PAKDD 2014 - ASUS Malfunctional Components Prediction

Implementation, Simulation, and Architectural Proposal

Muñoz Mariño David Eduardo[1], Vargas Arias Daniel[1], Moreno Durán Juan Esteban[1], Romero Buitrago Julián Darío[1]
[1] Programa de Ingeniería de Sistemas, Universidad Distrital Francisco José de Caldas, Bogotá, Colombia

Introduction

Operational failures in ASUS components generate high replacement and logistics costs. This poster summarizes the analytical and experimental processes carried out in the development of the predictive system, focusing on data engineering, predictive modeling, and chaos-driven simulations for evaluating system robustness under adverse conditions.

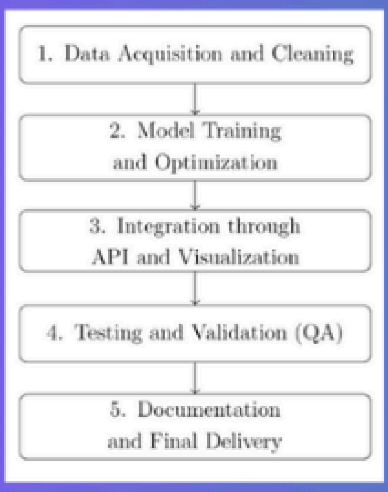
Objective

To build a scalable predictive system capable of anticipating malfunctioning electronic components and evaluating its resilience when exposed to noise, missing data, and unexpected operational events.

Methodology

- Data Preparation (ETL):
  - Column normalization and renaming.
  - Categorical imputation.
  - Extraction of year/month features.
  - Min–Max normalization.
- Simulations:
  - Data-Driven Simulation: Logistic Regression under progressive noise perturbations; retraining activated as a feedback control when accuracy drops below a defined threshold.
  - Event-Driven Simulation: Cellular automaton modeling of system states: NORMAL, HIGH_LOAD, BAD_DATA, PARTIAL_FAILURE, RECOVERY.
- Evaluation Metrics: Accuracy, F1-score, ROC-AUC, response times, recovery capability, and data consistency.

Workflow



Experimental Execution

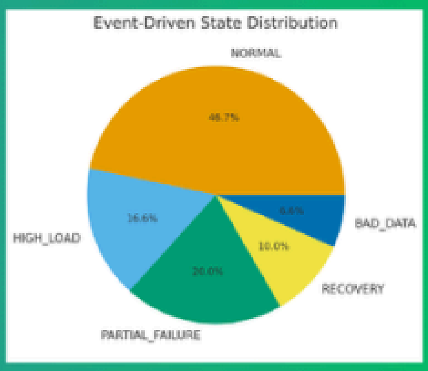
- Processed rows: ~10,500 (cleaned dataset)
- Tested noise levels: 0.01, 0.03, 0.05, 0.07, 0.09
- Retraining threshold: 75% of baseline (baseline = 84.5% → threshold = 63.3%)
- Event simulation: 30 total cycles; chaosRate increases by +0.005 per cycle

Main Results

A. Machine Learning (Data-Driven)

- Baseline Accuracy: 84.5%.
  - Accuracy degradation under noise:
  - Noise = 0.01 → 84.1%
  - Noise = 0.03 → 81.0%
  - Noise = 0.05 → 72.5%
  - Noise = 0.07 → 61.0% (retraining triggered)
  - Noise = 0.09 → 65.5% (partial recovery)
- Analytical Insight: Around $\varepsilon \approx 0.05$, the accuracy drop becomes nonlinear, indicating a structural fragility point. Full retraining improves performance but does not restore baseline accuracy.

B. Event-Driven Simulation



- Total cycles: 30
- Observed state distribution:
  - NORMAL: 46.6%
  - HIGH_LOAD: 16.6%
  - PARTIAL_FAILURE: 20.0%
  - RECOVERY: 10.0%
  - BAD_DATA: 6.6%
- Analytical Insight: The system shows early-stage stability (cycles 0–12), mid-stage bifurcation, and late-stage oscillation between FAILURE and RECOVERY, suggesting hysteresis and path-dependence.

Comparative Analysis

- Chaos Manifestation: ML degradation is continuous; event simulation transitions are abrupt.
- Feedback Mechanisms: ML uses costly retraining; event simulation resets states quickly.
- Emergent Behaviors: ML undergoes diminishing returns after retraining; event simulation exhibits oscillatory behavior and hysteresis.

Risks and Bottlenecks

- Expensive Retraining: Risk of retraining loops if noise fluctuates near the threshold.
- False Stability Perception: NORMAL states may arise by randomness within the automaton.
- Irreversibility: Increasing chaosRate reduces the probability of returning to NORMAL states.

Design Recommendations

1. Circuit Breaker Layer (ingestion/services): Prevent continuous oscillation between failures and recoveries.
2. Online/Incremental Learning: Use adaptive models (Hoeffding Trees, Online Logistic Regression) to reduce retraining cost.
3. Pre-Inference Anomaly Filtering: Reject inputs with noise > 0.05 using Z-score or statistical rules.
4. Integrated Chaos Engineering: Introduce controlled perturbations as a permanent stress-testing practice.
5. Model Registry + Canary Deployments: Safely manage model versioning and rollbacks.

Proposed Visualizations

- Curve: Performance vs Noise (decay curve with retraining threshold marker).
- Distribution Chart: State proportions from Event Simulation.
- Architecture Diagram: Layered system: ingestion → storage → training → inference → monitoring.

Strengths / Weaknesses

- Strengths: Industrial impact, scalable architecture, continuous improvement cycle.
- Weaknesses: High sensitivity to noise, costly retraining cycles, predictive uncertainty under chaotic conditions.

Conclusions

The proposed predictive system demonstrates strong performance under controlled environments. However, under chaotic perturbations it exhibits structural limitations. Integration of anomaly filtering, online learning, and circuit-breaker patterns significantly improves operational robustness.

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS
Acreditación Institucional en Alta Calidad

Bibliography

- PAKDD Cup 2014 — ASUS Malfunctional Components Prediction
- Workshop 1–4 Reports (ETL, architecture, simulations, and experimental results)