A Hybrid Machine Learning and Physics-Based Model for Quasi-Ballistic Nanotransistors

Qimao Yang and Jing Guo, *Senior Member, IEEE*

***Abstract*— We introduce a hybrid model that synergistically combines machine learning (ML) with semiconductor device physics to simulate nanoscale transistors. This approach integrates a physics-based (PB) ballistic transistor model with an ML model that predicts ballisticity, enabling flexibility to interface the model with device data. The inclusion of device physics not only enhances the interpretability of the ML model but also streamlines its training process, reducing the necessity for extensive training data. The model's effectiveness is validated on both silicon nanotransistors and carbon nanotube (CNT) field-effect transistors (FETs), demonstrating high model accuracy with a simplified ML component. We assess the impacts of various ML models—multilayer perceptron (MLP), recurrent neural network (RNN), and RandomForestRegressor (RFR)—on predictive accuracy and training data requirements. Notably, hybrid models incorporating these components can maintain high accuracy with a small training dataset, with the RNN-based model exhibiting better accuracy compared with the MLP and RFR models. The trained hybrid model provides significant speedup compared to device simulations and can be applied to predict circuit characteristics based on the modeled nanotransistors.**

***Index Terms*— Carbon nanotube (CNT) transistors, hybrid model, machine learning (ML), nanoscale transistors, silicon nanotransistors.**

I. Introduction

THE scaling of silicon transistors has led to metal–oxide–semiconductor field-effect transistors (MOSFETs) with a gate length in the 10-nm-scale regime. As silicon transistors near their scaling limits, exploration into transistors based on new nanoscale materials, such as carbon nanotubes (CNTs), graphene, and 2-D layered semiconductors, has intensified [1], [2]. Both silicon nanotransistors and nanotransistors based on new nanomaterials can operate in near-ballistic or quasi-ballistic transport regimes.

Transistor models have played an important role in semiconductor designs. Traditionally, transistor models are based on device physics. As the transistors scale down and new physical phenomena need to be incorporated, a larger amount of empirical fitting parameters have been induced. For example, the widely used Berkeley short-channel IGFET model (BSIM) [3] has hundreds of parameters and requires significant domain knowledge for parameter extraction. On the other hand, physics-based (PB) ballistic transistor models have a much smaller number of parameters, but a transistor does not operate perfectly at the ballistic limit. PB quasi-ballistic transistor models have also been developed [4], but these models have limited flexibility to accurately describe the device data.

Recently, machine learning (ML)-based models have emerged as a new tool for modeling transistor device characteristics [5], [6], [7], [8], [9]. Predominantly data-driven, these ML models harness extensive datasets to learn and predict transistor behaviors, displaying a promising aptitude for replicating detailed TCAD simulation results. Nonetheless, ML models in the physical domain face certain challenges. A challenging issue is their propensity to yield unphysical device characteristics, especially when extrapolating beyond the training data's scope. Furthermore, the complexity and effectiveness of these models are closely linked to the training data's volume and quality. Larger, more complex models require extensive datasets for optimal functionality. Conversely, smaller models, though less demanding of data, often struggle to fully capture complex data relationships. In addition, training these models, particularly with sparse or limited datasets, can be a painstaking process, necessitating meticulous tuning and validation to ensure accurate and reliable predictions. The choice of ML algorithm is another critical factor, as different algorithms may exhibit varying performance characteristics and sensitivities to the nature and quantity of training data.

In this work, we adopt a hybrid approach, merging the device physics knowledge of ballistic transistors with an ML methodology to interface the model with data. Through this approach, we demonstrate that the hybrid model can not only flexibly interface with nanotransistor data but also significantly alleviate the training complexity of the ML component, by simplifying the task. The hybrid physics-ML model is applied to two types of nanotransistors: a double-gated ultrathin-body (UTB) silicon field-effect transistor (FET) and a CNTFET. In semiconductor device technology, obtaining an accurate large dataset for ML model training is often

Manuscript received 9 April 2024; revised 15 May 2024; accepted 21 May 2024. Date of publication 11 June 2024; date of current version 23 August 2024. This work was supported by NSF under Grant 2203625. The review of this article was arranged by Editor S.-M. Hong. *(Corresponding author: Jing Guo.)*

The authors are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: qimao.yang@ufl.edu; guoj@ufl.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TED.2024.3408769.

Digital Object Identifier 10.1109/TED.2024.3408769

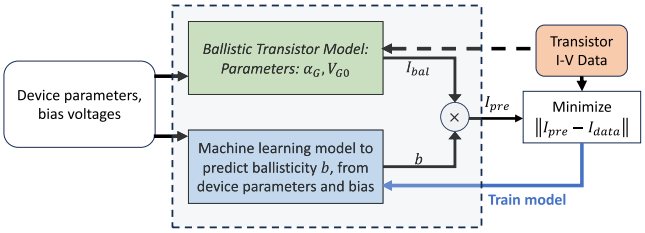

Fig. 1. Flowchart of the hybrid physics-ML model. The ballistic transistor model part is PB with two parameters extracted from data, which predicts ballistic transistor *I*–*V* characteristics. The ML model part predicts the ballisticity of a nanotransistor. The product of the ballistic current and the ballisticity gives the predicted transistor current. The model is trained by a cost function defined by the difference between the prediction and the data.

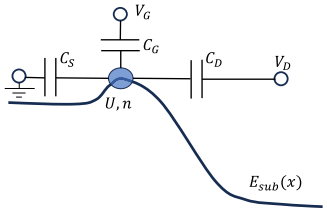

Fig. 2. Schematic of the ballistic transistor model. The top of the potential barrier is shown by the filled circle, with its potential energy as U and electron density as n. $E_{\text{sub}}(x)$ is the subband profile. The capacitances C_S, C_D, and C_G are shown schematically.

time-consuming and expensive. We also investigate the effects of ML model selection in training the hybrid model with limited data. The accuracy of various choices of the ML model components in the hybrid model is compared by varying and reducing the size of the training dataset. The results indicate the possibility of high model accuracy with small device training data by using the hybrid physics-ML device modeling approach.

