scientific reports

OPEN

Machine learning-powered compact modeling of stochastic electronic devices using mixture density networks

Jack Hutchins[1], Shamiul Alam[1], Dana S. Rampini[2], Bakhrom G. Oripov[2], Adam N. McCaughan[2] & Ahmedullah Aziz[1]✉

The relentless pursuit of miniaturization and performance enhancement in electronic devices has led to a fundamental challenge in the field of circuit design and simulation-how to accurately account for the inherent stochastic nature of certain devices. While conventional deterministic models have served as indispensable tools for circuit designers, they fall short when it comes to capturing the subtle yet critical variability exhibited by many electronic components. In this paper, we present an innovative approach that transcends the limitations of traditional modeling techniques by harnessing the power of machine learning, specifically Mixture Density Networks (MDNs), to faithfully represent and simulate the stochastic behavior of electronic devices. We demonstrate our approach to model heater cryotrons, where the model is able to capture the stochastic switching dynamics observed in the experiment. Our model shows 0.82% mean absolute error for switching probability. This paper marks a significant step forward in the quest for accurate and versatile compact models, poised to drive innovation in the realm of electronic circuits.

In the ever-evolving landscape of electronic devices, the pursuit of smaller, faster, and more efficient technology has led to groundbreaking innovations, unlocking new possibilities for various applications across industries[1]. However, as devices continue to shrink and push the boundaries of Moore's law[2], the inherent stochastic nature of certain electronic components has become increasingly significant[3]. These stochastic electronic devices exhibit inherent variability in their behavior, posing a formidable challenge for conventional deterministic modeling approaches[4].

In the pursuit of greater accuracy and ease of implementation in electronic circuit simulations, a significant leap has been taken by incorporating machine learning methodologies into compact modeling[5–10]. While various machine learning-based compact models have emerged, these models, thus far, have failed to address the implications of stochastic behavior in electronic devices. And while traditional physics-based compact models have been created with stochastic properties[11,12], these models lack the benefits of machine learning-based methods such as decreased turnaround time and little required device knowledge. The dynamics of electronic devices, traditionally considered deterministic, are increasingly proving to be intrinsically stochastic at the nanoscale. Consequently, conventional compact models that ignore these inherent stochastic processes fail to capture the true behavior of the devices, leading to inaccuracies in circuit simulation.

This paper presents a new approach that addresses this gap in the realm of compact modeling. We introduce the use of Mixture Density Networks (MDNs)[13] to capture the stochastic nature of electronic devices accurately. In addition to this neural network approach, we introduce a novel sampling methodology that uses inverse transform sampling[14] to make our model more accurate to the stochastic nature of devices. By doing these, we create compact models that account for the inherent stochasticity of certain devices, allowing the development of new circuits that are both reliable and innovative. Our approach represents a paradigm shift in electronic device modeling, as it for the first time encompasses the full spectrum of electronic device behavior, from deterministic to stochastic.

To demonstrate the effectiveness and versatility of our approach, we focus on the modeling of heater cryotron[15], a three-terminal device that shows gate-current controlled switching between superconducting and resistive states. Heater cryotrons, with their stochastic switching behavior, are a particularly compelling case

[1]Department of Electrical Engineering & Computer Science, University of Tennessee, Knoxville, TN 37996, USA. [2]National Institute of Standards and Technology, Boulder, Co 80305, USA. ✉email: aziz@utk.edu

study. Our methodology accurately captures the nuances of this stochasticity, providing a foundation for the development of novel circuits and the optimization of existing designs. It is worth noting that beyond digital devices, the flexibility of MDNs allows this approach to be used in a variety of applications where stochastic device models could be beneficial, such as analog or mixed-signal electronics.

The contributions of this paper are threefold:

- *Mixture Density Networks*: By employing MDNs, we equip our modeling framework with the capacity to capture complex probability distributions, thereby enabling a more faithful representation of device variability. This empowers designers to explore device behavior beyond traditional deterministic bounds.
- *Inverse Transform Sampling*: By leveraging inverse transform sampling, we can use our MDN to its full potential. This approach allows the model to provide smooth outputs in transient simulations while maintaining stochasticity. This in turn makes our model more realistic to device-to-device or cycle-to-cycle variations.
- *Demonstration with Heater Cryotrons*: To showcase the effectiveness of our approach, we focus on modeling heater cryotrons-a class of devices known for their stochastic switching behavior. Through this demonstration, we illustrate the practical applicability of our methodology in real-world electronic systems.

As we delve into the details of our approach, we will elucidate the intricacies of MDN-based modeling for stochastic electronic devices, providing a comprehensive understanding of the benefits it offers to circuit simulation and electronic design.

Results

The following section will describe our neural network architecture, sampling methodology, and the results of our approach using experimentally derived heater cryotron data.

