2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Leveraging AI/ML Models for Enhanced VLSI Design and Verification

Sudha Jain (sudhaj@synopsys.com)
Ayush Jain (ayush.jain@onsemi.com)

Abstract

This paper presents a dual-focused analysis aimed at fortifying the functionality, performance, and safety of digital circuits while optimizing verification processes. Firstly, leveraging machine learning algorithms, particularly Decision Trees, the study predicts input stimuli outcomes to address coverage gaps effectively. This predictive modeling not only enhances functional verification but also optimizes simulation cycles, thereby fostering a more efficient verification environment. Secondly, the investigation encompasses a comprehensive range of failure modes, including timing violations, power integrity issues, logic errors, memory access failures, and temperature-related challenges. Results showcase a notable reduction in simulation cycles required for full coverage closure compared to existing methodologies. Key takeaways underscore the augmented predictive capabilities, data-driven insights, and iterative refinement facilitated by Failure Mode Analysis using machine learning models. Future directions include validating the scalability of machine learning models with more complex Design Under Test (DUT) configurations and integrating multi-modal data sources to enhance reliability and accuracy in the design process. This interdisciplinary approach holds promise for advancing the robustness and efficiency of digital circuit designs across diverse industries and applications.

I. INTRODUCTION

Hardware systems play a pivotal role in our interconnected world, from consumer electronics to critical infrastructure. However, even the most meticulously designed systems can encounter unexpected failures. The challenge lies in identifying these failure modes early, allowing for timely mitigation and robust system development. Our research focuses on two critical objectives:

Predictive Models for Early Failure Mode Identification: The primary goal is to create predictive models capable of identifying potential failure modes during the design phase. By harnessing machine learning algorithms, such as the Random Forest Classifier, we aim to detect subtle patterns and features associated with failures. These models learn from labelled datasets, enabling them to proactively predict failure scenarios. The application of Failure Mode Analysis (FMEA) principles enhances our understanding of failure dynamics across various phases of a product or system lifecycle. FMEA, a versatile tool, provides actionable insights, early issue detection, and continuous improvement opportunities. Its adaptability allows us to address failure risks comprehensively. [4]

Functional Coverage Optimization with Reduced Simulation Cycles: Beyond failure prediction, we propose a novel approach to increase functional coverage while minimizing simulation cycles. Our Machine Learning model focuses on predicting correct input stimuli. By doing so, we optimize coverage metrics, ensuring that critical scenarios are thoroughly explored. We explore a range of machine learning techniques, including Artificial Neural Networks (ANN), Deep Neural Networks (DNN), and

Support Vector Regression (SVR). These models allow us to systematically randomize input values, hitting planned coverage targets efficiently. [3]

Scope and Future Directions: The field of Failure Mode Analysis is vast, and our research opens doors for deeper exploration. As we refine our models and expand their applicability, we anticipate breakthroughs in hardware reliability. By combining predictive modelling and FMEA, we pave the way for resilient systems that withstand real-world challenges.

II. HELPFUL HINTS

Why Random Forest?

- **Bootstrap Aggregation (Bagging):** Predicts results by combining multiple decision trees. Prevents overfitting. [6]
- **Variance Reduction:** Averages predictions from diverse trees. Yields accurate and robust models.
- **Flexibility:** Handles classification and regression tasks.
- **Noise Resilience:** Less impacted by noisy data.