II. Approach

The hybrid ML and PB device model for nanotransistors is discussed in Section II-A. The PB component, which is based on the ballistic transistor theory, is summarized in Section II-B. The ML model component is described in Section II-C. The training procedure of the hybrid physics-ML device model is described in Section II-D.

A. Hybrid ML and PB Model for Nanotransistors

The hybrid model proposed, as summarized in Fig. 1, combines a PB ballistic transistor model, which predicts the ballistic limit of the transistor, with an ML model that predicts ballisticity. The simple ballistic model captures the qualitative feature of the transistor device characteristics. For example, the source–drain current varies exponentially as a function of the gate voltage in the subthreshold region, but somewhere between linearly and quadratically in the above-threshold region. Such strongly nonlinear, bias-dependent, and orders of magnitude variation of the device *I*–*V* data impose challenges to ML regression but are captured by a PB model. The PB ballistic transistor model also provides predictive and interpolative power. On the other hand, the ballisticity of a nanotransistor is hard to accurately predict from device physics. Although PB quasi-ballistic transistor models have been developed, the important parameters in the model are very difficult to determine accurately. The hybrid model infuses device physics captured by the ballistic transistor model with the data-driven ML modeling approach for ballisticity.

B. Ballistic Transistor Model

Ballistic transistor theory computes the ballistic performance limits of a transistor, which assumes that carrier transport is ballistic and the source and drain contacts are ideal [10], [11]. Ballistic transistor models have been developed for various types of nanotransistors, such as silicon transistors and CNT transistors. Here, we summarize the ballistic transistor model below for completeness.

As schematically shown in Fig. 2, the potential at the top of the channel barrier U is expressed as follows:

$$U = -q[\alpha_G(V_G - V_{G0}) + \alpha_D V_D] + \frac{q^2 n}{C_\Sigma} \tag{1}$$

where $C_\Sigma = C_S + C_D + C_G$ and $\alpha_G = C_G/C_\Sigma$ which implies $C_\Sigma = C_G/\alpha_G$ and $\alpha_D = C_D/C_\Sigma$ which simplifies to $\alpha_D = (1 - \alpha_G)/2$ if $C_S = C_D$, and the capacitances are defined as shown in Fig. 2 [10], [11]. V_{G0} plays a similar role as the flat band voltage, which shifts the gate voltage V_G to an effective gate voltage value of $(V_G - V_{G0})$. The electron density at the top of the potential barrier is related to the potential

$$n = \frac{1}{\mathcal{N}}\left[\sum_{k_x>0} f(E(\mathbf{k}) - E_{\text{FS}}) + \sum_{k_x<0} f(E(\mathbf{k}) - E_{\text{FD}})\right] \tag{2}$$

where E_{FS} and E_{FD} are the source and drain Fermi energy levels, respectively. $\mathcal{N} = L$ or $\mathcal{N} = A$ is a normalization length for a quasi-1-D channel or a normalization area for a quasi-2-D channel.

After the potential U and charge density n are solved from (1) and (2), and the source–drain current is computed as follows:

$$I = \frac{1}{\mathcal{N}}\left[\sum_{k_x>0} f(E(\mathbf{k}) - E_{\text{FS}})\upsilon_x(\mathbf{k}) + \sum_{k_x<0} f(E(\mathbf{k}) - E_{\text{FD}})\upsilon_x(\mathbf{k})\right]$$

where $\upsilon_x(\mathbf{k}) = (1/\hbar)(\partial E)/(\partial k_x)$ is the band-structure-limited velocity along the channel direction.

In the above model, the gate capacitance C_G is computed from the gate oxide thickness and dielectric constant, and the E–k relation is determined by the band structure of the channel material. The above ballistic transistor model is simple and computationally efficient. It only contains two fitting parameters, α_G and V_{G0}. The model sets the ballistic performance limits of a transistor.

C. ML Model Component

In the hybrid physics-ML device model, the ML component is crucial in defining the ballisticity of nanotransistors, augmenting the PB ballistic model with data-driven insights. We evaluate three distinct ML models in our approach, each selected for its unique attributes and appropriateness in analyzing transistor behavior.

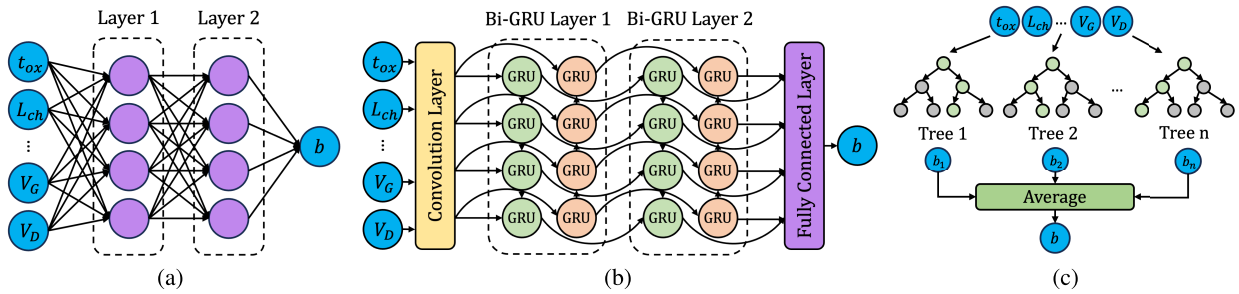

Fig. 3. Three ML models used in this work. (a) Framework of MLP. Two hidden layers and four neurons for each layer are shown in this diagram, while in practice, the number of hidden layers and neurons is adjustable. (b) Framework of BiGRU. The convolution kernel and the number of BiGRU layers are adjustable. (c) Framework of RFR. The number of trees is adjustable.