Mixture density network

Mixture density networks provide the perfect approach for modeling stochastic devices. Mixture density networks work similarly to standard multilayer perceptions (MLP)[16], but with three output layers connected to the last hidden layer instead of just one. Each output layer has the same number of neurons N, and we label the output layers as μ, σ, and α. In our model, we use 2 hidden layers with 512 neurons each using the ReLU activation function, and we set $N = 20$. The output of these layers is used as the parameters of a Gaussian mixture distribution probability density function (PDF) as defined in Eq. (1).

$$p(x) = \sum_{k=1}^{N} \alpha_k \cdot \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right). \tag{1}$$

Using this approach allows the model to learn the unique output distribution for any given input. The PDF is a combination of N different Gaussian distributions, which are multiplied by a scaling factor α. In order to maintain a valid probability distribution, we need to ensure that the sum of α is always equal to 1. To do that, we apply a softmax function to the output of the α layer, as defined in Eq. (2).

$$s(\alpha_i) = \frac{e^{\alpha_i}}{\sum_{j=1}^{N} e^{\alpha_j}}. \tag{2}$$

Additionally, we need to ensure that the output of the σ layer needs to be positive and non-zero since they represent the standard deviation of the Gaussian distributions in the mixture density PDF. In order to do this, we use the exponential linear unit (ELU) activation function plus 1 plus ϵ, where ϵ is the smallest value before loss of precision in floating point calculations. The activation function can be seen in Eq. (3).

$$ELU(\sigma_i) = \left\{ \begin{array}{ll} \sigma_i + 1 + \epsilon, & \text{if } \sigma \geq 0 \\ 0.5(e^{\sigma_i} - 1) + 1 + \epsilon, & \text{if } \sigma < 0 \end{array} \right\}. \tag{3}$$

Finally, we allow μ to be any value, so there is no need to use an activation function to restrict its output. For the hidden layers, we choose to use the rectified linear unit activation function, which is defined in Eq. (4).

$$ReLU(x) = \left\{ \begin{array}{l} x, \ \text{if } x \geq 0 \\ 0, \ \text{if } x < 0 \end{array} \right\} \tag{4}$$

In order to train the model, we need an appropriate loss function to measure the performance of the model. We choose to use Gaussian Negative Log-Likelihood (GNLL), since it effectively captures the performance with respect to probability. The GNLL for a single data point x is the negative logarithm of this PDF:

$$\text{GNLL}(x) = -\log\left(\sum_{k=1}^{N} \alpha_k \cdot \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right)\right) \tag{5}$$

The goal during training is to minimize the GNLL across all training data points. Therefore, the overall GNLL loss is the average GNLL over all data points:

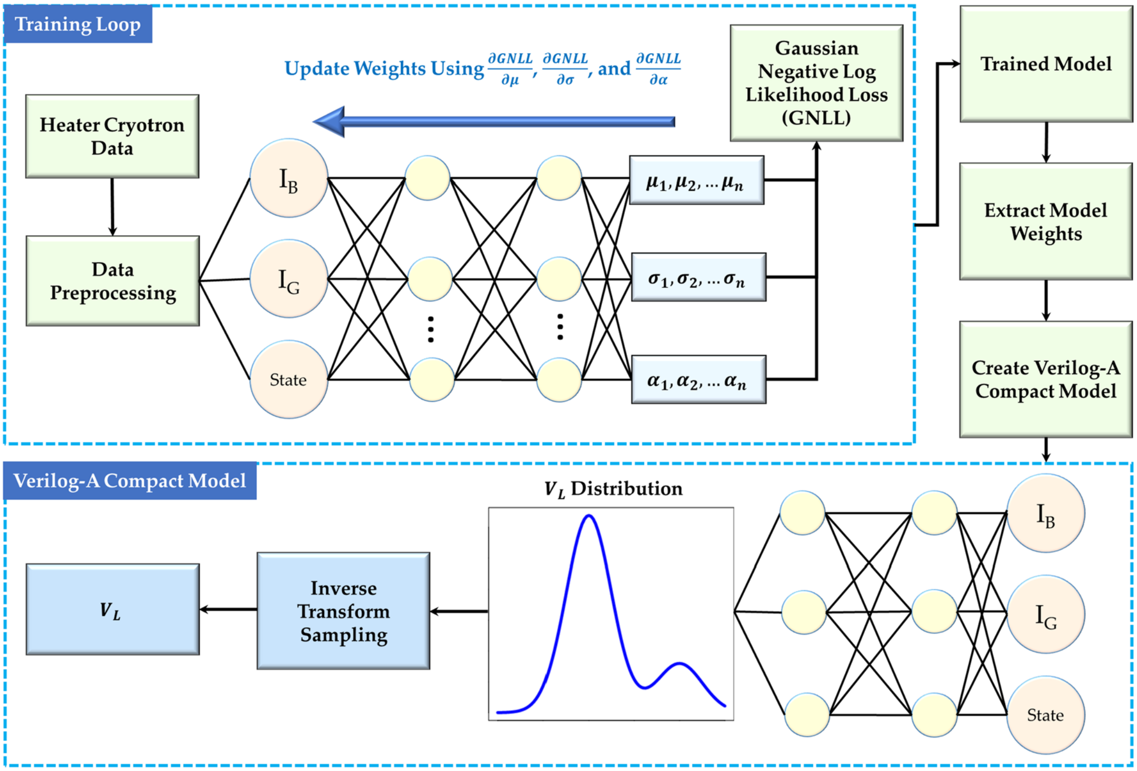
Training Loop
Update Weights Using $\frac{\partial GNLL}{\partial \mu}$, $\frac{\partial GNLL}{\partial \sigma}$, and $\frac{\partial GNLL}{\partial \alpha}$
Gaussian Negative Log Likelihood Loss (GNLL)
Heater Cryotron Data
Data Preprocessing
I_B
I_G
State
$\mu_1, \mu_2, \dots \mu_n$
$\sigma_1, \sigma_2, \dots \sigma_n$
$\alpha_1, \alpha_2, \dots \alpha_n$
Trained Model
Extract Model Weights
Create Verilog-A Compact Model
Verilog-A Compact Model
V_L Distribution
V_L
Inverse Transform Sampling
I_B
I_G
State

Figure 1. Overview of the machine learning-powered modeling approach for the stochastic behavior of heater cryotron. After training the mixture density network with the device characteristics obtained from experiments, the weights are extracted which are eventually used to create the Verilog-A-based compact model for circuit/system-level simulations.