The formula for accuracy is given by:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Samples}} \times 100$$

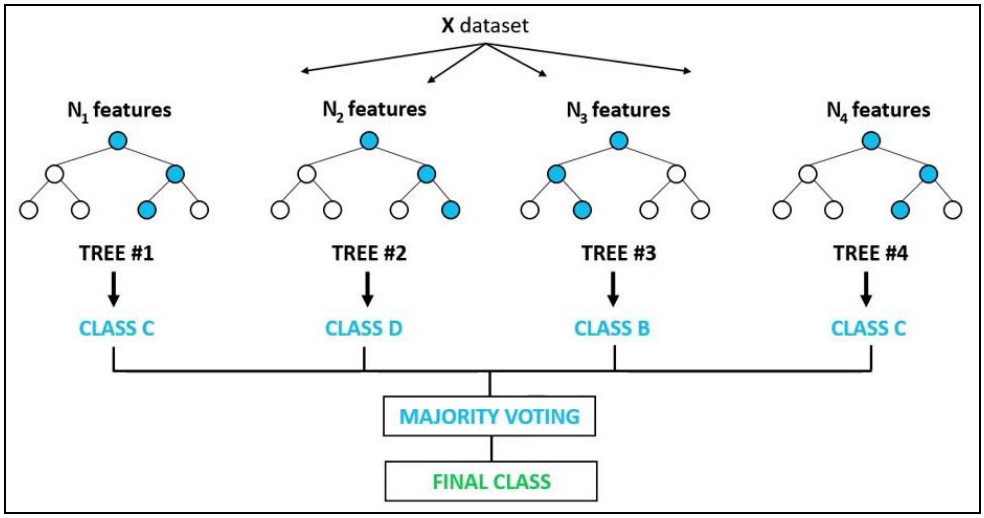

Fig 1: Random Forest Classifier [5]

Failure Mode Analysis using ML:

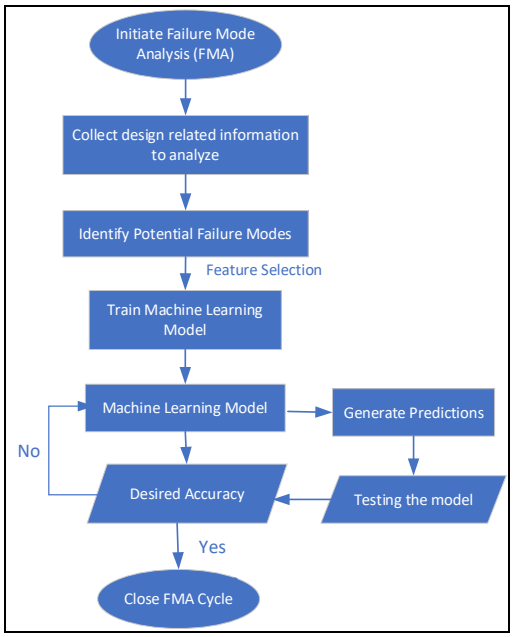

Fig 2: Failure Mode Analysis using Machine Learning Models

101 to 125 of 1000 entries Filter

index	ClockPeriod_ns	SetupTime_ns	HoldTime_ns	DataArrival_ns	ClockToQDelay_ns	TimingViolation
100	1.2828626711806081	0.2099010854704531	1.838671129973382	1.7410038644109376	2.0784383151206276	0
101	6.727693701374023	1.9412949972675415	1.1127548477676659	6.987074797409851	1.5659958920040564	0
102	3.82920382968694	1.7791931814305089	0.4001141633871612	8.316389310788471	1.870363006820055	0
103	5.577136220482325	1.8627293380709047	1.4222083245207247	4.472167781670071	0.9366179647408239	0
104	9.168098265334837	1.990324863028227	1.6071965658326834	3.0696931372435277	1.2398300735915404	0
105	3.2436300623398746	0.4304009735170259	0.7018471864283006	4.475226552052878	2.1595611871318128	0
106	4.693446307320667	0.8528598359150515	1.7286405882976306	6.787582542800008	2.9132568090759996	0
107	7.799960246887438	1.5406531038377735	1.8216721840549561	1.0724332926429678	0.626041934005523	0
108	3.059183489424602	1.4224391743022053	0.6261185260178369	5.7156319158775775	2.7259608021381654	0
109	1.6928191884591368	0.3924022220457246	1.968690798158922	5.9063334790990325	1.942203003278021	1
110	3.6077630762239123	1.6500829374821737	0.367351903292641	6.101505975167484	1.9097989168890592	1
111	2.4509915852860398	0.5264370864896553	0.4838297699647309	2.9361766398470177	1.750696338019401	0
112	9.367278871083158	0.5252534681629838	0.4500271945522496	6.712421787025014	0.6734852147604584	0
113	8.273083416079754	1.1202514034974858	1.7985804490149537	1.7311318906542112	0.7246921293404016	0
114	6.700633808593811	1.226585876199236	1.3431558494174276	2.5941626917344065	2.0022800141477153	0
115	8.84314531168946	1.2021637948918413	0.3889981381223184	8.019760373235854	1.3523964285888956	0
116	8.23304869209203	0.27382499105573427	0.936614494899863	6.909785205812531	2.7931701664442015	0
117	2.6791305299743224	1.7671756389977227	1.269066247632624	1.109964432090762	1.5164763433986805	0
118	9.0330309864098	0.6046400809185415	0.25858176147982825	2.9870100829584016	0.8582036467566028	0
119	5.854080177240856	0.34607835043732793	1.7765910999739436	2.7152518639453165	2.28679708375509	0
120	8.266961396476564	1.7886213517510567	1.626846717338917	3.1664506242957806	1.2333835536025246	0
121	9.06482169931144	1.9157378466365313	1.0598928697956127	2.9804781622390912	1.8136757683897329	1
122	3.8620312747467747	1.7380424728043562	1.9375405689924239	1.500210365942766	2.244581963762925	0
123	1.9904673207490908	1.6380805419772684	0.8937458606226893	4.671524132774563	2.750660278005176	0
124	3.051416462877475	1.3449597632141412	1.969809606169345 8	6.861867798289321	2.48047798700419	1

