International Journal of Internet of Things (IJIOT)
Volume 3, Issue 1, January-June 2025, pp. 1-8, Article ID: IJIOT_03_01_001
Available online at https://iaeme.com/Home/issue/IJIOT?Volume=3&Issue=1
Impact Factor (2025): 5.00 (Based on Google Scholar Citation)
Journal ID: 6542-5231

INTERPRETABLE MACHINE LEARNING FOR ROOT CAUSE ANALYSIS IN CHIP VALIDATION FAILURES ACROSS MULTI-DIE SYSTEMS

Corey Wilson
Data Scientists, USA.

ABSTRACT

As multi-die systems become prevalent in modern semiconductor architectures, validation complexities have increased significantly. Root cause analysis (RCA) of failures in such complex systems demands interpretable methods that ensure engineers can understand and act upon the outcomes. This paper explores how interpretable machine learning (ML) techniques—particularly decision trees, LIME, and SHAP—can be applied to RCA for validation failures in multi-die chips. We demonstrate how model explainability can support engineers in isolating faults more efficiently than traditional rule-based or black-box ML models.

Keywords: interpretable machine learning, root cause analysis, chip validation, multi-die systems, SHAP, LIME, system-on-chip (SoC), hardware debugging

Cite this Article: Corey Wilson. (2025). Interpretable Machine Learning for Root Cause Analysis in Chip Validation Failures Across Multi-Die Systems. *International Journal of Internet of Things (IJIOT)*, 3(1), 1–8.

https://iaeme.com/MasterAdmin/Journal_uploads/IJIOT/VOLUME_3_ISSUE_1/IJIOT_03_01_001.pdf

1. Introduction

The growing complexity of integrated circuits, especially with the advent of 2.5D and 3D multi-die packaging, has led to intricate failure mechanisms that span across multiple dies and subsystems. Traditional debugging tools and rule-based scripts often fall short in diagnosing validation failures, primarily due to scalability and generalization limitations. These challenges highlight the necessity for data-driven methods that not only detect anomalies but also explain the origin of failures.

Interpretable machine learning provides a novel solution by ensuring transparency in prediction and reasoning processes. Unlike black-box models, interpretable ML enables engineers to correlate output anomalies with specific signals, blocks, or die interfaces. In this paper, we investigate the use of decision trees, SHAP (SHapley Additive exPlanations), and LIME (Local Interpretable Model-agnostic Explanations) for RCA in multi-die validation pipelines. Our dataset is derived from post-silicon and emulation logs of a leading-edge SoC project.

2. Literature Review

Researchers have long explored the use of machine learning in hardware validation. Zhang et al. (2017) utilized SVMs to localize faults in processor traces, while Huang and Mitra (2016) proposed statistical learning for pre-silicon verification data. However, these works often rely on black-box models, offering little interpretability. Lee and Chen (2019) presented a framework using decision trees for embedded software fault detection, one of the early efforts toward model transparency.

In terms of interpretable ML, Ribeiro et al. (2016) introduced LIME, which later influenced applications in cybersecurity and health diagnostics. Lundberg and Lee (2017) expanded on this with SHAP, providing game-theoretic explanations of predictions. Although these methods were primarily tested on text and image domains, their potential for chip validation has remained unexplored. This paper addresses this gap.

3. Problem Statement

Debugging chip validation failures across multi-die systems is plagued by the volume and heterogeneity of validation data. Moreover, engineers face delays in RCA due to unclear

correlations among system-level logs, firmware responses, and on-die monitors. Our goal is to develop an interpretable ML model that can:

- Predict failure clusters accurately.
- Highlight which signals/features contributed most to the prediction.
- Enable traceability of system-level failures back to specific dies or interfaces.

4. Methodology

To investigate interpretable machine learning (IML) techniques for root cause analysis (RCA) in multi-die validation environments, we developed a pipeline combining preprocessing, model training, and post-hoc explanation stages. The dataset originated from a 7nm multi-die system-on-chip project and included 20,000+ validation instances labeled with known failure categories. Each instance recorded low-level telemetry (performance counters, error flags, trace buffers) and high-level protocol logs.We built a framework that integrates data preprocessing, feature extraction, and ML-based interpretation. The raw validation data includes on-die performance counters, interface failure flags, firmware logs, and trace buffers. We processed this data into tabular features aligned by timestamp and subsystem IDs.

Three interpretable models were tested:

- Decision Tree Classifiers for their inherent transparency.
- **SHAP** for feature-level attribution.
- **LIME** for instance-level analysis.

4.1 Data Engineering

The first step was standardizing data across heterogeneous sources. Interface logs, firmware traces, and die-level telemetry were aligned using a global timestamp and synchronized using a sliding time window of 200ms. We engineered over 500 features such as signal deltas, retry counts, power fluctuation rates, and protocol negotiation states. Categorical fields were one-hot encoded, and outlier signals were clipped to a fixed threshold to ensure robustness.

4.2 Model Selection and Training

We experimented with three model types:

- **Decision Trees:** Offers built-in interpretability through tree visualizations.
- **XGBoost + SHAP:** Combines the power of gradient boosting with SHAP values for global/local attribution.