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\alpha_k\cdot\frac{1}{\sqrt{2\pi\sigma_k^2}}\cdot\exp\left(-\frac{(x_i-\mu_k)^2}{2\sigma_k^2}\right)\right) \tag{6}$$

During the training process, the network undergoes parameter adjustments aimed at minimizing the GNLL loss. To achieve this, we employ the backpropagation algorithm coupled with the AdamW optimizer, as introduced by Loshchilov and Hutter in 2019[17]. This optimization technique facilitates the iterative refinement of the network's weights and biases. The gradients of GNLL with respect to key parameters, namely μ, σ, and α, are computed using the depicted framework in Fig. 1.

Model sampling methodology

```
Input:
  $K$ - Number of Gaussian components
  $\{\mu_1, \mu_2, \dots, \mu_K\}$ - Means of the Gaussians
  $\{\sigma_1, \sigma_2, \dots, \sigma_K\}$ - Stand deviation of the Gaussians
  $\{\alpha_1, \alpha_2, \dots, \alpha_K\}$ - Mixing coefficients, $\sum_{i=1}^{K}\pi_i = 1$
Output:
  Sample $x$ from the mixture distribution
Sample Function:
  sum = 0
  Sample $q$ from a uniform distribution [0, 1]
  for $k \in \{1, K\}$ do
    sum = sum + $\alpha_k$
    if sum $\geq$ $q$ then
      break
  end
  Sample $x$ from the Gaussian distribution $\mathcal{N}(\mu_k, \sigma_k)$
Return:
  $x$ - Sample from the Gaussian mixture density
```

Algorithm 1. Standard sampling from a Gaussian mixture density distribution.

In order to properly utilize our model, it is critical that we use an appropriate sampling methodology for the output distributions. The obvious approach for this would be to randomly sample from the distribution, however, this approach presents several issues. In order to sample in this way, we can use the approach outlined in Algorithm 1, where we randomly select a distribution k for the mixture density distribution and sample from a standard normal distribution $\mathcal{N}(\mu_k, \sigma_k)$. This approach allows us to utilize uniform and standard Gaussian sampling, which is built into most modeling languages including Verilog-A. The issue with this approach is that it results in sporadic currents in transient simulations since the current will jump around the distribution every time step. In addition, this approach will cause multi-state devices to switch sporadically between states near the critical value if the switching of the device is stochastic. While this approach may be valuable for some devices, for most devices we will need another sampling methodology that more accurately reflects the variation we see in devices. We propose using inverse transform sampling[14] to accomplish this.

Inverse transform sampling involves obtaining the cumulative distribution function (CDF) of the target distribution and finding its inverse. The CDF represents the probability that a random variable is less than or equal to a specific value. In this method, a uniform random number between 0 and 1 serves as the input, representing a quantile of the distribution. The inverse CDF maps this input to the corresponding value in the distribution.

As the distribution evolves over time, the CDF is updated to reflect these changes. Inverse transform sampling allows for the generation of samples that consistently align with a predetermined quantile (q) for each sweep of the device. This ensures that the model accurately captures the device's behavior at specific points in the distribution, providing a constant sampling point across different timesteps of the same sweep. This approach allows the model to provide a smooth curve while still demonstrating the stochasticity that we are trying to model and offering a coherent representation of the device's performance under dynamic conditions. To initiate this process, we can begin by examining the probability density function in Eq. (7).

$$p(x) = \sum_{k=1}^{K} \alpha_k \cdot \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(x-\mu_k)^2}{2\sigma_k^2}\right) \tag{7}$$

Our CDF will be equal to the integral from $-\infty$ to x as shown in Eq. (8).

$$F(x) = \int_{-\infty}^{x} \sum_{k=1}^{K} \alpha_k \cdot \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(t-\mu_k)^2}{2\sigma_k^2}\right) dt \tag{8}$$

While this is the CDF, it would be beneficial for us to manipulate this into a more calculable form. By bringing the integral inside the summation and leaving the scaling factor α outside the integral, we are left with Eq. (9).

$$F(x) = \sum_{k=1}^{K} \alpha_k \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot \exp\left(-\frac{(t-\mu_k)^2}{2\sigma_k^2}\right) dt \tag{9}$$

This leaves us with the weighted sum of the CDF of a standard normal distribution, which can be defined in terms of the error function as shown in equation (10).

$$F(x) = \sum_{k=1}^{K} \frac{\alpha_k}{2}\left[1 + erf\left(\frac{x-\mu_k}{\sigma_k\sqrt{2}}\right)\right] \tag{10}$$

Defining this in terms of the error function makes calculation much easier since the error function is well-defined and easy to calculate numerically. The error function is defined in Eq. (11).

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2}\, dt \tag{11}$$

The issue is that there is no closed-form solution to the inverse of this CDF. As such, we use a numerical root finder to find what value of x leads to $F(x) - q = 0$ where q is the percentile we sample at. We choose to use Brent's method[18], however, most numerical root finders should work here. For derivative-based root finders, the derivative of the CDF is equal to the PDF.

Now that we have established a way to perform inverse transform sampling, we can take full advantage of its properties to improve our model. To solve our issue with transient simulations, we can keep the same value of q for the entirety of a sweep. This will result in a continuous output that is more in line with the device's properties of cycle-to-cycle variation than traditional sampling. In this case, we can generate a new q every time the derivative of an input with respect to time crosses 0, which can easily be done in Verilog-A. Another benefit of this sampling approach is the ability to clip the probability distribution. For example, we could generate q only between 0.05 and 0.95 so that the model doesn't predict values more than 2 standard deviations away (or in any range). Another use could be modeling device-to-device variation by weighting the distribution when sampling for q, though this work doesn't explore this option.