1) *Multilayer Perceptron (MLP):* A fundamental form of artificial neural networks (ANNs), the MLP [12] comprises multiple fully connected layers [Fig. 3(a)] and is widely used in prior transistor simulation works [13], [14]. As shown in Fig. 3(a), it takes device parameters ($t_{\rm ox}$, $L_{\rm ch}$) and voltage biases (V_G, V_D) as input and predicts the corresponding ballisticity value b. Each neuron evaluates $\mathbf{h} = f(\mathbf{W}\mathbf{x} + \mathbf{b})$, where $\mathbf{x}$ is the input for the current layer, $\mathbf{W}$ and $\mathbf{b}$ are the weights and bias of the current neuron, and f is the nonlinear activation function. While the MLP is effective across various applications, its simple architecture may limit its ability to process complex dependencies in data, which is investigated in our experimental analysis.
2) *Recurrent Neural Network (RNN):* To discern the intricate interrelationships among device parameters, we utilize a bidirectional gated recurrent unit (BiGRU) [15], which is a variation of traditional RNNs. As shown in Fig. 3(b), BiGRU takes $t_{\rm ox}$, $L_{\rm ch}$, V_G, and V_D as input, and outputs the prediction of ballisticity value b. BiGRU's operation is represented as $h_t = \text{GRU}(\mathbf{h_{t-1}}, \mathbf{x_t}) \oplus \text{GRU}(\mathbf{h_{t+1}}, \mathbf{x_t})$, where $\oplus$ denotes the concatenation operation. Unlike MLP, BiGRU can capture dependencies between sequential elements, providing a thorough analysis of the relationships among various device parameters.
3) *RandomForestRegressor (RFR):* Fig. 3(c) shows the framework of RFR [16]. Functioning as a regression methods, RFR constructs an ensemble of decision trees, each trained on a random subset of the data and features. Its strength lies in diminishing prediction variance and reducing the risk of overfitting, particularly in scenarios with sparse data. The RFR's proficiency in managing complex, nonlinear relationships is invaluable for capturing the nuanced interplay of device parameters in FETs.

The selection of these models underscores their distinct roles in our hybrid approach, enabling a nuanced understanding of the complex dynamics in nanoscale transistor behavior. It is crucial for advancing the accuracy and applicability of our hybrid modeling approach.

D. Model Training Procedure

The training procedure for the model is shown in Fig. 1 to predict the transistor I–V characteristics is described as follows.

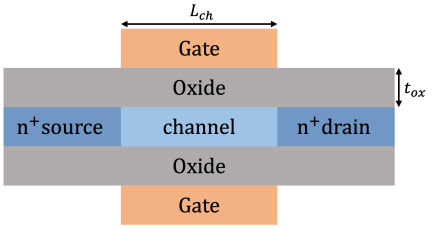

Fig. 4. Schematic of the double-gated UTB silicon MOSFET modeled.

1) From the transistor I_D versus V_G data at $V_D = V_{DD}$, compute the inverse subthreshold slope $S(V_G) = (dV_G)/(d(\log_{10}(I_D)))$.
2) Extract α of the ballistic transistor model as $\alpha_G = S_0/(\min(S(V_G)))$, where min() is the minimal value, and $S_0 = (kT)/q\ln 10$, which is the ideal subthreshold swing.
3) Extract the other parameter in the ballistic transistor model V_{G0} by minimizing a cost function defined as the difference of $S(V_G)$ from the data and the ballistic model. With both parameters of the ballistic transistor model obtained, the ballistic current $I_{\rm bal}$ is obtained from the ballistic model.
4) The ballisticity, b, as shown in Fig. 1, is obtained from an ML regression model, and the cost function is defined as the mean square error between the predicted I–V and the I–V data. The ML model is trained to minimize the cost function.

III. Results

In this section, we illustrate the application of our hybrid model to nanotransistor technologies through two distinct examples. The first example applies the proposed hybrid model to a silicon nanotransistor. The second example applies the model to a CNT transistor, showcasing the model's versatility and generality across different nanoscale devices. To further substantiate the practical utility of our hybrid model, we incorporate it into circuit-level simulations, specifically focusing on inverter and ring oscillator (RO) circuits.

A. Model Double-Gated Ultrathin-Body Silicon Transistor and FinFET

1) Data Generation: The modeled device is a double-gated UTB silicon FET, as shown in Fig. 4. The device data are obtained by numerical device simulations based on solving

the Poisson equation self-consistently with the nonequilibrium Green's function (NEGF) formalism, as implemented in the simulation tool nanoMOS [17], [18]. In nanoMOS, a Schrodinger equation is solved in the vertical confinement direction to obtain the confinement subbands. In the transport direction along the channel, the quantum transport equation is solved by using the NEGF formalism. Scattering is treated with the self-consistent Born approximation (SCBA) [18]. The NEGF device simulations are computationally expensive; we parallel the simulations over processors to obtain the device data of multiple silicon nanotransistor devices for model training and testing.

The simulated device in the data generation process has a gate oxide thickness (t_{ox}) equal to 2 nm with a dielectric constant of 20. The silicon body has a thickness of 3 nm. The gate length (L_{ch}) is 10 nm. Room temperature $T = 300$ K is assumed. For each individual device, the gate bias sweeps from 0 to 0.5 V with a step of 0.0125 V/step, which results in a total of $N_G = 41$ V_G points. The drain bias sweeps from 0.001 to 0.501 V with a step of 0.0125 V/step, which results in an $N_D = 41 V_D$ points. The total bias points in the dataset of an individual device is $N_D \times N_G = N^2 = 1681$ points. To investigate the possibility of small data training, we use a subset of the full data with a larger step and smaller N value to train the device model and investigate the dependence of the accuracy of the trained model as a function of the size of the dataset, as described later.

2) Model Evaluation: We divide the device data into two sets: a training set and a test set, each containing approximately half of the data points. The training set spanned V_D values from 0.001 to 0.501 V in 40 equal steps, and V_G values from 0 to 0.5 V in 20 equal steps. The test set also covered V_D values from 0.001 V to 0.501 V in 40 equal steps but V_G values from 0.0125 to 0.4875 V in 19 equal steps. The $V_D = 0$ bias was excluded since the ballistic transistor model gives $I_{D,bal}(V_D = 0) = 0$, which results in a zero current at zero drain bias for the hybrid model.

The above data partition ensures that the test data does not overlap with the training data, so that information leakage between training and testing is avoided. Three ML models as described before are trained, and the comparison between the model prediction with the test data is shown in Fig. 5(a) and (b), which illustrate the I_D–V_D and I_D–V_G characteristics, respectively. The MLP model comprises two hidden layers, each with ten nodes using ReLU activation, and a single-node output layer for I_D prediction. The RNN model, a two-layer BiGRU, includes an input layer with a 1×1 kernel for embedding each device parameter into an 8-D tensor, followed by a sequence processing model with eight nodes in each hidden layer, and a single-node output layer for I_D prediction. Both models utilize the Adam optimization algorithm with a learning rate of 0.001, a batch size of 32, and 5000 epochs. The RFR model is implemented using RFR in Scikit-learn with 100 estimators.

The results, as depicted in the figures, reveal that all models achieve remarkable accuracy, with half of the dataset used for training and the rest half for testing. This finding is particularly noteworthy considering that most ML studies

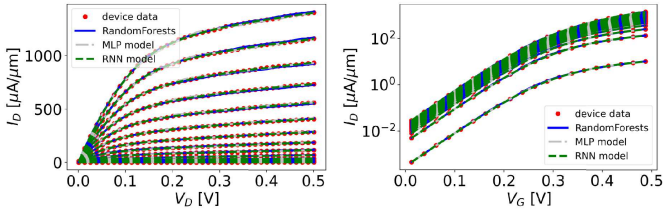

Fig. 5. Comparison between the hybrid physics-ML device models (lines) with the NEGF simulation data (dots) for the Si MOSFET as shown in Fig. 4. (a) I_D versus V_D characteristics at $V_G = 0.0125$–0.4875 V at 0.025 V/step and (b) I_D versus V_G characteristics at $V_D = 0.001$–0.501 V at 0.0125 V/step. Three different ML model components are tested: RFR (blue solid lines), MLP (gray dashed-dotted lines), and RNN (green dashed lines).

TABLE I
R^2 VALUES ACROSS DIFFERENT ML MODELS WITH SINGLE Si MOSFET DEVICE AND 55 Si MOSFET DEVICE INSTANCES

Model	MLP model	RNN model	RandomForests
Single device	99.801%	99.997%	99.575%
Cross devices	95.685%	99.995%	99.600%

typically require 70%–80% of the data for training. This efficiency demonstrates the proposed hybrid model's ability to reduce the demand for extensive training data, which is often time-consuming to acquire. Table I provides a summary of the hybrid models' performance in fitting both a single device and a dataset encompassing 55 device instances, with the channel length varying from 10 to 20 nm with 1 nm/step, and the oxide thicknesses varying from 1.5 to 3.5 nm with 0.5 nm/step. All other parameters are the same as the single-device simulation. As we scale the models to larger datasets, an increase in model size is usually necessary to accommodate the increased complexity of the data. However, as Table I illustrates, almost all models, particularly the BiGRU model, maintain consistently high accuracy across both single and multiple device datasets.

3) Why Does the Hybrid Device Model Work Well?: The reason that the hybrid model works well is that the approach significantly simplifies the ML training task. The ballistic device I–V characteristics are predicted by a PB model, and the ML part only describes the ballisticity of the transistor. To illustrate this point, we plot the ballistic current of the modeled device as shown in Fig. 6(a) and (b). One key challenge of modeling transistor data is that the current varies by orders of magnitude from the OFF-state to the ON-state in a highly nonlinear manner. The PB part of the hybrid model predicts the ballistic I–V, which captures the qualitative features and orders of magnitude variation of the data. The ML component only predicts ballisticity, as shown in Fig. 1, which is defined as the ratio of the device I–V data to its corresponding ballistic limit. The dependence of the ballisticity on V_D and V_G are shown in Fig. 6(c) and (d), respectively. Although the current changes orders of magnitude from the subthreshold to the above threshold regime, the ballisticity value has a relatively weak dependence on the applied gate and drain voltages in the entire bias regime from subthreshold to above threshold for V_G, and from the linear region to

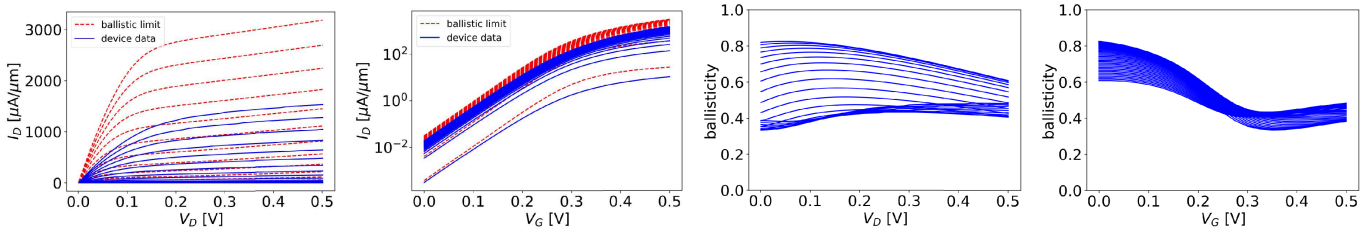

Fig. 6. Ballistic I–V and ballisticity of Si MOSFET. (a) I_D versus V_D and (b) logarithmic I_D versus V_G predicted by the ballistic transistor model (red dashed lines) and the NEGF simulation data (blue solid lines). (c) Ballisticity b versus V_D at different V_G values and (d) ballisticity b versus V_G at different V_D values. (a) and (c) are simulated at V_G = 0–0.5 V at 0.05 V/step and (b) and (d) are simulated at V_D = 0–0.5 V at 0.025 V/step.

saturation region for V_D. The ballisticity varies on a much smaller relative scale compared with the variations of I_D. As a result, by using the hybrid approach, the ML training task to fit the ballisticity is significantly simplified compared to directly fitting the transistor I–V data. This advantage can enable easier training with a simple ML model component and high prediction accuracy.