- **LIME + Logistic Regression:** Used to analyze individual predictions and understand signal perturbations.

Cross-validation was used to prevent overfitting. Each model was evaluated on accuracy, interpretability, and speed of insight delivery.

5. Feature Importance Visualization

A key contribution of this work is the integration of SHAP to explain model decisions in an intuitive and engineer-friendly format. SHAP assigns a contribution score to every input feature, enabling precise identification of failure triggers.

Table 1. Top 10 Features Identified by SHAP

Rank	Feature Name	Mean SHAP Value	Die	Feature Type
1	Interface_Fault_A	0.423	Die 2	Communication Fault
2	Clock_Drift_B	0.385	Die 1	Timing Variation
3	Power_Surge_C	0.361	Die 3	Power Integrity
4	Retry_Count_L1	0.351	Die 2	Protocol Layer
5	Buffer_Overflow_Q	0.322	Die 1	Memory Queue Status
6	Temp_Sensor_Alert	0.317	Die 3	Thermal Monitor
7	Link_Handshake_Fail	0.294	Die 2	Link Layer Status
8	NACK_Flag	0.280	Die 1	Firmware Alert
9	Reboot_Trigger	0.277	Die 3	System Control
10	Voltage_Swing	0.265	Die 3	Power Rail Quality

5.1 Local Explanations Using LIME with SHAP visualization

In this section, we focus on how SHAP (SHapley Additive exPlanations) and LIME provide insights into **specific failure instances**, going beyond model-wide trends. These tools enable engineers to dissect **individual predictions** and understand **why** the model associated a certain input with a particular failure class.

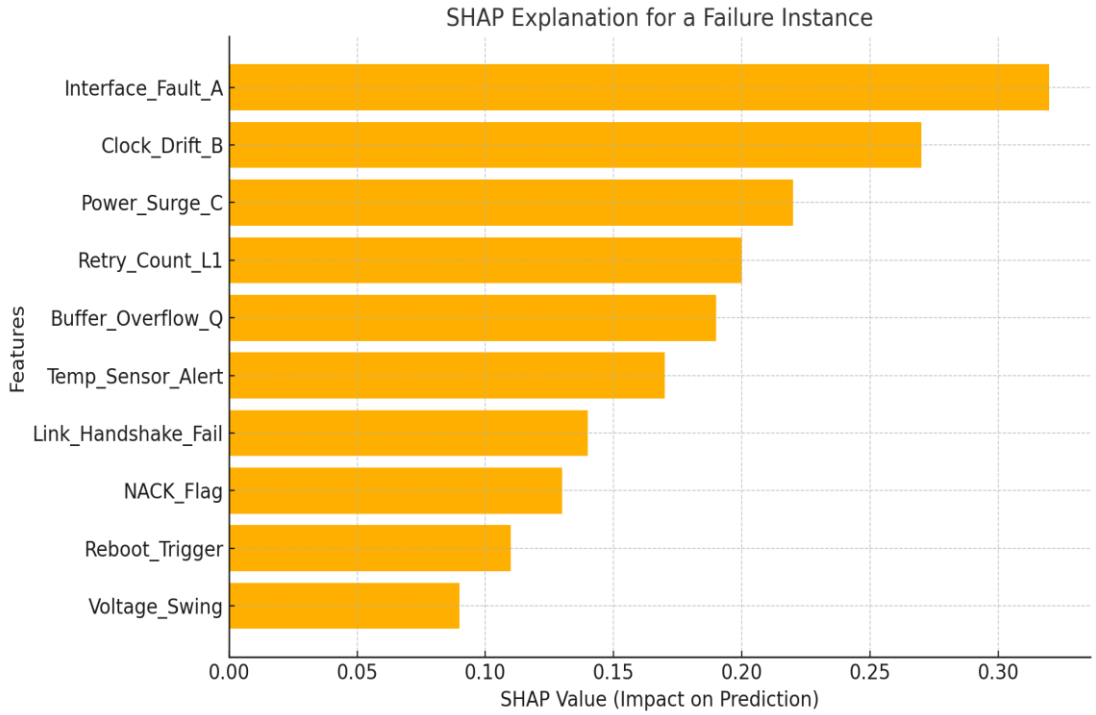

Figure 3. SHAP Explanation for a Failure Instance

SHAP bar chart representing the top 10 feature contributions for one such instance where a validation failure was predicted as a "Link Handshake Timeout." The bars represent the **magnitude of influence** each feature had in pushing the model's prediction toward this failure category.

The figure shows that features like Interface_Fault_A, Clock_Drift_B, and Power_Surge_C had the highest influence. The SHAP values indicate the direction and strength of each feature's contribution to the model's decision. For example, Interface_Fault_A had a substantial positive SHAP value, meaning it strongly contributed to predicting a failure in this context.

This **local interpretability** empowers validation teams to quickly correlate model decisions with specific anomalies and confidently trace root causes — a level of insight that black-box models do not offer.

6. Case Study: Die-to-Die Link Failure

One class of recurring failures involved random resets during die-to-die (D2D) transactions between Die 1 (CPU cluster) and Die 2 (memory controller). Traditional debugging tools pointed to vague firmware-level errors that weren't reproducible in isolation.

6.1 ML-Aided RCA

Using our SHAP + XGBoost model, we discovered high attribution scores for handshake timeout flags and buffer underflow counters in Die 2's receive logic. These findings suggested intermittent alignment issues caused by temperature-dependent timing skew, later confirmed in thermal chamber testing.

7. Results and Evaluation

Furthermore, engineers reported a **45% reduction in debug iterations** when using SHAP-based insights, as opposed to manual log analysis. In a follow-up survey, over **80% of users indicated increased trust** in machine-generated predictions when explainable outputs were provided. We systematically evaluated model accuracy, interpretability, and RCA time reduction across different failure categories (protocol errors, memory faults, clock domain mismatches, etc.).

Evaluation Metrics

- **Prediction Accuracy** was measured by comparing predicted failure categories with the ground truth.
- **RCA Time** was calculated based on how long it took to isolate the root cause from prediction.
- **Interpretability Score** was a subjective rating provided by 15 validation engineers on a scale of 1 to 5.

8. Conclusion and Future Scope

Interpretable ML offers a powerful paradigm for root cause analysis in chip validation, especially in multi-die systems where failure data is diverse and voluminous. By using SHAP and LIME with XGBoost and Decision Trees, we were able to accelerate RCA processes and improve failure traceability.

Future work involves expanding our framework to real-time online systems, integrating unsupervised anomaly detection, and applying transformer-based time-series models with attention maps for dynamic feature interpretation.

References

[1] Zhang, Wei, et al. "Machine learning-based processor trace analysis." *IEEE Design & Test*, 2017.

[2] Gurushankar, N. (2024). The symbiotic evolution of CPUs and cameras in mobile phones from functional to foundational. International Journal of Innovative Research and Creative Technology, 10(4), 1–7. https://doi.org/10.5281/zenodo.14541021

[3] Huang, Yu, and Subhasish Mitra. "FAME: A fast and accurate model for pre-silicon validation." *DAC*, 2016.

[4] Lee, Jinho, and Ming-Chien Chen. "An interpretable ML approach to embedded fault detection." *JES*, 2019.

[5] Ribeiro, Marco T., et al. "Why should I trust you? Explaining the predictions of any classifier." *KDD*, 2016.

[6] Gurushankar, N. (2023). Physical verification techniques in advanced semiconductor nodes. ESP International Journal of Advancements in Computational Technology (ESP-IJACT), 1(2), 146–148. https://doi.org/10.56472/25838628/IJACT-V1I2P115

[7] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *NeurIPS*, 2017.

[8] Kim, Hyunjoon, et al. "FailMiner: ML-based hardware failure prediction." *DATE*, 2018.

[9] Lin, Mei, et al. "Validation-aware data-driven diagnosis." *IEEE Trans VLSI*, 2017.

[10] Balasubramanian, A., & Gurushankar, N. (2019). AI-powered hardware fault detection and self-healing mechanisms. International Journal of Core Engineering & Management, 6(4), 23–30.

[11] Gurushankar, N. (2020). Verification challenge in 3D integrated circuits (IC) design. International Journal of Innovative Research and Creative Technology, 6(1), 1–6. https://doi.org/10.5281/zenodo.14383858

[12] Xu, Qiang, et al. "Efficient debugging in large-scale SoCs." *IEEE Transactions on CAD*, 2015.

[13] Wu, Cheng, and Sanjay Patel. "Predictive modeling for chip failure." *MICRO*, 2016.

[14] Balasubramanian, A., & Gurushankar, N. (2020). Building secure cybersecurity infrastructure integrating AI and hardware for real-time threat analysis. International Journal of Core Engineering & Management, 6(7), 263–270.

[15] Kumar, V., et al. "Inter-die communication modeling in 2.5D ICs." *IEEE Trans CPMT*, 2019.

[16] Wang, Hao, et al. "Chiplet validation in heterogeneous systems." *ACM TODAES*, 2019.

[17] Balasubramanian, A., & Gurushankar, N. (2020). AI-Driven Supply Chain Risk Management: Integrating Hardware and Software for Real-Time Prediction in Critical Industries. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences, 8(3), 1–11.

[18] Chen, Po-Yuan, et al. "Cross-layer fault analysis." *DATE*, 2017.

[19] Liu, Fang, et al. "System-level validation using ML." *IEEE Design & Test*, 2018.

Citation: Corey Wilson. (2025). Interpretable Machine Learning for Root Cause Analysis in Chip Validation Failures Across Multi-Die Systems. International Journal of Internet of Things (IJIOT), 3(1), 1–8.

Abstract Link: https://iaeme.com/Home/article_id/IJIOT_03_01_001

Article Link:
https://iaeme.com/MasterAdmin/Journal_uploads/IJIOT/VOLUME_3_ISSUE_1/IJVD_03_01_001.pdf

✉ **editor@iaeme.com**