Device characteristics of heater cryotron

The heater cryotron is a superconducting device designed to exploit the unique properties of superconducting nanowires[15]. This device consists of two superconducting nanowires separated by a dielectric, forming the gate and the channel of the device. Figure 2a shows a false-colored scanning electron microscope (SEM) image of the

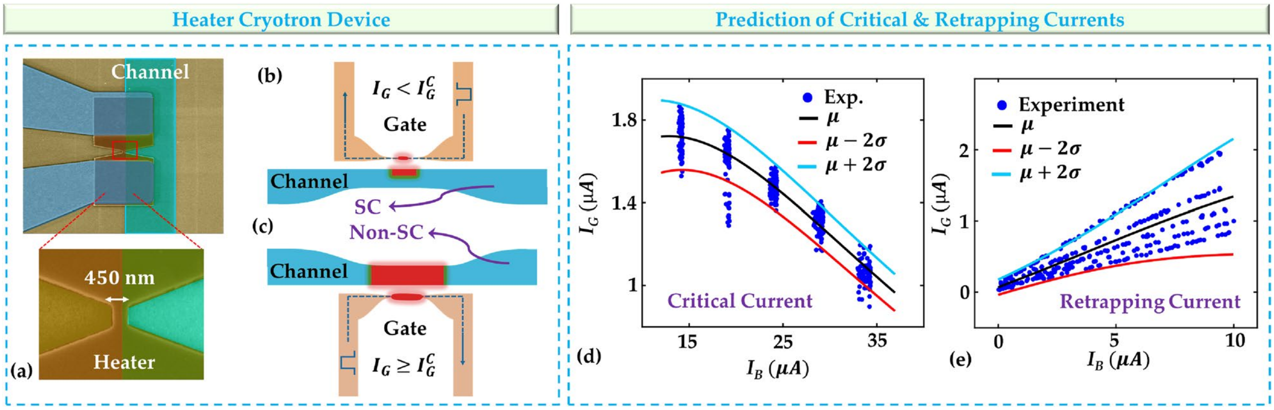

Figure 2. Prediction of Switching characteristics of heater cryotron. (**a**) A false-colored scanning electron microscope (SEM) image of heater cryotron device consisting two superconducting nanowires of tungsten silicide (WSi) and a dielectric spacer of SiO_2. (**b**,**c**) Illustration of the gate-current-controlled switching mechanism. An enough gate current (I_G) switches the gate nanowire to its resistive state and generates thermal phonos which eventually causes the switching of the channel from its superconducting to resistive state. (**d**) Switching model for critical current evaluated at the mean (μ) and plus or minus 2 standard deviations (σ) of the resultant distribution and (**e**) retrapping current of heater cryotron switching model evaluated at the same points.

fabricated heater cryotron device, with a channel width of 1 μm and a gate width of 450 nm. The gate and channel nanowires are separated by a SiO_2 dielectric spacer of 25 nm thickness. Initially, for a given channel current (I_B), both nanowires (gate and channel), when maintained at cryogenic temperatures, remain superconducting exhibiting zero resistance, and remain so until the gate nanowire becomes resistive (Fig. 2b). However, when a sufficient amount of current (I_G) is passed through the gate nanowire to make it resistive, it starts generating thermal phonons which transport to the channel nanowire with the help of the spacer. These thermal phonons suppress the superconductivity and reduce the critical current of the channel nanowire (I_{Ch}^C). Eventually, when the increase of gate current reduces the channel critical current below the applied channel current, the channel transitions from its superconducting state to a resistive state as shown in Fig. 2c. Conversely, removing the current from the heater enables the channel nanowire to revert to its superconducting state. The specific current levels at which superconducting to resistive and resistive to superconducting transitions occur are referred to as the critical current and retrapping current, respectively. The gate current-controlled switching of heater cryotrons has been used in a multitude of applications, including designing logic circuits[15,19–21], memory systems[22–26], and neuromorphic systems[27–31] for cryogenic environment, and in interfacing superconducting circuits with semiconducting technology[32].

Notably, the point at which the device switches between its superconducting and resistive states is characterized by stochastic behavior, which introduces an inherent element of randomness into its operation. This stochasticity makes this device ideal for testing our MDN-based compact modeling approach. The critical current value is not a fixed constant but instead depends on the applied channel current. For example, a higher bias current leads to a lower critical current for the gate, thus influencing the switching behavior of the device. Conversely, a higher bias current leads to a lower retrapping current. Additionally, a much lower bias current is required to switch from a resistive to a superconducting state since the nanowire produces its own heat when resistive among other factors. Understanding this dependence on the channel current is pivotal in characterizing and modeling the behavior of the heater cryotron.

To sample the characteristics of the device, we sweep the gate at different bias currents. By performing each of the sweeps multiple times and recording the resulting current-voltage characteristics of the device, we can have enough data for the MDN to learn the stochastic switching behavior of the device. This experimental approach allows us to explore the device's response under varying conditions and analyze the interplay between the gate and channel currents.

