

MACHINE LEARNING – BASED AUTOMATED VERIFICATION AND PREDICTIVE BUG DETECTION IN RTL AND PHYSICAL LAYOUT DESIGNS: A LITERATURE SURVEY

Sahana Patil
Assistant Professor
Dept. of Electronics and Communication Engg.,
C. Byre Gowda Institute of Technology Kolar, India
patilsahana005@gmail.com

Dr. G K Venkatesh
Head of the Department
Dept. of Electronics and Communication Engg.,
C. Byre Gowda Institute of Technology Kolar, India
gkvlshiv@gmail.com

Prasad Shastry
Professor
Dept. of Electrical and Computer Engg.,
Bradley University, Peoria, Illinois, USA
snp@bradley.edu

Kavya S Assistant Professor
Dept. of Electronics and Communication Engg.,
C. Byre Gowda Institute of Technology
Kolar, India ghanavigk@gmail.com

Abstract: The growing complexity of modern VLSI systems has significantly increased the burden on conventional verification and validation techniques. Traditional approaches struggle to meet today's stringent requirements related to time-to-market, cost efficiency, and design reliability. In recent years, Machine Learning (ML) has emerged as a powerful enabling technology capable of transforming verification workflows. By learning from historical design and verification data, ML techniques can automate regression analysis, guide intelligent test generation, and predict defects early at both the Register Transfer Level (RTL) and physical layout stages. This paper presents a comprehensive literature survey of ML-driven automated verification and predictive bug detection methodologies. The reviewed research spans supervised, unsupervised, ensemble, graph-based, and reinforcement learning techniques applied to verification planning, directed stimulus generation, RTL defect prediction, security-aware RTL analysis, routing congestion estimation, design rule checking (DRC) prediction, placement optimization, and early power-performance-area (PPA) estimation. The survey also critically examines modeling strategies, dataset limitations, evaluation metrics, deployment constraints, and integration challenges within industrial Electronic Design Automation (EDA) flows. Finally, research gaps related to dataset scarcity, cross-design generalization, interpretability, and lifecycle integration are identified, highlighting promising directions for future intelligent verification ecosystems.

Keywords: Machine learning, hardware verification, RTL bug prediction, defect prediction, EDA, reinforcement learning, DRC prediction, layout optimization.

I. INTRODUCTION

All Functional verification accounts for a major portion of hardware development effort, often consuming more than 60% of the overall design cycle. Conventional verification strategies such as simulation-based testing, constrained-random verification, formal verification, assertion-based checking, and static linting have served the industry for decades. However, as designs grow in complexity, include heterogeneous architectures, and target advanced technology nodes, these approaches increasingly face scalability challenges. State-space explosion, long regression cycles, and coverage closure difficulties make traditional methodologies time-consuming and resource-intensive.

Recent advances in machine learning show that verification artifacts—such as logs, waveforms, and coverage data—contain learnable patterns. Using this data, ML models can predict defects, optimize testing, and identify design bottlenecks early. This survey presents a unified view of ML-based verification across both RTL and physical design stages.

II. Foundational Surveys and ML Taxonomy in Verification

The structured study of ML applications in hardware verification begins with several foundational surveys.

Work in [1] categorizes supervised, unsupervised, and reinforcement learning approaches applied to problems such as SAT solving, coverage improvement, and assertion generation, effectively framing verification tasks as learnable optimization problems.

Building on this taxonomy, [2] expands the discussion by incorporating industrial case studies and highlighting practical deployment challenges such as class imbalance, feature engineering complexity, and integration within regression infrastructures.

A broader perspective is presented in [8], which reviews the historical evolution of ML-assisted hardware verification and discusses issues related to scalability, dataset reproducibility, and cross-project generalization. Similarly,

[10] positions ML as a cross-disciplinary enabler across EDA stages, emphasizing the growing importance of Graph Neural Networks (GNNs) for modeling structural hardware representations such as netlists and routing graphs.

Together, these surveys establish the conceptual and methodological foundation for ML-driven verification research.

III. ML-Based Verification Planning and Test Optimization

Verification planning aims to reduce regression cost while maximizing coverage efficiency. Several works demonstrate how ML can optimize this process.