Show 25 per page 1 2 3 4 5 6 10 30 40

Fig 3: Sample Dataset for Timing Violations

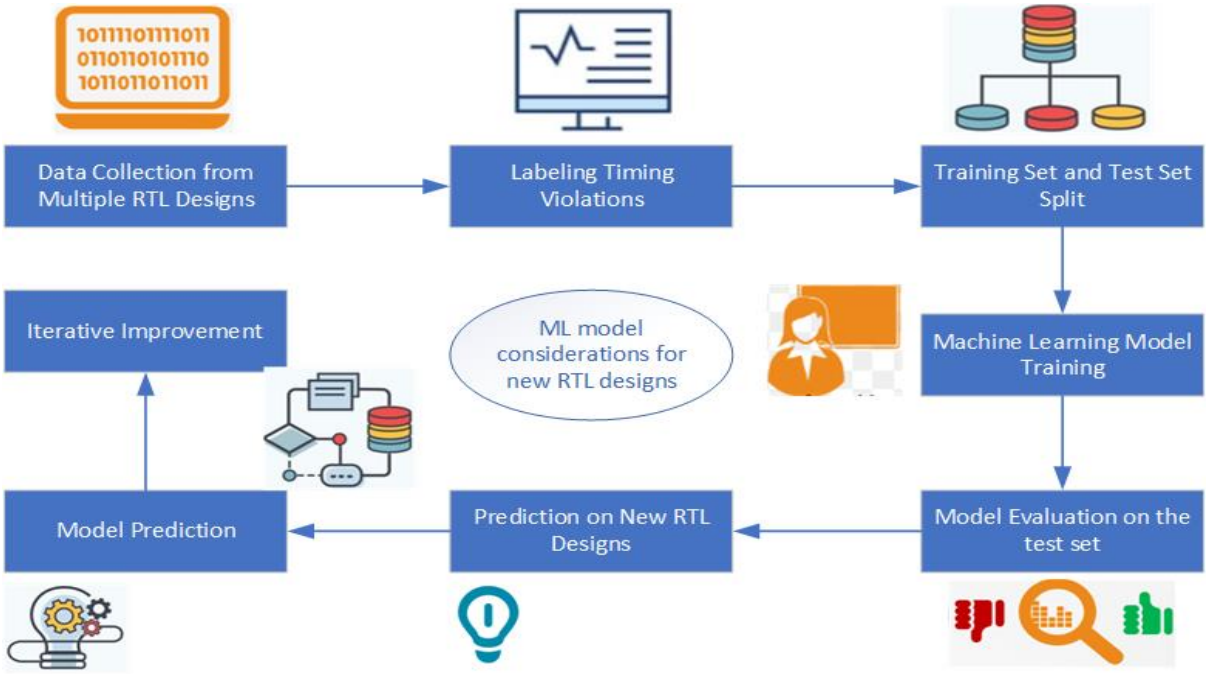

Fig 4: ML Model Approach to Detect Failures in New Designs

Obtained Results using Random Forest Classifier:

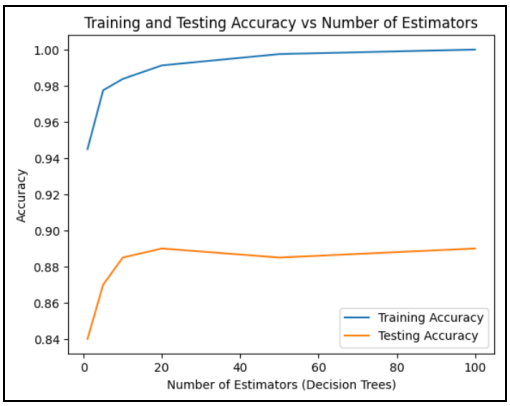

Fig 5: Accuracy Plot Training vs Testing for Failure Mode Analysis

Neural Network Results:
Accuracy: 0.865
Classification Report:

	Precision	recall	f1-score
0	0.87	0.99	0.95

Support Vector Machine Results:
Accuracy: 0.875
Classification Report:

	Precision	recall	f1-score
0	0.88	0.99	0.95

Random Forest Results:
Accuracy: 0.885
Classification Report:

	Precision	recall	f1-score
0	0.89	0.99	0.95

Fig 6: Accuracy Scores for FMA using different ML Models

NOTE: Number of Estimators represents No. of Decision Trees used in Random Forest

Constrained Random Verification using ML:

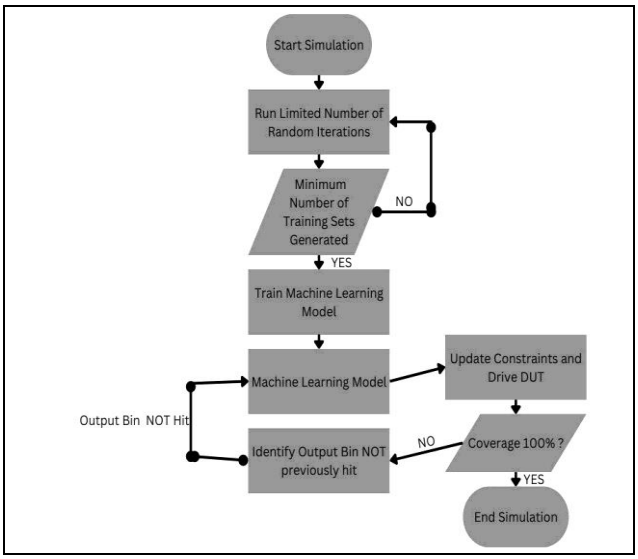

Fig 7: Constrained Random Verification using Machine Learning Models [1]

Obtained Results using Random Forest Classifier:

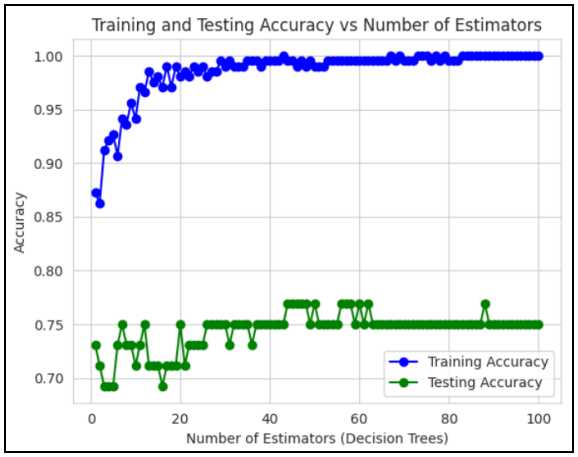

Fig 8: Accuracy Plot Training vs Testing for Constrained Random Verification

```
Original Shape: (4,)
Reshaped Shape: (1, 4)
Iteration 1 - Input: ['1' '0' '0' '0'], Output: 0001
Current Coverage: {'output_bin_coverage': 0}
Original Shape: (4,)
Reshaped Shape: (1, 4)
Iteration 2 - Input: ['1' '0' '1' '0'], Output: 0010
Current Coverage: {'output_bin_coverage': 0}
Original Shape: (4,)
Reshaped Shape: (1, 4)
Iteration 3 - Input: ['0' '0' '1' '0'], Output: 0001
Current Coverage: {'output_bin_coverage': 0}
Original Shape: (4,)
Reshaped Shape: (1, 4)
Iteration 4 - Input: ['1' '0' '0' '1'], Output: 0010
Current Coverage: {'output_bin_coverage': 0}
Original Shape: (4,)
Reshaped Shape: (1, 4)
Iteration 5 - Input: ['0' '0' '0' '0'], Output: 0000
Current Coverage: {'output_bin_coverage': 100}
Coverage reached 100%. Ending simulation.
```

Fig 9: Iterations showing output bin is hit and coverage value is updated

Results showing Reduced Number of Simulations:

Width of Inputs	No. of Iterations	Optimized No. of Iterations
1	3	3
2	33	16
3	140	23
4	800	26
5	3981	90

Fig 10: Result showing No. of Iterations required to hit coverage for different width input without using ML and with using Random Forest Classifier

III. SUMMARY

Applying Failure Mode Analysis (FMEA) to digital Very-Large-Scale Integration (VLSI) design proactively identifies potential issues, enhancing chip reliability. FMA serves as an essential risk management tool, allowing targeted optimization. Failure modes encompass timing violations, power integrity problems, logic errors, memory access failures, temperature-related issues, and communication failures. Leveraging machine learning (ML) models, including Random Forest, predicts input stimuli coverage. Benefits include enhanced predictive capabilities, data-driven insights, improved accuracy, and iterative refinement. Future directions involve testing ML models on complex designs and integrating multi-modal data sources for reliability and accuracy enhancement.

III. ABBREVIATIONS AND ACRONYMS

FMEA: Failure Mode Analysis
DUT: Design Under Test
ANN: Artificial Neural Networks (ANN)
DNN: Deep Neural Networks
SVR: Support Vector Regression

V. ACKNOWLEDGMENT

We extend our sincere appreciation to all those who supported and contributed to this research. Special thanks to our advisors, colleagues, and the research community for their valuable insights and feedback. Their guidance and encouragement have been instrumental in shaping this work.

VI. REFERENCES

[1] Sarath Mohan Ambalakkat, Eldon Nelson," Simulation Runtime Optimization of Constrained Random Verification using Machine Learning Algorithms", Design and Verification Conf. (DVCON), 2019.

[2] https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.

[3] Deepthi Amuru, Harsha V. Vudumula "AI/ML Algorithms and Applications in VLSI Design and Technology ". https://arxiv.org/abs/2202.10015

[4] https://quality-one.com/dfmea/

[5] https://www.kaggle.com/code/prashant111/random-forest-classifier-tutorial

[6] https://www.ibm.com/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems.