10.48047/jocaaa.2026.35.01.41

A Learning-Based Reliability Platform for Automated GPU Fault Classification

Kalyan Inturi
Independent Researcher, USA

Abstract

Large-scale artificial intelligence (AI) training and inference systems rely on highly available accelerator infrastructure, where hardware faults directly reduce effective compute capacity and prolong recovery times. As GPU clusters grow in scale and architectural diversity, fault diagnosis becomes increasingly challenging due to partial observability, overlapping failure signatures, and the limitations of deterministic, rule-based repair policies.

We propose a machine learning–driven framework for automated GPU fault classification and repair recommendation using heterogeneous in-band telemetry (e.g., driver logs, kernel events, and performance counters) and out-of-band diagnostics (e.g., firmware health indicators and environmental sensors). Our approach performs structured feature engineering and extraction to transform noisy and ambiguous signals into diagnostically meaningful representations, and trains a supervised multiclass classifier to infer latent fault states and map them to actionable repair decisions. We evaluate the framework on a large synthetic dataset designed to reflect realistic fleet behavior, incorporating probabilistic ground-truth labels and class overlap to model operational uncertainty.

Experimental results demonstrate that the proposed approach consistently outperforms a representative rule-based baseline across accuracy, balanced accuracy, and macro F1 metrics, with particularly strong gains on rare but high-impact failure classes. These results suggest that learning-based fault inference provides a principled and effective alternative to static rules for reliability management in large-scale AI infrastructure.

1. Introduction

Modern AI workloads place unprecedented demands on accelerator infrastructure, where reliability directly determines effective compute availability and system throughput [1]. Contemporary GPU clusters consist of multi-GPU compute nodes—typically integrating 4–8 accelerators per server—alongside host CPUs, system memory, and local storage. These components are interconnected through PCIe and highbandwidth GPU fabrics and deployed at rack scale with shared power, cooling, and networking subsystems.

Failures in such environments span multiple domains, including thermal stress, power instability, memory degradation, interconnect faults, firmware errors, and environmental anomalies [2]. These faults frequently exhibit overlapping symptoms and evolve over time, making root-cause analysis difficult under partial observability. Production monitoring systems therefore rely heavily on static thresholds and hand-crafted rules, which are precise for well-understood conditions but brittle in the presence of ambiguity and noise.

As cluster scale increases, reliance on manual triage and conservative repair actions leads to prolonged Mean Time To Repair (MTTR) and underutilized compute capacity. This motivates learning-based approaches that can infer latent fault states from heterogeneous telemetry and recommend repair actions probabilistically rather than deterministically [3].

In this work, we investigate whether supervised machine learning can outperform traditional rule-based repair classification under realistic operational uncertainty. We focus on automated inference of repair

10.48047/jocaaa.2026.35.01.41

actions rather than explicit fault labeling, reflecting how decisions are made in practice. Using a synthetically generated but structured dataset, we compare a learning-based classifier against a representative rule-based baseline under identical observability constraints.

