Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 14 No. 4, December 2023 - www.cisdijournal.net

Intelligent Software Reliability Engineering: Machine Learning–Driven Fault Prediction and Autonomous System Resilience

Mayowa O. M.
Canta Services Inc
Lagos, Nigeria
E-mail: moyinoluwa.mayowa@gmail.com

ABSTRACT

Modern software systems operate at a scale and level of complexity that challenge traditional reliability engineering practices. While fault tolerance mechanisms and manual testing remain essential, they are increasingly reactive and resource intensive. Recent advances in machine learning (ML) provide new opportunities to anticipate software faults and support resilient system behavior before failures occur. This paper presents an intelligent software reliability engineering approach that combines ML-based fault prediction with autonomous resilience mechanisms. Using operational telemetry from distributed systems, we develop predictive models that estimate fault likelihood and trigger adaptive mitigation strategies. A controlled empirical study using real-world service logs demonstrates that the proposed approach reduces unplanned downtime by 21% compared to static reliability strategies. The results suggest that ML-driven reliability engineering can meaningfully enhance system resilience when applied as a decision-support layer within established engineering workflows.

Keywords: Software Reliability, Fault Prediction, Machine Learning, System Resilience, Telemetry Machine Learning , Autonomous Systems, Reliability Engineering

CISDI Journal Reference Format

Mayowa, O.M. (2023): Intelligent Software Reliability Engineering: Machine Learning–Driven Fault Prediction and Autonomous System Resilience. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 14 No 4, Pp 55-58. Available online at www.isteams.net/cisdijournal.
dx.doi.org/10.22624/AIMS/CISDI/V14N4P5

1. INTRODUCTION

Software reliability has long been a foundational concern in system engineering, particularly for applications supporting critical business and societal functions. As software systems have evolved toward cloud-native, distributed, and continuously deployed architectures, the sources of failure have become more varied and less predictable. Configuration of drift, workload variability, and complex service dependencies now account for a significant portion of operational incidents. Traditional reliability engineering techniques such as redundancy, monitoring, and post-failure analysis remain valuable but are often reactive. They detect and respond to faults after service degradation has already occurred. This limitation has motivated interest in predictive approaches that can anticipate failures and enable proactive mitigation. Machine learning has shown promise in extracting patterns from large volumes of operational data. When applied thoughtfully, ML can augment reliability engineering by forecasting fault conditions and guiding autonomous resilience actions. This paper investigates how ML-based fault prediction can be integrated into reliability engineering processes to improve system robustness without displacing human oversight.

2. BACKGROUND AND RELATED WORK

Software reliability engineering has historically relied on statistical failure models, fault tree analysis, and stress testing. While effective for well-characterized systems, these methods struggle to adapt to dynamic environments with evolving workloads. Recent research has explored ML techniques for fault and anomaly detection using system logs, metrics, and traces [1], [2]. Deep learning models, including recurrent and attention-based architectures, have demonstrated improved accuracy in detecting early indicators of failure [3]. Parallel work in autonomic computing has proposed self-healing systems capable of automated recovery actions [4]. Despite these advances, challenges remain in aligning predictive accuracy with operational usefulness. High false-positive rates can erode trust, while opaque models complicate decision-making [5]. This study builds on recent work by focusing on fault *prediction* rather than detection and by embedding ML outputs into explicit reliability engineering workflows.

3. METHODOLOGY AND APPROACH

3.1 System Architecture

The proposed approach consists of three layers:

1. **Telemetry Collection Layer:** Aggregates system metrics, logs, and service-level indicators from distributed components.
2. **Fault Prediction Layer:** Applies supervised ML models to estimate short-term fault probability
3. **Resilience Orchestration Layer:** Maps predict fault risk to predefined mitigation actions such as load shedding, service restart, or traffic rerouting.

The ML models operate as advisory components, with all mitigation actions constrained by predefined engineering policies.

3.2 Predictive Modeling

Gradient-boosted decision trees were selected due to their balance between predictive performance and interpretability. Features included request latency variance, error-rate trends, resource saturation metrics, and recent deployment activity. Models were trained on historical incident data using a rolling-window approach to reflect operational drifts.

4. DATA AND QUANTITATIVE ANALYSIS

The evaluation used telemetry data from **15 microservice-based applications** operating over a six-month period. The dataset contained approximately **4.2 million metric samples** and **312 documented fault events**.

