TraceLens: Early Detection of Software Anomalies Using Critical Path Analysis

Masoumeh Nourollahi, Amir Haghshenas, Michel Dagenais
École Polytechnique de Montréal
ICPE 2025, Toronto, Canada

Motivation & Problem Statement

- Modern IoT, Edge, and Cloud systems evolve continuously and must run reliably
- Performance anomalies can cause serious degradation or failures
- Current trace-based methods are costly and lack real-time responsiveness
- We need a scalable, low-overhead approach for early anomaly detection.

Related Work

Existing performance anomaly detection approaches:

- Static thresholds
- System call-based ML techniques

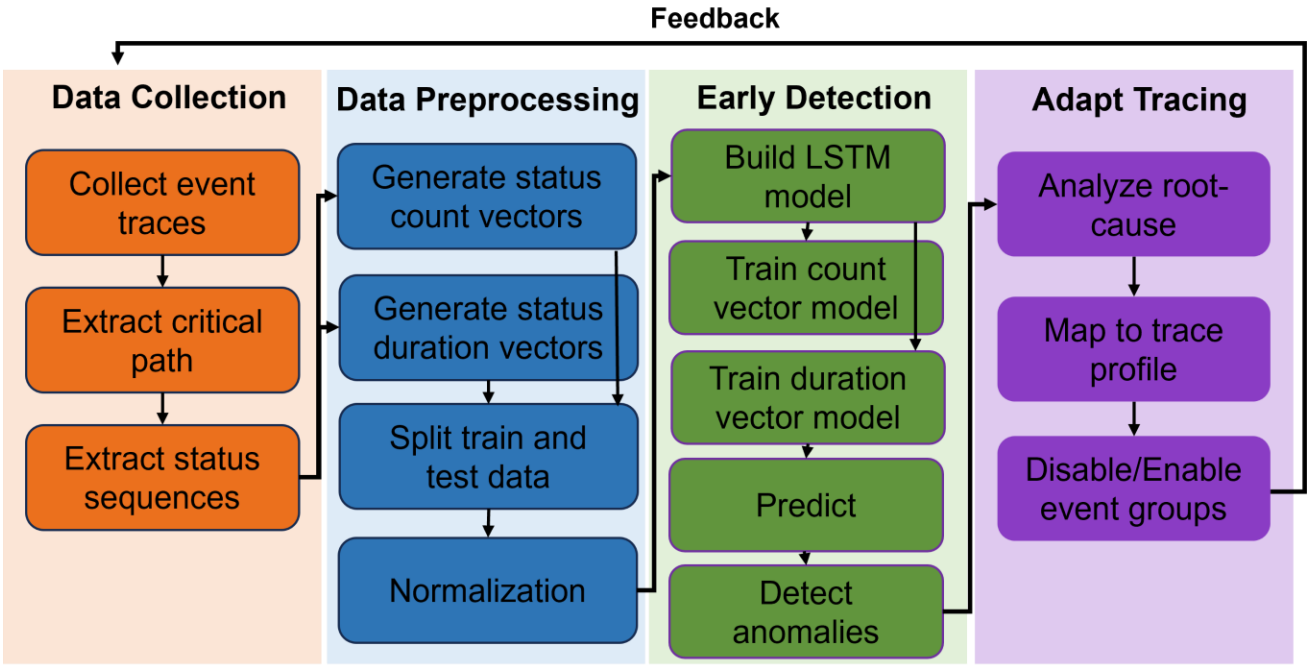
Recent methods, like LSTM models and clustering algorithms improved detection accuracy but are still limited by trace size and overhead.

Key limitations:

- Generation of large volumes of trace data
- Often used offline
- Lack adaptive real-time capability

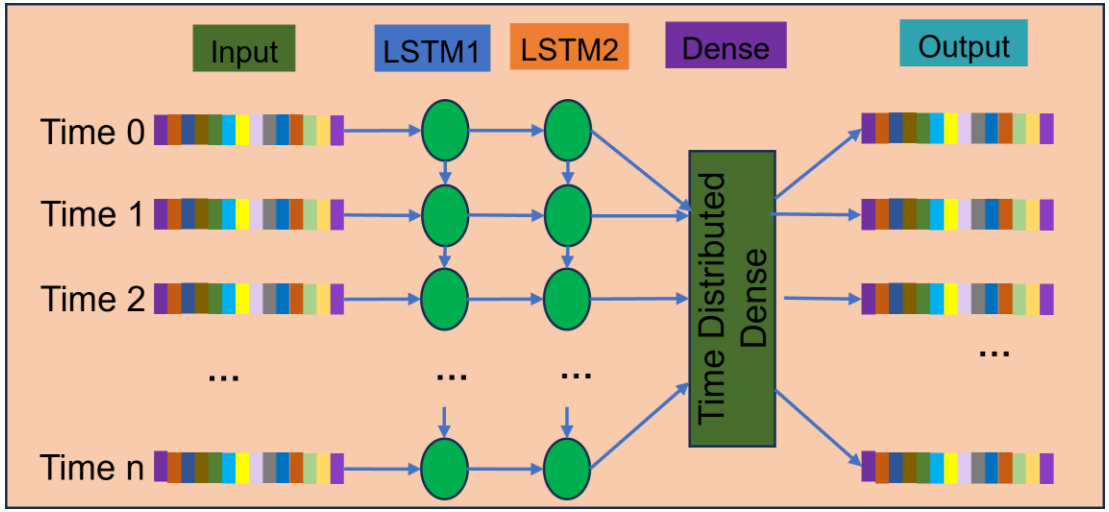
Proposed Approach- Deep learning-based anomaly detection using critical path analysis

