Annals of Process Engineering and Management

www.apem.reapress.com

Ann. Proc. Eng. Manag. Vol. 1, No. 1 (2024) 86–105.

Paper Type: Original Article

Machine Learning-Driven Design Optimization of CNFET-Based SRAM Cells

Mohammad Hasan Jali[1], Seyed Mohammad Ali Zanjani[2,*], Mehdi Dolatshahi[2], Behrang Barekatain[1]

[1] Department of Computer Science, Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran; Jali_hasan@sco.iaun.ac.ir; behrang_barekatain@iaun.ac.ir.

[2] Department of Electrical Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran; sma_zanjani@pel.iaun.ac.ir; dolatshahi@pel.iaun.ac.ir.

Citation:

Received: 23 August 2023 *Revised: 13 November 2023* *Accepted: 10 January 2024*	Jali, M. H, Zanjani, S. M. A., Dolatshahi, M., & Barekatain, B. (2024). Machine learning-driven design optimization of CNFET-Based SRAM cells. *Annals of process engineering and management, 1*(1), 86-105.

Abstract

The rapid evolution of nanotechnology has positioned Carbon Nanotube Field-Effect Transistors (CNFETs) as a promising alternative to traditional Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) in the design of low-power, high-performance Static Random-Access Memory (SRAM) cells. However, optimizing the design of CNFET-based SRAM cells to meet stringent requirements, such as stability, power efficiency, and scalability, remains a significant challenge. In this paper, we explore the potential of Machine Learning (ML) algorithms to address these challenges by leveraging their capabilities in complex design optimization tasks. This paper comprehensively reviews various ML models, including supervised, unsupervised, reinforcement, and deep learning. It examines their application in optimizing key performance metrics of CNFET-based SRAM cells. Mathematical models and formulas for design optimization are presented alongside case studies demonstrating ML techniques' effectiveness in improving SRAM stability and reducing power consumption. Finally, we discuss the challenges of integrating ML into circuit design workflows and propose future research directions, highlighting the transformative potential of ML in shaping the future of CNFET-based SRAM design.

Keywords: Carbon nanotube field-effect transistor-based static random-access memory, Machine learning optimization, Design automation, Low-power circuits, Circuit stability, Deep learning in circuit design, Electronic design automation.

1 | Introduction

The primary goal of this review is to explore how ML algorithms can be effectively utilized to optimize the design of Carbon Nanotube Field-Effect Transistor (CNFET)-based Static Random-Access Memory (SRAM) cells. Due to their promising properties, the increasing demand for low-power, high-performance SRAM cells has driven extensive research into CNFET-based designs. However, optimizing these designs for power, stability, and area efficiency poses significant challenges [1], [2]. This review examines various Machine

Corresponding Author: sma_zanjani@pel.iaun.ac.ir

https://doi.org/10.48314/apem.v1i1.27

Learning (ML) models and their mathematical formulations employed to enhance the performance of CNFET-based SRAM cells. CNFETs are considered a promising alternative to conventional Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) in SRAM design due to their superior electrical properties, such as higher carrier mobility, lower power consumption, and scalability to nanoscale dimensions.

CNFETs offer the potential for improved speed and energy efficiency, which are crucial for modern memory applications. Recent studies, such as those by Parvizi and Zanjani [3], have demonstrated that CNFETs can achieve significant power savings and speed improvements over traditional Complementary Metal-Oxide-Semiconductor (CMOS) technology. Moreover, CNFETs are less susceptible to Short-Channel Effects (SCEs) and offer greater flexibility regarding channel length scaling, making them ideal candidates for next-generation memory technologies [4], [5].

The design of CNFET-based SRAM cells presents unique challenges, including variability in carbon nanotube properties, process variations, and the need to balance multiple performance parameters such as read/write stability, access time, and leakage current. Previous studies have explored various design configurations, such as differential and single-ended SRAM cells, to address these challenges [6], [7]. However, achieving optimal performance across all design metrics remains an open problem, prompting the exploration of advanced optimization techniques.

Design optimization is critical for CNFET-based SRAM cells to ensure reliable operation, minimal power consumption, and efficient use of silicon area. Traditional optimization methods, such as circuit simulations and heuristic algorithms, have been employed to improve CNFET-based SRAM designs [8]. However, these methods often require extensive computation and may not fully capture the complex, multi-dimensional design space inherent in CNFET technologies. For example, optimization techniques focusing solely on minimizing power consumption may inadvertently compromise stability or increase read/write delay [9], [10]. Emerging studies suggest that ML algorithms offer a promising solution to these challenges by providing more efficient and scalable approaches to Design Space Exploration (DSE) and optimization. For instance, ML models can learn from vast datasets generated from circuit simulations to predict optimal design parameters that balance power, stability, and performance [11], [12]. This capability is particularly valuable in CNFET-based SRAM designs, where the interplay between various parameters is highly non-linear and difficult to model using traditional methods.

ML has emerged as a powerful tool for optimizing the design of CNFET-based SRAM cells. The use of ML algorithms in this domain is motivated by their ability to handle complex design spaces, model non-linear relationships, and make accurate predictions based on limited or noisy data [13]. Recent studies have explored a variety of ML models, ranging from supervised learning techniques like Support Vector Machines (SVM) and Artificial Neural Networks (ANNs) to unsupervised learning methods such as clustering and Reinforcement Learning (RL) algorithms [14], [15]. Based on design parameters, supervised learning algorithms have been effectively used to predict critical performance metrics of CNFET-based SRAM cells, such as power consumption and read/write delay [16]. For example, Kenarangi and Partin-Vaisband [17] demonstrated the use of SVM to optimize the threshold voltage and channel length of CNFETs to achieve minimal power dissipation while maintaining stability. Similarly, Deep Neural Network (DNN) have been employed to model complex, non-linear dependencies between design variables and performance outcomes, providing a more accurate and comprehensive optimization framework [18]. Unsupervised learning approaches, such as clustering, have been used to identify patterns in design data and group similar SRAM configurations, enabling more targeted optimization strategies [19]. Meanwhile, RL has shown promise in dynamically exploring the design space and learning optimal design policies through iterative simulations and feedback [20]. These approaches leverage the strengths of ML to provide robust and adaptive solutions to the multi-objective optimization problems inherent in CNFET-based SRAM cell design.

This article will systematically examine the application of these ML models to the design optimization of CNFET-based SRAM cells, analyzing their performance, benefits, and limitations. By reviewing recent

advancements and mathematical formulations, this paper aims to provide a comprehensive overview of the state-of-the-art ML-driven techniques for optimizing CNFET-based SRAM designs.

2 | Overview of Carbon Nanotube Field-Effect Transistor-Based Static Random-Access Memory Cells

2.1 | Fundamentals of Carbon Nanotube Field-Effect Transistor Technology

CNFETs represent an emerging technology that leverages the unique electrical properties of Carbon Nanotubes (CNTs) to create high-performance transistors. CNFETs differ from conventional MOSFETs, which use semiconducting Single-Walled Carbon Nanotubes (SWCNTs) as the channel material instead of silicon. This fundamental difference provides several advantages, including higher carrier mobility, lower power consumption, and the potential for miniaturization to nanoscale dimensions.

CNFETs consist of a semiconducting CNT channel positioned between source and drain terminals, with a gate terminal controlling the current flow through the channel. The device operates on the principle of modulating the electrostatic potential of the CNT channel by applying a voltage to the gate. When a voltage is applied, the gate modulates the carrier density in the channel, allowing current to flow (On-state) or preventing it (Off-state), similar to traditional Field-Effect Transistors (FETs). However, due to the quasi-one-dimensional nature of CNTs, CNFETs exhibit ballistic transport characteristics, which significantly reduce scattering effects and enhance carrier mobility [21].

The key parameters affecting CNFET performance are as follows:

I. Threshold voltage V_{th}: The voltage at which the CNFET begins to conduct. This parameter influences the CNT diameter, doping level, and gate dielectric thickness. For example, the threshold voltage is inversely proportional to the CNT diameter, which allows for adjustable Vth through precise control of CNT properties [22].

II. Subthreshold Swing (SS): Defined as the slope of the subthreshold region's current voltage (I-V) curve. A lower SS is desirable for faster switching and lower power consumption. CNFETs exhibit lower SS values than MOSFETs, attributed to the reduced density of states in CNTs [23].

III. Channel length L_{ch}: The distance between the source and drain terminals. Shorter channel lengths enable faster- switching speeds but can introduce SCEs. Due to the ballistic transport properties of CNTs, CNFETs can achieve superior performance even at nanoscale channel lengths, reducing susceptibility to SCEs [24].

IV. Contact resistance R_c: The resistance between the metal contacts and the CNT channel. Reducing Rc is critical for achieving low power consumption and high performance. Optimizing the interface between CNTs and metal contacts can address this [25].

V. Gate dielectric material and thickness: The choice of gate dielectric material and its thickness affect the channel's electrostatic control and leakage current. High-k dielectrics are often used to improve gate control without significantly increasing leakage currents [26].

2.2 | Static Random-Access Memory Cell Design Considerations

Designing SRAM cells using CNFET technology involves several critical considerations to optimize performance metrics such as read/write stability, leakage current, access time, and power consumption. The following sections outline the primary design challenges and the types of SRAM cells commonly used in CNFET-based designs.

Design criteria and challenges are as follows:

I. Read/write stability: Stability during read and write operations is a significant concern in SRAM cell design. The read stability is often characterized by the Static Noise Margin (SNM), the maximum noise voltage

that can be tolerated without flipping the cell's state. CNFET-based SRAM cells must be optimized to maintain a high SNM while minimizing power consumption. Write stability, defined by the Write Margin (WM), is another critical parameter, ensuring the cell can be reliably written under various operating conditions [27].

II. Leakage current: Minimizing leakage currents is crucial for reducing standby power consumption, especially in battery-operated devices. In CNFET-based SRAM cells, leakage current is primarily due to subthreshold leakage and gate tunneling leakage. Optimizing the threshold voltage and gate dielectric material can help mitigate these leakages [28].

III. Access time: Access time is required to perform a read or write operation. This parameter is influenced by the transistor switching speed, which is determined by the carrier mobility, channel length, and threshold voltage of CNFETs. A key design challenge is to balance low access time and high stability [29].

IV. Power consumption: Both dynamic power (Related to charging and discharging capacitive loads during switching) and static power (Due to leakage currents) are important considerations. CNFET-based SRAM cells benefit from CNFETs' low switching power, but careful optimization is needed to reduce static power without compromising other performance metrics [30].

Types of SRAM cells and their performance trade-offs are as follows:

I. 6T SRAM cell: The standard 6-transistor (6T) SRAM cell consists of two cross-coupled inverters (Each consisting of a pull-up and a pull-down CNFET) and two access CNFETs. This design is commonly used due to its simple architecture and low power consumption. However, it may suffer from poor read stability and be sensitive to variations in CNFET parameters, necessitating optimization using ML algorithms [31].

II. 8T SRAM cell: The 8-transistor (8T) SRAM cell introduces two additional transistors to separate the read and write paths, improving read stability and reducing the likelihood of read disturbance. This design increases the area overhead, enhances the read margin, and allows faster read operations [32].

III. 10T SRAM cell: The 10-transistor (10T) SRAM cell further decouples the read and write operations and includes extra transistors to reduce leakage currents and enhance robustness against process variations. This cell type is particularly suitable for low-power applications, where maintaining stability and minimizing power consumption are critical [33].

IV. 12T SRAM cell: The 12-transistor (12T) SRAM cell further enhances stability and performance by incorporating additional transistors to separate read and write operations completely, providing independent control over both. This architecture significantly improves read and write stability, especially under low-voltage operation, while reducing soft error rates. Including extra transistors also minimizes leakage currents, enhancing noise immunity and reliability in process variations. While the 12T cell increases area and design complexity, it is ideal for high-performance, low-power applications requiring robust operation in harsh environments [34].

The design optimization of CNFET-based SRAM cells requires mathematical models that accurately represent the electrical characteristics of CNFETs. For example, the drain current I_d of a CNFET can be expressed as:

$$I_d = \frac{W}{L} \cdot \mu \cdot C_{ox} \cdot \left(V_{gs} - V_{th}\right) \cdot V_{ds}, \quad \textbf{(1)}$$

where: 1) W is the width of the CNT channel, 2) L is the channel length, 3) μ is the carrier mobility, 4) Cox is the oxide capacitance per unit area, 5) Vgs is the gate-source voltage, 6) Vth is the threshold voltage, and 7) Vds is the drain-source voltage.

These models, along with other key formulas, will be critical in understanding the optimization potential offered by ML algorithms, as discussed in subsequent sections.

3 | Machine Learning in Circuit Design: A Brief Overview

3.1 | Introduction to Machine Learning Algorithms

ML has emerged as a powerful tool for optimizing and automating complex design processes in Electronic Design Automation (EDA). ML algorithms enable the analysis and extraction of patterns from large datasets, particularly useful for optimizing circuit designs, such as CNFET-based SRAM cells. This section overviews various ML algorithms relevant to circuit design optimization.

Supervised learning

Supervised learning algorithms are trained using labeled datasets, where the input features (e.g., design parameters) and the corresponding output labels (e.g., power consumption, delay) are known.

Common supervised learning algorithms include:

I. Linear regression predicts continuous output variables based on input features. For example, linear regression can model the relationship between a CNFET's width and length and its resulting power consumption or access time [35].

II. SVM: Effective for classification problems, such as categorizing different SRAM cell designs based on their stability or power efficiency. SVM can create a hyperplane in a high-dimensional space to separate different classes of designs [36].

III. ANN: ANNs are computational models inspired by the human brain's structure and function. They are particularly useful for capturing complex, non-linear relationships between design variables and performance metrics. In CNFET-based SRAM design, ANNs can predict multiple performance characteristics simultaneously [37].

Unsupervised learning

Unsupervised learning algorithms work with unlabeled datasets to identify inherent patterns or groupings within the data. Examples include:

Clustering algorithms (e.g., K-Means): These are used to group similar SRAM designs based on performance metrics like stability, power consumption, or leakage current. Clustering can help identify optimal design clusters or regions within the design space [38].

Principal Component Analysis (PCA) is a dimensionality reduction technique that reduces the number of variables under consideration, simplifying the complexity of the design space. PCA can help identify the most influential parameters in CNFET SRAM design [39].

Reinforcement learning

RL involves training an agent to make sequential decisions to maximize a reward. RL can optimize SRAM design in circuit design by learning the best sequence of design modifications to achieve a target performance metric, such as minimizing power consumption or maximizing read/write stability [40].

Q-learning: A model-free RL algorithm used to learn the value of actions taken in specific states, useful for iterative design optimization tasks where direct supervision is not feasible [41].

Deep Q-Networks (DQN): This extension of Q-Learning uses DNN to approximate the optimal action-value function, optimizing high-dimensional design spaces, such as those encountered in CNFET SRAM design [42].

Deep learning

Deep learning is a subset of ML that uses DNNs to model complex patterns and relationships.

Convolutional Neural Networks (CNNs): Traditionally used in image recognition, CNNs can extract spatial hierarchies and patterns in circuit layouts, aiding in optimizing CNFET-based SRAM cells [43].

Recurrent Neural Networks (RNNs) are useful for time-series prediction tasks. RNNs can predict temporal variations in SRAM cell performance, such as degradation over time due to aging effects [44].

3.2 | Application of Machine Learning in Electronic Design Automation

ML techniques are increasingly integrated into EDA tools to enhance circuit design and optimization. The use of ML in EDA encompasses a range of applications, such as DSE, optimization, and prediction of circuit performance. Below are some specific use cases:

Design space exploration

DSE involves evaluating various design configurations to identify the optimal solution. ML algorithms like Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO) can efficiently explore significant and complex design spaces by mimicking natural evolution and social behavior, respectively [45]. For instance, GAs can optimize CNFET SRAM cell design by iteratively selecting, crossing, and mutating design parameters (e.g., channel length, CNT diameter) to converge on the best-performing solution [46]. The objective function to be optimized can be formulated as:

$$\begin{aligned}&\text{Objective Function} = \alpha \cdot \text{Power Consumption} \\ &+\beta \cdot \text{Read Stability} - \gamma \cdot \text{Access Time},\end{aligned} \tag{2}$$

where α, β, γ are weighting factors representing the relative importance of each parameter.

Circuit performance prediction

ML models, such as ANNs or Gaussian Process Regressions (GPR), can be trained to predict the performance metrics (e.g., delay, power, area) of CNFET-based SRAM cells based on various input design parameters [47]. This enables rapid estimation of circuit performance without the need for exhaustive simulations. For example, an ANN model can take the following inputs:

Input vector: $[L_{ch}, W_{ch}, V_{th}, V_{dd}, \epsilon_{ox}]$.

Output: $[\text{Power}, \text{Delay}, \text{SNM}]$.

The trained ANN model can accurately predict outputs, significantly reducing the time required for performance analysis [48].

Optimization of power, performance, and area

Power, Performance, and Area (PPA) optimization is critical in circuit design. ML algorithms such as RL can dynamically adjust design parameters to achieve an optimal balance between PPA metrics [49]. An RL agent can be trained to maximize a reward function defined as:

$$\begin{aligned}&R(s,a) = w_1 \cdot (\text{Reduction in Power}) \\ &+w_2 \cdot (\text{Improvement in Stability}) - w_3 \cdot (\text{Increase in Area}),\end{aligned} \tag{3}$$

where: s represents the state (Current design configuration), a represents the action (Modification to design parameters), and w_1, w_2, and w_3 are weighting factors.

Layout optimization and physical design

ML models and intense learning techniques like CNNs can assist in optimizing the layout and physical design of CNFET-based SRAM cells by identifying optimal transistor placements and routing paths. This reduces parasitic effects and improves overall cell performance [50]. A CNN model can be trained on a dataset of layout designs, learning the spatial correlations between component placements and performance metrics. The output can guide designers in choosing the most efficient layout configuration. By integrating ML algorithms into the EDA process, designers can automate and enhance the optimization of CNFET-based

SRAM cells, leading to better-performing, power-efficient, and stable circuits. The following section will delve into specific case studies where these ML techniques have been successfully applied to optimize CNFET SRAM designs.

4 | Machine Learning Models for Carbon Nanotube Field-Effect Transistor-Based Static Random-Access Memory Design Optimization

In this section, we will explore various ML models that are particularly effective for optimizing the design of CNFET-based SRAM cells. We will discuss the application of supervised, unsupervised, RL, and deep learning approaches, highlighting their mathematical formulations and specific use cases in design optimization.

4.1 | Supervised Learning Algorithms

Supervised learning algorithms are particularly useful for predicting performance metrics of CNFET-based SRAM cells based on input design parameters. These models are trained on datasets containing labeled examples, where the input features (Such as transistor dimensions, supply voltage, etc.) are associated with known output performance metrics (Like power consumption, delay, and stability).

Linear regression and support vector machine

Linear regression: Linear regression can establish a linear relationship between the design parameters (e.g., channel length, threshold voltage) and performance metrics (e.g., delay, power). For instance, linear regression can model the impact of increasing the channel length L_{ch} of a CNFET on the overall power consumption P of an SRAM cell. The general form of the linear regression model is:

$$P = \beta_0 + \beta_1 L_{ch} + \beta_2 V_{th} + \beta_3 W_{ch} + \ldots + \epsilon \quad \textbf{(4)}$$

where:

I. P represents the predicted power consumption.

II. β_0, β_1, β_2, … are the regression coefficients.

III. L_{ch}, V_{th}, and W_{ch} are the design parameters (Channel length, threshold voltage, and channel width).

IV. ε is the error term.

By minimizing the error term ε through least squares fitting, the regression model can provide insights into the influence of individual design parameters on performance metrics. This can guide optimization by indicating which parameters significantly impact power or delay.

SVM: SVMs can classify SRAM cell designs into different categories based on their performance characteristics (e.g., stable vs. unstable designs). The SVM algorithm aims to find the optimal hyperplane that maximizes the margin between different classes in a high-dimensional feature space. The general form of the SVM model is:

$$\text{minimize} \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i . \quad \textbf{(5)}$$

s.t.

$$y_i (w \cdot x_i + b) \geq 1 - \xi_i , \xi_i \geq 0. \quad \textbf{(6)}$$

I. w is the weight vector of the hyperplane.

II. C is the regularization parameter.

III. ξ_i are the slack variables for soft margin classification.

IV. y_i are the class labels, and x_i are the feature vectors representing design parameters.

SVM can help optimize CNFET SRAM designs by categorizing them based on performance metrics and identifying the optimal class boundaries [51], [52].

Artificial neural network

ANNs are highly effective for modeling complex non-linear relationships between CNFET design parameters and performance outcomes, such as power consumption, delay, and stability. They are capable of learning from data and generalizing to unseen design configurations. An ANN consists of multiple layers (Input, hidden, and output layers) with interconnected neurons. The mathematical representation of a simple feedforward neural network with one hidden layer can be given by

$$\begin{cases} h = \sigma(W_1 x + b_1), \\ \hat{y} = W_2 h + b_2, \end{cases} \quad (7)$$

where:

I. x is the input vector representing design parameters.

II. W_1 and W_2 are weight matrices for the hidden and output layers.

III. b_1, and b_2 are bias vectors.

IV. $\sigma(.)$ is the activation function (e.g., ReLU, Sigmoid).

V. $\hat{y}$ is the predicted output (e.g., power consumption).

The ANN can be trained using backpropagation to minimize the Mean Squared Error (MSE) between the predicted and actual performance metrics:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad (8)$$

where: 1) y_i is the actual value, 2) $\hat{y}$ is the predicted value, and 3) n is the number of training samples. This approach can predict performance metrics for new CNFET SRAM designs, accelerating the optimization process [53].

4.2 | Unsupervised Learning Algorithms

Unsupervised learning algorithms can identify patterns or clusters within design data without requiring labeled datasets. These methods are particularly useful for grouping similar design configurations based on performance characteristics.

K-Means clustering: K-Means clustering can partition CNFET SRAM designs into distinct groups based on similarities in design parameters or performance metrics (e.g., leakage current, read stability). The objective of K-Means is to minimize the Within-Cluster Sum of Squares (WCSS):

$$\text{minimize} \sum_{k=1}^{K}\sum_{i \in C_k} \| x_i - \mu_k \|^2, \quad (9)$$

where:

I. K is the number of clusters.

II. x_i is the data point representing a design.

III. μ_k is the centroid of cluster C_k.

By clustering designs with similar performance metrics, designers can identify regions in the design space likely to yield optimal results.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN): DBSCAN is useful for identifying clusters of high-density points in the design space, representing optimal or near-optimal design configurations. DBSCAN defines clusters based on the density of points and identifies noise (Outliers). It requires ε (Maximum distance between points in a cluster) and MinPts (Minimum number of points to form a cluster). DBSCAN can effectively handle noise in design data and identify dense regions that represent promising design configurations [54].

4.3 | Reinforcement Learning for Design Optimization

RL techniques can be employed to optimize design parameters by learning from iterative exploration and feedback [55], [56].

Policy-based methods (e.g., policy gradients): Policy-based RL methods learn a parameterized policy $\pi(a|s;\theta)$ that directly maps design states s (e.g., current parameter values) to actions a (e.g., adjustments to parameters) to maximize a cumulative reward. The objective is to maximize the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T}\gamma^{t} r_{t}\right], \quad (10)$$

where:

I. θ represents policy parameters.

II. γ is the discount factor.

III. $-r_t$ is the reward at time step t.

Policy gradients can be used to find an optimal policy that improves the performance of CNFET-based SRAM designs over time through trial and error.

Value-based methods (e.g., Q-learning): Value-based methods like Q-learning aim to learn the optimal value function Q(s, a) that estimates the maximum cumulative reward achievable from a state s by taking an action. The Q-learning update rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right], \quad (11)$$

where:

I. α is the learning rate.

II. r is the immediate reward.

III. s' is the next state.

Q-learning can optimize SRAM design parameters by iteratively adjusting them to maximize performance metrics.

4.4 | Deep Learning Approaches

Deep learning models like CNNs and RNNs can extract complex features from design data and predict long-term reliability and performance metrics.

CNN*:* CNNs can be used to analyze spatial patterns in the design parameter space, such as spatial correlation in transistor layouts. A CNN consists of convolutional layers that apply filters to input data to detect features:

$$x_{ij}^{l} = \sigma\left(\sum_{m,n} x_{(i+m)(j+n)}^{l-1} \cdot w_{mn}^{l} + b^{l} \right), \quad (12)$$

where:

I. x^{l}_{ij} is the convolutional layer output at position (i, j).

II. w^{l}_{mn} are the weights of the filter.

III. σ is the activation function.

CNNs can predict which SRAM designs will have favorable performance characteristics based on spatial data patterns [57], [58].

RNN*:* RNNs are suitable for predicting temporal reliability or degradation over time due to their capability to handle sequential data [59]. An RNN computes hidden states that capture temporal dependencies:

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h), \quad (13)$$

where:

I. h_t is the hidden state at time t.

II. x_t is the input at time t.

III. W_h, U_h are weight matrices.

IV. σ is the activation function.

RNNs can model the long-term reliability of CNFET-based SRAM designs, allowing designers to optimize for durability and stability over time. By leveraging these ML models, the design and optimization of CNFET-based SRAM cells can be significantly accelerated, leading to designs that offer enhanced performance, reliability, and energy efficiency [60].

5 | Case Studies and Comparative Analysis

This section delves into two specific case studies that highlight the application of ML algorithms in optimizing CNFET-based SRAM cell designs. The first case study focuses on enhancing SRAM stability, while the second addresses reducing power consumption. A detailed comparative analysis follows to evaluate the strengths and weaknesses of different ML models in various design optimization scenarios [36], [61]–[65].

5.1 | Case Study 1. Optimization of Static Random-Access Memory Stability Using Machine Learning Algorithms

Stability is a paramount design objective for CNFET-based SRAM cells, particularly when aiming for reliable operation under scaled-down technology nodes. Stability optimization maximizes the SNM while balancing parameters like delay and power consumption.

Objective

To maximize the SNM of a 6T CNFET-based SRAM cell while maintaining acceptable read and write delays.

Machine learning algorithms employed

Random Forest (RF): Utilized to predict the SNM based on input design parameters (Such as channel length, threshold voltage, etc.).

GA: Used to optimize the design parameters to achieve the highest possible SNM with minimal delay trade-offs.

Optimization approach

Data preparation: A dataset of simulated SRAM designs was generated using HSPICE, varying key parameters like channel length (L_{ch}), threshold voltage (V_{th}), channel width (W_{ch}), and supply voltage (V_{dd}). These parameters were chosen because they significantly impact SRAM stability.

RF model training: The RF model was trained to predict SNM values based on the input features. The model's feature importance analysis indicated that V_{th} and L_{ch} were the most influential factors in determining SNM.

GA optimization: The GA was initialized with a diverse population of SRAM designs, and a fitness function was defined to maximize SNM while penalizing configurations with excessive delay. The algorithm iteratively evolved the population over 150 generations to converge to an optimal design.

The optimal design achieved a 15% increase in SNM compared to the baseline, with only a 5% increase in delay. The GA's convergence demonstrated the effectiveness of

combining predictive modeling (RF) with heuristic optimization (GA) for stability optimization.

5.2|Case Study 2. Power Consumption Reduction in Carbon Nanotube Field-Effect Transistor-Based Static Random-Access Memory Design

Minimizing power consumption in CNFET-based SRAM cells is crucial, especially for portable and battery-operated devices. This case study explores ML techniques to optimize power efficiency while maintaining acceptable performance levels.

Objective

To minimize static and dynamic power consumption in an 8T CNFET-based SRAM cell while ensuring a minimum acceptable speed and stability.

Machine learning techniques applied

SVM: Applied for classification to identify feasible low-power designs based on initial constraints.

DNN: Employed for regression tasks to predict power consumption metrics and guide the optimization process.

PSO: Utilized for multi-objective optimization to fine-tune design parameters based on DNN predictions.

Optimization approach

Data collection: The dataset was generated by simulating numerous SRAM configurations using Cadence Virtuoso, with variations in gate length, oxide thickness, V_{dd}, and temperature.

SVM Classification: An SVM with a Radial Basis Function (RBF) kernel was trained to classify designs as "Low Power" or "High Power," achieving an 89% classification accuracy.

DNN Regression: A DNN with three hidden layers was trained to predict static and dynamic power consumption, achieving a Mean Absolute Percentage Error (MAPE) of 3.5%.

PSO Optimization: PSO was used to search for the optimal parameter set to minimize power consumption while maintaining performance constraints.

The optimized SRAM design achieved a 22% reduction in static power and a 17% reduction in dynamic power, maintaining desired speed and stability metrics. PSO effectively refined the parameter search space based on DNN predictions.

5.3|Comparative Analysis of Different Machine Learning Models

This section compares the strengths and weaknesses of the different ML algorithms used in the two case studies and evaluates their effectiveness in various CNFET-based SRAM design optimization scenarios. The results of these comparisons are shown in *Tables 1-3*.

Table 1. Comparison of machine learning algorithms for static random-access memory stability optimization.

ML Model	Strengths	Weaknesses	Performance in Stability Optimization
RF	High interpretability, robust to overfitting, good for feature importance ranking	Requires large datasets, less effective for high-dimensional data	Effective for predicting SNM, identifying key parameters
GA	Effective for global optimization, handles multi-objective problems well	Computationally expensive, slow convergence in complex spaces	Achieved significant SNM improvement but slow convergence
SVM	Strong for binary classification tasks, handles non-linear separations well	Sensitive to parameter tuning, less effective with noisy data	Suitable for classifying feasible SRAM configurations
DNN	Excellent at capturing complex, non-linear relationships with high accuracy	Requires substantial training data, risk of overfitting, computationally intensive	Effective for predicting performance metrics, power consumption
PSO	Fast convergence, suitable for continuous optimization problems	It may get stuck in local minima and requires good parameter initialization	Efficient in power optimization, balancing multiple objectives

Table 2. Comparative analysis: Effectiveness of machine learning algorithms in power optimization.

Criteria	RF	GA	SVM	DNN	PSO
Computational efficiency	Moderate (Training and inference fast)	Low (Computationally intensive optimization)	High (Fast classification)	Low to moderate (Depending on model size)	High (Fast convergence, low compute cost)
Accuracy	High (For initial predictions)	High (After convergence)	Moderate (Depends on data quality)	Very high (With sufficient data)	High (Given good initial parameters)
Ease of implementation	Moderate (Requires parameter tuning)	Moderate to high (Complex optimization setup)	High (Straightforward binary classification)	Low to moderate (Complex models, training)	Moderate (Requires domain expertise)
Data Requirements	Large datasets needed for training	Moderate (Depends on fitness function design)	Low to Moderate (Small to medium datasets)	Very High (Large training datasets required)	Moderate (Sufficient initial samples needed)
Scalability	Moderate (Scales well with data volume)	Low (Difficult to scale due to computational cost)	High (Scales well with additional features)	Moderate to High (Can scale with neural architecture)	High (Adaptable to different scales)
Interpretability	High (Provides feature importance)	Low (Black-box nature)	Moderate to High (With linear/non-linear kernels)	Low (Complex to interpret neural weights)	Moderate (Depends on implementation details)

Table 3. Summary of machine learning models' performance across case studies.

Optimization Goal	Best Performing ML Model	Rationale
SRAM Stability Optimization	Combination of RF + GA	RF provides insight into key parameters affecting SNM, while GA effectively explores the global design space for optimal solutions.
Power Consumption Reduction	Combination of SVM + DNN + PSO	SVM quickly identifies feasible low-power configurations, DNN accurately predicts power metrics, and PSO efficiently fine-tunes the design.

5.4|Detailed Observations and Insights

Trade-offs in model selection:

I. RF is useful for initial exploration and feature selection due to its interpretability, but it is less suited for high-dimensional optimization tasks without additional heuristic methods.

II. GA is powerful for global optimization but may require extensive computational resources, especially in highly complex design spaces.

III. SVM provides fast, reliable classification but may not capture complex relationships unless paired with more robust models.

IV. DNN offers high accuracy in predicting complex non-linear outcomes but requires careful tuning and large datasets to avoid overfitting.

V. PSO is efficient for continuous optimization tasks and can complement predictive models by fine-tuning design parameters based on their outputs.

Combining models for enhanced results:

I. Combining ML models like RF with GA or DNN with PSO can leverage their strengths, such as using RF for rapid feature selection and GA for extensive parameter optimization.

II. Hybrid approaches, where different models handle different aspects of the design process (e.g., SVM for initial classification and PSO for optimization), effectively balance performance metrics like speed, accuracy, and power consumption.

6|Challenges and Future Directions

Numerous opportunities have arisen with the advancement of CNFET technology and the growing interest in using ML techniques for SRAM design optimization. However, researchers and engineers must address several challenges. This section outlines the primary obstacles in applying ML to CNFET-based SRAM design, followed by a discussion of future trends and emerging techniques in this interdisciplinary domain.

6.1|Challenges in Applying Machine Learning to Carbon Nanotube Field-Effect Transistor-Based Static Random-Access Memory Design

The integration of ML models into CNFET-based SRAM design presents several unique challenges, which stem from both the technology's complexity and the limitations of current ML approaches.

Data availability and quality

One of the primary challenges in applying ML to CNFET-based SRAM design is the availability of high-quality datasets. Extensive datasets of simulated or experimental SRAM designs with varied parameters (e.g., transistor dimensions, threshold voltages, supply voltages) are required to train ML models effectively. These datasets should cover various operational conditions, including temperature variations, process variations, and different workload scenarios.

Challenge: CNFET technology is still in the research phase, and experimental data is relatively scarce compared to traditional MOSFET technology. Simulation tools for CNFETs, such as HSPICE or Cadence, are computationally intensive, limiting the generation of large datasets.

Solution directions: Developing efficient surrogate models to approximate the behavior of CNFETs can reduce the computational load associated with simulations. Techniques like generative models (e.g., variational autoencoders or GANs) could create synthetic datasets that mimic real design scenarios.

Model Interpretability

ML models, intense learning approaches, are often considered "black-box" models. This lack of interpretability can be problematic in the highly controlled and precise environment of circuit design, where understanding the relationship between input parameters and output performance metrics is critical.

Challenge: Engineers must understand how specific parameters (e.g., channel length, gate capacitance) influence performance metrics (e.g., power consumption, SNM). Complex ML models like DNNs do not provide direct insights into these relationships.

Solution directions: Techniques like SHapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) can improve model interpretability by providing insights into each feature's contribution to the final prediction.

Another potential direction is the integration of simpler, more interpretable models such as decision trees or RFs alongside more complex ML methods for optimization tasks, balancing accuracy with transparency.

Computational complexity

Optimizing CNFET-based SRAM cells with ML techniques can be computationally expensive due to the vast parameter search space and the need for iterative simulations or optimizations. This computational burden is further compounded when exploring multi-objective optimization problems (e.g., power, delay, and stability trade-offs).

Challenge: The high computational cost associated with training models and running optimization algorithms for large-scale designs limits the practical application of ML in real-world environments.

Solution directions: Distributed computing and cloud-based platforms can be leveraged to parallelize the training and optimization processes.

In addition, model compression techniques, such as pruning or quantization, can reduce the computational overhead of using large neural networks without significantly sacrificing accuracy.

Integration with existing electronic design automation tools

Integrating ML techniques into existing EDA workflows for CNFET-based SRAM designs remains a significant challenge. EDA tools such as Cadence, Synopsys, and HSPICE are traditionally used for physical layout, timing analysis, and verification, but most do not natively support ML-based design optimization.

Challenge: The lack of a seamless interface between ML frameworks (e.g., TensorFlow, PyTorch) and EDA tools makes it difficult for designers to apply ML-driven optimizations in their existing workflows.

Solution directions: Future development of specialized plugins or APIs to integrate ML models directly into EDA tools could streamline the design process. For instance, ML models could be embedded within the synthesis and simulation steps, allowing real-time optimization as part of the standard design flow.

6.2 | Future Trends in Machine Learning for Circuit Design Optimization

Despite the challenges, ML-based design optimization for CNFET-based SRAM cells holds immense promise for future research and practical applications. This section explores emerging trends and potential advancements in the field.

Emerging machine learning algorithms for circuit design

As ML continues to evolve, new algorithms and frameworks are being developed to address the specific needs of circuit design.

RL: RL has shown significant promise in autonomous decision-making and optimization problems. In CNFET-based SRAM design, RL could optimize circuit parameters in a dynamic, iterative fashion, where the system learns to optimize for multiple objectives (e.g., power, delay, area) by interacting with the environment.

Federated Learning (FL): FL is a decentralized ML approach that allows models to be trained across multiple devices without sharing raw data. This can be particularly beneficial in CNFET-SRAM optimization, where design data from different fabrication plants can improve model accuracy while maintaining data privacy.

Hybrid machine learning models

Hybrid ML models, which combine the strengths of different algorithms, are emerging as a powerful tool in circuit design optimization.

A hybrid approach combining RL for dynamic parameter tuning and GA for global exploration could achieve better design outcomes for CNFET-based SRAM cells by efficiently navigating the complex design space while minimizing computational overhead. Let's assume a fitness function F(x) for optimizing the delay and power consumption of an SRAM cell:

$$F(x) = w_1 \times \text{delay}(x) + w_2 \times \text{power}(x), \quad \textbf{(14)}$$

where w_1 and w_2 are weighting factors. A hybrid RL-GA algorithm can optimize this function by:

I. Using RL to adjust the parameters x dynamically.

II. Utilizing GA to explore different initial configurations based on the fitness function F(x).

Quantum computing for machine learning in circuit design

Quantum computing is poised to revolutionize various fields, and its integration with ML could provide significant advances in circuit design optimization.

Potential applications: Quantum computing could dramatically reduce the time required for simulating complex circuits, such as CNFET-based SRAM cells. By leveraging Quantum Machine Learning (QML), designers could solve optimization problems that are currently computationally intractable using classical approaches.

Quantum Support Vector Machines (QSVMs) and Quantum Neural Networks (QNNs) could be applied to classify and optimize SRAM designs much faster than their classical counterparts, especially when dealing with high-dimensional parameter spaces.

Automated design space exploration

As the complexity of CNFET-based SRAM designs increases, automated DSE tools that leverage ML will become indispensable. These tools will automate the exploration of large design spaces, significantly reducing the time needed for optimization.

Automated DSE: By integrating ML models into EDA workflows, automated DSE could evaluate millions of design variations in parallel. Techniques like Bayesian optimization could efficiently search for optimal design parameters by balancing exploration and exploitation of the design space.

Sustainability and energy-efficient design

Future research in ML-driven design optimization for CNFET-based SRAMs will increasingly focus on sustainability and energy efficiency. The growing demand for low-power, environmentally friendly devices will drive the development of optimization algorithms that prioritize energy efficiency without sacrificing performance.

Green computing trends: ML models designed to optimize CNFET-based SRAM cells for low power consumption will be aligned with green computing initiatives, further emphasizing the importance of developing energy-efficient circuits.

Despite the challenges of data availability, computational complexity, and integration into existing design tools, ML has shown tremendous potential for optimizing CNFET-based SRAM designs. Future research will likely focus on developing more interpretable, efficient, and integrated ML models. Emerging trends such as hybrid models, RL, and quantum computing will redefine the landscape of circuit design optimization, paving the way for more advanced and efficient CNFET-based SRAM technologies.

7 | Conclusion

The application of ML to the design optimization of CNFET-based SRAM cells offers significant advancements in stability, power efficiency, and performance. Designers can efficiently explore complex design spaces and address multi-objective optimization challenges by leveraging algorithms like neural networks, RL, and SVM. ML models enable the fine-tuning of SRAM parameters, reducing power consumption, improving stability, and better performance, especially in low-power applications such as IoT devices. The transformative potential of ML in SRAM design is clear, as it automates traditionally manual and time-consuming processes. ML accelerates the design process and allows for the discovery of novel solutions that maximize performance while minimizing costs. With the future integration of quantum computing and advanced ML techniques, the role of ML in optimizing CNFET-based SRAM cells is expected to grow, driving further innovation in the design of next-generation electronic systems.

Funding

This research was conducted without the support of any external grants or institutional funding. The work reflects the independent efforts and academic initiative of the research team, driven by the goal of advancing the state-of-the-art in low-power memory design and machine learning applications in nanoelectronics.

Conflicts of Interest

The authors declare that there are no financial or personal relationships that could have influenced the work reported in this paper. All analyses and interpretations have been conducted with academic neutrality and integrity.

References

[1] Kavitha, S., Kumar, C., Fayek, H. H., & Rusu, E. (2023). Design and implementation of CNFET SRAM cells by using multi-threshold technique. *Electronics*, *12*(7), 1–23. https://doi.org/10.3390/electronics12071611

[2] Gadgil, S., Sandesh, G. N., & Vudadha, C. (2023). Power efficient designs of CNTFET-based ternary SRAM. *Microelectronics journal*, *139*, 105884. https://doi.org/10.1016/j.mejo.2023.105884

[3] Parvizi, M., & Zanjani, S. M. A. (2020). An efficient, current-mode full-adder based on majority logic in CNFET technology. *2020 10th international conference on computer and knowledge engineering (ICCKE)* (pp. 301–305). IEEE. http://dx.doi.org/10.1109/ICCKE50421.2020.9303623

[4] Chen, R., Chen, L., Liang, J., Cheng, Y., Elloumi, S., Lee, J., … & Todri-Sanial, A. (2022). Carbon nanotube SRAM in 5-nm technology node design, optimization, and performance evaluation—Part I: CNFET transistor optimization. *IEEE transactions on very large scale integration (VLSI) systems*, *30*(4), 432–439. https://doi.org/10.1109/TVLSI.2022.3146125

[5] Mohammed, M. U. (2019). *Novel high performance ultra low power static random access memories (SRAMs) based on next generation technologies* [Thesis].

[6] Patel, P. K., Malik, M. M., & Gupta, T. K. (2020). A novel high-density dual threshold GNRFET SRAM design with improved stability. *Microprocessors and microsystems*, *73*, 102956. https://doi.org/10.1016/j.micpro.2019.102956

[7] Valluria, A., & Musala, S. (2022). Comparative analysis of CNTFET based SRAM cells. *ECS transactions*, *107*(1), 19177. https://doi.org/10.1149/10701.19177ecst

[8] Sajjadi, S. A., Sadrossadat, S. A., Moftakharzadeh, A., Nabavi, M., & Sawan, M. (2024). Yield maximization of flip-flop circuits based on deep neural network and polyhedral estimation of nonlinear constraints. *IEEE access*, *12*, 113944–113959. https://doi.org/10.1109/ACCESS.2024.3443343

[9] Zahoor, F., Jaber, R. A., Isyaku, U. B., Sharma, T., Bashir, F., Abbas, H., … & Hanif, M. (2024). Design implementations of ternary logic systems: A critical review. *Results in engineering*, *23*, 102761. https://doi.org/10.1016/j.rineng.2024.102761

[10] Alekhya, Y., & Nanda, U. (2022). Investigation of CNTFET based energy efficient fast SRAM cells for edge AI devices. *Silicon*, *14*(14), 8815–8830. https://doi.org/10.1007/s12633-021-01589-0

[11] Tabrizchi, S., Angizi, S., & Roohi, A. (2022). Design and evaluation of a robust power-efficient ternary sram cell. *2022 IEEE 65th international midwest symposium on circuits and systems (MWSCAS)* (pp. 1–4). IEEE. https://doi.org/10.1109/MWSCAS54063.2022.9859306

[12] Zhang, A. Q., Tosson, A. M. S., Ma, D., Fang, R., & Wei, L. (2023). Stuck-at faults tolerance and recovery in MLP neural networks using imperfect emerging CNFET technology. *IEEE journal on exploratory solid-state computational devices and circuits*, *9*(2), 168–175. https://doi.org/10.1109/JXCDC.2023.3313127

[13] Kim, Y. B. (2023). *A Novel CNFET SRAM-based computing-in-memory design and low power techniques for AI accelerator*. https://www.proquest.com/openview/ccaa11e388b9d59e7e674e1a7848cd88/1?pq-origsite=gscholar&cbl=18750&diss=y

[14] Birla, S., Dargar, S. K., Singh, N., & Sivakumar, P. (2023). *Low power designs in nanodevices and circuits for emerging applications*. CRC Press. https://doi.org/10.1201/9781003459231

[15] Zhang, A. Q. (2021). *Struck-at fault tolerance with emerging technology RAM in the neurosim MLP neural network system* [Thesis]. http://hdl.handle.net/10012/17391

[16] Islam, A., & Hasan, M. (2011). Power optimized variation aware dual-threshold SRAM cell design technique. *Nanotechnology, science and applications*, *4*, 25–33. https://doi.org/10.2147/NSA.S15719

[17] Kenarangi, F., & Partin-Vaisband, I. (2021). A single-MOSFET analog high resolution-targeted (SMART) multiplier for machine learning classification. *IEEE journal on emerging and selected topics in circuits and systems*, *11*(4), 816–828. https://doi.org/10.1109/JETCAS.2021.3124940

[18] Yin, S., Kim, Y., Han, X., Barnaby, H., Yu, S., Luo, Y., … & Seo, J. (2019). Monolithically integrated RRAM- and CMOS-based in-memory computing optimizations for efficient deep learning. *IEEE micro*, *39*(6), 54–63. https://doi.org/10.1109/MM.2019.2943047

[19] Kenarangi, F., Hu, X., Liu, Y., Incorvia, J. A. C., Friedman, J. S., & Partin-Vaisband, I. (2020). Exploiting dual-gate ambipolar CNFETs for scalable machine learning classification. *Scientific reports*, *10*(1), 5735. https://doi.org/10.1038/s41598-020-62718-0

[20] Pal, S., Bose, S., Ki, W. H., & Islam, A. (2020). A highly stable reliable SRAM cell design for low power applications. *Microelectronics reliability*, *105*, 113503. https://doi.org/10.1016/j.microrel.2019.113503

[21] Prakash, P., Sundaram, K. M., & Bennet, M. A. (2018). A review on carbon nanotube field effect transistors (CNTFETs) for ultra-low power applications. *Renewable and sustainable energy reviews*, *89*, 194–203. https://doi.org/10.1016/j.rser.2018.03.021

[22] Cho, G., Kim, Y. B., Lombardi, F., & Choi, M. (2009). Performance evaluation of CNFET-based logic gates. *2009 IEEE instrumentation and measurement technology conference* (pp. 909–912). IEEE. https://doi.org/10.1109/IMTC.2009.5168580

[23] Hailiang, Z., Yue, H., & Minxuan, Z. (2010). Numerical study of the sub-threshold slope in T-CNFETs. *Journal of semiconductors*, *31*(9), 94005. https://doi.org/10.1088/1674-4926/31/9/094005

[24] Sankar, P. A. G., & Udhayakumar, K. (2014). MOSFET-like CNFET based logic gate library for low-power application: A comparative study. *Journal of semiconductors*, *35*(7), 75001. https://doi.org/10.1088/1674-4926/35/7/075001

[25] Su, S. K., Sanchez-Soares, A., Chen, E., Kelly, T., Fagas, G., Greer, J. C., … & Radu, I. P. (2022). Impact of metal hybridization on contact resistance and leakage current of carbon nanotube transistors. *IEEE electron device letters*, *43*(8), 1367–1370. https://doi.org/10.1109/LED.2022.3185991

[26] Dixit, A., & Gupta, N. (2018). Analysis of different gate dielectric materials in carbon nanotube field effect transistor (CNFET) using optimization technique. *2018 IEEE electron devices kolkata conference (EDKCON)* (pp. 354–357). IEEE. https://doi.org/10.1109/EDKCON.2018.8770503

[27] Arif, S., & Pal, S. (2015). CNFET based 8-t sram cell to improve stability with reduced power consumption. *2015 international conference on advanced computing and communication systems* (pp. 1–5). IEEE. https://doi.org/10.1109/ICACCS.2015.7324112

[28] Madhavi, B. K., & Kishore, K. L. (2012). Low leakage-power sram cell design using CNTFETs at 32nm technology. *International conference on advances in communication, network, and computing* (pp. 165–171). Springer. https://doi.org/10.1007/978-3-642-35615-5_24

[29] Zanjani, S. M., & Toghian, P. (2024). Design and simulation of a low-power, universal & multi-mode filter for the commercial FM band in 20-nm CNFETs. *Analog integrated circuits and signal processing*, *120*(1), 9–20. https://doi.org/10.1007/s10470-024-02287-8

[30] Srinivasan, V., Varshad Venkatraman, R., & Senthil kumar, K. K. (2013). Schmitt trigger based SRAM cell for ultralow power operation- A CNFET based approach. *Procedia engineering*, *64*, 115–124. https://doi.org/10.1016/j.proeng.2013.09.082

[31] Kureshi, A. K., & Hasan, M. (2009). Performance comparison of CNFET- and CMOS-based 6T SRAM cell in deep submicron. *Microelectronics journal*, *40*(6), 979–982. https://doi.org/10.1016/j.mejo.2008.11.062

[32] Arif, S., & Pal, S. (2015). Variation-resilient cnfet-based 8T sram cell for ultra-low-power application. *2015 international conference on signal processing and communication engineering systems* (pp. 147–151). IEEE. https://doi.org/10.1109/SPACES.2015.7058235

[33] Kanhaiya, P. S., Lau, C., Hills, G., Bishop, M. D., & Shulaker, M. M. (2019). Carbon nanotube-based CMOS SRAM: 1 kbit 6T SRAM arrays and 10T SRAM cells. *IEEE transactions on electron devices*, *66*(12), 5375–5380. https://doi.org/10.1109/TED.2019.2945533

[34] Soni, L., & Pandey, N. (2023). A novel CNTFET based schmitt-trigger read decoupled 12T SRAM cell with high speed, low power, and high Ion/Ioff ratio. *AEU-international journal of electronics and communications*, *167*, 154669. https://doi.org/10.1016/j.aeue.2023.154669

[35] Kim, Y., Patel, S., Kim, H., Yadav, N., & Choi, K. K. (2021). Ultra-low power and high-throughput SRAM design to enhance AI computing ability in autonomous vehicles. *Electronics*, *10*(3), 1–22. https://doi.org/10.3390/electronics10030256

[36] Aslam, M. F. (2024). *An inverter based OTA with novel tunable pseudoresistor and its application in arrhythmia detection using machine learning algorithms*. https://doi.org/10.21203/rs.3.rs-4957694/v1

[37] Gupta, S. K. (2024). Special topic on physics-based modeling and simulation of materials, devices, and circuits of beyond-CMOS logic and memory technologies for energy-efficient computing. *IEEE journal on exploratory solid-state computational devices and circuits*, *9*(2),1-3. https://doi.org/10.1109/JXCDC.2023.3340557

[38] Singh, J., & Singh, D. (2024). Applications of AI/ML algorithms in VLSI design and technology. In *Integrated devices for artificial intelligence and VLSI* (pp. 157–191). Wiley Online Library. https://doi.org/10.1002/9781394205158.ch7

[39] Maqbool, M., Sharma, V. K., & Kaushik, N. (2024). CNTFET-based SRAM cell design using INDEP technique. *E-prime-advances in electrical engineering, electronics and energy*, *7*, 100477. https://doi.org/10.1016/j.prime.2024.100477

[40] Elangovan, M., Sharma, K., Sachdeva, A., & Gupta, L. (2024). Read improved and low leakage power CNTFET based hybrid 10t SRAM cell for low power applications. *Circuits, systems, and signal processing*, *43*(3), 1627–1660. https://doi.org/10.1007/s00034-023-02529-6

[41] Marani, R., & Perri, A. G. (2024). A procedure to compare CNTFET and CMOS technologies through the design of a SRAM cell: A review. *International journal of nanoscience and nanotechnology*, *20*(2), 113–127. https://doi.org/10.22034/ijnn.2024.2021428.2472

[42] Chu, C., Xu, D., Wang, Y., & Chen, F. (2022). Canopy: A cnfet-based process variation aware systolic DNN accelerator. *Proceedings of the acm/IEEE international symposium on low power electronics and design* (pp. 1–6). IEEE. https://doi.org/10.1145/3531437.3539703

[43] Sachdeva, A., Sharma, K., Bhargava, A., & Abbasian, E. (2024). A CNTFET based stable, single-ended 7T SRAM cell with improved write operation. *Physica scripta*, *99*(3), 35011. https://doi.org/10.1088/1402-4896/ad24a8

[44] Darabi, A., Salehi, M. R., & Abiri, E. (2023). Single-sided gate-wrap-around CNTFET SRAM cell for utilization in reliable IoT-based platforms. *AEU-international journal of electronics and communications*, *163*, 154605. https://doi.org/10.1016/j.aeue.2023.154605

[45] Sadeghi, A., Ghasemi, R., Ghasemian, H., & Shiri, N. (2023). Efficient and optimized approximate GDI full adders based on dynamic threshold CNTFETs for specific least significant bits. *Frontiers of information technology & electronic engineering*, *24*(4), 599–616. https://doi.org/10.1631/FITEE.2200077

[46] Darabi, A., Salehi, M. R., & Abiri, E. (2021). Newly energy-efficient SRAM bit-cell using GAA CNT-GDI method with asymmetrical write and built-in read-assist schemes for QR code-based multimedia applications. *Microelectronics journal*, *114*, 105117. https://doi.org/10.1016/j.mejo.2021.105117

[47] Sajjadi, S. A., Sadrossadat, S. A., Moftakharzadeh, A., Nabavi, M., & Sawan, M. (2024). DNN-Based optimization to significantly speed up and increase the accuracy of electronic circuit design. *IEEE transactions on circuits and systems I: Regular papers*, *71*(3), 1273–1284. https://doi.org/10.1109/TCSI.2023.3347688

[48] Raja, G. B., & Madheswaran, M. (2013). Design and analysis of 5-T SRAM cell in 32nm CMOS and CNTFET technologies. *International journal of electronics and electrical engineering*, *1*(4), 256–261. https://doi.org/10.12720/ijeee.1.4.256-261

[49] Yadav, A., & Wairya, S. (2021). Performance and area optimization of SRAM Cell in nanocomputing application. *International journal of computing and digital system*, *11*(1), 1009–1026. http://dx.doi.org/10.12785/ijcds/110182

[50] Gopinath, A., Cochran, Z., Ytterdal, T., & Rizkalla, M. (2022). SRAM design leveraging material properties of exploratory transistors. *Materials today: Proceedings*, *57*, 368–374. https://doi.org/10.1016/j.matpr.2021.10.327

[51] Arora, S., & Tripathi, S. L. (2024). A comprehensive analysis of ultra low power GNRFET based 20T hybrid full adder for computing applications. *Physica scripta*, *99*(8), 85008. https://doi.org/10.1088/1402-4896/ad5c0b

[52] Ghorbani, A., Dolatshahi, M., Zanjani, S. M., & Barekatain, B. (2022). A new low-power dynamic-GDI full adder in CNFET technology. *Integration*, *83*, 46–59. https://doi.org/10.1016/j.vlsi.2021.12.001

[53] Valdés, R. C., García, F., García, R. Z., López, A., & Hernández, N. (2023). Parameter extraction in thin film transistors using artificial neural networks. *Journal of materials science: Materials in electronics*, *34*(6), 555. https://doi.org/10.1007/s10854-023-09953-z

[54] Bhuvaneswari, T., & Kathirvelu, M. (2024). A comprehensive analysis on the intelligent VLSI systems design and applications. *Indian journal of natural sciences*, *15*(83). 71174-71180. https://www.researchgate.net/publication/382335425

[55] Ghorbani, A., Dolatshahi, M., Zanjani, S. M., & Barekatain, B. (2022). A new low power, area efficient 4-bit carry look ahead adder in CNFET technology. *Majlesi journal of electrical engineering*, *16*(1), 1-9. https://oiccpress.com/mjee/article/view/4943

[56] Chen, F., Song, L., Li, H., & Chen, Y. (2021). Marvel: A vertical resistive accelerator for low-power deep learning inference in monolithic 3D. *2021 design, automation & test in europe conference & exhibition (DATE)* (pp. 1240–1245). IEEE. https://doi.org/10.23919/DATE51398.2021.9474208

[57] Mahida, N., Parikh, K. A., Darji, K., Gupta, R., Tanwar, S., Shahinzadeh, H., & Mohammad-Hosseini, M. (2024). PneuTeleCNN: Deep learning-assisted framework for pneumonia identification in web-enabled telemedicine systems. *2024 10th international conference on web research (ICWR).* IEEE. http://dx.doi.org/10.1109/ICWR61162.2024.10533369

[58] Shahinzadeh, H., Mahmoudi, A., Asilian, A., Sadrarhami, H., Hemmati, M., & Saberi, Y. (2024). Deep learning: A overview of theory and architectures. *2024 20th csi international symposium on artificial intelligence and signal processing (AISP)* (pp. 1–11). IEEE. https://doi.org/10.1109/AISP61396.2024.10475265

[59] Bhavsar, A., Patel, V., Patel, Y., Geddam, R., Gupta, R., Tanwar, S., … & Shahinzadeh, H. (2024). Transfer learning based breast cancer detection for telemedicine systems in healthare environment. *2024 8th*

international conference on smart cities, internet of things and applications (SCIOT) (pp. 156–161). IEEE. https://doi.org/10.1109/SCIoT62588.2024.10570101

[60] Vijay, M. M., Kumar, O. P., Francis, S. A. J., Stalin, A. D., & Vincent, S. (2024). Enhancing high-speed digital systems: MVL circuit design with CNTFET and RRAM. *Journal of king saud university - computer and information sciences*, *36*(4), 102033. https://doi.org/10.1016/j.jksuci.2024.102033

[61] Dutt, A. (2023). *Modelling, exploration and optimization of hardware accelerators for deep learning applications*. https://dr.ntu.edu.sg/entities/publication/ddc4a592-3921-4c45-8a40-e44232913d94

[62] Das, R. R., Rajalekshmi, T. R., & James, A. (2024). FinFET to GAA MBCFET: A review and insights. *IEEE access*, *12*, 50556–50577. https://doi.org/10.1109/ACCESS.2024.3384428

[63] Ghasemi Nejad Raeini, A., Kordrostami, Z., & Hamedi, S. (2021). Coupled NEGF-PSO method for maximizing the current ratio of CNTFETs based on oxide thickness optimization. *Journal of computational electronics*, *20*(5), 1657–1665. https://doi.org/10.1007/s10825-021-01743-2

[64] Sandhya Rani, S., Pasha Mohammad, H., & Masthan, S. (2024). Remora optimized dynamic neural network to design low-power CNTFET logic gate realization with multiple-valued circuits. *Electric power components and systems*, 1–14. https://doi.org/10.1080/15325008.2024.2326854

[65] Mohammaden, A., Fouda, M. E., Alouani, I., Said, L. A., & Radwan, A. G. (2022). CNTFET-based ternary multiply-and-accumulate unit. *Electronics*, *11*(9), 1–10. https://doi.org/10.3390/electronics11091455