Integrating Traditional Low-Power Techniques with AI/ML for Low-Power and High-Efficiency VLSI Chips

[1]M L Sharma, [2]Neelam Sharma, [3]Sunil Kumar, [4]Chittral Rastogi, [5]Garv Gupta

[1,2,3]Professor, Maharaja Agrasen Institute of Technology, Delhi

[4,5]Research Scholar, Electronics and Communication Department, Maharaja Agrasen Institute of Technology, Delhi
[1]madansharma.20@gmail.com, [2]neelam_sr@yahoo.com, [3]sunilkumar@mait.ac.in, [4]chittralrw@gmail.com, [5]garv9325@gmail.com

ABSTRACT

Continuous miniaturization in VLSI fabrication has enhanced integration density and computational capability but has also intensified challenges related to energy efficiency and thermal reliability. Conventional design strategies such as voltage scaling, clock gating, power gating, and multi-threshold CMOS remain vital for managing dynamic and leakage power. However, growing circuit complexity and heterogeneous workloads limit the effectiveness of traditional methods alone. Consequently, leading semiconductor industries are investing extensively in low-power design research, highlighting its industrial importance.

This paper presents an integrated approach that combines traditional low-power strategies with Artificial Intelligence (AI) and Machine Learning (ML) to achieve higher optimization efficiency in VLSI chips. ML-based predictive models estimate power-performance trade-offs and optimize resource allocation, while reinforcement learning dynamically controls parameters such as voltage and frequency. The comparative evaluation demonstrates that this hybrid integration improves scalability, flexibility, and efficiency—signaling a promising direction for next-generation VLSI design.

KEYWORDS

VLSI, Machine Learning, Artificial Intelligence, Low-Power Design, CMOS, Neural Networks, Power Estimation, High-Level Synthesis (HLS), Reinforcement Learning Power, Performance, and Area (PPA), Optimization

1. INTRODUCTION

VLSI technology enables the integration of millions of transistors on a single chip, driving continuous improvements in performance and functionality. Yet, as devices scale, power consumption becomes a major concern affecting thermal stability and overall reliability. Power dissipation in VLSI circuits generally consists of dynamic power—caused by signal transitions and capacitive charging—and static or leakage power, which stems from subthreshold and gate leakage currents.

Accurate power estimation remains crucial. Circuit-level tools like SPICE deliver high accuracy but are computationally demanding for large designs, while RTL-level methods are faster but less precise. Classic techniques such as voltage scaling, clock gating, power gating, and multi-threshold CMOS continue to play key roles, though their impact diminishes as circuits grow in complexity.

To address these limitations, machine learning has emerged as a data-driven solution for optimizing power, performance, and area (PPA). Predictive models such as random forests, gradient boosting, and neural networks have shown strong capability in estimating power directly from design attributes. Reinforcement learning further allows real-time

DOI: 10.55041/IJSREM53987

optimization of design parameters like placement and gate sizing, supporting adaptive trade-off management across large design spaces. Integrating these ML methods with established circuit-level approaches offers a scalable path toward low-power, high-efficiency VLSI systems.

.

$$P_{\text{dynamic}} = \alpha \cdot C_{\text{eff}} \cdot V_{\text{dd}}^2 \cdot f$$

$$P_{\text{leakage}} = I_{\text{leak}} \cdot V_{\text{dd}} \approx (I_0 \cdot e^{\frac{-V_{\text{th}}}{nV_T}}) \cdot V_{\text{dd}}$$

2. Machine Learning and its implication :

Machine Learning (ML) refers to computational approaches that identify relationships within data and enhance performance through experience rather than explicit programming. ML techniques include supervised, unsupervised, and reinforcement learning.