Heater cryotron model

For our first iteration of the model, we choose to use our MDN to model the critical gate current for a given bias current. This approach allows the use of a simpler neural network structure for our MDN when compared to learning the I-V characteristics directly. To accomplish this, we use bias current and current state of the device as input with the MDN trained to predict gate critical current (if the heater cryotron is in its superconducting state) and retrapping current if the heater cryotron is in its resistive state. The results of this model can be seen in Fig. 2d and e, where we show the experimental data as well as our model's output distribution at the mean (μ), and with a variation of twice the standard deviation (σ) in either side ($\mu - 2\sigma$ and $\mu + 2\sigma$) of the mean. We can see that the model is able to capture the critical and retrapping current at different bias currents as well as the variance in these switching points. Another benefit to this approach is that in addition to reduced model complexity, we do not have to run the model every time step as long as the bias current is constant, making the model even more efficient in certain applications.

While the switching model could work well for some devices, it comes with some major drawbacks. Most notably, it requires separate models for the I-V characteristics of the device in its different states in addition to the switching model. This means that this approach is unable to capture the variance in the device's I-V characteristics outside of switching. Going forward, we are going to use the MDN to directly learn the I-V characteristics of the heater cryotron. This means that we will be using I_G, I_B, and state as input to the model, and directly predicting the voltages drop across the load resistor (V_L) as the output.

One challenge with MDNs is that there is no good way to provide easily interpretable numerical metrics for performance. Log-likelihood can be used to compare different models, but it is not a useful metric on its own. Since there are not any existing approaches to compare against, this metric will not be useful for us. Because of this, we will need to rely on graphical comparisons between the model and the experimental data. First, we can look at a comparison between the distribution from the model vs a histogram of the experimental data for different values of I_G and I_B. This is shown in Fig. 3, where we are using I_B of $23.5\mu A$ and values of I_G from $1.35\mu A$ to $1.6\mu A$. Using this range of I_G allows us to see how the switching probability distributions change around the critical current. Figure 3 shows that our probability distributions closely match the distributions obtained from the experiment.

Next, we can compare the switching probability of the model. To do this, we use a sweep of I_G from 0 μ A to 3 μ A for I_B values of 14 μ A, 16.5 μ A, 23.5 μ A, 28 μ A, and 33 μ A. To evaluate the switching probability of the model, we can use the CDF evaluated at the voltage midway between the load voltage when the heater cryotron is in superconducting and resistive states. Figure 4 (left) shows the results of switching probability observed in the experimental data compared to the MDN model. Framing the model in this way allows us to use traditional regression performance metrics, which can be seen in Table 1. On the entire dataset, our model achieves a mean absolute error (MAE) of 0.82% for switching probability with an R^2 of 0.9891. Here, the R^2 value represents the percentage of variance in the switching probability explained by the model. We can compare this to a deterministic neural network such as a multiplayer perception (MLP), which has been proposed for compact modeling by several pervious works[6,7,10]. To do this, we can train the model to predict the threshold gate current for a given bias voltage (I_B). Since this model isn't stochastic, we consider the switching probability to be 0% for any value below the threshold current and 100% above. After testing in this way, the model's MAE is 2.61%, a significant drop in performance compared to our probabilistic approach.

Finally, we can evaluate the performance of the model in transient simulation. So that we can match the output of the model with experimental results, we mirror I_G and I_B from the experimental data as shown in Fig. 4 (right). We use the inverse transform sampling technique as described in section II to achieve the smooth, continuous curves as seen in Fig. 4 (right). To show the distribution of the model in the transient simulation, we report μ, $\mu - 2\sigma$, and $\mu + 2\sigma$ conditions. We can do this by setting the quantile value (q) to 0.5 for the mean and 0.05 and 0.95 for the inverse transform sampling. Figure 4 (right) shows that our MDN model accurately captures the variance in the I-V characteristics of the devices, including the switching dynamics.

Discussion

Our approach utilizing mixture density networks has demonstrated its remarkable ability to accurately model the variance of I-V characteristics and stochastic switching behavior of heater cryotrons. With a 0.82% mean absolute error in the switching probability and an R^2 value of 0.9891, our method has proven its efficacy in

