



A Comparative Analysis of Machine Learning Techniques for Power, Performance, and Area Optimization in Low-Power VLSI Design

Maneesh Pawar

Ph.D. Scholar – Electronics Engineering

School of Electrical and Electronics Engineering

SAM Global University, Raisen, Madhya Pradesh, India.

Abstract – The continuous scaling of semiconductor technology has significantly increased the complexity of Very Large-Scale Integration (VLSI) design. Modern integrated circuits are required to deliver high computational performance while operating within strict power and area constraints. Traditional optimization approaches based on exhaustive design-space exploration, manually tuned heuristics, and iterative Electronic Design Automation (EDA) tool execution can become computationally expensive as the number of design parameters increases. Machine Learning (ML) provides an alternative data-driven approach for predicting design quality, reducing expensive evaluations, and guiding the search toward promising design configurations. This paper presents a comparative analysis of major machine learning techniques applicable to Power, Performance, and Area (PPA) optimization in low-power VLSI design. Supervised learning methods, ensemble learning, artificial neural networks, deep learning, reinforcement learning, and graph-based learning are analyzed with respect to their suitability for PPA prediction and optimization. The study considers power consumption, timing performance, silicon area, computational cost, interpretability, scalability, training-data requirements, and generalization capability as major evaluation dimensions. A unified ML-assisted VLSI optimization framework is proposed in which circuit and synthesis features are extracted from RTL and EDA-generated reports, machine-learning models estimate PPA metrics, and an optimization engine searches for configurations satisfying multi-objective constraints. The comparative analysis indicates that no single ML technique is universally optimal for all VLSI optimization tasks. Tree-based ensemble models are attractive for structured tabular design data because of their relatively low training cost and interpretability, whereas neural and graph-based models are better suited to highly nonlinear and structural circuit relationships. Reinforcement learning is particularly promising for sequential design-space exploration and physical-design optimization, although its computational cost and dependence on reward formulation remain important limitations. The paper therefore recommends a task-oriented hybrid strategy rather than the indiscriminate use of a single ML algorithm. The proposed framework provides a reproducible foundation for future experimental validation using open VLSI benchmarks and publicly available ML-for-EDA datasets.

Keywords: Machine Learning, VLSI, Low-Power Design, Power-Performance-Area, PPA Optimization, Design Space Exploration, Electronic Design Automation, Reinforcement Learning, Neural Networks, Random Forest, Graph Neural Networks.

I. INTRODUCTION

Very Large-Scale Integration (VLSI) technology has become the foundation of modern computing, communication, artificial intelligence, Internet of Things (IoT), mobile electronics, automotive systems, and edge-computing platforms. The increasing number of transistors integrated into a single chip has enabled substantial improvements in computational capability. However, the continuous growth in circuit complexity has also created significant challenges in power management, timing closure, physical implementation, thermal management, and silicon-area utilization.

Historically, VLSI optimization has focused on three closely coupled objectives: power, performance, and area. These objectives are commonly represented by the acronym PPA. Improving one objective frequently affects the other two. For example, increasing the operating frequency may improve performance but increase dynamic power consumption, while aggressive area reduction can create routing congestion and timing degradation. Consequently, modern VLSI optimization is inherently a multi-objective problem.

Power consumption is particularly important for battery-operated and thermally constrained systems. In CMOS circuits, dynamic power is strongly influenced by switching activity, capacitance, supply voltage, and operating frequency, while leakage power becomes increasingly significant with technology scaling. Therefore, low-power VLSI design requires optimization across several abstraction levels, including architecture, logic, circuit, and physical design.

Conventional VLSI design flows depend heavily on EDA tools. A designer typically modifies design parameters, performs synthesis or physical implementation, evaluates the generated reports, and then selects another configuration. This iterative procedure may require a large number of expensive tool executions. As the design space becomes multidimensional, exhaustive exploration becomes impractical.

Machine Learning has emerged as a promising approach for addressing this challenge. Instead of executing a complete EDA flow for every possible design configuration, an ML model can learn the relationship



between design features and PPA outcomes. The trained model can subsequently estimate the likely quality of unexplored configurations and help the optimizer concentrate on promising regions of the design space.

Recent research demonstrates the growing importance of ML in VLSI CAD. Public datasets such as CircuitNet have been introduced specifically to support reproducible ML research in electronic design automation. CircuitNet contains thousands of samples generated from different design-tool runs and supports tasks such as congestion and other physical-design prediction problems.

Deep reinforcement learning has also demonstrated the ability to address difficult chip-floorplanning problems. Mirhoseini et al. demonstrated a graph-based reinforcement-learning approach capable of automatically generating chip floorplans while considering important physical-design objectives.

Similarly, recent research has investigated ML-based PPA estimation, design-space exploration, and physical-synthesis optimization. An IEEE study reported an ML-based framework for early PPA estimation directly from RTL/HDL information, highlighting the potential of ML for reducing dependence on expensive downstream EDA evaluations.

Despite these developments, different ML algorithms possess different advantages and limitations. A model that performs well for power prediction may not necessarily be suitable for placement optimization. Similarly, an algorithm with excellent predictive accuracy may require substantial training data or computational resources. Therefore, this paper focuses on a comparative analysis of machine-learning techniques for PPA optimization in low-power VLSI design.

The major contributions of this paper are:

- A unified classification of ML techniques for low-power VLSI PPA optimization.
- A comparative analysis of supervised, ensemble, deep-learning, graph-based, and reinforcement-learning approaches.
- Identification of the relationship between ML algorithm characteristics and VLSI optimization requirements.
- Development of a unified ML-assisted PPA optimization framework.
- Identification of practical challenges involving data availability, model generalization, interpretability, computational cost, and EDA integration.
- Development of a reproducible experimental methodology suitable for future implementation on open VLSI benchmarks.

II. BACKGROUND OF LOW-POWER VLSI DESIGN

1. Power Consumption in CMOS Circuits

Total CMOS power can generally be represented as:

$$P_{total} = P_{dynamic} + P_{short-circuit} + P_{leakage}$$

Dynamic power is commonly approximated by:

$$P_{dynamic} = \alpha C_L V_{DD}^2 f \quad \text{where:}$$

α = switching activity factor,

C_L = effective load capacitance,

V_{DD} = supply voltage,

f = operating frequency

The equation illustrates why voltage scaling is particularly effective for reducing dynamic power. However, voltage reduction can also affect circuit delay and noise margins. Therefore, power optimization cannot be performed independently of performance. Leakage power becomes increasingly important as transistor dimensions decrease. Threshold-voltage selection, transistor sizing, power gating, multi-voltage design, clock gating, and architectural optimization are therefore commonly considered in low-power design.

Recent reviews emphasize that low-power VLSI optimization increasingly combines conventional techniques with ML-driven optimization and prediction approaches.

2. Performance

Performance in digital circuits is generally associated with propagation delay, maximum operating frequency, critical-path delay, setup/hold timing, and related timing metrics. A simplified relationship between frequency and clock period is:

$$T_{clock} \geq T_{critical-path} + T_{setup} + T_{skew} \quad \text{and:}$$

$$f_{max} \approx \frac{1}{T_{clock}}$$

Improving timing performance may require larger cells, buffering, additional routing resources, or higher operating voltage, which can negatively affect power or area. Consequently, PPA optimization is a multi-objective problem rather than a single-variable optimization problem.

3. Area

The silicon area occupied by a design depends on the number and size of standard cells, memories, macros,



routing resources, and physical implementation constraints.

At the synthesis stage, area can be related to the number and size of logic cells. During physical implementation, however, congestion, routing resources, macro placement, utilization, and design-rule constraints also influence the final area. Therefore, an ML model intended for physical-design optimization must consider structural information rather than relying only on high-level scalar parameters.

III. MACHINE LEARNING IN VLSI DESIGN

ML can be incorporated into VLSI design in two principal ways:

1. Predictive Modeling

In predictive modeling, the ML model receives circuit or design features and predicts one or more target metrics:

$X \rightarrow \{P, A, T\}$

where:

- P= power,
- A= area,
- T= timing/performance metric.

The predicted metrics can be used to reject poor configurations before expensive EDA evaluation.

2. Optimization

In optimization-oriented ML, the model is used to identify promising design configurations:

$X^* = \arg \min_X F(P, A, T)$

subject to design constraints.

A normalized multi-objective cost function may be defined as:

$$F(X) = w_p P_n + w_a A_n + w_t T_n$$

where w_p , w_a , and w_t represent the relative importance of power, area, and timing.

The normalized metrics can be calculated as:

$$P_n = \frac{P - P_{min}}{P_{max} - P_{min}}$$

with equivalent expressions for area and timing.

IV. LITERATURE REVIEW

1. ML-Based PPA Estimation

One of the most important applications of ML in VLSI is early prediction of PPA metrics. Traditional PPA estimation requires synthesis, place-and-route, simulation, or other EDA operations. These operations can consume significant computational resources.

ML-based prediction attempts to estimate PPA before complete implementation.

Recent IEEE research has proposed early PPA estimation directly from RTL/HDL information and library-related features. Such approaches seek to reduce the time required for design iteration.

This application is particularly suitable for regression algorithms such as:

- Linear Regression,
- Support Vector Regression,
- Decision Trees,
- Random Forest,
- Gradient Boosting,
- Artificial Neural Networks,
- Deep Neural Networks.

2. Tree-Based Models

Decision Trees and Random Forest models are attractive for structured VLSI datasets.

Random Forest constructs multiple decision trees and combines their predictions. Its major advantages include:

- relatively low preprocessing requirements,
- nonlinear modeling capability,
- resistance to overfitting compared with individual decision trees,
- feature-importance analysis,
- suitability for tabular datasets.

Recent research has also investigated Random Forest-based power estimation in CMOS circuits. However, tree-based models may be less effective when the fundamental relationship between circuit topology and PPA is highly complex and structural.

3. Artificial Neural Networks

Artificial Neural Networks (ANNs) can model nonlinear relationships between circuit characteristics and PPA metrics. A basic feed-forward neural network can be represented as:

$$h = \sigma(WX + b)$$

where:

- X is the input feature vector,
- W represents learned weights,
- b represents bias,
- σ is the activation function.

ANNs can provide high prediction accuracy when sufficient training data are available. Research on neural-network-based power estimation has demonstrated that neural models can provide accurate estimates for digital circuits under appropriate experimental conditions. The



major limitation is the requirement for appropriate training data and careful hyperparameter selection.

V. DEEP LEARNING AND GRAPH NEURAL NETWORKS

Traditional ML methods generally treat circuit information as a fixed feature vector. However, a circuit is fundamentally a graph consisting of:

- cells,
- gates,
- nets,
- pins,
- connections,
- hierarchy.

Graph Neural Networks (GNNs) can represent this structural information directly.

A graph may be defined as:

$$G = (V, E)$$

where:

V= set of circuit nodes,

E= set of electrical/logical connections.

A simplified graph-convolution operation is:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)})$$

where:

$\hat{A}$ is the normalized adjacency matrix,

$H^{(l)}$ is the node representation,

$W^{(l)}$ is the trainable parameter matrix.

GNNs are particularly attractive for VLSI because the underlying design naturally has graph structure.

CircuitNet was developed to facilitate ML research in VLSI CAD and includes graph and physical-design

information. Its authors also identified data imbalance and transferability as important challenges for ML in CAD.

VI. REINFORCEMENT LEARNING FOR VLSI OPTIMIZATION

Reinforcement Learning (RL) differs from supervised learning because the agent learns through interaction with an environment.

The fundamental components are:

- State S
- Action A
- Reward R
- Policy π

For VLSI optimization:

State: current design configuration.

Action: change a design parameter.

Reward: improvement in PPA.

A possible reward formulation is:

$$R = -(w_p P_n + w_a A_n + w_t T_n)$$

A constraint penalty can also be introduced:

$$R = F(X) - \lambda C(X)$$

where $C(X)$ represents constraint violations.

Deep reinforcement learning has demonstrated strong potential for chip floorplanning. The Google/academic work on graph-based chip placement showed that reinforcement learning could generate competitive floorplans while optimizing important physical-design objectives. more recent research has investigated hybrid reinforcement-learning frameworks incorporating graph neural networks and transformers for physical-design parameter tuning.

Technique	Power Prediction	Area Prediction	Timing Prediction	DSE	Structural Awareness	Training Cost	Interpretability
Linear Regression	Moderate	Moderate	Moderate	Low	Low	Very Low	High
Decision Tree	Good	Good	Good	Good	Low	Low	High
Random Forest	Very Good	Very Good	Very Good	Good	Low	Low–Moderate	High
Gradient Boosting	Very Good	Very Good	Very Good	Good	Low	Moderate	Medium
ANN	Very Good	Very Good	Very Good	Good	Low	Moderate	Low
Deep Neural Network	Excellent*	Excellent*	Excellent*	Good	Medium	High	Low
GNN	Excellent*	Excellent*	Excellent*	Very Good	Very High	High	Low–Medium
Reinforcement Learning	Indirect	Indirect	Indirect	Excellent	High	Very High	Low
Hybrid ML/RL	Excellent*	Excellent*	Excellent*	Excellent	Very High	High–Very High	Medium



*Performance is highly dependent on dataset quality, architecture, feature representation, and training methodology.

The comparison shows that Random Forest and Gradient Boosting are attractive initial baselines because they provide a strong balance between prediction quality, training cost, and interpretability. Neural networks become more attractive as dataset size and nonlinear complexity increase. GNNs are especially relevant when circuit topology must be represented directly.

RL is fundamentally different: its primary advantage is sequential decision-making and design-space exploration rather than simple prediction.

VII. PROPOSED ML-ASSISTED PPA OPTIMIZATION FRAMEWORK

This study proposes a five-stage framework.

Stage 1: Design Generation

A set of RTL designs or benchmark circuits is selected.

Possible benchmark sources include:

- ISCAS benchmarks,
- ITC benchmarks,
- OpenROAD-compatible designs,
- RISC-V designs,
- CircuitNet datasets

CircuitNet currently provides multiple technology-specific datasets and ML-oriented EDA data, making it a useful source for reproducible research

Stage 2: Feature Extraction

The following features can be extracted:

- RTL-level features
- number of inputs,
- number of outputs,
- number of registers,
- number of combinational operators,
- number of arithmetic units,
- number of multiplexers,
- number of logic gates,
- hierarchy depth.

Synthesis-level features

- cell count,
- sequential-cell count,
- combinational-cell count,
- estimated area,
- timing slack,
- critical-path depth,
- switching activity.

Physical-design features

- utilization,

- congestion,
- wire length,
- macro count,
- routing demand,
- cell density,

VIII. MACHINE LEARNING MODELS

The proposed comparative study considers six principal model categories:

Model 1: Linear Regression

Used as the baseline.

Model 2: Random Forest Regression

Used for nonlinear tabular relationships.

Model 3: Gradient Boosting

Used to evaluate boosted decision-tree performance.

Model 4: Artificial Neural Network

Used for nonlinear PPA prediction.

Model 5: Graph Neural Network

Used to exploit circuit topology.

Model 6: Reinforcement Learning

Used for sequential design-space exploration.

IX DATASET PREPARATION

The dataset should be divided into:

- Training set: 70%
- Validation set: 15%
- Test set: 15%

However, random splitting should not be the only evaluation strategy. A stronger experiment should also use design-wise splitting, in which entire circuits are kept exclusively in the test set. This prevents information leakage caused by having closely related configurations of the same design in both training and test datasets.

Feature normalization should be applied where appropriate:

$$x' = (x - \mu) / \sigma$$

where μ is the feature mean and σ is the standard deviation.

X. PERFORMANCE METRICS

The following metrics are recommended.

1. Mean Absolute Error



$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

2. Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}$$

3. Coefficient of Determination

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

4. Mean Absolute Percentage Error

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

These metrics should be independently reported for power, area, and timing.

Multi-Objective Optimization

The central optimization problem can be formulated as:

$$\min_X [P(X), A(X), T(X)]$$

subject to:

$$P(X) \leq P_{max}$$

$$A(X) \leq A_{max}$$

$$T(X) \leq T_{max}$$

A weighted objective can alternatively be defined as:

$$J(X) = w_P \frac{P(X)}{P_{ref}} + w_A \frac{A(X)}{A_{ref}} + w_T \frac{T(X)}{T_{ref}}$$

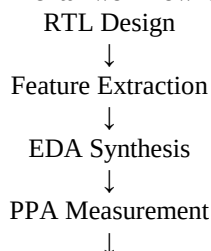
where:

$$w_P + w_A + w_T = 1$$

This formulation allows the optimization objective to be adjusted according to application requirements. For an IoT device, power may receive the highest weight. For a high-performance processor, timing may receive greater importance.

Experimental Workflow

The proposed experimental workflow is:



Dataset Construction

↓
ML Training

↓
PPA Prediction

↓
Design-Space Exploration

↓
Candidate Selection

↓
EDA Verification

↓
Final PPA Evaluation

The most important feature of the proposed workflow is that ML predictions should not replace final EDA verification. Instead, ML should reduce the number of expensive EDA evaluations.

Proposed Comparative Experiment

A rigorous implementation can evaluate the following six models:

- Linear Regression
- Random Forest
- Gradient Boosting
- ANN
- GNN
- Reinforcement Learning

For each model, three target variables should be considered:

- Power
- Area
- Timing

The experiment should answer four research questions.

RQ1-Which ML model provides the most accurate PPA prediction?

RQ2-Which model requires the least computational training cost?

RQ3-Which model provides the most effective design-space exploration?

RQ4- Which approach provides the best overall trade-off between prediction accuracy and optimization cost?

XI. EXPECTED COMPARATIVE INTERPRETATION

The expected behavior of the models can be explained conceptually rather than by fabricating numerical results. Linear Regression should provide the simplest baseline but may struggle with nonlinear relationships.

Decision Trees can capture nonlinear interactions but may suffer from instability when used individually.



Random Forest is expected to provide a strong baseline for structured PPA datasets because it can model nonlinear relationships and estimate feature importance.

Gradient Boosting can potentially achieve higher predictive performance than individual trees by sequentially correcting prediction errors.

ANNs can capture complex nonlinear relationships but require careful selection of architecture, learning rate, regularization, and training epochs.

GNNs have an important advantage when circuit connectivity and topology influence PPA outcomes. Their performance, however, depends strongly on graph representation and dataset scale.

RL is expected to be particularly useful for sequential optimization. Instead of merely predicting PPA, an RL agent can learn which design action should be taken next. This makes RL particularly suitable for design-space exploration, floorplanning, placement, and parameter tuning.

XII. DISCUSSION

- The comparative analysis demonstrates that the choice of ML technique should be determined by the nature of the VLSI problem.
- For early-stage PPA prediction with tabular features, Random Forest and Gradient Boosting provide practical solutions. They can operate on relatively small datasets and provide useful feature-importance information.
- For large datasets containing highly nonlinear relationships, neural networks become more attractive.
- For problems in which circuit connectivity is fundamental, graph-based learning provides a more natural representation. Rather than converting the entire circuit into a flat feature vector, GNNs preserve information about relationships among circuit components.
- For sequential design optimization, RL provides a fundamentally different paradigm. The agent learns a sequence of actions rather than producing only a prediction.
- This distinction is important. A highly accurate PPA prediction model does not automatically guarantee an optimal design. Prediction and optimization are related but different problems.

Advantages Of The Proposed Framework

The proposed framework provides several advantages.

1. Reduced EDA Evaluations

An ML model can eliminate clearly poor configurations before expensive synthesis or physical-design evaluation.

2. Faster Design-Space Exploration

ML-guided exploration can focus computational resources on promising configurations.

3. Multi-Objective Optimization

Power, performance, and area can be optimized simultaneously.

4. Reusability

A trained model can potentially be reused for related designs, although transferability must be carefully validated.

5. Automated Optimization

RL-based approaches can automate sequential parameter selection.

Limitations

Despite the advantages, several challenges remain.

1.Dataset Dependence

ML performance is strongly influenced by the quantity and diversity of training data.

2. Generalization

A model trained on one design family may not generalize well to another technology node or architecture.

3. EDA Tool Dependence

PPA values depend on:

- technology library,
- synthesis settings,
- clock constraints,
- physical constraints,
- placement algorithms,
- routing algorithms,
- optimization settings.

Therefore, an ML model trained under one EDA environment may not directly transfer to another.

4. Computational Cost

Deep learning and reinforcement learning can require substantial computational resources.

5. Interpretability

Deep neural networks and GNNs are often difficult to interpret compared with tree-based models.

6. Reward Design

For RL, an inappropriate reward function can cause the agent to optimize one metric at the expense of other PPA objectives.



XIII. REPRODUCIBILITY CONSIDERATIONS

For a strong journal submission, the following information should be reported:

RTL source or benchmark identifier.

- Technology node.
- Standard-cell library.
- EDA tool and version.
- Synthesis constraints.
- Clock frequency.
- Supply voltage.
- Training/test split.
- Feature list.
- ML hyperparameters.
- Random seeds.
- Number of EDA evaluations.
- Hardware used for training.
- PPA measurement methodology.

Public datasets such as CircuitNet can improve reproducibility because they provide standardized ML-for-EDA data and benchmark information.

XIV. FUTURE RESEARCH DIRECTIONS

Several research directions can extend the proposed framework

1. Hybrid GNN-RL Optimization

A GNN can encode circuit structure while an RL agent performs design-space exploration.

2. Transfer Learning

Models trained on one technology or design family can be adapted to another with limited additional data.

3. Explainable AI

SHAP-based analysis and feature attribution can help designers understand why a particular configuration produces better PPA.

4. Multi-Fidelity Optimization

Low-cost estimates can initially be used to eliminate poor candidates, followed by higher-fidelity EDA evaluation of promising configurations

5 Active Learning

The ML model can identify which new circuit configurations would provide the greatest information gain

6 Foundation Models for EDA

Large pretrained models may eventually learn transferable representations of RTL, netlists, placement information, and physical-design data.

XV. CONCLUSION

This paper presented a comparative analysis of machine-learning techniques for Power, Performance, and Area optimization in low-power VLSI design. The analysis shows that machine learning can significantly change the conventional design-space exploration process by providing early PPA estimation and intelligent search capabilities.

Traditional regression models provide useful baselines, while ensemble techniques such as Random Forest and Gradient Boosting offer attractive solutions for structured design data. Neural networks are suitable for complex nonlinear relationships, whereas graph neural networks provide a more natural representation of circuit topology. Reinforcement learning provides an additional advantage for sequential design-space exploration and physical-design optimization.

However, the results of ML-based optimization should always be validated using the underlying EDA flow. The main value of ML is therefore not to eliminate conventional VLSI tools but to reduce unnecessary tool evaluations and guide designers toward promising regions of the design space.

The most promising direction is a hybrid architecture combining predictive ML, graph representation, and reinforcement learning. Such a system can use ML models for rapid PPA estimation, GNNs for circuit-structure representation, and RL for sequential optimization

Ultimately, the effectiveness of ML-based VLSI optimization depends not only on algorithmic sophistication but also on dataset quality, technology-specific information, reproducibility, generalization, and integration with existing EDA workflows. A carefully designed hybrid ML-EDA framework therefore represents a practical pathway toward faster, energy-efficient, and increasingly autonomous VLSI design.

Declarations

Funding

No external funding was received for this study.

Conflict of Interest

The author declares that there is no conflict of interest.

Data Availability

The proposed study is designed to use publicly available VLSI benchmark circuits and/or publicly available ML-for-EDA datasets. Dataset identifiers and experimental configurations should be reported with the final experimentally validated version of the manuscript.



REFERENCES

1. Mirhoseini, A., Goldie, A., Yazgan, M., Jiang, J. W., Songhori, E., Wang, S., Lee, Y. J., Johnson, E., Pathak, O., Nazi, A., Pak, J., Tong, A., Srinivasa, K., Hang, W., Tuncer, E., Le, Q. V., Laudon, J., Ho, R., Carpenter, R., & Dean, J. (2021). A graph placement methodology for fast chip design. *Nature*, 594, 207–212. <https://doi.org/10.1038/s41586-021-03544-w>.
2. Chai, Z., Zhao, Y., Liu, W., Lin, Y., Wang, R., & Huang, R. (2023). CircuitNet: An open-source dataset for machine learning in VLSI CAD applications with improved domain-specific evaluation metric and learning strategies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(12), 5034–5047. <https://doi.org/10.1109/TCAD.2023.3287970>.
3. Jiang, X., Chai, Z., Zhao, Y., Lin, Y., Wang, R., Huang, R., et al. (2024). CircuitNet 2.0: An advanced dataset for promoting machine learning innovations in realistic chip design environment. *International Conference on Learning Representations (ICLR)*.
4. Agnesina, A., Rajvanshi, P., Yang, T., Pradipta, G., Jiao, A., Keller, B., Khailany, B., & Ren, H. (2023). AutoDMP: Automated DREAMPlace-based macro placement. *International Symposium on Physical Design*.
5. Khailany, B., Ren, M. H., Dai, S., Godil, S., Keller, B., Kirby, R., Klinefelter, A., Venkatesan, R., Zhang, Y., Catanzaro, B., & Dally, W. (2020). Accelerating chip design with machine learning. *IEEE Micro*.
6. Celik, E., & Dal, D. (2025). A systematic review of machine learning-driven design space exploration in high-level synthesis. *Very Large Scale Integration (VLSI) Systems*. <https://doi.org/10.1016/j.vlsi.2025.102513>.
7. Sabarad, J. M., Iyer, S., & Pandya, R. J. (2026). AI/ML for VLSI chip design automation: A comprehensive survey. *Proceedings of the Indian National Science Academy*. <https://doi.org/10.1007/s43538-026-00744-8>.
8. Papadopoulou, M., Dossis, M., & Karvounis, E. (2026). A comprehensive review of low-power design techniques in VLSI systems: From device physics to algorithmic optimization. *Analog Integrated Circuits and Signal Processing*, 128, Article 4. <https://doi.org/10.1007/s10470-026-02612-3>.
9. Govindaraj, V., Sahaya Sheela, M., Muthu Krishnammal, P., Hari Kumar, S., Jayanthi, T., et al. (2026). A hybrid random forest–ANFIS framework for accurate power estimation in CMOS VLSI circuits. *Discover Computing*, 29, Article 542. <https://doi.org/10.1007/s10791-026-10430-4>.
10. Gowrishankar, R., Govindaraj, V., Sharmila Devi, S., Ramesh, C., et al. (2025). Intelligent power estimation of digital circuits using random forest and neural network models. *International Journal of Computational Intelligence Systems*, 18, Article 263.
11. Machine Learning Framework for Early Power, Performance, and Area Estimation of RTL. (2025). *IEEE Conference Publication*. DOI: 10.1109/ICCC64910.2025.11077283.
12. Physical Synthesis Optimization Prediction Using Machine Learning. (2025). *IEEE VLSI Design and Embedded Systems Conference*. DOI: 10.1109/VLSID64188.2025.00018.
13. Optimizing Power, Performance, and Area in Digital Design Using ML Based Design Space Exploration. (2025). *IEEE Conference Publication*. DOI: 10.1109/CSITSS67709.2025.11295118.
14. A Hybrid Reinforcement Learning Framework for Efficient Physical Design Parameter Tuning. (2026). *ACM Transactions on Design Automation of Electronic Systems*, 31(3), Article 50. <https://doi.org/10.1145/3779423>.
15. Carothers, J. D., & Radjassamy, R. (1998). Low power VLSI design techniques—The current state. *Integrated Computer-Aided Engineering*, 5(2), 153–176. <https://doi.org/10.3233/ICA-1998-5206>.
16. Zhai, J., Ling, Z., Bai, C., Zhao, K., & Yu, B. Machine learning for microarchitecture power modeling and design space exploration: A survey. *Research publication on ML-assisted microarchitecture optimization*.