4) Application of the Hybrid Physics-ML Model to FinFET Dataset: To demonstrate the adaptability of the hybrid model to datasets created for industrial FinFET devices, we generated the device I–V characteristics data by using the BSIM-CMG model [19]. The BSIM-CMG model has a large number of model parameters and has been shown to agree with experimental and TCAD simulation data for FinFET devices [19]. We train the hybrid model through the same procedure as shown in Fig. 1 and compare the trained model with the MLP, RNN, and RFR ML model components with the device data as shown in Fig. 7. The results confirm that the hybrid models with the RNN ML model and RFR can describe the FinFET dataset with high accuracy.

We further explore the extrapolation capability of the hybrid model. Fig. 7 also shows the comparison between the data and the hybrid models that extend into extrapolated regions, illustrated by the bold sections in our figures. Although the extrapolation performance of MLP and random forest in the extrapolation regions is not good enough, especially with a high V_G, the results show that the RNN-based hybrid model retains its predictive accuracy even when applied to the I–V data outside its immediate training range.

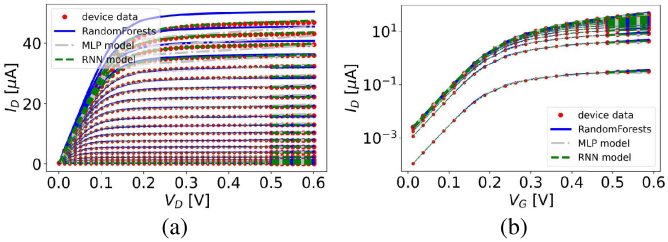

Fig. 7. Comparison between the hybrid physics-ML model with the BSIM-CMG data. (a) I_D–V_D characteristics at V_G = 0.0125–0.5875 V at 0.025 V/step. (b) I_D–V_G characteristics at V_D = 0.001–0.601 V at 0.0125 V/step. The modeled n-type FinFET has a single fin with a gate length of 20 nm and a Fin thickness of 5 nm, a gate oxide thickness of 1.2 nm with a relative dielectric constant of 3.9, and a metal gate work function of 4.3 eV. All other parameters have default values of the BSIM-CMG model. The hybrid models were trained with data in the range of $0 < V_G < 0.5$ V and $0.001 < V_D < 0.501$ V. The model prediction out of this range is extrapolation.

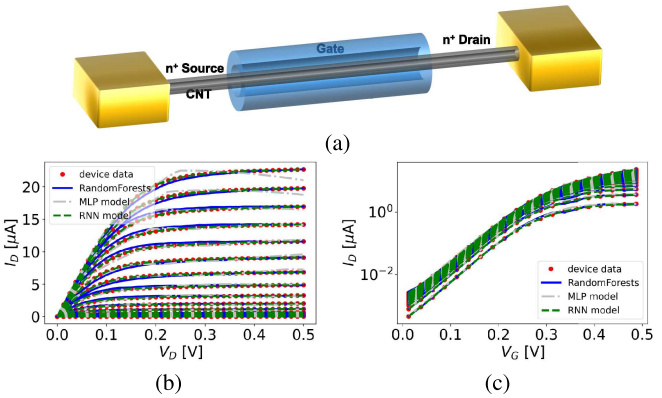

Fig. 8. (a) Schematic device structure of the modeled CNTFET. Comparison of the test data (not used in ML model training) from NEGF simulations to the physics-ML device model prediction for (b) I_D–V_D characteristics at V_G = 0.0125–0.4875 V at 0.025 V/step and (c) I_D–V_G characteristics at V_D = 0.001–0.501 V at 0.0125 V/step. The gate length is L_G = 12 nm.

B. Apply the Hybrid Physics-ML Device Model to CNTFETs

1) Device Structure and Data Generation: The hybrid model approach can be applied to other types of nanotransistors. In this section, we apply the hybrid model to CNTFETs. The modeled CNTFET, as shown in Fig. 8, is a MOSFET-like CNTFET with doped source and drain extensions [20] and a gate–all–around structure [21].

NEGF device simulations are performed in a p_z-orbital tight-binding Hamiltonian, and scattering by acoustic and optical phonons are treated in the NEGF simulations by using the SCBA [22], [23]. The batch of devices simulated has a gate oxide thickness varying from 1 to 3 nm with 0.5 nm/step. The dielectric constant is set to 20. The gate length varies from 10 to 30 nm with 2 nm/step. The NEGF device simulation data generated are partitioned to the training and test data in a similar manner as in the previous example of silicon nanotransistors.

2) Model Evaluation: The comparative analysis of different ML models for CNTFETs is presented in Fig. 8(b) and (c). Despite the small variation in their R^2 scores (Table II), a significant performance disparity is observed with CNTFET data. This discrepancy is attributed to the dense clustering of

TABLE II
R^2 VALUES ACROSS DIFFERENT ML MODELS WITH SINGLE CNTFET DEVICE AND 55 CNTFET DEVICE INSTANCES

Model	MLP model	RNN model	RandomForests
Single device	96.580%	99.997%	98.657%
Cross devices	98.123%	99.986%	99.935%

TABLE III
MODEL PERFORMANCE QUANTIFIED BY R^2 VALUES VERSUS THE SIZE OF THE TRAINING DATASET. THE GRID POINTS $N = N_G = N_D$ AND THE TOTAL NUMBER OF TRAINING DATA $= N^2$

Num of N		21	11	7	6	5
ΔV_G and ΔV_D [V]		0.025	0.05	0.075	0.1	0.125
MLP (Pytorch)	max	0.945	0.866	0.599	0.564	0.348
	min	-0.619	-0.509	-0.779	-0.479	-0.969
	avg.	0.688	0.552	-0.447	-0.443	-0.858
RNN (Pytorch)	max	0.999	0.999	0.998	0.996	0.926
	min	0.999	0.999	0.990	0.983	0.772
	avg.	0.999	0.999	0.995	0.992	0.882
RFR (Scikit-learn)	max	0.975	0.939	0.826	0.774	0.507
	min	0.974	0.936	0.810	0.751	0.402
	avg.	0.974	0.938	0.822	0.767	0.470

data points with almost zero current, which artificially inflates the R^2 scores while diminishing the impact of outliers.

3) Reduce the Training Data Size for Small Data Training: A large set of training data is either computationally or experimentally expensive to generate. Achieving high model accuracy with a small training dataset is preferred to reduce the cost of obtaining training data. To further investigate the capabilities of the three ML models under varying training set sizes, we adjust the step sizes for both V_G and V_D from 0.025 to 0.125 V. The number of grid points N_G and N_D are approximately inversely proportional to the step size, and the total size of the data is $N_G \times N_D$. To account for the influence of randomness in the ML model training, each ML training experiment was repeated 10 times, recording the minimum, maximum, and average R^2 scores, and the results are displayed in Table III. It is evident that the MLP is more susceptible to random variations. However, the RNN model displayed remarkable accuracy, which could achieve an accuracy of nearly 99.9% even under extremely low training data density. Furthermore, the average performance of the RNN model consistently outperformed the RFR by an order of magnitude, indicating a reduced dependency on extensive datasets compared with the RFR. As shown in Table III, a small data training with $N = N_D = N_G = 6$ and a total data size of $N^2 = 36$ can still maintain the high accuracy of the model by using the RNN ML model.

4) Model Inference and Training Time: The trained hybrid physics-ML device model predicts I–V characteristics much faster compared to the NEGF simulations. We sampled a large number of (V_G, V_D) bias points to obtain I_D from both the NEGF simulations and the trained hybrid models. On average, hybrid model inference is 18 000× faster than the NEGF device simulations.

On the other hand, the major cost of the hybrid ML model is a one-time investment in data generation and training. We have discussed the possibility of small data training above. Next, we examine the training time. The training duration for each model is summarized in Table IV. Due to the inherent characteristics of their gradient descent learning algorithms, ANNs generally require longer training times compared to the RFR.

TABLE IV
TRAINING TIME IN SECOND ACROSS DIFFERENT ML MODELS WITH SINGLE CNTFET DEVICE AND DIFFERENT V_G AND V_D STEP SIZE

Num of N	21	11	7	6	5
ΔV_G and ΔV_D [V]	0.025	0.05	0.075	0.1	0.125
MLP (Pytorch)	99.804	85.352	77.902	72.430	71.050
RNN (Pytorch)	140.004	97.523	85.614	75.612	76.464
RFR (Scikit-learn)	0.050	0.036	0.032	0.031	0.033

While RFR models train substantially faster due to simpler computations and parallel estimator training, the performance gap between it and BiGRU especially under different training set sizes is also important because retrieving device data is really time-consuming. The performance disparity between BiGRU and RFR, particularly under varying training data densities, will be further explored in the subsequent circuit simulation section.

C. Application of the Model to Circuit Simulations

We conduct simulations on two circuits to validate the proposed hybrid model and to explore the performance differences between RNN-based and RFR-based hybrid models. The circuit simulations were performed by implementing KCL and KVL laws in Python. The circuits included an inverter with 1 p-CNTFET and 1 n-CNTFET, featuring $R_S = R_D = 100$ kΩ parasitic source and drain resistances and $C_{GS} = C_{GD} = 20$ aF parasitic gate-to-source and gate-to-drain capacitance as depicted in Fig. 9(a), and an RO comprising three inverters to examine transient operation reliability. Detailed simulation and model training can also be processed in a similar manner for p-CNTFETs. Due to symmetric conduction and valence bands of a CNT, we, however, simplify the p-CNTFET characteristics by assuming it is perfectly balanced with n-CNTFET, $I_{D,p}(-V_G, -V_D) = -I_{D,n}(V_G, V_D)$. The NEGF simulation is used to generate Q–V characteristics data, $Q(V_G, V_D)$, where Q is the total charge in the CNTFET channel. A ridge regression model with RBF kernel is used to train and fit the NEGF Q–V data of the intrinsic CNTFET, and it is used in the transient RO simulations. Parasitic gate-to-source and gate-to-drain capacitance often have charges significantly larger than the intrinsic channel in a nanoscale CNTFET. As the terminal bias varies, charge variation on the parasitic capacitors often dominates over that of the intrinsic CNTFET, especially in the subthreshold region.

In addressing the ground truth for circuit characteristics for assessing the ML model, we encountered the challenge that directly performing NEGF simulations is exceedingly time-consuming within the context of circuit simulation. To circumvent this, we generated a dataset with very high density and large data points and trained a device model from this dense dataset. We generated a data grid comprising 81 × 81 points. The interpolated I_D values, derived from this grid data, served as our ground truth. To validate the reliability of our approach, we quantify its uncertainty by varying the data size and interpolation model. We examine various subsets of this grid data—specifically, a half-size data corresponding to a reduction according to twice the

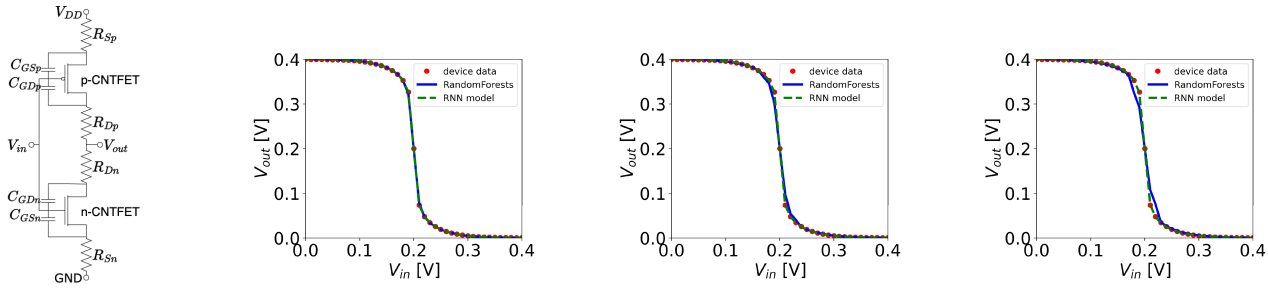

Fig. 9. Apply the hybrid physics-ML model of CNTFET to inverter simulations. (a) Schematic circuit of the modeled inverter with transistors. The parasitic source and drain resistance are shown. The parasitic resistance is $R_S = R_D = 100$ kΩ for both n-type and p-type CNTFETs. Comparison of simulated inverter transfer characteristics between different ML model components and the hybrid device models trained with different total training data sizes of N^2 with $N = N_G = N_D$. (b) $N = 21$, (c) $N = 7$, and (d) $N = 5$.

step size of V_G, a half-size data pertaining to a reduction according to twice the step size of V_D, and a quarter-size data resulting from halving both V_G and V_D. We also explore different interpolation methods, including multiquadric, cubic, and linear methods, to assess performance. The average error $(1 - R^2)$ value less than 10^{-5} for the inverter, along with the average amplitude discrepancy less than 10^{-4} and average period difference less than 10^{-14} underscore the sufficiency of our device data density. These uncertainties in the ground truth are order-of-magnitude smaller compared to the scale of model error investigated next, which indicates the sufficiency of the above approach.

Fig. 9(b)–(d) illustrates the performance evaluation of the hybrid models using RNN and RFR, varying the V_G and V_D step sizes, which equivalently varies the training data size for the hybrid device model. The results indicate that both models align closely with the ground truth at a step size of 0.025 V, demonstrating effective learning with dense training datasets. At this step size, the training data size is $N^2 = 441$, indicating the promise of small data training. However, as the step size increases to 0.075 V and further to 0.125 V, the RFR model begins to diverge from the actual device data, struggling to accurately replicate device behavior. In contrast, the RNN models consistently match the ground truth across all tested step sizes. Notably, at a step size of 0.125 V, the training dataset for a single device diminishes to merely $N^2 = 25$ data points in the hybrid model training, a challenging scenario for most ML models to discern valid patterns.

The simulation results for a three-stage RO are presented in Fig. 10(a) and (b), comparing the RNN-based and RFR-based hybrid models at 0.025- and 0.1-V V_G and V_D step sizes, respectively, with an initial voltage set to 0.25 V. These findings corroborate the earlier observation of the RNN model's robustness against low training data density. Its structure, capable of modeling interelement relationships, facilitates a more accurate prediction of ballisticity. Moreover, our hybrid model successfully captures the nonlinear transistor behavior with remarkable accuracy, even with a simplistic RFR model. The implementation of a two-layer RNN model in circuit simulations demonstrates impressive accuracy under extremely low training data density, potentially reducing the time required to gather device data significantly. Fig. 11 shows

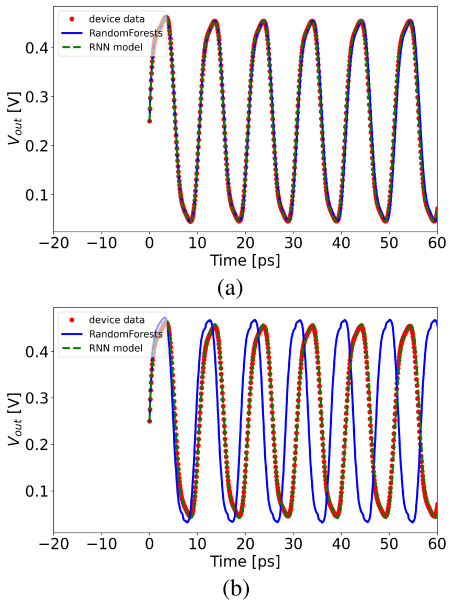

Fig. 10. Apply the hybrid device model to RO simulations: Comparison of a simulated three-stage RO with different ML model components with different training dataset sizes of N^2 with $N = N_G = N_D$ for (a) $N = 21$, and (b) $N = 6$. The CNTFETs have a gate length of 12 nm.

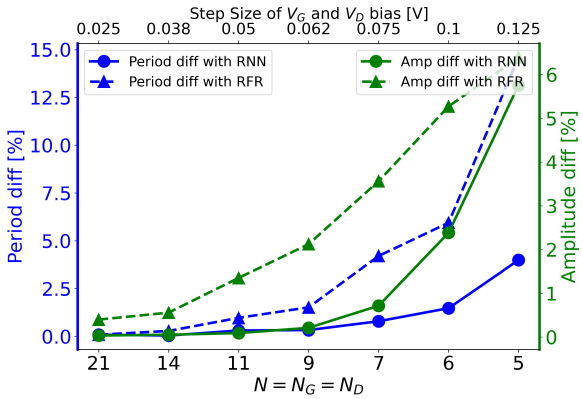

Fig. 11. Error of period and amplitude in RO simulations versus the data size for training the hybrid device model of the CNTFET. The bottom axis shows $N = N_G = N_D$ for equally spaced training data points with the corresponding step size shown on the top axis. The RNN and RFR model results are shown.

the amplitude discrepancies and period differences under different training data sizes. It is clear that the RNN model

still outperforms the RFR model in circuit simulations. The RNN model can achieve a high accuracy with a period error of $<1.52\%$ and an amplitude error of $<0.34\%$ with a relatively small data of $N_D = N_G = 9$ in the application of the hybrid device model to RO circuit simulations.

IV. Conclusion

A hybrid physics-ML model that combines a ballistic transistor model with an ML model component is developed to model nanoscale transistors. The approach captures the physics of quasi-ballistic transport in nanoscale transistors and can flexibly interact with nanotransistor data. Although the transistor current values vary orders of magnitude from the OFF-state to the ON-state and have a strongly nonlinear dependence on the bias voltages, the ballisticity has a much weaker dependence on the bias voltage. The ML component, focusing on ballisticity prediction, significantly simplifies ML training and mitigates the requirement for extensive training data in transistor modeling. Comparative analysis of various ML models, including MLP, BiGRU, and RFR, revealed the exceptional capability of the BiGRU model in processing sparse datasets and maintaining high accuracy, even with limited training data.

The application of this hybrid model to both silicon and CNT transistors demonstrated its versatility and high accuracy for modeling nanoscale transistors. Compared with NEGF device simulations, the trained model predicts device characteristics with a factor of $>18\,000$ speed improvement and is useful in circuit simulations. Overall, this hybrid approach not only achieves a balance between empirical ML methods and PB modeling but also opens new avenues for efficient and precise modeling of nanoscale transistors with small data training.

Acknowledgment

The authors thank Prof. Hiu-Yung Wong from San Jose State University, San Jose, CA, USA, for helpful technical discussions.

References

[1] G. Fiori et al., "Electronics based on two-dimensional materials," *Nature Nanotechnol.*, vol. 9, no. 9, pp. 768–779, Aug. 2014.

[2] A. D. Franklin, M. C. Hersam, and H.-S.-P. Wong, "Carbon nanotube transistors: Making electronics from molecules," *Science*, vol. 378, no. 6621, pp. 726–732, Nov. 2022.

[3] Y. S. Chauhan et al., "BSIM—Industry standard compact MOSFET models," in *Proc. ESSCIRC*, Sep. 2012, pp. 30–33.

[4] A. Rahman and M. S. Lundstrom, "A compact scattering model for the nanoscale double-gate MOSFET," *IEEE Trans. Electron Devices*, vol. 49, no. 3, pp. 481–489, Mar. 2002.

[5] J. Wang, Y.-H. Kim, J. Ryu, C. Jeong, W. Choi, and D. Kim, "Artificial neural network-based compact modeling methodology for advanced transistors," *IEEE Trans. Electron Devices*, vol. 68, no. 3, pp. 1318–1325, Mar. 2021.

[6] K. Mehta and H.-Y. Wong, "Prediction of FinFET current-voltage and capacitance-voltage curves using machine learning with autoencoder," *IEEE Electron Device Lett.*, vol. 42, no. 2, pp. 136–139, Feb. 2021.

[7] M. Kao, H. Kam, and C. Hu, "Deep-learning-assisted physics-driven MOSFET current-voltage modeling," *IEEE Electron Device Lett.*, vol. 43, no. 6, pp. 974–977, Jun. 2022.

[8] N. Chatterjee, J. Ortega, I. Meric, P. Xiao, and I. Tsameret, "Machine learning on transistor aging data: Test time reduction and modeling for novel devices," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Mar. 2021, pp. 1–9.

[9] F. Klemme, J. Prinz, V. M. van Santen, J. Henkel, and H. Amrouch, "Modeling emerging technologies using machine learning: Challenges and opportunities," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Nov. 2020, pp. 1–9.

[10] A. Rahman, J. Guo, S. Datta, and M. S. Lundstrom, "Theory of ballistic nanotransistors," *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1853–1864, Sep. 2003.

[11] M. Lundstrom and J. Guo, *Nanoscale Transistors: Device Physics, Modeling and Simulation*. Berlin, Germany: Springer, 2006.

[12] S. Amari, "A theory of adaptive pattern classifiers," *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 3, pp. 299–307, Jun. 1967.

[13] Y.-W. Ho et al., "Neuroevolution-based efficient field effect transistor compact device models," *IEEE Access*, vol. 9, pp. 159048–159058, 2021.

[14] H. M. Xie, D. Y. Lei, Z. C. Zhang, Y. Q. Chen, Z. H. He, and Y. Liu, "Compact modeling of metal–oxide TFTs based on the Bayesian search-based artificial neural network and genetic algorithm," *AIP Adv.*, vol. 13, no. 8, Aug. 2023.

[15] C. Li, Y. He, X. Li, and X. Jing, "BiGRU network for human activity recognition in high resolution range profile," in *Proc. Int. Radar Conf. (RADAR)*, Sep. 2019, pp. 1–5.

[16] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.

[17] Z. Ren, R. Venugopal, S. Goasguen, S. Datta, and M. S. Lundstrom, "nanoMOS 2.5: A two-dimensional simulator for quantum transport in double-gate MOSFETs," *IEEE Trans. Electron Devices*, vol. 50, no. 9, pp. 1914–1925, Sep. 2003.

[18] Z. Ren et al., "Nanomos," 2016. [Online]. Available: https://nanohub.org/resources/nanomos

[19] J. P. Duarte et al., "BSIM-CMG: Standard FinFET compact model for advanced circuit design," in *Proc. 41st Eur. Solid-State Circuits Conf. (ESSCIRC)*, 2015, pp. 196–201.

[20] A. Javey, R. Tu, D. B. Farmer, J. Guo, R. G. Gordon, and H. Dai, "High performance n-type carbon nanotube field-effect transistors with chemically doped contacts," *Nano Lett.*, vol. 5, no. 2, pp. 345–348, Feb. 2005.

[21] A. D. Franklin et al., "Carbon nanotube complementary wrap-gate transistors," *Nano Lett.*, vol. 13, no. 6, pp. 2490–2495, Jun. 2013.

[22] J. Guo, S. Datta, M. Lundstrom, and M. P. Anantam, "Toward multiscale modeling of carbon nanotube transistors," *Int. J. Multiscale Comput. Eng.*, vol. 2, no. 2, pp. 257–276, 2004.

[23] J. Guo, "A quantum-mechanical treatment of phonon scattering in carbon nanotube transistors," *J. Appl. Phys.*, vol. 98, no. 6, Sep. 2005, Art. no. 063519.