc.ca) on the Design and Architecture of Advanced Microelectronic Systems, which he held until June 2015. He also received the Synergy Award from Canada's Natural Sciences and Engineering Research Council in 2006. Since June 2019, he has been the NSERC-Kaloom-Intel-Noviflow (KIN) Chair Professor.