Hybrid ML-EMT Based Digital Twin for Device-Level HIL Real-Time Emulation of Ship-Board Microgrid on FPGA

Songyang Zhang, *Student Member, IEEE,* Tian Liang, *Member, IEEE,* and Venkata Dinavahi, *Fellow, IEEE*

***Abstract*—Maritime industries desire high speed and reliability, low lifespan cost, and environmental impact shipping for transportation. Compared to highly congested land shipments and high-cost air freight, all-electric ship (AES) can reduce the lifespan energy consumption and transport a considerable freight volume at a lower rate. Recently, the medium voltage DC (MVDC) topology, recommended by IEEE standard, pushes the AES to the next stage in considering space and weight constraints with the reduction of bulky transformers and simplified parallel connections. However, device-level modeling of this massive parallel MVDC-based ship-board microgrid (SBM) is challenging to both the state-of-the-art general-purpose compute unit and traditional electromagnetic transient (EMT) based emulation. With the rapid development of machine learning (ML) algorithm and its dedicated execution unit, accelerated parallel emulation becomes achievable in different levels of this paralleled connected SBM. Applying the ML-aided technique can help to improve the emulation execution efficiency and reduce the consumption of hardware resource on the field-programmable gate arrays (FPGAs). This work proposes a real-time hybrid ML-EMT based digital twin of the complete SBM at the subsystem-level and equipment-level with validated results from PSCAD/EMTDC®, and device-level with validated results from SaberRD®.**

***Index Terms*—All-electric ship (AES), artificial intelligence (AI), digital twin, electromagnetic transient (EMT), field-programmable gate arrays (FPGAs), gated recurrent units (GRU), hardware-in-the-loop (HIL), machine learning (ML), power electronics, ship-board microgrid (SBM), recurrent neural network (RNN), real-time systems, silicon carbide (SiC).**

List of Abbreviations

ATRUS	Auto-Transformer Rectifier Unit Subsystem
DRNN	Dependent Recurrent Neural Network
EMT	Electromagnetic Transient
ESS	Energy Storage Subsystem
FNN	Forward Neural Network
FPGA	Field-Programmable Gate Array
GRU	Gated Recurrent Unit
HIL	Hardware-In-the-Loop
IDCDC	Isolated DC-DC Equipment
IGBT	Insulated-Gate Bipolar Transistor
IRNN	Independent Recurrent Neural Network
MAE	Mean Absolute Error
ML	Machine Learning
MMC	Modular Multilevel Converter
MVDC	Medium-Voltage DC
PMSM	Permanent Magnet Synchronous Motor
RNN	Recurrent Neural Network
SBM	Ship-Board Microgrid
TLC	Two-Level Converter
TLM	Transmission Line Modeling
TSFNN	Time-Series Forward Neural Network

This work is supported by the Natural Science and Engineering Research Council of Canada (NSERC), Mitacs Accelerate program, and RTDS Technologies Inc. (Corresponding author: Tian Liang)

S. Zhang and V. Dinavahi are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada. Email: zhang.songyang@ualberta.ca, dinavahi@ualberta.ca.

T. Liang was with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada. He is now with RTDS Technologies Inc., Winnipeg, Manitoba R3T 2E1, Canada. Email: tliang5@ualberta.ca.

I. Introduction

Global economic output plummeted by 3.5% in 2020 as a result of the pandemic, as did commercial commerce, while international maritime exports fell by 3.8% to 10.65 billion tonnes [1]. However, global marine trade has recovered by 4.3% in 2021, with growth expected to continue over 2022-2026, albeit at rates that will be moderated due to an easing of world economic production. Meanwhile, the International Maritime Organization (IMO) set a sustainable development goal to reduce greenhouse gas emissions by at least 50% before the year 2050, compared with the year 2008 [2]. Energy-efficiency based design in the marine industry becomes a preferable choice for future ship application. Thus, the medium-voltage DC (MVDC) based all-electric ship (AES) [3], [4] has drawn the attention of the researchers and manufacturers due to its high energy-efficiency, reliability, reconfigurability, low lifespan cost, and the reduction of the bulky AC power equipment. With the proliferation of novel power electronic-based equipment in MVDC-based AES, there has been an increasing focus on modeling and simulation techniques for these microgrid systems [5]. These methodologies provide a virtual platform to validate various issues and control strategies pertaining to islanded microgrids. However, conventional simulation tools and models employed on computers face significant challenges in terms of interactive validation with real-world physical devices and control strategies. In response to these challenges, considerable research efforts have been dedicated to hardware-in-the-loop (HIL) applications for power system evaluation [8] and assessment of power switch stress performance [9], [10]. Consequently, there is an urgent and compelling need to develop real-time emulation of comprehensive ship-board microgrid (SBM)

systems, incorporating device-level modeling [11] - [13].

The traditional way of simulating large-scale and paralleled connected electrified transportation systems [14] - [16] is to solve the physical nonlinear differential equations with full matrix nodes integrated. Then, a heavy computational burden (e.g., Newton-Raphson iterative method) is needed by solving these matrices or other complex iterative mathematical calculations. Separating the traditional power system into multiple sections without theoretical loss is a favourable technique for accelerating the calculation with parallel execution (e.g. transmission line modeling method) [17], [18]. Many studies [19], [20] on parallel computing in power systems have been conducted to accomplish real-time emulation of system dynamics. However, a complex combined all level real-time emulation for these large-scale and paralleled connected MVDC-based microgrids is not yet reported in literature and practice.

With the rapid development of machine learning (ML) technology and its accelerated hardware, it is a possible solution for integrating all levels of microgrid models into multiple complex black-box models by different ML methods. These methods were widely researched on fault detection [21], load estimation [22], and control algorithm [23], HIL emulation for microgrid systems (e.g, more electric aircraft power systems [24] and the high-speed rail microgrid [25]). Previous studies utilize ML to replace traditional electromagnetic transient (EMT) equipment and investigate the benefits, which facilitates in achieving the optimal trade-off point when considering execution efficiency, adaptability, variety, and accuracy [24], [25]. Nonetheless, equations-based EMT models have advantages over ML methods in terms of being succinct, efficient, and straightforward to comprehend. Both EMT-based and ML-based modeling methods have been explored independently, but little attention is given to the comprehensive and detailed comparison between EMT and ML models, let alone the ML-EMT-based hybrid models. As a result of merging traditional EMT and ML models in the microgrid system, comparing and selecting or analyzing hybrid models at the subsystem, equipment, and device levels will benefit from both EMT and ML methodologies. Microgrid HIL emulation can be accelerated and resource-efficient by combining classical EMT with ML technology without sacrificing robustness and stability.

This paper proposes the real-time hierarchical digital twin for the MVDC-based SBM on the Xilinx® Ultrascale+ architecture FPGA hardware platform with the hybrid ML-EMT modeling techniques. The paper is organized as follows: Section II describes the overview of the MVDC-based SBM power system. Section III discusses different neural networks (NN) structures for ML modeling. Section IV propose the various level hybrid ML-EMT based models. Section V gives an introduction to the ML training processes and explains the details of the SBM digital twin hardware implementation. Section VI shows the real-time emulation results and verification using experimentally verified software PSCAD/EMTDC® (system-level, subsystem-level, and equipment-level) and SaberRD® (device-level). Section VII gives the conclusion of this paper.

II. Overall Architecture of SBM Power System

In Fig. 1 (d), the whole SBM power system consists of generator subsystems, energy storage subsystems (ESS), propulsion subsystems, and load zones. In this paper, the emulation models can be classified into system-level, subsystem-level equipment-level and device-level, based on the complexity of topology and the purpose of models. Regardless of their levels, these models may be represented using the standard nodal voltage approach. This section focuses on typical EMT modeling techniques for these distinct kinds of equipment. Following the introduction of the EMT theoretical base, ML approaches can be used to create hybrid models in section IV. The TLM method [26] is used link numerous subsystems, allowing each component of overall SBM to run in parallel.

A. Energy Storage Subsystem

In the ESS, the isolated DC-DC equipment (IDCDC) and battery are included, shown in Fig. 1 (a), where the IDCDC can be divided into two parts: the left side connects to a battery as input, and the right side connects to the DC bus for the loads. Then, the transformer can be equivalent to inductances, and the topology in the simulation system is shown in Fig. 2 (a). In order to accelerate the simulation system, two parts are separated and executed in parallel by the TLM method, shown in Fig. 2 (b). The left side is investigated further, and it is found to be equivalent to a 7-node network with equivalent current sources and resistances, and the equations are displayed:

$$\mathbf{I}_e^{t-\Delta t} = \mathbf{R}_e \cdot \mathbf{I}^{t-\Delta t}, \quad \mathbf{V}^t = \mathbf{G}^{-1}\mathbf{I}_e^{t-\Delta t}, \quad \mathbf{I}^t = \mathbf{R} \cdot \mathbf{V}^t, \quad (1)$$

where $\mathbf{I}_e$ and $\mathbf{R}_e$ represent the history equivalent current and resistance matrix of nodes; $\mathbf{V}$, $\mathbf{I}$ and $\mathbf{R}$ mean the voltage, current and resistance matrix of the nodes; $\mathbf{G}$ is the admittance matrix for the network; the superscripts, $t-\Delta t$ and t, represent the previous time-step and the current time-step. Following that, the circuit on the right may be modeled in a similar way, and all node voltage and current are determined.

B. Generator Subsystem

In this subsystem, a synchronous generator (SG), an auto-transformer rectifier units (ATRU), and a filter are included, shown in Fig. 1 (b). 250 kVA generator is control to provide stable AC voltage, then converter to 6 kV DC voltage supply by ATRU. The EMT modeling equations of SG can be obtained based on the following assumptions: 1) stator abc three-phases windings have the same structure, are rotationally symmetric, and are spatially separated by 120 electrical degrees; 2) the rotor structure is symmetric with respect to the $d-$ and $q-$axis; 3) the magnetic potential generated by the currents in the stator and rotor windings is sinusoidally distributed with the fundamental waveform in the air gap, disregarding harmonics; 4) the magnetic circuit is linear, without saturation, hysteresis, and the eddy current losses, and the skin effect is

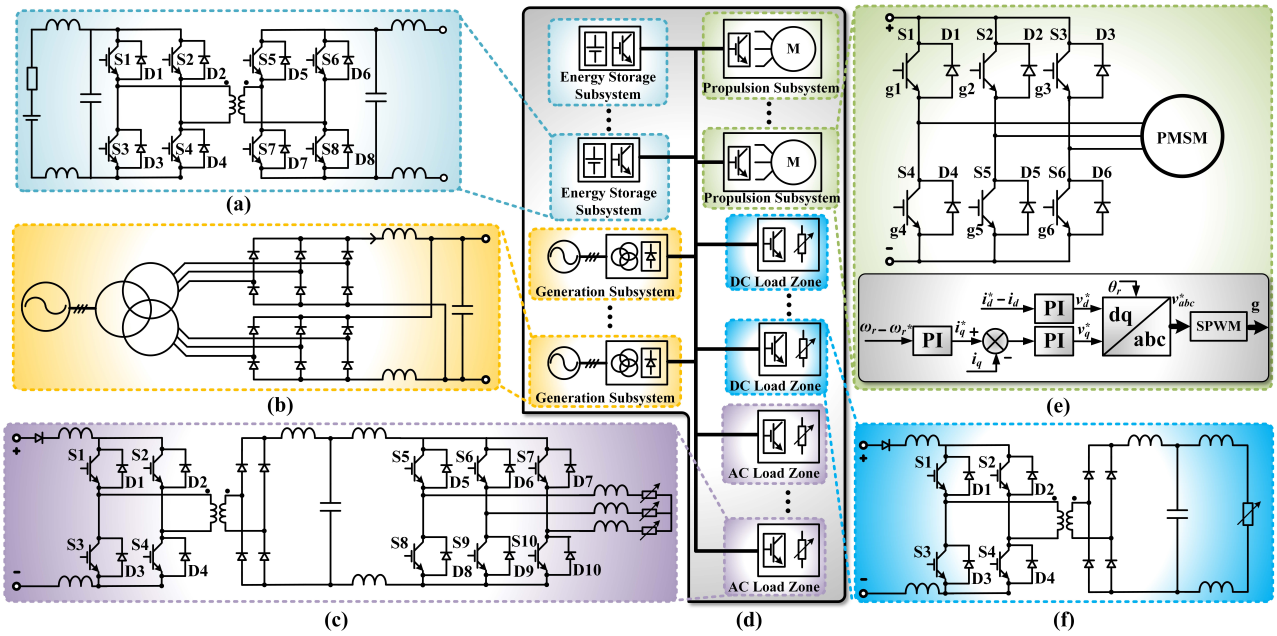

Fig. 1. SBM power system: (a) energy storage subsystem; (b) generator subystem; (c) AC load zone; (d) overall system topology; (e) propulsion subsystem; (f) DC load zone.

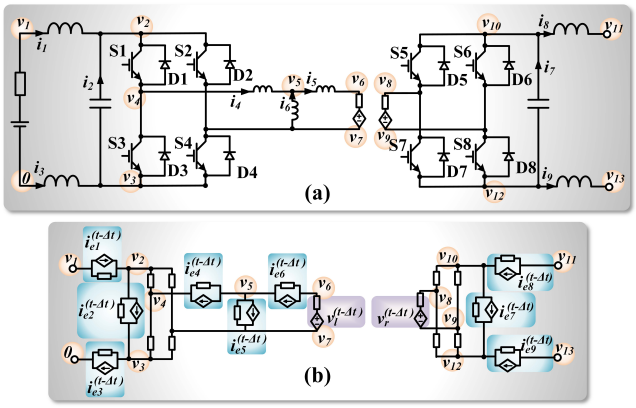

Fig. 2. Energy storage subsystem: (a) divided-transformer topology; (b) equivalent circuit.

neglected. The equations of SG with damping windings are presented to analyze modeling processes:

$$v_d = R_s i_d - \omega_r \lambda_q + \frac{d\lambda_d}{dt}, \quad v_q = R_s i_q + \omega_r \lambda_d + \frac{d\lambda_q}{dt}, \tag{2}$$

$$v_{fd} = R_{fd} i_{fd} + \frac{d\lambda_{fd}}{dt}, \quad R_{kd} i_{kd} = -\frac{d\lambda_{kd}}{dt}, \tag{3}$$

$$R_{kq1} i_{kq1} = -\frac{d\lambda_{kq1}}{dt}, \quad R_{kq2} i_{kq2} = -\frac{d\lambda_{kq2}}{dt}, \tag{4}$$

$$\lambda_d = L_d i_d + L_{md}(i_{kd} + i_{fd}), \quad \lambda_q = L_q i_q + L_{mq} i_{kq}, \tag{5}$$

$$\lambda_{fd} = L_{fd} i_{fd} + L_{md}(i_d + i_{kd}), \tag{6}$$

$$\lambda_{kd} = L_{kd} i_{kd} + L_{md}(i_d + i_{fd}), \tag{7}$$

$$\lambda_{kq1} = L_{kq1} i_{kq1} + L_{mq} i_q, \quad \lambda_{kq2} = L_{kq2} i_{kq2} + L_{mq} i_q. \tag{8}$$

where v_d, v_q, i_d, i_q, λ_d, and λ_q mean the dq axis voltage, current and flux linkage; v_{fd}, i_{fd} and λ_{fd} represent the field winding voltage, current and flux linkage; v_{kd}, v_{kq1}, v_{kq2}, i_{kd}, i_{kq1}, i_{kq2}, λ_{kd}, λ_{kq1} and v_{kq2} are the dq axis damper voltage, current and flux linkage; R_s, R_{fd}, R_{kd}, R_{kq1}, and R_{kq2} are the stator, field winding, and dq axis damper winding resistors, respectively; and L_d, L_q, L_{fd}, L_m, L_{kd}, L_{kq1}, and L_{kq2} are the dq axis stator, field winding, magnetizing, and dq axis damper winding inductances, respectively.

The typical simulation computation is separated into two sections: the first procedure employs the Forward Euler technique to determine the flux linkage of formula (2)-(8) based on the voltage and current values at the last time-step. which are shown as (9)-(14):

$$\lambda_d^t = \Delta t(v_d^{t-\Delta t} - R_s i_d^{t-\Delta t} + \omega_r^{t-\Delta t} \lambda_q^{t-\Delta t}) + \lambda_q^{t-\Delta t}, \tag{9}$$

$$\lambda_q^t = \Delta t(v_q^{t-\Delta t} - R_s i_q^{t-\Delta t} + \omega_r^{t-\Delta t} \lambda_d^{t-\Delta t}) + \lambda_d^{t-\Delta t}, \tag{10}$$

$$\lambda_{fd}^t = \Delta t(v_{fd}^{t-\Delta t} - R_{fd} i_{fd}^{t-\Delta t}) + \lambda_{fd}^{t-\Delta t}, \tag{11}$$

$$\lambda_{kd}^t = -\Delta t R_{kd} i_{kd}^{t-\Delta t} + \lambda_{kd}^{t-\Delta t}, \tag{12}$$

$$\lambda_{kq1}^t = -\Delta t R_{kq1} i_{kq1}^{t-\Delta t} + \lambda_{kq1}^{t-\Delta t}, \tag{13}$$

$$\lambda_{kq2}^t = -\Delta t R_{kq2} i_{kq2}^{t-\Delta t} + \lambda_{kq2}^{t-\Delta t}. \tag{14}$$

Then, in the second step, the magnetic linkage acquired from the previous solution is utilized to solve the homogeneous linear equations to obtain the current value at the current time-step, which involves inverting the impedance matrix:

$$\mathbf{I}^t = \mathbf{Z}^{-1} \mathbf{A}_\lambda^t, \tag{15}$$

where $\mathbf{A}_\lambda$, $\mathbf{Z}$, and $\mathbf{I}$ are the magnetic linkage matrix, the impedance matrix, and the current matrix, respectively.

C. Propulsion Subsystem

As illustrated in Fig. 1 (d), the system has propulsion subsystems, each of which consists of a two-level converter (TLC) and a 60 kVA PMSM with a dampening circuit. Fig. 1 (e) depicts the PMSM's control scheme, which controls

TLC to regulate the PMSM's dq axis current. PMSM's motor equations are similar to SG's, updated by dq axis current and flux linkage, and the particular formula is described in detail by the prior research [24]. It is worth noting that PMSM modeling takes into account both the electromagnetic torque and the mechanical component, whereas the electrical part is calculated solely for the SG. The procedure of computing the current and flux linkage is identical, except that the PMSM permanent magnet flux linkage λ_{pm} substitutes the synchronous motor field winding voltage v_{fd}, current i_{fd} and flux linkage λ_{fd}. Then, electromagnetic torque and the mechanical part are shown as follows:

$$T_e = \frac{3}{2}\frac{p}{2}\left[\lambda_{pm} i_q + \left(L_d - L_q\right) i_d i_q\right], \tag{16}$$

$$J\frac{d\omega_r}{dt} = T_e - T_m - B\omega_r, \qquad \frac{d\theta_m}{dt} = \omega_r, \tag{17}$$

where i_d, i_q, L_d, L_q are the dq axis stator current and inductances, respectively; T_e, T_m, ω_r, J, B, θ_m are the electrical torque, mechanical torque, rotor speed, moment of inertia, friction factor, and rotor position angle, respectively.

D. Load Zone

There are two sorts of load zones, which are characterized primarily by the type of load linked. The DC load zone is depicted in Fig. 1 (f). Before driving the DC load, the required DC voltage is obtained through the IDCDC. The IDCDC is identical to that in the ESS, but the energy flows in one direction, so diodes rather than IGBTs are employed on the rectifier side. The voltage generated by the IDCDC can be changed to meet the demands of the load, however, is generally less than 1 kV.

Fig. 1 (c) depicts the AC load zone, which differs from the DC load zone in that the TLC generates three-phase AC power to drive the AC load after obtaining the DC bus voltage. The AC level is generally a commonly used load voltage, such as 110 V 50 Hz AC, which can be controlled and adjusted according to actual needs.

III. Machine Learning Modeling Methods

In this section, ML modeling methods for power electronics digital twin are discussed. Based on the complexity and the function of equipment, NN models can be classified as component-level, device-level, and system-level models, which is introduced in the previous research [24]. These models can be constructed by single NN only, called single-NN models. Then, some of them can be composed of multiple single-NNs, named multi-NN models. Finally, other models that consist of traditional EMT methods and NNs are called hybrid models.

A. NN Topology

To build a single-NN model for the power electronics digital twin, several stable and accurate NN topologies can be utilized, including traditional fully connected artificial NNs (ANNs), classical recurrent NNs (RNNs), long short-term memory NN (LSTMs), gated recurrent units NNs (GRUs), etc. Most power electronic ML digital twins require not long sequence prediction nor coarse prediction but near future prediction. Hence, the complex NNs (e.g., the transformer NN [27] and the informer NN [28]) for the long sequence (larger than 100 sequences) time-series prediction are unsuitable for our applications, considering the computational burden and sub-microsecond real-time execution requirements. According to comparisons between these NN methods in [24], the traditional ANN and classical RNN are highlighted with a simple and efficient structure, while GRU and LSTM may improve accuracy but cost more hardware resource.

Nevertheless, for models with time-series signal input, there will be some discrepancies in the specifics of the data processing structure of traditional ANN and classical RNN. As shown in Fig. 3 (a), the first line shows the simplest ANN model, forward neural network (FNN), which calculates the data at the current time-step based on the signal at the previous time-step. It can be expressed as the following function:

$$\mathbf{Y}^t = f\left(\mathbf{X}^{t-\Delta t}\right), \tag{18}$$

where $\mathbf{Y}^t$ and $\mathbf{X}^{t-\Delta t}$ are output and input signals, respectively. The EMT models similar to this ANN are displayed in Fig. 3 (b), showing how a typical closed-loop system simulation process works.

However, in many cases it is not possible to calculate the current time-step only from the previous time-step, because the derivative of the input signal is necessary, such as the $\frac{d\omega_r}{dt}$ and $\frac{d\theta_m}{dt}$ in equations 17. Then, the expression in this case can be rewritten as:

$$\mathbf{Y}^t = f\left(\mathbf{X}^{t-\Delta t}, \mathbf{X}^{t-2\Delta t} ... \mathbf{X}^{t-n\Delta t}\right), \tag{19}$$

where n means the number of previous time-step, is depended on the algorithm design. In the second line of Fig. 3 (a), the structure of ANN with multi-time-step ($n = 3$) inputs are shown, and this type of NN is named the time-series forward neural network (TSFNN).

TSFNN becomes exceedingly complicated and bloated as time-step n rises since TSFNN is fully connected ANN, and the computation matrix becomes massive. Compared with TSFNN, RNN can process each time-step with a smaller matrix and fewer parameters, and it can continue the procedure with the same structure and parameters to determine the final $\mathbf{Y}^t$. TSFNN will undoubtedly outperform RNN in hardware parallel computation, but RNN surpasses TSFNN the flexibility, usability, and accuracy. Although the RNN cells are run in serial, each cell can be calculated in parallel. The parameter class of TSFNN increases by n^3, while the RNN just needs to be executed n times without any parameter number increase, which makes RNN easier to be trained than TSFNN.

RNNs are classified into two types: dependent RNN (DRNN) and independent RNN (IRNN). The distinction is whether or not the historical information $\mathbf{H}$ is shared by multiple RNN computation processes. For each RNN cell, its output $\mathbf{Y}^t$ and $\mathbf{H}^t$ can be calculated as:

$$\left\{\mathbf{Y}^t, \mathbf{H}^t\right\} = f\left(\mathbf{X}^{t-\Delta t}, \mathbf{H}^{t-\Delta t}\right). \tag{20}$$

The IRNN utilizes the input of the previous n time steps for each computation, calculates all historical information from $\mathbf{H}^{t-n\Delta t}$ to $\mathbf{H}^t$, and outputs $\mathbf{Y}^t$ at time t in the last cell, shown

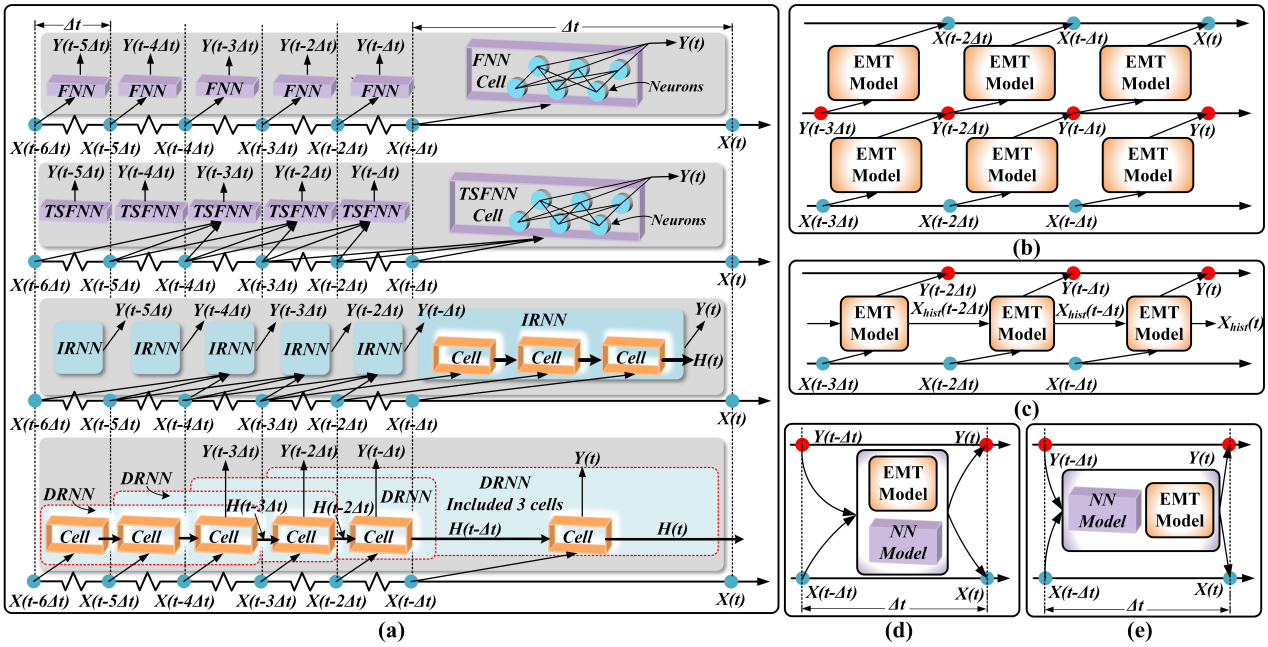

Fig. 3. Structure of time serried emulation algorithm: (a) NNs models; (b) EMT models without history information; (c) EMT models with history information; (d) ML-EMT parallel hybrid model; (e) ML-EMT serial hybrid model.

as the third line in Fig. 3 (a). The IRNN process executes the RNN cell n times each run, whereas the DRNN only runs the cell once for $\mathbf{Y}^t$. DRNN will output the historical information $\mathbf{H}$ at each time-step, and all DRNNs share this $\mathbf{H}$. Therefore, the DRNN only needs to execute the cell once rather than repeating it n times in a time-step, shown as the fourth line in Fig. 3 (a). Clearly, the DRNN is more efficient in execution (it no longer has to perform n cell loops in serial each time-step), which allows it to be executed faster than the IRNN and more accurate than TSFNN. However, because each computation output is conducted independently, and the sequence of input data can be shuffled for improved generality before training, the IRNN is more resilient and easier to be trained. Then, as illustrated in Fig. 3 (c), EMT discrete model has a similar structure to the DRNN, and undoubtedly both ML and EMT models can perform the same tasks.

Overall, different ML models have different trade-offs in terms of complexity, scalability, and performance. Although complex models like LSTM may achieve better performance, they require much more computational resources and much longer training time. The choice of ML model may also depend on the hardware architecture, its limitations, and system communication bandwidth. Depending on the specific requirements and constraints of the system, a simpler ML model may still be preferable, especially in the context of real-time inferencing applications for modeling high-complexity hierarchical systems.

B. Multi-NN Models

In [24], multi-layer single-NN models can simulate the system-level modeling function relatively accurately. Increasing the number of NN layers or recycling structure can reduce the number of neurons in each layer and improve accuracy but sacrifice execution speed. Hence, RNN is a compromise solution, taking accuracy, resource consumption, and training speed into account. The research [29] concludes that a single-layer neural network with enough neurons is sufficient to simulate any functions. To further enhance emulation speed, a single-layer FNN, the fastest method, should be utilized to replace RNN in some cases.

Typically, a piece of equipment is represented with a single NN so that the full digital twin system may be assembled like Lego bricks. Following the construction of each piece of equipment, the entire microgrid system is created by single-NN models block-by-block. However, there will always be some equipment that cannot be efficiently simulated by a single-NN model. In [25], the modular multilevel converter modeling (MMC) was used as an example: if the entire MMC was treated as a single-NN model, it would be inefficient and cumbersome. There are several unrelated variables and signals in the same model. Then, by isolating and operating the submodules in parallel, the operation efficiency and latency may be improved. This notion is vital in multi-NN models: the MMC submodule may be treated as a NN model. The whole MMC network is made up of tens of hundreds of NN submodule models as well as the main circuit NN model.

Besides, the equipment can be decomposed into different units based on their functions, and various types of NN can be utilized to model these units accordingly (e.g. PMSM multi-NN models in [24] and SG multi-NN models in section IV). This method mathematically decomposes a huge matrix into block matrices and allows each to compute in parallel, which is another critical reason for the application of multi-NN models instead of single-NN models.

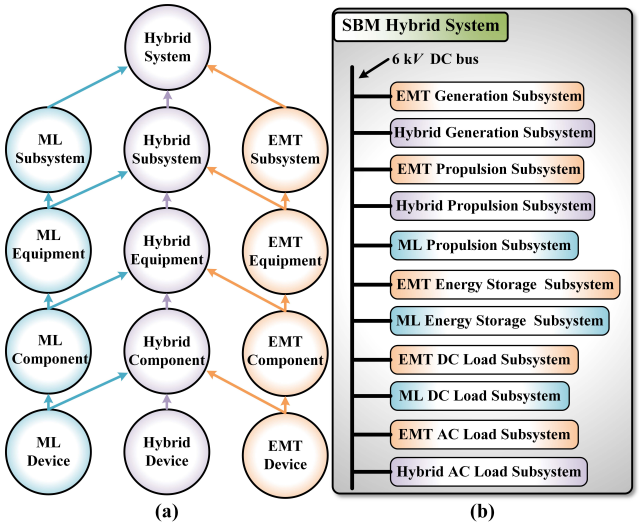

Fig. 4. Collection and classification of digital twin for SBM: (a) hierarchy; (b) classification based on function.

C. Hybrid Models

Undeniably, almost every piece of equipment in the power system can be modeled as NN models, whether single-NN or multi-NN models, which has been discussed in our research [24], [25]. However, it makes a simple problem complicated when only NN models are applied without traditional models because sometimes traditional EMT methods can efficiently work out the linear models. Compared with conventional equations-based methods, NN-only models may waste hardware resources and cause a heavy computational burden when applied to replace simple linear multiple equations in traditional ways. What is worse, it also results in unnecessary latency and hardware resource consumption during training processes. The modeled objective can be divided into different parts by empirical knowledge, where suitable NN or traditional methods will be decided to model based on the characteristics and functions of these parts. When both the NN and traditional methods are utilized and combined to build the models, these hybrid ML-EMT based models can take advantage of both modeling skills, which is the interdisciplinary utilization of power electronics and ML. Inside the hybrid ML-EMT based model, these NN and EMT parts can be connected in parallel (shown in Fig. 3 (d)) or in serial (shown in Fig. 3 (e)), respectively. The hybrid model can provide a higher degree of accuracy and efficiency in comparison to EMT models, especially in scenarios where the parameters of the models are unknown or may change due to specific operating conditions. When compared with the multi-NN or all-NN model in [24], this method utilizes the EMT model for simple partial calculation in order to reduce calculation burden and guarantee generality, stability, and accuracy. With the strengths of each approach, hybrid models can overcome the limitations of individual models and achieve superior results in practical scenarios.

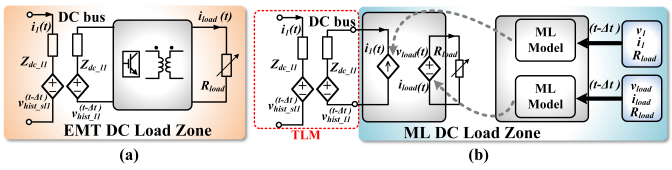

Fig. 5. DC load zone digital twin: (a) EMT model; (b) ML model.

IV. SBM Hybrid System Emulation

The ML models can be connected in series or parallel in SBM system, and their data ports could be different when they link to the traditional model and ML models themselves, respectively. System-level, subsystem-level, equipment-level, and device-level hybrid models, shown in Fig. 4 (a), are the foremost concern in this section since the component-level is pre-discussed in [24]. The details are given as: 1) The hybrid system includes many subsystem-level ML, EMT, and hybrid models; 2) The hybrid propulsion subsystems are compared with ML and EMT subsystems to demonstrate the topology and equipment differences; 3) The equipment-level hybrid SG model is constructed with ML and EMT parts, which are decomposed by function; 4) The device-level SiC IGBT hybrid model is separated into two parts, the EMT steady-state and ML transient, to improve the execution efficiency.

A. Hybrid System

Fig. 4 (b) depicts a hybrid system made up of subsystems of various functional categories. The same subsystem (e.g., propulsion subsystem) can be built as a classic EMT model, a hybrid model, or just an ML model. As an instance, the entire EMT DC load subsystem (illustrated in Fig. 5 (a)) is modeled as an ML model (illustrated in Fig. 5 (b)). The output of the ML subsystem are the load side voltage and bus side current, and the ML model does not require to calculate the internal signals. Because the process of ML model does not calculate the voltage and current of each switch and node in the subsystem but simply concentrates on the input and output of the entire subsystem, this technique may considerably conserve hardware resources and accelerate the emulation. The EMT subsystem can be utilized in conjunction with the ML model. When several DC load subsystems are presented, one or two EMT models can be utilized to derive the precise theoretical voltage and current of the internal nodes. Other similar subsystems can replace the whole subsystem with the ML model, which aids in reaching the optimal point while accounting for computational accuracy, resource consumption, and latency.

B. Hybrid Subsystem

Fig. 6 (a) depicts the propulsion subsystem, which incorporates ML, EMT, and hybrid models, and according to Fig. 4 (a), the subsystem consists of equipment. This section will utilize the propulsion system as an example to demonstrate the construction of the hybrid subsystems. Fig. 6 (a) shows the distinction between the ML (turquoise block), EMT (orange block), and hybrid (purple block) subsystems. Hybrid

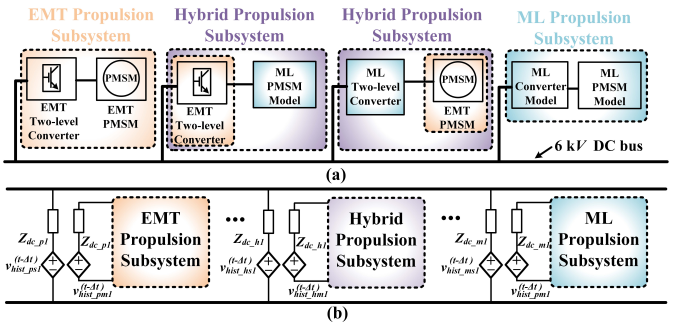

Fig. 6. Propulsion subsystem: a) general topology; b) equivalent circuit.

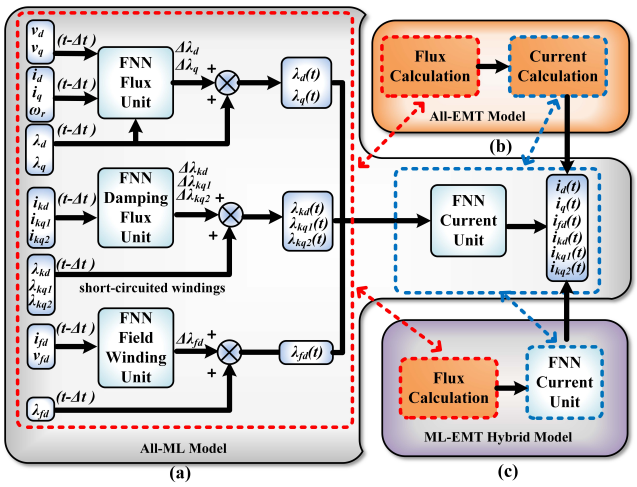

Fig. 7. SG modeling structure: (a) ML model; (b) EMT model; (c) hybrid model.

subsystems can be constructed to take advantage of unique benefits from ML and EMT models. The hybrid subsystem also has several possible combinations: one is made up of ML TLC equipment-level models and the EMT PMSM equipment-level model, while the other is composed of the EMT TLC equipment-level model and the ML PMSM equipment-level model. Section V will provide the comparative results.

C. Hybrid Equipment

To conserve resources and accelerate procedures, it is required to create a hybrid model that employs both conventional and ML methodologies, such as the SG model. Fig. 7 illustrates the hybrid equipment composition method: some use conventional computer models, some use ML models, and others employ multidisciplinary knowledge and experience to complete the modeling and fully utilize the benefits of these approaches. The EMT model calculation procedure is shown in Fig. 7 (a), where the flux linkage is computed first, followed by the current. The ML model using the ML technique for all calculation operations is shown in Fig. 7 (b), while the hybrid ML-EMT based model is displayed in Fig. 7 (c).

The classical Euler method for calculating flux linkage is straightforward and can readily deconstruct and compute each flux linkage value in parallel. The formula calculates the more straightforward linear multiplication portion, which does not require a significant quantity of data to train the NN. Furthermore, these parallel linear formulas are simple to implement by utilizing hardware parallelism, which has a slightly faster execution than ML approaches. An ML model can be used for the second phase, updating the current by inferencing the system dynamics. Based on the standard paradigm, the NN may eliminate matrix inversion to significantly increase execution speed. Traditional matrix inversion is challenging to hardware-accelerated parallel computing, whereas NN matrix multiplication and addition can be easily hardware accelerated.

Nevertheless, when the matrix elements are unaltered, this process of matrix inversion can be applied once and reused to improve execution efficiency. On the other hand, the NN training procedure is analogous to discovering a general approximation solution to the equation system in advance, which may be instrumental for hardware acceleration.

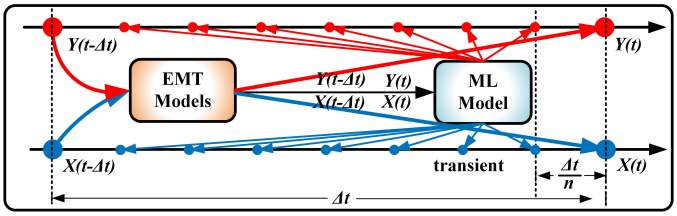

Fig. 8. Device-level ML-EMT hybrid IGBT model.

D. Hybrid Device

The silicon carbide (SiC) insulated-gate bipolar transistor (IGBT) model in [24] [25] employs two NNs: one for the steady-state waveform (a system-level model with 1 μs time-step) and another for the transient waveform (a device-level model with 50 ns time-step). The system-level model can achieve faster execution than the device-level model by utilizing reduced nodes to solve the circuit equations. Thus, the employment of the EMT steady-state portion and the ML transient hybrid model can minimize training time, training data collecting, and training hardware resource consumption while maintaining the original system-level execution efficiency and device-level accuracy. As shown in Fig. 8, the resistive EMT model calculates steady-state waveforms, which can adapt to different, larger time-steps while not being complicated to achieve. Concurrently, the ML model is only trained for transient waveforms with high accuracy and less hardware resource consumption. The transient model will output multiple data points in a time-step cycle to achieve the simulation of tiny time-step waveforms.

E. Selection of EMT, ML, and EMT-ML Hybrid Models

The selection of different models is often based on practical needs, and will usually take the following principles into consideration:

1) In situations where high-frequency transients and high accuracy events need to be captured, the EMT model is preferable. Compared to the ML model, the EMT model is more straightforward and simpler as it relies on empirical knowledge. On the other hand, the ML model may be more

complex and require heavy training cost and high hardware resource consumption to achieve high precision. Therefore, the EMT model is often given higher priority when the system stability relies on an ultra-high accuracy model.

2) To improve emulation accuracy and efficiency, ML models are ideal for capturing complex nonlinear interactions between different components of a comprehensive system. These models are especially useful in real-time inferencing scenarios. They can be employed when model parameters are unknown or may change due to different operating conditions like temperature and humidity, as long as input and output data are readily available. In contrast, traditional approaches for solving nonlinear component in EMT may require more iterations to handle, resulting in significant latency and challenging simulation criteria. On the other hand, EMT calculation may result in inaccuracies due to the misuse of the large time-step.

3) Hybrid models that combine the benefits of EMT and ML models can provide superior results in real-time hardware emulation. Such models are designed based on empirical knowledge and specific requirements for the intended application. For instance, the device-level IGBT model can benefit from a hybrid approach by utilizing the EMT model to ensure the stability and accuracy of the system-level working conditions while incorporating the ML model to analyze and output device-level transient waveforms. This approach can provide a higher degree of accuracy and efficiency in comparison to traditional methods, especially in scenarios where the parameters of the models are unknown or may change due to specific operating conditions. By leveraging the strengths of each approach, hybrid models can overcome the limitations of individual models and achieve superior results in real-world scenarios.

V. FPGA-Based Digital Twin Implementation for Real-Time SBM HIL Emulation

The conventional EMT formula-based model is straightforward to implement on hardware, and the calculation process is unrolled and pipelined as much as possible utilizing FPGA parallel computing. The ML model requires parameter gathering and training before referencing the system dynamics for HIL application, which is discussed further in this section: 1) Data processes and hyperparameters for offline training; 2) Details for tuning the parameters for models; 3) Hardware resource consumption and latency introduced by the selection of the methods and parameters.

A. Hyper-Parameters and Design for Training

To enhance the versatility, scalability, and performance of the ML model, it is crucial to utilize diversified work-condition data during training. To this end, a large amount of data has been collected on the modeling object in various working situations. The acquired data is split into training and testing sets, with 70% used for offline training and the remaining 30% used for validation and testing. For example, when training the PMSM motor model, various combinations of operating conditions with different speeds and torques were utilized, such as (0.5 p.u. T_e, 1 p.u. w_r), (1.5 p.u. T_e, 0.8 p.u. w_r), and (1.2 p.u. T_e, 1.2 p.u. w_r), along with steady-state and some dynamic data (such as maintaining a speed of 1 p.u. while reducing torque from 1 p.u. to 0). The test data were selected from other reasonable speeds and torques that were not included in the aforementioned training operating conditions (e.g., dynamic changes in torque from 0.8 p.u. to 1.2 p.u. at a speed of 0 to 1.2p.u. or static operating conditions with 1.6 p.u T_e and 0.4 p.u. w_r). The normalization or standardization of the data has a significant impact on the training results of the models.

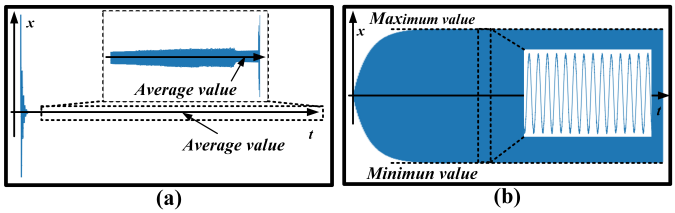

Fig. 9. Datasets utilization for: (a) standardization; and (b) normalization.

In this work, the primary application of the ML model is the replacement or enhancement of EMT process. While online training of ML models can enhance their accuracy and make them more realistic in terms of the physical models [25], it involves significant resource consumption, and makes itself unsuitable for SBM systems that encompass various levels. Therefore, this paper primarily employs offline training ML models. Additionally, the ML model is predominantly utilized for modeling EMT hybrid SBM systems instead of relying entirely on ML modeling. The EMT model and system are given priority, and the corresponding ML model is built according to their specific requirements. The most commonly compatible method involves training the ML model into a nonlinear controlled voltage source or current source, which has been proven to achieve high accuracy and generalization performance of the proposed design.

The standardization can focus on most data changes while ignoring a limited number of extreme values. Then, these datasets are better suited for normalization when the data varies within the steady-state range, where maximum and minimum values may be modified manually, which is beneficial for preprocessing in conjunction with professional experience and expertise. Fig. 9 shows the examples suitable for standardization and normalization.

$$standardization: x' = \frac{x - mean(x)}{std(x)}, \tag{21}$$

$$normalization: x' = \frac{x - min(x)}{max(x) - min(x)}, \tag{22}$$

where $mean(x)$, $std(x)$, $max(x)$, and $min(x)$ represent the average value, standard deviation, maximum, and minimum of dataset x, respectively.

After normalization or standardization, the training data will be sampled out of order and sent to the offline training program. During each epoch training, the parameter is optimized 50,000 times, and the training is repeated for 100 epochs to get the tuned weights and biases of NNs. Adam algorithm [30], utilized in this work, is a training NN method that automatically modifies the learning rate in comparison to

the classical stochastic gradient descent algorithm. The mean absolute error and the mean squared error have comparable impacts when calculating prediction and reference error [24].

$$MAE: e = \sum_{i=1}^{n} \frac{\left|y_i^{pre} - y_i\right|}{n}. \tag{23}$$

Designing a hybrid model requires expertise and experience to determine the optimal utilization of EMT modeling and ML methods, considering factors like computation speed, latency, accuracy, resource consumption, and interface compatibility. While hybrid models offer partial analytical insights, they outperform solely NN models by incorporating both analytical and ML components, although thorough testing is necessary to ensure their general applicability. To obtain the optimal hyperparameters for the ML model, a comprehensive analysis of the SBM was conducted, taking its specific characteristics and requirements into consideration. A trial-and-error approach was then adapted to identify the hyperparameters that yielded the best performance for the model. Regarding the consistency of hyperparameters across different cases and time-steps, it should be noted that the selection of hyperparameters is a case-specific process. Different cases and time-steps may necessitate different hyperparameters to achieve the desired level of accuracy and reliability. Thus, careful consideration was given to the selection of hyperparameters for each case and time-step to ensure optimal performance of the ML model.

B. Parameters for Models

Previous research [24] [25] have compared a reasonably workable range for the number of hidden layers and neurons in the equipment-level NN model: For all NN models, the layer number is 1 or 2; The number of neurons in the hidden layer range from 10 to 30, and the RNN sequence size is between 3-5. Parameters within this range not only ensure the accuracy of the output but also reduce hardware resource consumption and latency as much as possible. The consideration of NN model parameters at subsystem-level and device-level is hence the primary emphasis of this work. The hidden layer and hidden size should still be in the previously indicated range for subsystem-level IRNN or GRU models, but the sequence size should be increased by at least 20. This is due to the necessity of output prediction being based on the comparatively lengthy time of input signals. The enormous number of output data points for device-level models necessitates a considerable increase in hidden size. Hence, the hidden size for the IGBT transient model is 25, 30, 40, 50, and 60 for 10 ns, 20 ns, 30 ns, 40 ns, and 50 ns time-step, respectively. ML models for subsystem-level DC load zones offer insights into how parameters can be manipulated. For example, the time-step of the ML model can be varied between 10, 15, 20, 25, and 30. Additionally, the number of neurons in the model can be increased from 20 to 40, and the number of layers can be decreased from 3 to 1. The training process also involves adjustments to the learning rate and the number of training epochs based on feedback results. Moreover, hardware resources can be controlled to optimize the training and implementation of the model.

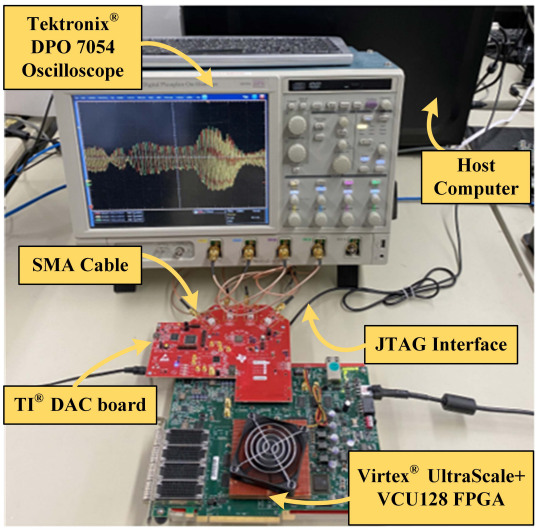

Fig. 10. Hardware setup of the real-time SBM digital twin.

TABLE I
MODEL HARDWARE RESOURCE CONSUMPTION ON FPGA

Device	BRAM	DSP	FF	LUT	Latency
EMT DC Load Zone	2%	13%	3%	15%	7.63 μs
DRNN DC Load Zone	5%	4%	1%	3%	0.39 μs
GRU DC Load Zone	11%	7%	3%	5%	0.83 μs
EMT Converter	0%	1%	≈0%	1%	0.12 μs
IRNN Converter	4%	6%	≈0%	3%	0.64 μs
EMT PMSM	≈0%	4%	2%	1%	0.41 μs
Multi-FNN PMSM	0%	3%	1%	2%	0.84 μs
EMT SG	1%	7%	2%	9%	4.84 μs
Multi-FNN SG	2%	17%	2%	11%	0.96 μs
Hybrid SG	1%	8%	1%	5%	0.96 μs
EMT Resistive IGBT	≈0%	≈0%	≈0%	≈0%	0.1 μs
FNN IGBT($10ns \times 150$)	0%	2%	≈0%	33%	3.88 μs
FNN IGBT($20ns \times 75$)	0%	6%	≈0%	40%	0.6 μs
FNN IGBT($50ns \times 30$)	0%	≈0%	≈0%	9%	0.82 μs
Available	4332	9024	2607k	1304k	

C. Hardware Platform

In Fig. 10, the hardware connection is displayed. As can be shown in Table I, the subsystem-level DRNN model only requires 4% DSP and 3% LUT, whereas the subsystem-level EMT DC load zone model costs 13% DSP and 15% LUT resources, incurring 7.63 μs latency. Compared with DRNN models, the subsystem-level GRU model increases resource consumption (11% BRAMs, 7% DSP, and 5% LUT) and slightly improves the accuracy with long-term prediction ability. The EMT PMSM and TLC models are less expensive than ML models in terms of the equipment-level model in electromagnetic domain only. However, ML models can predict multi-physics domain details, which is close to the actual working conditions. Then, the hybrid equipment-level SG model demonstrates the transdisciplinary modeling benefit of traditional EMT and newer ML methods. The execution of this hybrid model is completed in 1 μs as opposed to

TABLE II
MAE COMPARISON OF TLC-PMSM SUBSYSTEMS

Output	I_{DC}	T_e	ω_r
EMT-TLC-ML-PMSM	1.41%	0.91%	0.14%
ML-TLC-EMT-PMSM	0.21%	0.09%	≈0%
ML-TLC-ML-PMSM	1.49%	0.93%	0.14%

TABLE III
MAE COMPARISON OF SG AND IGBT

Output	V	I
Hybrid SG	0.41%(V_{AC})	0.41%(I_{AC})
EMT Resistive IGBT(ON)	14.14%	11.26%
EMT Resistive IGBT(OFF)	10.13%	15.68%
FNN IGBT(10 $ns \times 150$)(ON)	0.09%	0.27%
FNN IGBT(10 $ns \times 150$)(OFF)	0.19%	0.30%
FNN IGBT(20 $ns \times 75$)(ON)	0.07%	0.24%
FNN IGBT(20 $ns \times 75$)(OFF)	0.28%	0.23%
FNN IGBT(50 $ns \times 30$)(ON)	1.20%	0.99%
FNN IGBT(50 $ns \times 30$)(OFF)	2.83%	1.86%

the EMT model's 5 μs with approximate hardware resource consumption. When compared with multi-FNN model, the hybrid model consumes less hardware resource but still keep the same execution efficiency. The execution time for resistive switching IGBT EMT model is 100 ns for a single time-step without detailed switching transients while the FNN-based IGBT model can predict multiple device-level time-steps transients by only one referencing operation with reasonable hardware resource consumption. The multiple device-level time-steps are given: 1) 10 ns multiply by 150 points; 2) 20 ns multiply by 75 points; 3) 50 ns multiply by 30 points.

VI. RESULTS AND DISCUSSION

This section compares the emulation results of ML, EMT, and hybrid models, whose system-level, subsystem-level, and equipment-level results are validated by PSCAD/EMTDC® while device-level results are validated by SaberRD®.

Fig. 11 depicts the subsystem-level EMT model, as well as the DRNN and GRU models with varying sequence lengths, where the current is increased from 10 A to 20 A, then to 30 A. With the same sequence length, the DRNN and GRU models have similar effects. The larger sequence length will result in closer outputs to EMT model. Although the GRU model has a slightly better performance than the DRNN model, the mean absolute error of both models is below 2% when the sequence size is more than 30 (1.6% for DRNN and 1.2% for GRU).

Fig. 12 contrasts ML, EMT, and two hybrid subsystems, where the PMSM mechanical torque changes with its load to 0 p.u. at 1 s, 0.8 p.u. at 2 s, and 0.5 p.u. at 3 s, while the PMSM's referenced speed increases from 0.5 p.u. to 0.7 p.u. at 4 s and lowers to 0.3 p.u. at 5 s. Although the ML model may introduce a designated training target error, it can take into consideration features of complex physical working conditions with the capability of utilizing adaptive parameter updates. The MAEs of ML and ML-EMT hybrid subsystems' outputs are all below 2% when compared to the EMT subsystem. As shown in Table II, the MAEs of the EMT-TLC-ML-PMSM subsystem are 1.41% for I_{dc}, 0.91% for T_e, and 0.14% for w_r, while the MAEs of the ML-TLC-EMT-PMSM subsystem are 0.21% for I_{dc}, 0.09% for T_e, and approximately 0 for w_r. As for the ML-TLC-ML-PMSM subsystem, its MAEs are close to the EMT-TLC-ML-PMSM subsystem, those are 1.49% for I_{dc}, 0.93% for T_e, and 0.14% for w_r. Furthermore, the hybrid equipment-level SG model flexibly combines the benefits of the EMT and ML models. As shown in Fig. 13, the hybrid model's resource usage is comparable to that of the EMT model with reduced latency, while the steady-state errors are less than 1%, as demonstrated in Table III. These results indicate the potential of the proposed hybrid model to provide improved performance and efficiency in power system digital twin applications.

Fig. 14 is for the validation of different device-level models. Since real-time IGBT waveform output of 10 ns and 20 ns is challenging to achieve with the EMT method, the SaberRD® waveform is used for comparison. The hybrid model can output a waveform close to SaberRD®: The steady-state waveform is generated by the EMT resistive model because ML model can not overweight EMT model in terms of hardware resource consumption and latency for a single time-step, while the ML model outputs the switching on and off transient waveforms because ML model can inference multiple time-steps waveform to achieve a faster execution overall. As shown, the real-time emulation with the tiny time-step is realized, revealing five cycles of harmonics. As shown in Table III, the EMT resistive IGBT model results in unacceptable MAEs during the switch on and off transients, while the FNN IGBT transient models perform much better. For instance, the FNN IGBT model with (50 ns $\times$30 outputs) can limit the MAEs to less than 3% and achieve the results which mimic the SaberRD® device-level transient waveform, as shown in Fig. 14. Furthermore, increasing the number of output points (also increasing the number of neurons) can lead to excellent performance, with all turn-on and turn-off MAEs for V and I being less than 0.5%. The models with (20 ns $\times$75 points) and (10 ns $\times$150 points) show similar high accuracy.

VII. CONCLUSION

This paper proposes hybrid ML-EMT based real-time models of the SBM digital twin for FPGA-based emulation under different operation scenarios. The entire SBM is segregated according to function, then the typical EMT modeling calculation method is introduced, and following the various NN topologies are comprehensively studied. Finally, the ML and EMT approaches are integrated, and the purpose of the hybrid model is explained at the system, subsystem, equipment, and device levels. The obtained results confirm the benefits of the real-time hybrid modeling method on the Xilinx® VCU128 hardware platform and are validated by PSCAD/EMTDC® (system-level, subsystem-level, and equipment-level) and SaberRD® (device-level): 1) The real-time hybrid SBM digital twin emulation is implemented to conserve hardware resources while maintaining system details and accuracy; 2) The hybrid

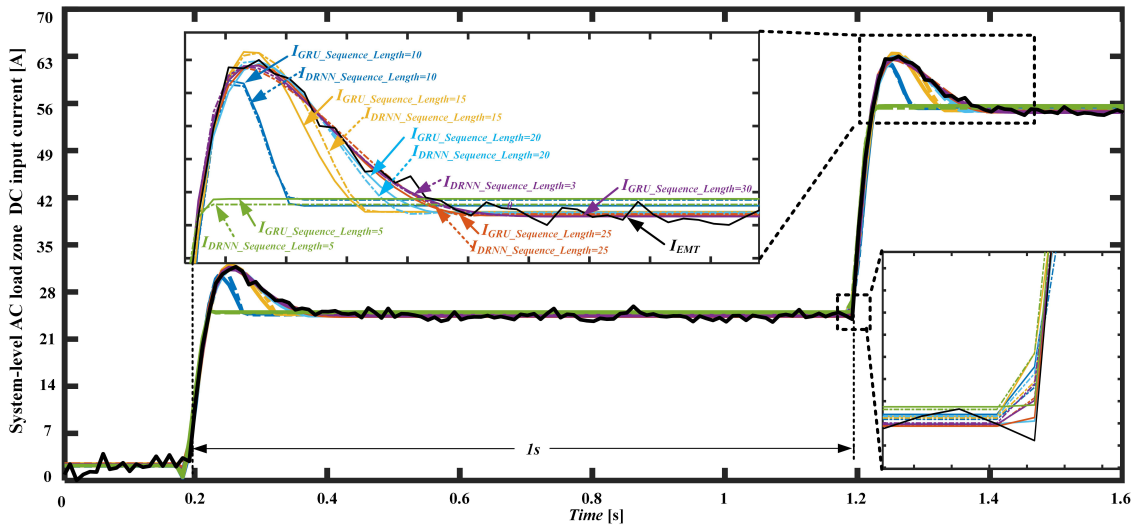

Fig. 11. EMT and ML subsystem-level DC load zone results.

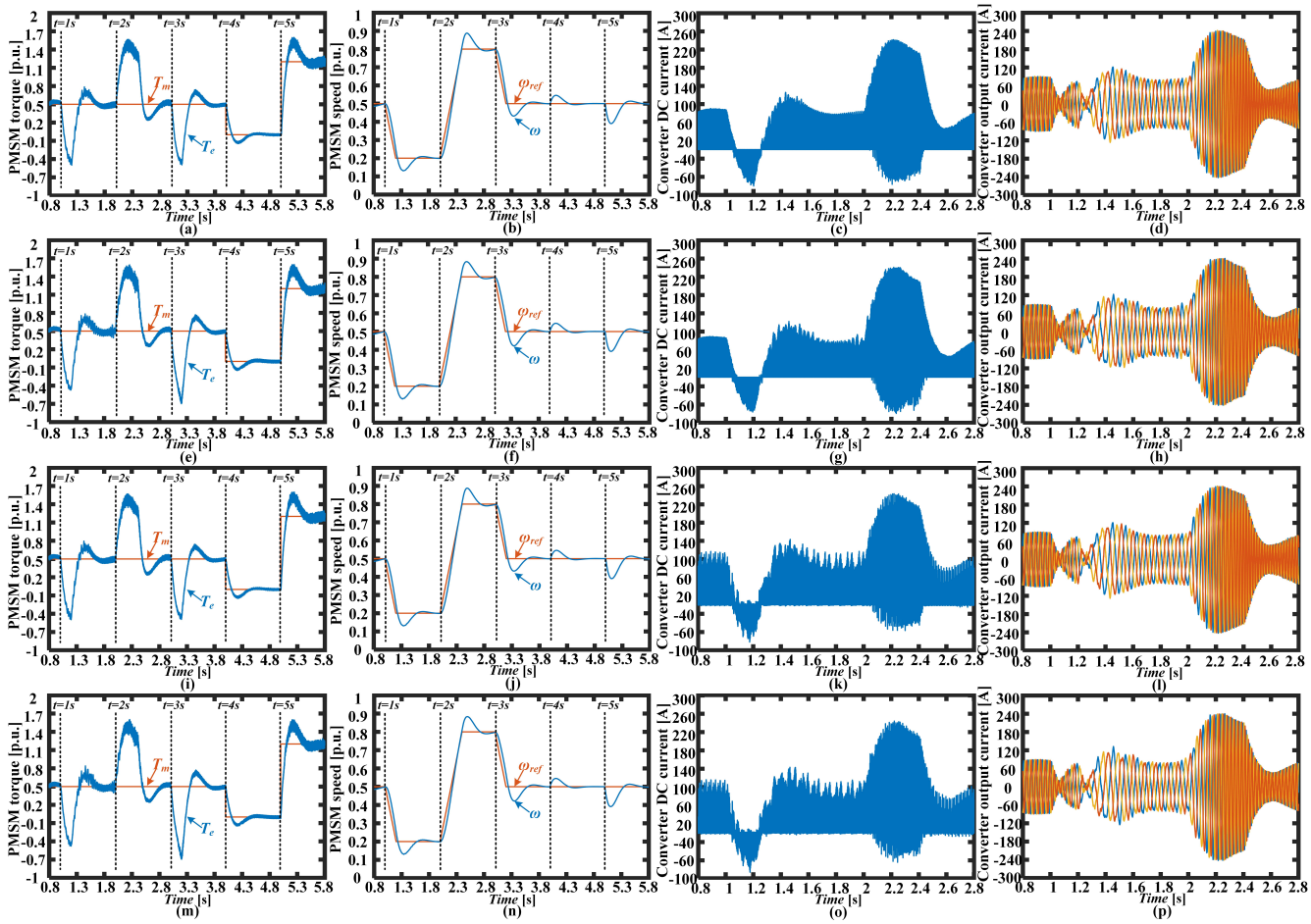

Fig. 12. ML, EMT, and hybrid propulsion subsystems results (top line: EMT-TLC-EMT-PMSM; seconde line: EMT-TLC-ML-PMSM; third line: ML-TLC-EMT-PMSM; bottom line: ML-TLC-ML-PMSM): (a), (e), (i), and (m) PMSM torque; (b), (f), (j), and (n) PMSM speed; (c), (g), (k), and (o) TLC DC current; (d), (h), (l), and (p) TLC output three-phase current.

subsystem-level modeling can make the entire model closer to the real physical environment due to the data-driven ML part; 3) The hybrid equipment-level model can not only leverage the ML model's general and flexible structure to execute parallel computing with high execution efficiency, but it can also maintain the EMT model's stability and high calculation

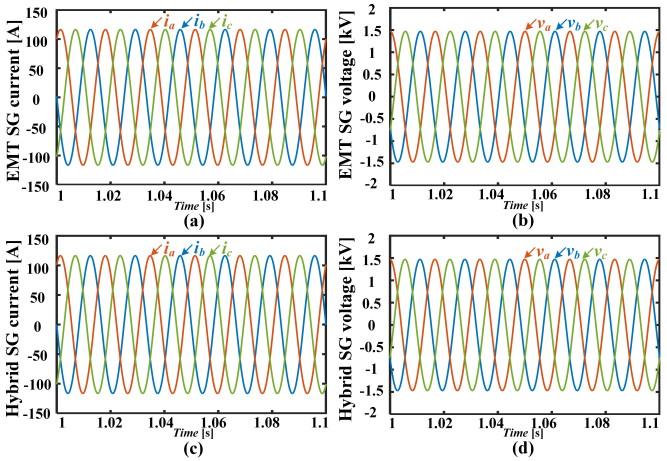

Fig. 13. EMT and hybrid equipment-level SG steady-state results.