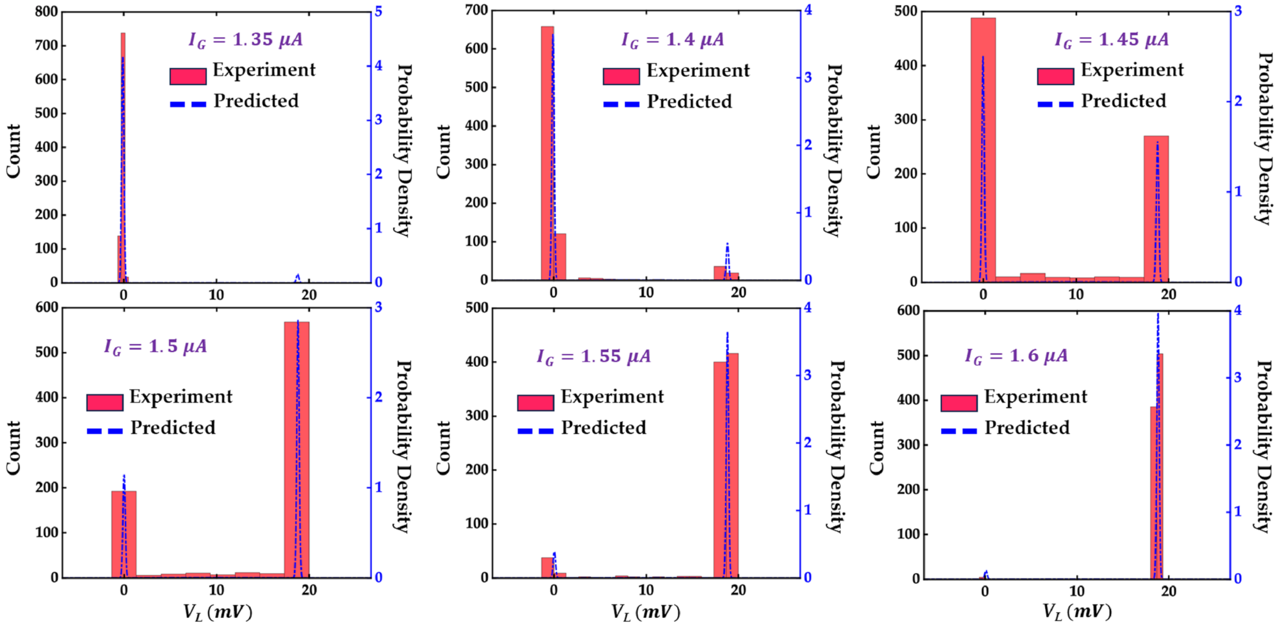

Figure 3. Model validation for load voltage (V_L) against the experiment. The predicted vs experimental distributions of V_L (dropped across a 1 $k\Omega$ load resistor) at a bias current of 23.5 μA and gate current of (**a**) 1.35 μA, (**b**) 1.4 μA, (**c**) 1.45 μA, (**d**) 1.5 μA, (**e**) 1.55 μA, and (**f**) 1.6 μA.

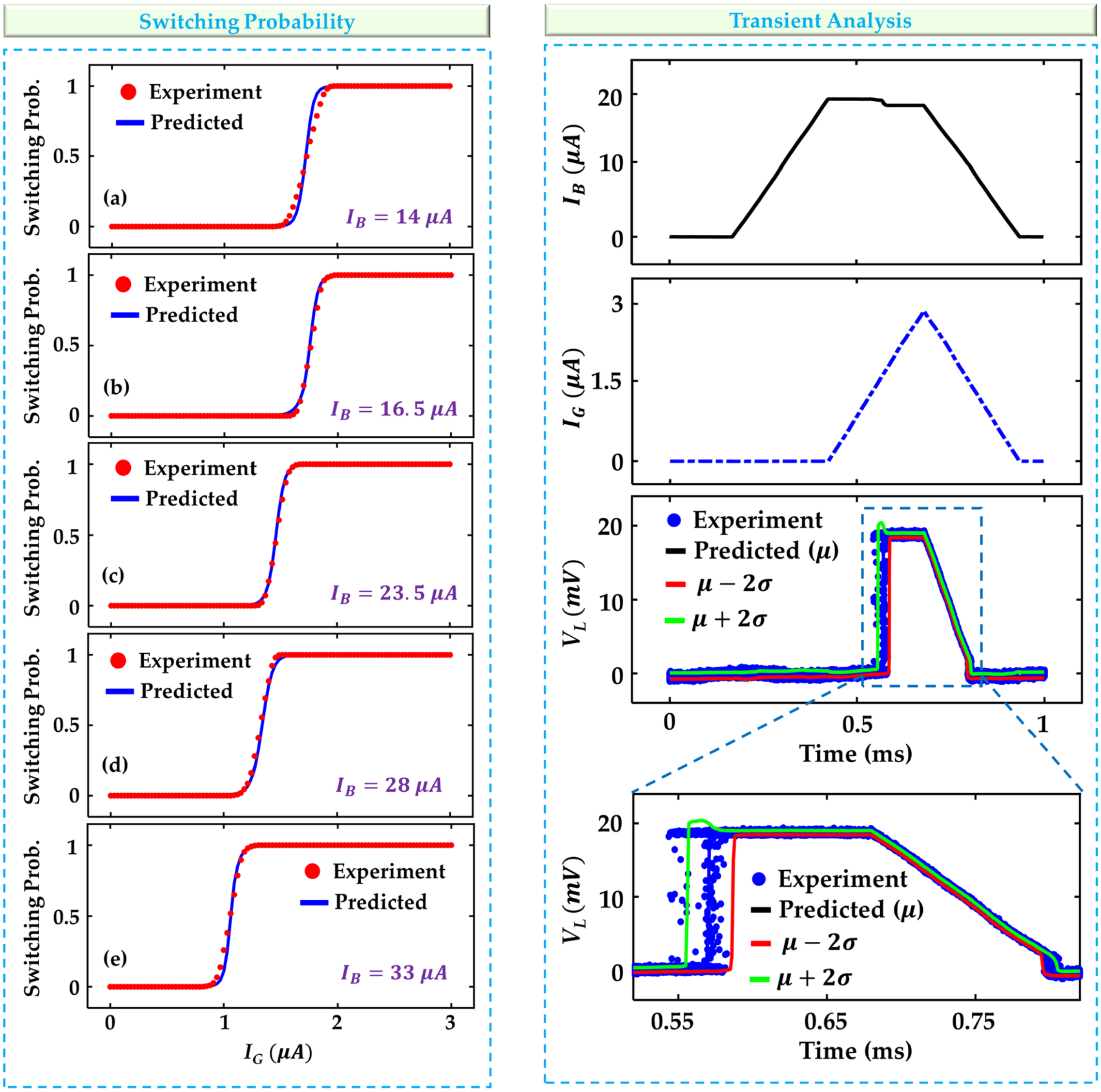

Figure 4. Switching probabilities at varying bias currents and transient simulation results. The predicted vs experimental switching probability for gate current (I_G) between 0 μA and 3 μA at a bias current (I_B) of (**a**) 14 μA, (**b**) 16.5 μA, (**c**) 23.5 μA, (**d**) 28 μA, and (**e**) 33 μA. The increase in I_B leads to a reduction in the gate switching current. Transient dynamics of (**f**) bias current and (**g**) gate current which cause switching of the channel after exceeding the corresponding thresholds. (**h**) Switching is shown and validated with the experiment by the time dynamics of the load voltage (V_L) at mean (μ) and plus or minus 2 standard deviations (σ). (**i**) A zoomed-in view of V_L shown in (**h**).