In [3], a supervised learning framework predicts the expected coverage contribution of individual tests based on historical regression and design data. By ranking tests according to anticipated coverage gain, redundant simulations can be eliminated, significantly reducing verification runtime.

Reinforcement learning is explored in [5], where verification is modeled as a sequential decision-making problem. An RL agent dynamically adjusts stimulus strategies based on coverage feedback, learning to maximize verification efficiency while maintaining computational feasibility.

Directed test generation techniques are surveyed in [6], comparing formal constraint-solving approaches with ML-guided stimulus strategies. The study highlights how learning-based prioritization enhances corner-case exploration.

Regression pruning techniques in [21] use coverage delta prediction models to remove low-impact tests without compromising defect detection quality. Adaptive stimulus ranking in [22] applies classification models to identify high-impact verification scenarios, accelerating coverage closure.

Failure clustering methodologies in [23] analyze historical bug traces to predict regression scenarios likely to expose latent defects. Hybrid frameworks such as [31] combine supervised prediction and reinforcement learning to continuously refine regression strategies based on real-time outcomes.

Collectively, these works demonstrate measurable reductions in simulation cycles and improved regression efficiency.

IV. RTL-Level Predictive Bug Detection and Refactoring

Early identification of RTL defects significantly reduces downstream debugging effort. Trend-based modeling in

[7] predicts RTL refactoring needs by analyzing historical verification outcomes. Similarly, [9] proposes a feedback-loop framework in which verification results are continuously fed back into RTL improvement models, creating a closed-loop learning environment between design and verification.

The transfer of software defect prediction methodologies to hardware is explored in [12] and [13], where metrics-based and ensemble learning models identify fault-prone modules. These studies provide foundational principles applicable to RTL-level predictive analytics. Further research in [24], [25], and [26] investigates structural and complexity-based features such as fan-out, FSM states, module coupling, and assertion density to model defect likelihood. These approaches demonstrate that well-engineered structural metrics can significantly enhance prediction accuracy.

V. ML-Based Sequence Generation for Hardware Verification

Because verification involves sequential stimulus application, temporal learning models are particularly suitable. In [14], stimulus generation is formulated as a sequence learning problem, leveraging historical failure traces to generate optimized verification sequences.

Advanced architectures explored in [26] and [27] use recurrent and transformer-based models to predict next-best verification actions based on coverage progression patterns. These techniques improve exploration efficiency and enhance corner-case detection compared to heuristic-driven methods.

VI. Data-Centric ML Frameworks for Verification

Model performance strongly depends on data quality.

The data-centric pipeline proposed in [15] emphasizes dataset curation, normalization, labeling consistency, and iterative retraining to improve robustness. Early PPA estimation using ML at the RTL stage is addressed in [20], enabling designers to assess implementation risks before physical synthesis. Cross-project generalization is tackled in [28] and [29] through dataset augmentation, transfer learning, and synthetic defect injection techniques. These strategies aim to overcome limited labeled datasets and improve scalability across diverse design environments.

VII. Security-Aware RTL Analytics

Security vulnerabilities represent a critical class of hardware defects. The survey in [4] reviews AI-augmented secure RTL design methodologies for detecting hardware Trojans. Techniques such as anomaly detection and structural fingerprinting are applied to identify malicious modifications prior to fabrication, strengthening design security in the supply chain.

VIII. ML in Physical Layout and Design Rule

Machine learning approaches have increasingly influenced backend physical design tasks, particularly in early-stage prediction and optimization. In CongestionNet [16], deep graph neural networks are utilized to estimate routing congestion prior to global routing, enabling designers to identify problematic regions at an early stage. Overall by enhancing graph-based congestion modeling with improved feature representation and spatial learning mechanisms, leading to better prediction stability across complex layouts.

Unsupervised clustering strategies reported in [17] focus on identifying potential DRC hotspots during the initial layout phases, reducing the need for

repeated rule-check iterations. Similarly, ensemble-based classification techniques discussed in [18] enhance DRC prediction accuracy by combining multiple learning models, thereby improving generalization across different design blocks. Placement optimization has also benefited from learning-based strategies; the reinforcement learning framework described in [19] demonstrates improved power, performance, and area trade-offs when compared with conventional heuristic-driven placement methods.

Furthermore, the multi-resolution congestion estimation framework MEDUSA [25] introduces hierarchical spatial analysis to refine congestion prediction across varying layout granularities.