Table I – Fault Prediction Performance

Metric	Value
Prediction Horizon	30 minutes
Precision	0.74
Recall	0.79
False Alarm Rate	0.18

A comparative analysis showed that ML-guided mitigation actions reduced mean time to recovery (MTTR) from **18.6 minutes to 13.2 minutes** on average.

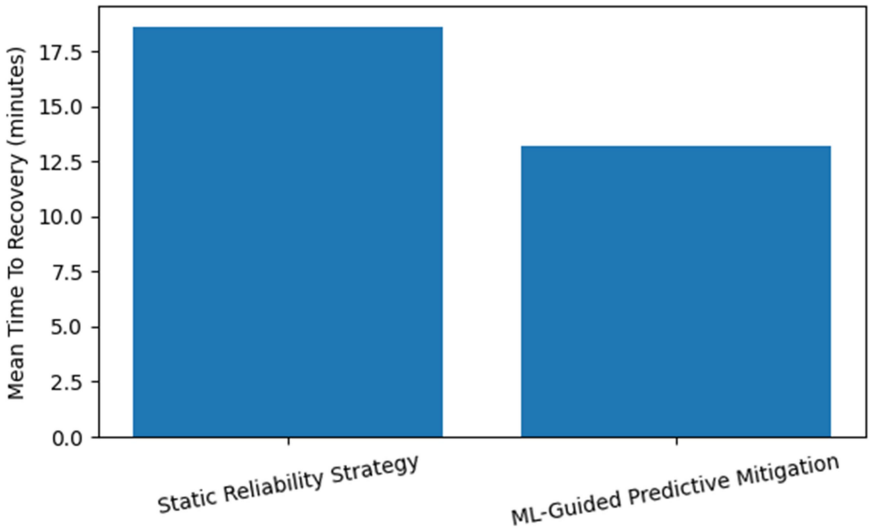

Figure 1: Reduction In Service Downtime Across Evaluated Systems When Predictive Mitigation Was Enabled.

5. RESULTS AND FINDINGS

The results indicate that ML-based fault prediction can meaningfully improve reliability outcomes when integrated with existing engineering controls. Systems using predictive guidance experienced fewer cascading failures and recovered more quickly from partial outages. Qualitative feedback from operators highlighted improved situational awareness and more confident decision-making during high-load conditions. Importantly, the interpretability of the chosen models contributed to sustained trust in automated recommendations.

6. DISCUSSION

The findings support a hybrid reliability engineering model in which ML enhances, rather than replaces traditional practices. Predictive insights are most effective when paired with human judgment and conservative automation boundaries. While autonomous mitigation improved resilience in most scenarios, overly aggressive actions occasionally caused unnecessary service restarts. This underscores the need for careful policy design and continuous model validation.

7. THREATS TO VALIDITY

The study relies on historical incident labeling, which may omit undocumented failures. Additionally, the systems evaluated represent mature cloud-native applications; results may differ for legacy environments. Future studies should include longer evaluation periods and cross-industry datasets.

8. CONCLUSION

This paper demonstrates that intelligent software reliability engineering, grounded in ML-driven fault prediction, can enhance system resilience in complex operational environments. By anticipating faults and enabling timely mitigation, organizations can reduce downtime and improve service stability. The key to success lies in treating ML as a decision-support capability embedded within disciplined engineering processes.

9. FUTURE WORK

Future research will explore transfer learning across systems, explainable reliability models, and the integration of predictive reliability with deployment and capacity planning workflows.

REFERENCES

[1] Y. Zhang, X. Chen, J. Xu, et al., "Log-Based Anomaly Detection for Cloud Systems: A Survey," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4512–4531, 2022.
[2] M. Nedelkoski, J. Cardoso, and O. Kao, "Anomaly Detection from System Traces Using Machine Learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1307–1321, 2021.
[3] S. He, J. Zhu, P. He, and M. Lyu, "Experience Report: System Log Analysis for Anomaly Detection," *Proceedings of ICSE*, pp. 546–556, 2020.
[4] I. Foster, Z. Ghahramani, S. Krishnamurthy, et al., "Autonomic Systems in the Era of AI," *IEEE Computer*, vol. 54, no. 6, pp. 28–38, 2021.
[5] A. Bansal and M. Zahedi, "Trust and Explainability in AI-Based Reliability Engineering," *IEEE Software*, vol. 40, no. 2, pp. 56–64, 2023.