	MDN (Ours)		MLP	
$I_B(\mu A)$	MAE (%)	R^2	MAE (%)	R^2
14	**1.149**	**0.9941**	2.793	0.9611
16.5	**0.534**	**0.9986**	2.876	0.9334
23.5	**1.344**	**0.9572**	2.874	0.9340
28	**0.611**	**0.9986**	2.054	0.9734
33	**0.662**	**0.9971**	2.457	0.9538

Table 1. MDN and MLP performance of switching probability mean absolute error (MAE) and R^2. Significant values are in bold.

capturing the true variance of device behavior. While we've only shown the model using heater cryotrons, the generalizing power of neural networks means this approach will be easily adaptable to other devices. This achievement marks a significant stride towards more precise and reliable electronic circuit simulations, offering unprecedented opportunities for the development of cutting-edge technologies in an era where stochasticity plays an increasingly pivotal role.

Methods

Heater cryotron fabrication

The fabrication process began with the deposition of a 4 nm layer of superconducting tungsten silicide (WSi) onto a silicon wafer through magnetron sputtering. Following this, contact pads consisting of 5 nm Ti and 30 nm Au were applied using optical lithography and a liftoff procedure. Subsequently, electron beam lithography was employed to define and etch the WSi layer in an RIE process with SF6 chemistry to create the device's channel. A 25 nm layer of SiO_2 was then sputter-deposited across the entire wafer to serve as the dielectric spacer separating the heater from the channel. Another 4 nm layer of WSi was subsequently sputter-deposited, patterned, and etched to form the heater input gate, with contact pads added to this upper layer through the same liftoff process as the initial layer. Finally, openings were etched through the SiO_2 dielectric layer using an RIE process with CHF_3:O_2 chemistry to establish electrical contact with the underlying channel layer.

Heater cryotron characterization

The measurements in this study employed an arbitrary waveform generator (AWG) equipped with two channels and incorporated 10 $k\Omega$ resistors in series with each channel. In the data acquisition process, one channel of the AWG was configured to maintain a constant voltage, thereby delivering a fixed bias current, while the other channel was programmed to incrementally ramp up from zero current to 3 μA. Simultaneously, the voltage of the device channel was meticulously recorded on an oscilloscope. The experimental conditions covered six distinct bias current levels, spanning from 14 μA to 33 μA. Following the ramp-up of gate current, a subsequent phase involved the concurrent reduction of gate and bias currents to get a larger breadth of data. This entire procedure was repeated 1000 times for each bias current setting, ensuring the capture of the stochastic nature of the device.

Model creation

We use Python with TensorFlow to create the mixture density network. We train the model using the data derived in the previous section. We implement a custom loss function to optimize the model for minimizing Gaussian negative log-likelihood. In addition, we implement a custom activation function for exponential linear unit plus 1 plus epsilon as described in equation (3). To integrate this model into circuit simulation, we translate the neural network architecture and the inverse transform sampling methodology into Verilog-A. Subsequently, the trained model weights are extracted using a Python script and seamlessly integrated into the Verilog-A model. This finalized Verilog-A model is compatible with various circuit simulators that support Verilog-A models, such as HSPICE or Spectre. This process is derived from Hutchins et al.[10].

Data availability

The data that support the plots within this paper and other findings of this study are available from the corresponding author upon reasonable request.

Received: 8 December 2023; Accepted: 11 March 2024
Published online: 16 March 2024

References

1. Salahuddin, S., Ni, K. & Datta, S. The era of hyper-scaling in electronics. *Nat. Electron.* **1**, 442–450 (2018).
2. Lundstrom, M. Moore's law forever?. *Science* **299**, 210–211 (2003).
3. Alawad, M. & Lin, M. Survey of stochastic-based computation paradigms. *IEEE Trans. Emerg. Top. Comput.* **7**, 98–114 (2016).
4. Hamilton, T. J., Afshar, S., van Schaik, A. & Tapson, J. Stochastic electronics: A neuro-inspired design paradigm for integrated circuits. *Proc. IEEE* **102**, 843–859 (2014).
5. Li, M., Irsoy, O., Cardie, C. & Xing, H. G. Physics-inspired neural networks for efficient device compact modeling. *IEEE J. Explorat. Solid-State Computat. Devices Circuits* **2**, 44–49. https://doi.org/10.1109/JXCDC.2016.2636161 (2016).
6. Zhang, L. & Chan, M. Artificial neural network design for compact modeling of generic transistors. *J. Comput. Electron.* **16**, 825–832. https://doi.org/10.1007/s10825-017-0984-9 (2017).
7. Wang, J. *et al.* Artificial neural network-based compact modeling methodology for advanced transistors. *IEEE Trans. Electron Devices* **68**, 1318–1325. https://doi.org/10.1109/TED.2020.3048918 (2021).
8. Abouelyazid, M. S., Hammouda, S. & Ismail, Y. Fast and accurate machine learning compact models for interconnect parasitic capacitances considering systematic process variations. *IEEE Access* **10**, 7533–7553. https://doi.org/10.1109/ACCESS.2022.3142330 (2022).
9. Lin, A. S. *et al.* Rram compact modeling using physics and machine learning hybridization. *IEEE Trans. Electron Devices* **69**, 1835–1841. https://doi.org/10.1109/TED.2022.3152978 (2022).
10. Hutchins, J. *et al.* A generalized workflow for creating machine learning-powered compact models for multi-state devices. *IEEE Access*https://doi.org/10.1109/ACCESS.2022.3218333 *(2022).*
11. Wang, Y. *et al.* Compact model of magnetic tunnel junction with stochastic spin transfer torque switching for reliability analyses. *Microelectron. Reliab.* **54**, 1774–1778. https://doi.org/10.1016/j.microrel.2014.07.019 (2014).
12. Becle, E., Talatchian, P., Prenat, G., Anghel, L. & Prejbeanu, I.-L. Fast behavioral veriloga compact model for stochastic mtj. 259–262, https://doi.org/10.1109/ESSDERC53440.2021.9631829 (IEEE, 2021).
13. Bishop, C. *Mixture density networks* (Aston University, 1994).
14. Devroye, L. *Non-Uniform Random Variate Generation* (Springer, 1986).