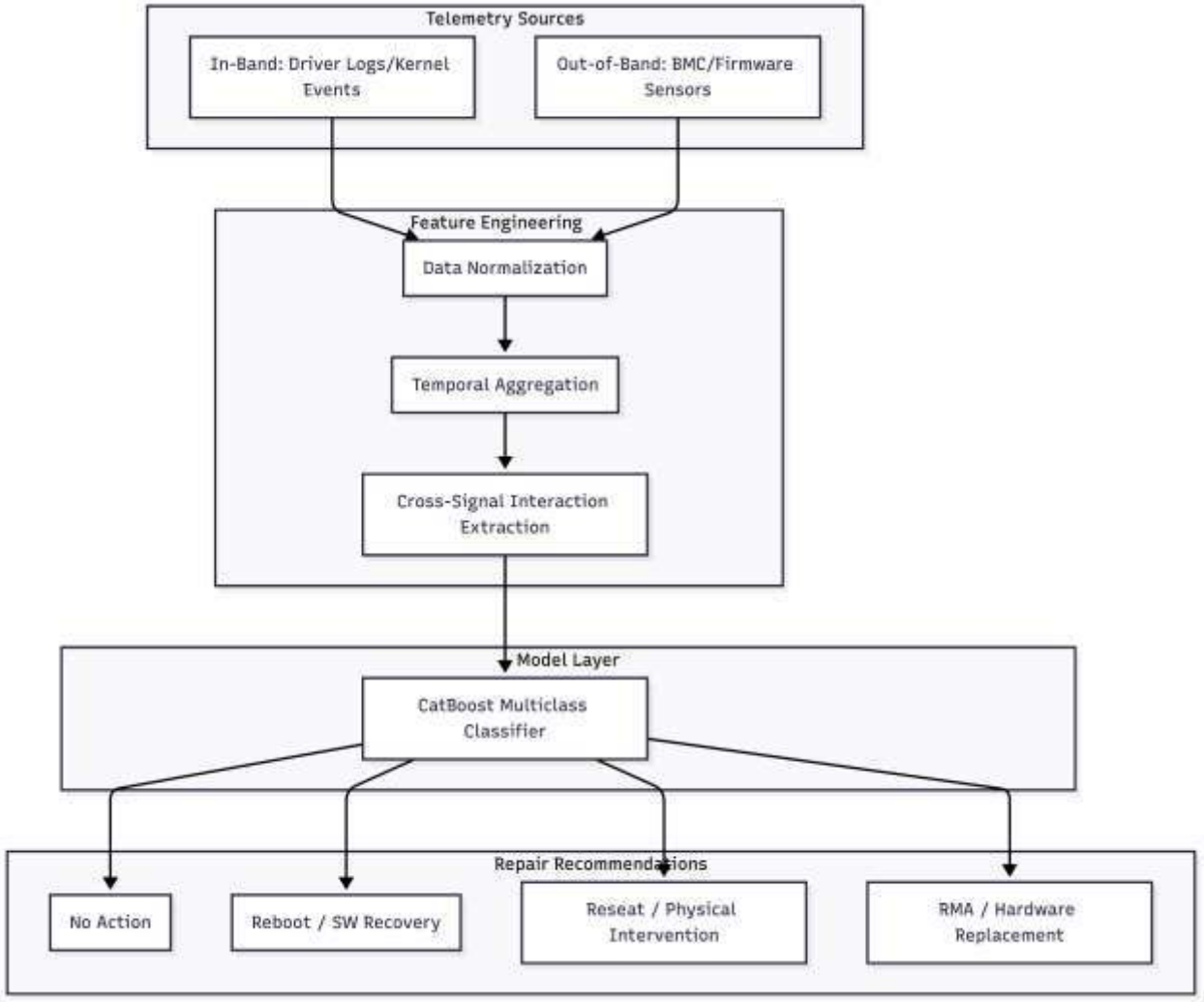

Figure 1 outlines the system architecture for the learning-based reliability platform, showing the flow from raw telemetry ingestion and feature engineering with CatBoost Multiclass classifier.

2. Methodology

Our objective is to learn a mapping from heterogeneous telemetry observations to actionable repair decisions under uncertainty. The system ingests telemetry from two complementary sources:

- **In-band observability**, collected through the host operating system and GPU drivers (e.g., runtime counters, kernel logs, error events).
- **Out-of-band diagnostics**, obtained from firmware, baseboard management controllers (BMCs), and platform sensors, independent of host software state.

These signals are normalized and transformed through a feature engineering and extraction pipeline before being consumed by a supervised multiclass classifier that outputs one of several repair actions.

10.48047/jocaaa.2026.35.01.41

3. Feature Engineering and Extraction

3.1 Feature Categories

We design the feature space to capture both instantaneous device state and longer-term degradation trends:

- **Thermal and Cooling Signals:** GPU temperature, fan speed, cooling configuration, and throttling indicators.
- **In-Band Runtime Telemetry:** Utilization metrics, power draw, driver-reported error events, and kernel log signals.
- **Interconnect and Signal Integrity:** PCIe link width and generation, retimer error counters, and bus-related fault events.
- **Memory and Reliability Indicators:** Correctable and uncorrectable ECC error counts, error accumulation rates, and memory fault flags.
- **Firmware and Platform Health:** Firmware error flags, reset histories, and motherboard-level health indicators.
- **Historical and Recovery Context:** Node age, prior fault frequency, and outcomes of previous recovery attempts.

Together, these features provide a comprehensive view of GPU health across software, hardware, and infrastructure layers.

3.2 Feature Extraction

Raw telemetry signals are often ambiguous when interpreted in isolation. For example, elevated temperature may result from legitimate workload behavior, cooling degradation, or power instability— each requiring different remediation. To resolve this ambiguity, we apply structured feature extraction.

We extract:

- **Temporal features:** Rolling averages, peak values, rates of change, and error burst frequencies over fixed observation windows.
- **Cross-signal interactions:** Relationships such as temperature relative to fan response, voltage stability under sustained load, and interconnect errors in the absence of memory faults.
- **Event abstractions:** Kernel logs and device error codes mapped to semantic failure classes.
- **Persistence indicators:** Repeated failures, unsuccessful recoveries, and escalation history.
- **Topology-aware features:** Comparisons across peer GPUs within the same node or rack.

These extracted features enable the model to distinguish transient anomalies from progressive or irreversible failures.

4. Experimental Evaluation

The goal of our evaluation is to determine whether a supervised learning model can more accurately classify repair actions than a traditional rule-based approach under partial observability and noisy telemetry.

4.1 Dataset Construction and Ground Truth

Obtaining large-scale, consistently labeled production fault data is challenging. We therefore construct a synthetic but structured dataset designed to reflect realistic GPU fleet behavior.

Each sample corresponds to a single GPU device instance and includes heterogeneous telemetry derived from the feature categories described above. Telemetry is generated from a **latent fault domain**

10.48047/jocaaa.2026.35.01.41

representing underlying causes such as software faults, signal integrity degradation, thermal stress, or silicon aging. These latent domains are not observable by either the rule-based system or the learning model.

4.2 Ground-Truth Repair Actions

In real operations, identical telemetry may lead to different repair decisions depending on severity, persistence, and historical context. To model this uncertainty, repair labels are sampled probabilistically from domain-specific distributions rather than assigned deterministically. Additional label noise is injected to simulate missing logs, delayed signals, and operator variance.

The classification task includes four repair actions:

1. No Action
2. Reboot / Software Recovery
3. Reseat / Physical Intervention
4. RMA / Hardware Replacement

4.3 Baseline: Rule-Based Classification

The baseline implements a deterministic repair policy representative of production monitoring systems. Example rules include:

- Uncorrectable ECC errors or sustained high error rates → RMA
- PCIe width degradation or excessive retimer errors → Reseat
- Thermal throttling or driver faults → Reboot
- Otherwise → No Action

These rules are **not** used to generate ground truth and are applied only as an inference baseline on heldout test data.

4.4 Machine Learning Model

We train a supervised multiclass classifier using CatBoost [4], selected for its effectiveness on heterogeneous tabular data and native support for categorical features. The dataset is split into training (70%), validation (15%), and test (15%) partitions using stratified sampling. Class imbalance is addressed through class weighting, with emphasis on rare but high-impact repair actions.

5. Results and Discussion

Source code: https://github.com/kalyanchakri02/ml-latest/tree/main

5.1 Quantitative Results

Table 1: Repair Action Classification Performance

Method	Accuracy	Balanced Accuracy	Macro F1	Weighted F1
Rule-Based (Baseline)	0.723	0.534	0.536	0.728
ML (Proposed)	0.783	0.670	0.644	0.786

10.48047/jocaaa.2026.35.01.41

The learning-based approach outperforms the rule-based baseline across all metrics, with especially strong gains in Balanced Accuracy and Macro F1, indicating improved performance on minority and high-impact classes.

5.2 Discussion

Rule-based systems are effective for well-defined conditions but struggle under overlapping symptoms and partial observability. The proposed model learns probabilistic decision boundaries that approximate latent fault states rather than reacting to isolated signals. Importantly, these gains persist despite intentional label ambiguity and noise, suggesting genuine inference capability rather than overfitting to synthetic patterns.

6. Limitations and Future Work

This study relies on synthetic telemetry and probabilistic labels, which may not fully capture long-tail behaviors observed in production fleets. Future work will incorporate real-world failure data and explore domain adaptation techniques. Extensions to sequential decision-making and uncertainty-aware models may further improve repair optimization under fleet-level constraints.

Conclusion

We present a learning-based framework for GPU fault classification and repair recommendation under operational uncertainty. By integrating heterogeneous telemetry and applying structured feature extraction, the system infers latent fault states and consistently outperforms deterministic rule-based baselines.

Our results demonstrate that learning-based reliability management can significantly improve repair decision quality, particularly for rare but operationally critical failures. This work supports a broader shift from static thresholding toward probabilistic inference in large-scale AI infrastructure.

References

[1] J. Dean, D. Patterson, and C. Young, “A New Golden Age in Computer Architecture,” *Communications of the ACM*, 2018. https://doi.org/10.1145/3282307

[2] L. Barroso, U. Hölzle, and P. Ranganathan, *The Datacenter as a Computer*, 3rd ed., 2018. https://doi.org/10.2200/S00874ED3V01Y201809CAC046

[3] S. Kandula et al., “Diagnosing Performance Changes by Comparing Request Flows,” *NSDI*, 2012 https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final116.pdf

[4] L. Prokhorenkova et al., “CatBoost: Unbiased Boosting with Categorical Features,” *NeurIPS*, 2018 https://arxiv.org/abs/1706.09516