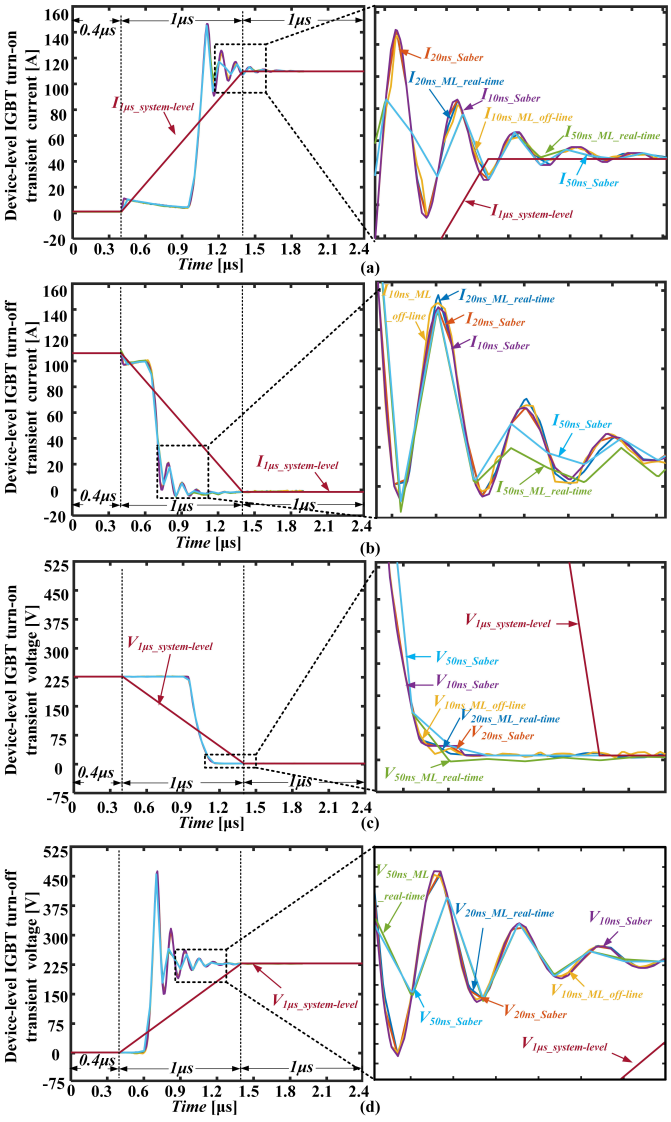

Fig. 14. Hybrid device-level SiC IGBT transient results: (a) turn-on current; (b) turn-off current; (c) turn-on voltage; and (d) turn-off voltage.

accuracy (less than 1% error); 4) The hybrid device-level model can enable real-time emulation with ultra-small time-steps (20 $n.s$). The hybrid ML-EMT based modeling aids in reaching the best trade-off point with consideration of real-time execution efficiency, adaptability, generality, variety, and accuracy.

REFERENCES

[1] Review of Maritime Transport 2021. [Online]. Available: https://unctad.org/system/files/official-document/rmt2021_en_0.pdf

[2] 2018 Initial IMO GHG Strategy. [Online]. Available: https://wwwcdn.imo.org/localresources/en/OurWork/Environment/Documents/Resolution%20MEPC.304%2872%29_E.pdf

[3] N. Doerry, J. Amy and C. Krolick, "History and the status of electric ship propulsion, integrated power systems, and future trends in the U.S. Navy," *in Proc. IEEE.,* vol. 103, no. 12, pp. 2243-2251, Dec. 2015.

[4] M. S. Agamy et al., "A high power medium voltage resonant dual active bridge for MVDC ship power networks," *IEEE Trans. Ind. Electron.,* vol. 5, no. 1, pp. 88-99, Mar. 2017.

[5] B. Zahedi and L. E. Norum, "Modeling and simulation of all-electric ships with low-voltage DC hybrid power systems" *IEEE Trans. Power Electron.,* vol. 28, no. 10, pp. 4525-4537, Oct. 2013.

[6] B. Ning, Q. -L. Han and L. Ding, "Distributed finite-time secondary frequency and voltage control for islanded microgrids with communication delays and switching topologies" *IEEE Trans. Cybern.,* vol. 51, no. 8, pp. 3988-3999, Aug. 2021.

[7] H. Xin, L. Zhang, Z. Wang, D. Gan and K. P. Wong, "Control of island AC microgrids using a fully distributed approach" *IEEE Trans. Smart Grid,* vol. 6, no. 2, pp. 943-945, Mar. 2015.

[8] W. Chen et al., "DC-distributed power system modeling and hardware-in-the-loop (HIL) evaluation of fuel cell-powered marine vessel" *J. Emerging Sel. Top. Ind. Electron.,* vol. 3, no. 3, pp. 797-808, Jul. 2022.

[9] C. Liu, R. Ma, H. Bai, Z. Li, F. Gechter and F. Gao, "FPGA-based real-time simulation of high-power electronic system with nonlinear IGBT characteristics" *IEEE J. Emerging Sel. Top. Power Electron.,* vol. 7, no. 1, pp. 41-51, Mar. 2019.

[10] T. Liang and V. Dinavahi, "Real-time system-on-chip emulation of electrothermal models for power electronic devices via hammerstein configuration" *IEEE J. Emerging Sel. Top. Power Electron.,* vol. 6, no. 1, pp. 203-218, Mar. 2018.

[11] M. Milton, A. Benigni and J. Bakos, "System-level, FPGA-based, real-time simulation of ship power systems" *IEEE Trans. Energy Convers.,* vol. 32, no. 2, pp. 737-747, Jun. 2017.

[12] C. S. Edrington, M. Steurer, J. Langston, T. El-Mezyani and K. Schoder, "Role of power hardware in the Loop in modeling and simulation for experimentation in power and energy systems" *Proc. IEEE,* vol. 103, no. 12, pp. 2401-2409, Dec. 2015.

[13] A. Hasanzadeh, C. S. Edrington, N. Stroupe and T. Bevis, "Real-Time emulation of a high-speed microturbine permanent-magnet synchronous generator using multiplatform hardware-in-the-loop realization" *IEEE Trans. Ind. Electron.,* vol. 61, no. 6, pp. 3109-3118, Jun. 2014.

[14] T. Liang and V. Dinavahi, "Real-time device-level simulation of MMC-based MVDC traction power system on MPSoC," *IEEE Trans. Transp. Electrific.,* vol. 4, no. 2, pp. 626-641, Jun. 2018.

[15] Z. Huang and V. Dinavahi, "An efficient hierarchical zonal method for large-scale circuit simulation and its real-time application on more electric aircraft microgrid," *IEEE Trans. Ind. Electron.,* vol. 66, no. 7, pp. 5778-5786, Jul. 2019.

[16] X. Feng, K. L. Butler-Purry and T. Zourntos, "A multi-agent system framework for real-time electric load management in MVAC all-electric ship power systems," *IEEE Trans. Power Syst.,* vol. 30, no. 3, pp. 1327-1336, May 2015.

[17] S. Y. R. Hui and K. K. Fung, "Fast decoupled simulation of large power electronic systems using new two-port companion link models," *IEEE Trans. Power Electron.,* vol. 12, no. 3, pp. 462-473, May 1997.

[18] N. Lin and V. Dinavahi, "Behavioral device-level modeling of modular multilevel converters in real time for variable-speed drive applications," *IEEE J. Emerging Sel. Top. Power Electron.,* vol. 5, no. 3, pp. 1177-1191, Sept. 2017.

[19] A. Benigni and A. Monti, "A parallel approach to real-time simulation of power electronics systems," *IEEE Trans. Power Electron.,* vol. 30, no. 9, pp. 5192-5206, Sept. 2015.

[20] T. Cheng, N. Lin and V. Dinavahi "Hybrid Parallel-in-Time-and-Space Transient Stability Simulation of Large-Scale AC/DC Grids," *IEEE Trans. Power Syst.,* vol. 37, no. 6, pp. 4709-4719, Nov. 2022.

[21] Q. Liu, T. Liang and V. Dinavahi, "Deep learning for hardware-based real-time fault detection and localization of all electric ship MVDC Power system," *IEEE Open J. Ind. Appl.,* vol. 1, pp. 194-204, Oct. 2020.

[22] Y. Chen, M. G. Fadda and A. Benigni, "Decentralized load estimation for distribution systems using artificial neural networks," *IEEE Trans. Instrum. Meas.,* vol. 68, no. 5, pp. 1333-1342, May 2019.

[23] S. Zhao, F. Blaabjerg and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Trans. Power Electron.,* vol. 36, no. 4, pp. 4633-4658, Apr. 2021.

[24] S. Zhang, T. Liang and V. Dinavahi, "Machine learning building blocks for real-time emulation of advanced transport power systems," *IEEE Open J. Power Electron.,* vol. 1, pp. 488-498, Nov. 2020.

[25] S. Zhang, T. Liang, T. Cheng and V. Dinavahi, "Machine learning based modeling for real-time inferencer-in-the-loop hardware emulation of high-speed rail microgrid," *J. Emerging Sel. Top. Ind. Electron.,* vol. 3, no. 4, pp. 920-932, Oct. 2022.

[26] V. Dinavahi, and N. Lin, *Real-Time electromagnetic transient simulation of AC-DC networks,* Wiley-IEEE Press, 2021.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems 30,* 2017.

[28] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: beyond efficient transformer for long sequence time-series forecasting," *In Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 35, no. 12, pp. 11106-11115. 2021.

[29] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks,* 1989, 2(5), pp.359-366.

[30] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)* San Diego, CA, USA, May 2015, pp. 1–15

Songyang Zhang (S'20) received both the B.Eng. degree and M.Eng. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, Hubei, China, in 2017 and 2019 respectively. He is currently pursuing his Ph.D. degree in electrical and computer engineering department at the University of Alberta, Edmonton, Alberta, Canada. His research interests include machine learning, real-time simulation, power electronics and field programmable gate arrays.

Tian Liang (S'16, M'20) received the B.Eng. degree in electrical engineering from Nanjing Normal University, Nanjing, Jiangsu, China, in 2011, the M.Eng. degree from Tsinghua University, Beijing, China, in 2014, the Ph.D. degree in energy systems from the University of Alberta, Edmonton, AB, Canada, in 2020. He is now with RTDS Technologies Inc.. His research interests include real-time simulation of power systems, power electronics, artificial intelligence, field-programmable gate arrays, and system on chip.

Venkata Dinavahi (Fellow, IEEE) received the B.Eng. degree in electrical engineering from Visvesvaraya National Institute of Technology (VNIT), Nagpur, India, in 1993, the M.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Kanpur, India, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Ontario, Canada, in 2000. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada. He is a Fellow of the Engineering Institute of Canada. His research interests include real-time simulation of power systems and power electronic systems, electromagnetic transients, devicelevel modeling, large-scale systems, and parallel and distributed computing.