15. McCaughan, A. N. & Berggren, K. K. A superconducting-nanowire three-terminal electrothermal device. *Nano Lett.* **14**, 5748–5753 (2014).
16. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366. https://doi.org/10.1016/0893-6080(89)90020-8 (1989).
17. Loshchilov, I. & Hutter, F. Decoupled weight decay regularization arXiv:1711.05101 (2019).
18. Brent, R. P. *Chapter 4: An Algorithm with Guaranteed Convergence for Finding a Zero of a Function* (Prentice-Hall, 1973).
19. Alam, S., Hossain, M. S., Ni, K., Narayanan, V. & Aziz, A. Voltage-controlled cryogenic boolean logic family based on ferroelectric squid. Preprint at arXiv:2212.08202 (2022).
20. Alam, S., Rampini, D. S., Oripov, B. G., McCaughan, A. N. & Aziz, A. Cryogenic reconfigurable logic with superconducting heater cryotron: Enhancing area efficiency and enabling camouflaged processors. *Appl. Phys. Lett.***123** (2023).
21. Alam, S., McCaughan, A. N. & Aziz, A. Reconfigurable superconducting logic using multi-gate switching of a nano-cryotron. In *2023 Device Research Conference (DRC)* (ed. Alam, S.) 1–2 (IEEE, 2023).
22. Alam, S., Hossain, M. S., Srinivasa, S. R. & Aziz, A. Cryogenic memory technologies. *Nat. Electron.* **6**, 185–198 (2023).
23. Alam, S., Hossain, M. S. & Aziz, A. A cryogenic memory array based on superconducting memristors. *Appl. Phys. Lett.***119** (2021).
24. Alam, S. *et al.* Cryogenic memory array based on ferroelectric squid and heater cryotron. In *2022 Device Research Conference (DRC)* (ed. Alam, S.) 1–2 (IEEE, 2022).
25. Alam, S., Islam, M. M., Hossain, M. S. & Aziz, A. Superconducting josephson junction fet-based cryogenic voltage sense amplifier. In *2022 Device Research Conference (DRC)* (ed. Alam, S.) 1–2 (IEEE, 2022).
26. Alam, S. *et al.* Cryogenic in-memory matrix-vector multiplication using ferroelectric superconducting quantum interference device (fe-squid). In *2023 60th ACM/IEEE Design Automation Conference (DAC)* (ed. Alam, S.) 1–6 (IEEE, 2023).
27. Islam, M. M., Alam, S., Hossain, M. S., Roy, K. & Aziz, A. A review of cryogenic neuromorphic hardware. *Journal of Applied Physics***133** (2023).
28. Islam, M. M., Alam, S., Hossain, M. S. & Aziz, A. Dynamically reconfigurable cryogenic spiking neuron based on superconducting memristor. In *2022 IEEE 22nd International Conference on Nanotechnology (NANO)* (ed. Islam, M. M.) 307–310 (IEEE, 2022).
29. Islam, M. M., Alam, S., Schuman, C. D., Hossain, M. S. & Aziz, A. A deep dive into the design space of a dynamically reconfigurable cryogenic spiking neuron. *IEEE Transactions on Nanotechnology* (2023).
30. Islam, M. M., Alam, S., Shukla, N. & Aziz, A. Design space analysis of superconducting nanowire-based cryogenic oscillators. In *2022 Device Research Conference (DRC)* (ed. Islam, M. M.) 1–2 (IEEE, 2022).
31. Islam, M. M., Alam, S., Udoy, M. R. I., Hossain, M. S. & Aziz, A. A cryogenic artificial synapse based on superconducting memristor. *Proc. Great Lakes Sympos. VLSI* **2023**, 143–148 (2023).
32. McCaughan, A. N. *et al.* A superconducting thermal switch with ultrahigh impedance for interfacing superconductors to semiconductors. *Nat. Electron.* **2**, 451–456 (2019).

Acknowledgements

This research was funded by the University of Tennessee Knoxville (https://ror.org/020f3ap87) and NIST (https://ror.org/05xpvk416). The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon.

S. A. was supported with funds provided by the Science Alliance, a Tennessee Higher Education Commission center of excellence administered by The University of Tennessee-Oak Ridge Innovation Institute on behalf of The University of Tennessee, Knoxville.

This research was supported in part by seed funding from the AI Tennessee Initiative at the University of Tennessee, Knoxville.

Author contributions

J.H. conceived the idea and developed the modeling framework. J. H. and S.A. created the figures. D.S.R and B.G.O. fabricated and characterized the heater cryotron device. A.M. and A.A. supervised the project. All authors analyzed the results and contributed to writing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2024