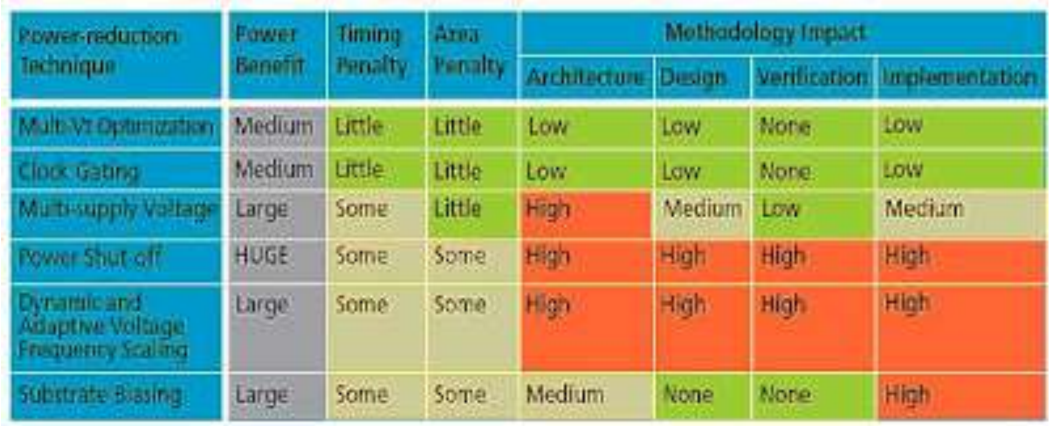
In **supervised learning**, algorithms such as regression models, decision trees, SVMs, and KNN learn mappings from labeled data to predict unknown outcomes. **Unsupervised learning** methods like clustering (K-Means) and dimensionality reduction (PCA) reveal hidden structures in unlabeled datasets. **Reinforcement learning (RL)** involves an agent improving its strategy through interaction and reward feedback, proving effective in dynamic optimization tasks such as energy control in VLSI systems.

Advances in GPUs and high-performance computing have expanded ML applications across electronic design automation (EDA), enabling its integration with conventional low-power methodologies to enhance predictive accuracy and automation.

3. Previously and Currently used Low-Power Techniques used

Early VLSI circuits employed strategies such as clock gating, logic optimization, and transistor sizing to limit dynamic power consumption. As technology entered deep-submicron regimes, leakage power became significant, leading to the adoption of **power gating** and **multi-threshold CMOS (MTCMOS)**. Power gating disconnects inactive modules from the supply rail to reduce leakage, while MTCMOS uses high-threshold transistors in idle states and low-threshold ones in active operation to balance performance and power.

Dynamic Voltage and Frequency Scaling (DVFS) leverages the quadratic dependence of dynamic power on supply voltage, lowering voltage and frequency under light workloads to conserve energy. Modern systems often combine several approaches—voltage islands, substrate biasing, operand isolation, and DVFS—to achieve optimal performance–efficiency trade-offs across diverse SoC applications.



Power-reduction Technique	Power Benefit	Timing Penalty	Area Penalty	Methodology Impact			
				Architecture	Design	Verification	Implementation
Multi-Vt Optimization	Medium	Little	Little	Low	Low	None	Low
Clock Gating	Medium	Little	Little	Low	Low	None	Low
Multi-supply Voltage	Large	Some	Little	High	Medium	Low	Medium
Power Shut off	HUGE	Some	Some	High	High	High	High
Dynamic and Adaptive Voltage Frequency Scaling	Large	Some	Some	High	High	High	High
Substrate Biasing	Large	Some	Some	Medium	None	None	High

4. Methodologies in ML-Based Power Analysis

Integrating machine learning into the VLSI design process seeks to enhance traditional optimization workflows rather than replace them. The growing complexity of design spaces and the limitations of purely analytical models have encouraged researchers to embed ML models at multiple abstraction levels—ranging from pre-synthesis estimation to post-layout optimization. Each level offers unique trade-offs in data granularity, accuracy, and computational cost. The following subsections present three prominent approaches—graph-based neural networks, attribute-driven ensemble learning, and deep reinforcement learning—that together form a comprehensive framework for intelligent power analysis and optimization.

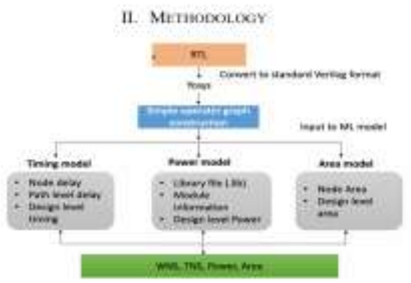
4.1 Pre-Synthesis Estimation via Graph-Based Neural Networks

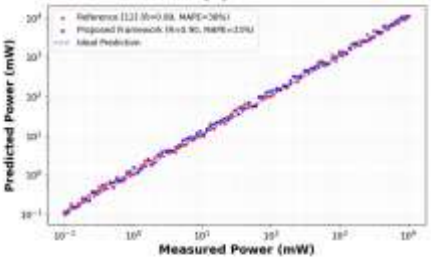
Power estimation in the earliest design stages remains a persistent challenge because register-transfer-level (RTL) or behavioral descriptions provide little structural correlation with final physical layouts. Traditional high-level estimators either oversimplify switching activity or rely on exhaustive simulation, both of which compromise accuracy or efficiency.

To bridge this gap, researchers have introduced **graph-centric representations**, most notably the *Simple Operator Graph (SOG)*. The SOG translates the HDL into a fine-grained, bit-level graph whose nodes represent fundamental logic operations and storage elements. This structure captures the eventual synthesis topology while remaining abstract enough for early-stage analysis. Each node can be annotated with data such as toggle rates, capacitances, and load information extracted from technology libraries.

These annotated graphs serve as direct input to **Graph Convolutional Networks (GCNs)**, which learn complex spatial dependencies between circuit components. By propagating information through graph convolutions, a GCN can infer how local gate interactions contribute to global power consumption. When trained on diverse circuit datasets, such models have achieved prediction accuracies exceeding 90 % relative to detailed simulation benchmarks. The result is a fast, learning-based estimator that eliminates the need for time-consuming SPICE or RTL power simulations while retaining strong correlation with post-synthesis measurements.

Beyond accuracy, the graph-based approach provides interpretability: saliency analysis on trained GCNs highlights which nodes or subcircuits dominate energy usage, enabling designers to target critical regions early in the design flow. This feature makes graph-based neural estimation a valuable complement to conventional EDA power tools.





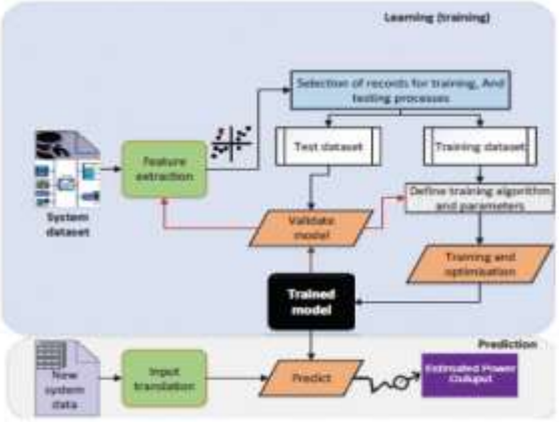
4.2 High-Level Estimation via Attribute-Based Random Forest

While graph-based techniques rely on structural information, an alternative strategy abstracts the circuit into a feature vector of descriptive parameters. The **Random Forest (RF)** algorithm, an ensemble of decision trees, is particularly effective for this level of modeling because it captures non-linear relationships between design attributes and measured power values.

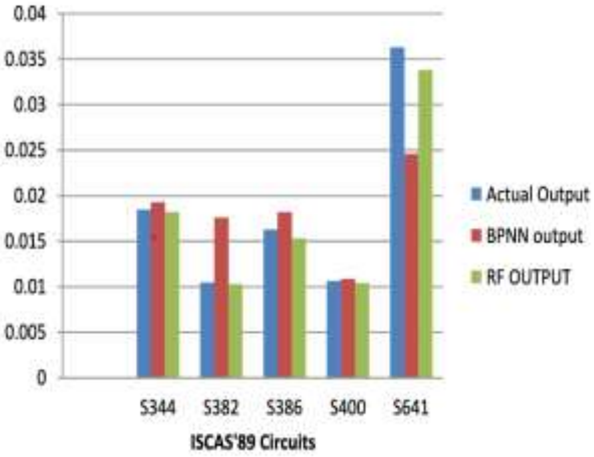
The attribute set typically includes quantities such as total gate count, number of inputs and outputs, counts of flip-flops, and distributions of logic gate types (AND, OR, NAND, NOR, etc.). These metrics are extracted automatically from RTL or gate-level representations and normalized across multiple benchmarks to form a consistent dataset. The RF model learns how combinations of these features influence total power and can generalize to unseen circuits with high fidelity.

To achieve optimal performance, hyperparameters—number of trees, maximum depth, and feature subset size—are tuned using **multi-objective evolutionary algorithms** such as NSGA-II. This tuning balances prediction accuracy against model complexity and training time. When properly optimized, the RF estimator provides near-perfect correlation coefficients (R ≈ 0.999) compared to simulation ground truth while requiring only milliseconds for inference.

Because it does not depend on structural connectivity data, the RF approach scales well for large SoCs and allows designers to perform *what-if* analyses rapidly. For instance, changes in module count or gate distribution can be evaluated almost instantly to predict their effect on total power consumption. Thus, attribute-based ensemble learning provides a practical, high-speed solution for architectural exploration and early design-space screening.



$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(Y_i^O - Y_i^C)^2}{N}}$$



4.3 Active Optimization via Deep Reinforcement Learning

Unlike estimation models, which passively predict power, Deep Reinforcement Learning (DRL) actively drives optimization by learning how to modify design parameters to achieve minimal power consumption. The central idea is to treat power optimization as a Markov Decision Process (MDP), where an intelligent agent interacts with a virtual design environment.

State (S): Represents the current configuration of the design, defined by metrics such as total power, timing slack, and chip area.

Actions (A): Correspond to available low-power techniques—gate sizing, threshold voltage swapping, placement refinement, and cell clustering.

Reward (R): A numerical signal evaluating how beneficial an action is, expressed as:

$R(s,a)=-(\alpha T+\beta P+\gamma A)$

The Deep Q-Network (DQN) serves as the learning agent, iteratively exploring the design space and receiving feedback from the environment after each action. Over millions of iterations, it constructs an optimal policy that maximizes cumulative rewards, effectively learning how and when to apply traditional low-power techniques.

Experimental studies have demonstrated that such DRL-based frameworks can reduce overall power by up to 22%, improve timing by around 18%, and cut area by nearly 12%, depending on the dataset and process node. The approach's major advantage lies in its ability to perform non-greedy, global optimization—the agent can sometimes increase power locally in one block to enable larger system-level savings later.

Moreover, DRL drastically improves computational tractability. Traditional "what-if" optimization scripts might take days due to repeated synthesis iterations, while a trained RL agent can perform equivalent evaluations in seconds through inference. The framework's scalability also enables it to adapt to new technologies or constraints through fine-tuning rather than retraining from scratch.

By integrating the reinforcement learning paradigm with classical circuit techniques, this methodology converts manual design tuning into an autonomous, learning-driven optimization loop, paving the way for intelligent, energy-aware VLSI design automation.

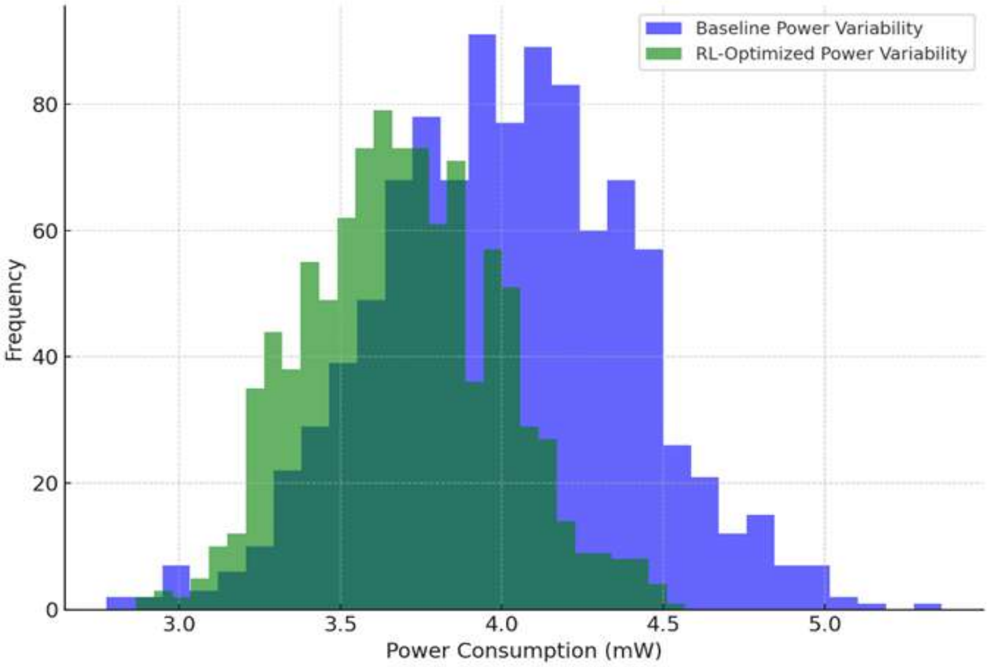


Figure 4: Power Variability Analysis Across PVT Conditions

5. Proposed Framework: Deep Reinforcement Learning for Power Optimization

Building upon the methodologies extant in the literature, we propose a framework that centers on the Deep Reinforcement Learning (RL) approach for active power optimization. This method is singled out because it directly integrates and automates the application of traditional, proven low-power design techniques, using ML as an intelligent control policy to navigate the vast, multi-dimensional design space.

5.1 The MDP Formulation for Power Optimization

Building upon the methodologies discussed above, this work proposes an integrated **Deep Reinforcement Learning (DRL)** framework that automates the application of traditional low-power design techniques through adaptive policy learning. Rather than relying on static heuristics or hand-crafted scripts, the DRL agent dynamically explores the design space to identify the sequence of actions that minimizes power while meeting performance and area constraints.

5.1 MDP Formulation for VLSI Power Optimization

At the heart of the framework lies the **Markov Decision Process (MDP)** representation of the physical design environment. The design is modeled as a set of measurable states, allowable actions, and scalar rewards that guide the learning process.

- **State (S):** Describes the current status of the chip, represented by metrics such as total power, timing violations (worst negative slack or total negative slack), and occupied silicon area. These attributes collectively define the environment's current condition.

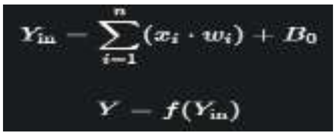
- **Actions (A):** Denote discrete modifications that the agent can apply—such as resizing a gate, changing threshold voltage levels, adjusting cell placement, or inserting clock-gating elements.

- **Transition (T):** Defines how an action changes the state of the design; for example, downsizing a non-critical gate might reduce power but slightly degrade timing.

- **Reward (R):** Provides quantitative feedback for each action. A multi-objective function balances power reduction, timing preservation, and area efficiency:

R(s,a)=−(αT+βP+γA)R(s,a)=-(\alpha T+\beta P+\gamma A)R(s,a)=−(αT+βP+γA)

During training, the agent interacts with a simulation-based design environment, repeatedly applying actions and observing their effects. Over time, the DRL model learns the mapping between design states and optimal optimization actions that maximize long-term cumulative reward.



$$Y_{in} = \sum_{i=1}^{n} (x_i \cdot w_i) + B_0$$

$$Y = f(Y_{in})$$

5.2 Integrating Traditional Techniques via the Reward Function

The DRL framework does not discard conventional approaches; rather, it **encodes them as learnable actions** within its policy space. Techniques such as gate sizing, power gating, and threshold-voltage adjustment become atomic moves that the agent learns to combine adaptively.

The reward-driven formulation ensures that the RL agent automatically identifies where each technique is most beneficial. For example, the model learns that applying high-Vt cells on non-critical paths reduces leakage without violating timing, or that clock gating is most effective in modules exhibiting high idle ratios. This adaptive capability allows the system to **replicate expert-level design intuition** without explicit rule encoding.

Simulation results from similar research frameworks have demonstrated that DRL agents can achieve simultaneous improvements across multiple metrics—reducing total power by more than 20 %, improving timing margins by around 15–18 %, and lowering silicon area by roughly 10–12 %. Such outcomes confirm that reinforcement learning can internalize the intricate dependencies among PPA parameters that traditional static optimization overlooks.

5.3 Advantages over Conventional Optimization

This RL-based framework provides a clear path to overcoming the limitations of both manual design and computationally expensive heuristics.

- **Holistic, Non-Greedy Optimization:** Conventional optimization scripts often follow greedy heuristics, focusing on short-term gains and missing global trade-offs. The DRL agent, trained through exploration, develops a long-term strategy that balances multiple objectives. It can, for example, temporarily increase area in one region to relieve timing bottlenecks that later yield larger power reductions elsewhere.
- **Computational Tractability:** Traditional "what-if" analyses require multiple synthesis and place-and-route iterations, consuming days of computation. Once trained, a DRL agent can evaluate thousands of virtual design scenarios in seconds, delivering near-real-time decision support for designers..
- **Scalability and Resilience:** The learned policy is generalizable—it can be fine-tuned for different designs or process nodes with minimal retraining. This makes the framework resilient to process, voltage, and temperature (PVT) variations and reusable across product generations.

6. CONCLUSION

With technology nodes shrinking and transistor counts soaring, controlling power consumption has become one of the most critical challenges in modern VLSI design. While traditional low-power design strategies—such as voltage scaling, clock gating, power gating, and multi-threshold CMOS—remain fundamental, they increasingly struggle to deliver optimal results under the complexity of heterogeneous and high-performance workloads.

This paper presented a comprehensive framework that merges **conventional low-power techniques with Machine Learning (ML) and Artificial Intelligence (AI)** to achieve more effective power optimization. Through the discussion of methodologies and the proposed DRL-based approach, several important conclusions emerge:

- **Early-Stage Power Estimation:** ML models such as Graph Convolutional Networks (GCNs) and Random Forest (RF) estimators enable fast and accurate power prediction without time-consuming simulations, maintaining high correlation with ground-truth results.

- **Adaptive Optimization:** Deep Reinforcement Learning (DRL) provides an automated mechanism to apply traditional power-saving actions intelligently, learning how to trade off between power, performance, and area.

- **Synergistic Integration:** Rather than competing with conventional techniques, ML complements them—guiding their application more efficiently and allowing broader exploration of design possibilities.

The study underscores that AI/ML integration represents not just a technological enhancement but a paradigm shift in how low-power VLSI systems are conceived and optimized. As design sizes continue to grow, these intelligent frameworks will be indispensable in achieving sustainable energy efficiency without compromising performance

7. FUTURE WORK

Future research will aim to advance and refine the proposed Deep Reinforcement Learning (DRL) framework across several critical dimensions:

1. Expanded Action Space:
The current model primarily focuses on physical design optimizations such as gate sizing, cell placement, and threshold voltage adjustment. In upcoming studies, the agent's action space will be broadened to incorporate higher-level techniques, including Dynamic Voltage and Frequency Scaling (DVFS) and adaptive clock gating. Integrating these methods will enable the agent to develop power-efficient strategies that operate cohesively across both design-time and runtime domains.

2. Transfer Learning and Scalability:
Enhancing the scalability of the DRL framework is another key objective. Future efforts will explore the use of Transfer Learning, allowing a policy trained on one circuit to be effectively reused and fine-tuned for different or larger designs with minimal retraining overhead. This direction is crucial for extending the framework's applicability to modern, industrial-scale Systems-on-Chip (SoCs) while maintaining computational efficiency.

7. REFERENCES

1. Friedman, E. G. (2001). Clock distribution networks in synchronous digital integrated circuits. Proceedings of the IEEE, 89(5), 665-692.
2. Chang, K., Pamarti, S., Kaviani, K., Alon, E., Shi, X., Chin, T. J., ... & Yuan, C. (2005, February). Clocking and circuit design for a parallel I/O on a first-generation CELL processor. In Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International (pp. 526-615). IEEE.
3. Najm, F. N. (1994). A survey of power estimation techniques in VLSI circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2(4), 446-455.
4. Benini, L., Bogliolo, A., & De Micheli, G. (2000). A survey of design techniques for system-level dynamic power management. IEEE transactions on very large scale integration (VLSI) systems, 8(3), 299-316.
5. W. Zhao, et al., F-CNN: An FPGA-based framework for training Convolutional Neural Networks, in: 2016 IEEE 27th...
6. R. Zhao, et al., Accelerating Binarized Convolutional Neural Networks with Software-Programmable FPGAs, in: Proceedings...

7. SchaferB.C. et al. High-level synthesis design space exploration: Past, present and futureLe, H.-B. Do, X.-D.

8. Lee, S.-G." Compact low-power high-slew-rate buffer for LCD driver application",IEEE Electronics Letters ,Volume: 46 Issue: 20 ,On page(s): 1370 - 1371 ISSN:0013-5194

9. V. Govindaraj and B. Arunadevi, "Machine Learning Based Power Estimation for CMOS VLSI Circuits,", vol. 35, no. 13, pp. 1043-1055, 2021.

10. A. Chattopadhyay and V. K. Sutrakar, "Machine Learning Framework for Early Power, Performance, and Area Estimation of RTL," 2023.

11. N. G. Rani, D. J, T. A. Fatihi, and J. A. Prajapati, "VLSI Design Automation: Tools and Techniques for Enchancing Circuit Performance and Reliability," *J. Electrical Systems*, vol. 20-11s, pp. 3142-3150, 2024.

12. Y. Attaoui, M. Chentouf, Z. E. A. A. Ismaili, and A. El Mourabit, "Machine Learning in VLSI Design: A Comprehensive Review," *Journal of Integrated Circuits and Systems*, vol. 19, no. 2, 2024.

13. H. Esmaeilzadeh et al., "An Open-Source ML-Based Full-Stack Optimization Framework for Machine Learning Accelerators," *ACM Transactions on Design Automation of Electronic Systems*, vol. 29, no. 4, art. 68, 2024.

14. P. Goswami, B. C. Schaefer, and D. Bhatia, "Machine learning based fast and accurate High Level Synthesis design space exploration: From graph to synthesis," *Integration*, vol. 88, pp. 116-124, Jan. 2023.

15. D. Sarkar, P. Bhattacharjee, and A. Majumder, "Data-Dependent Clock Gating approach for Low Power Sequential System," in *5th International Conference on 'Microelectronics, Circuits and Systems' (Micro2018)*, 2018.

16. T. Kapilachander, I. H. Shanavas, and V. Venkataraman, "Technical Study on Low Power VLSI methods," *I.J. Information Engineering and Electronic Business*, vol. 1, pp. 60-70, 2012.

17. B. Calhoun, J. Kao, and A. Chandrakasan, "Chapter 3: POWER GATING AND DYNAMIC VOLTAGE SCALING," in *Leakage in Nanometer CMOS Technologies*.