Beyond performance improvements, recent investigations such as [30] address the broader implications of integrating ML into EDA workflows, particularly emphasizing model reliability, robustness, and security considerations. These studies underline the importance of developing trustworthy and verifiable ML models before their widespread adoption in physical design automation.

IX. FPGA-Specific Perspectives

ML integration in FPGA design flows presents unique constraints. As discussed in [11], challenges include limited publicly available datasets, vendor-specific tool variability, and reduced reproducibility across FPGA platforms. These deployment considerations require tailored ML solutions for FPGA-based EDA ecosystems.

X. Comparative Analysis and Research Gaps

Despite notable progress, several critical challenges remain. A primary limitation is the scarcity of publicly available RTL and layout defect benchmark datasets, which restrict reproducibility and large-scale evaluation. Additionally, many models struggle with cross-project generalization, as performance often degrades when applied to different architectures or technology nodes. Explainability remains insufficient, particularly for safety-critical systems where interpretable predictions are essential. Most existing solutions operate in isolation, addressing either RTL or physical design stages independently, resulting in fragmented lifecycle coverage. Furthermore, the cost of labeling defect data for supervised learning limits scalability. Future research must therefore focus on developing unified, lifecycle-aware ML frameworks capable of continuous learning across design stages while ensuring robustness, interpretability, and industrial compatibility.

XI. Expected Results

Based on the surveyed literature, ML-based automated verification frameworks are expected to deliver measurable improvements across RTL and physical design stages. At the RTL level, supervised regression ranking and reinforcement learning-based stimulus optimization can reduce redundant simulation cycles and accelerate coverage closure. Predictive defect models can identify fault-prone modules early, minimizing late-stage debugging effort. Sequence-learning approaches can

enhance bug discovery by intelligently navigating complex state spaces. At the physical design level, graph-based models and ensemble predictors enable early routing congestion and DRC violation detection.

Reinforcement learning-driven placement tuning can achieve improved PPA trade-offs compared to manual optimization. Data-centric ML pipelines improve robustness and generalization through systematic dataset management. Overall, integrating ML across the RTL-to-layout lifecycle is expected to reduce verification turnaround time, lower computational cost, improve defect localization accuracy, and enhance predictability in achieving coverage and PPA targets within industrial EDA workflows.

XII. Conclusion

This survey has examined machine learning approaches for automated verification and predictive defect detection across rtl and physical layout stages. The literature demonstrates that historical design and

Verification data can be effectively leveraged to reduce redundant simulation cycles, accelerate coverage closure, and enable early defect prediction.

Future research must prioritize scalable, explainable, and lifecycle-integrated ml frameworks that can be seamlessly embedded within industrial eda tool chains. Such

Advancements will pave the way toward intelligent, adaptive, and self-optimizing hardware verification ecosystems.

REFERENCES

- [1]. A, Jayasudha, and D, Pugazhenth, "Classification of Textures using Patch Based Energy Features of Selected Wavelet Coefficients", Journal of Applied Sciences, Vol. 14, No. 7, 2014, pp. 709-713.
- [2]. K. A. Ismail and M. A. A. El Ghany, "Survey on machine learning algorithms enhancing the functional verification process," Electronics, vol. 10, no. 21, p. 2688, 2021.
- [3]. D. Yu, H. Foster, and T. Fitzpatrick, "A survey of machine learning applications in functional verification," DVCon US, 2023.
- [4]. H. Jang, S. Yim, S. Choi, S. B. Choi, and A. Cheng, "Machine learning based verification planning methodology using design and verification data," Design and Verification Conference (DVCon), 2022.
- [5]. R. Parikh and K. Parikh, "A survey on AI-augmented secure RTL design for hardware Trojan prevention," 2025.
- [6]. A. Dinu and P. L. Ogrutan, "Reinforcement learning made affordable for hardware verification engineers," Micromachines, vol. 13, no. 11, p. 1887, 2022.
- [7]. A. Jayasena and P. Mishra, "Directed test generation for hardware validation: A survey," ACM Computing Surveys, vol. 56, no. 5, pp. 1–36, 2024.

- [10]. H. Agoro, "Predicting RTL refactoring needs using trend analysis from historical verification outcomes and ML models," 2024.
- [11]. N. Wu et al., "Survey of machine learning for software-assisted hardware design verification: Past, present, and prospect," *ACM Transactions on Design Automation of Electronic Systems*, vol. 29, no. 4, pp. 1–42, 2024.
- [12]. H. Agoro, "Creating a feedback loop between verification results and RTL design improvements using machine learning," 2024.
- [13]. G. Huang et al., "Machine learning for electronic design automation: A survey," *ACM Transactions on Design Automation of Electronic Systems*, vol. 26, no. 5, pp. 1–46, 2021.
- [14]. A. Biscontini, E. Popovici, and A. Temko, "Machine learning for FPGA electronic design automation," *IEEE Access*, 2024.
- [15]. G. Esteves, E. Figueiredo, A. Veloso, M. Viggiano, and N. Ziviani, "Understanding machine learning software defect predictions," *Automated Software Engineering*, vol. 27, no. 3, pp. 369–392, 2020.
- [16]. M. K. Thota, F. H. Shajin, and P. Rajesh, "Survey on software defect prediction techniques," *International Journal of Applied Science and Engineering*, vol. 17, no. 4, pp. 331–344, 2020.
- [17]. M. Gad, M. Aboelmaged, M. Mashaly, and M. A. Abd El Ghany, "Efficient sequence generation for hardware verification using machine learning," *IEEE Access*, 2021.
- [18]. H. Shin, "Data-centric machine learning pipeline for hardware verification," in *Proc. IEEE 35th Int. System-on-Chip Conf. (SOCC)*, 2022.
- [19]. R. Kirby, S. Godil, R. Roy, and B. Catanzaro, "CongestionNet: Routing congestion prediction using deep graph neural networks," in *Proc. IFIP/IEEE 27th Int. Conf. on Very Large Scale Integration (VLSI-SoC)*, 2019.
- [20]. R. Islam and D. Challagundla, "Pre-global routing DRC violation prediction using unsupervised learning," in *Proc. IEEE 23rd Interregional NEWCAS Conf. (NEWCAS)*, 2025.
- [21]. R. Islam, "Early stage DRC prediction using ensemble machine learning algorithms," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 45, no. 4, Fall 2022.
- [22]. A. Agnesina, K. Chang, and S. K. Lim, "Parameter optimization of VLSI placement through deep reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 4, pp. 1295–1307, Apr. 2023.
- [23]. A. Chattopadhyay and V. K. Sutrar, "Machine learning framework for early power, performance, and area estimation of RTL," in *Proc. 6th Int. Conf. on Control, Communication and Computing (ICCC)*, 2025.
- [24]. S. Gogri et al., "Machine learning-guided stimulus generation for functional verification," *DVCon Proceedings*, 2020.
- [25]. N. Masamba, K. Eder, and T. Blackmore, "Supervised learning for coverage-directed test selection in simulation-based verification," 2022.
- [26]. S. Kumari, D. N. Gadde, and A. Kumar, "Optimizing coverage-driven verification using machine learning and PyUVM," 2025.
- [27]. Y. Attaoui et al., "Machine learning in VLSI design: A comprehensive review," *Journal of Integrated Circuits and Systems*, 2024.
- [28]. Z. Zhou et al., "MEDUSA: Multi-resolution machine learning method for congestion estimation,"
- [29]. *IEEE/ACM Design Automation Conference (DAC)*.

- [33]. R. Islam and D. Challagundla, "Pre-global routing DRC violation prediction using unsupervised learning,"
- [34]. *IEEE NEWCAS*, 2025.
- [35]. R. Islam, "Early stage DRC prediction using ensemble machine learning algorithms," *IEEE Canadian Journal of Electrical and Computer Engineering*, 2022.
- [36]. V. B. Pawar, "Application of machine learning to physical design," ScholarWorks, 2022.
- [37]. V. A. Chhabria *et al.*, "A machine learning approach to improving timing prediction and
- [38]. optimization," *ACM/IEEE* publication, 2023.
- [39]. B. Nadimi *et al.*, "Coverage-driven direct preference optimization for Verilog stimulus generation," 2025.
- [40]. Z. Xie *et al.*, "Security and reliability concerns in machine learning for EDA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (preprint/2025).