- Focus on key execution sequences to minimize trace overhead
- Critical path shows runtime dependencies and bottlenecks
- This analysis enables efficient tracing and deep insight into performance anomalies



LSTM Model Architecture & Training

- Extracted lightweight event data includes:
  - Critical path state count vector
  - Critical path state duration vector
- LSTM model processes state vectors over time windows



Experiments and Results

- Dataset
  - Experimental data created with web workload put on a flask application
  - Real world dataset
    - Noferesti, M. and Ezzati-Jivan, N., 2024. Enhancing empirical software performance engineering research with kernel-level events: A comprehensive system tracing approach. Journal of Systems and Software, 216, p.112117.

DataSet	Number of records	Vector size	Time window
Train	4,847	28-ary	100 sec
Test	1,212	28-ary	100 sec

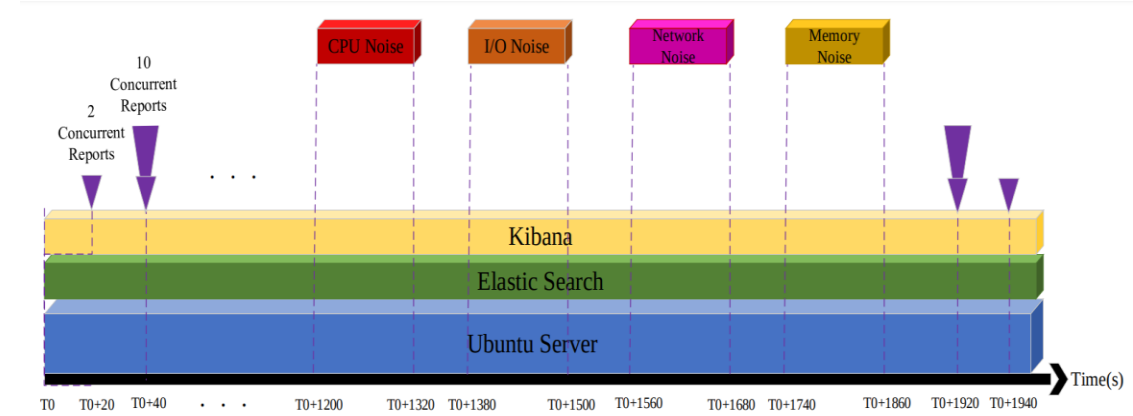
Experiments and Results

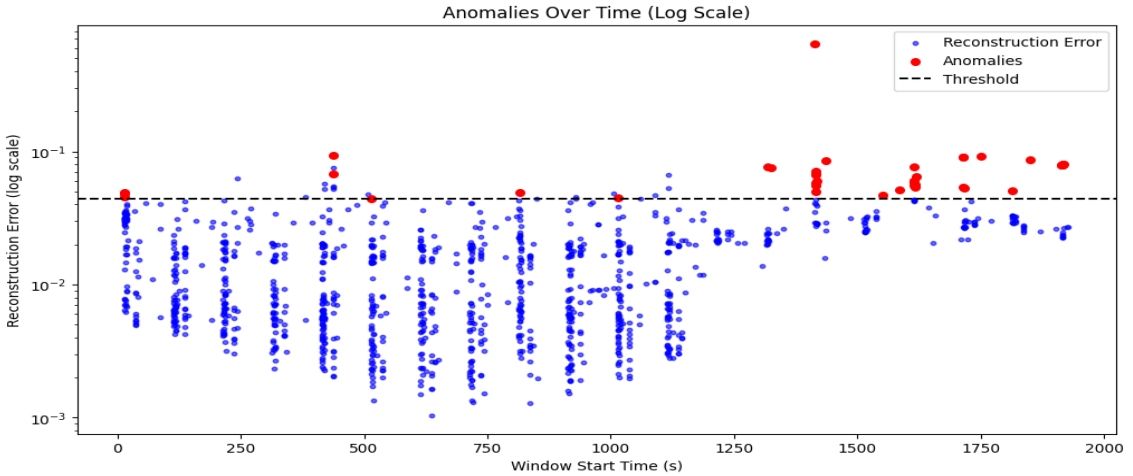
Comparable accuracy to system call methods with minimal overhead.

DataSet	Accuracy	Recall	Precision	Overhead	Data Size	Train Time
Critical path state count and duration	90.24%	92.30%	85.97%	5.59%	280.3 KB	35.92s
System call count and duration	91.46%	89.12%	84.33%	25.92%	576491.01 KB	3952.3s

Experiment Outcomes & Anomaly Detection

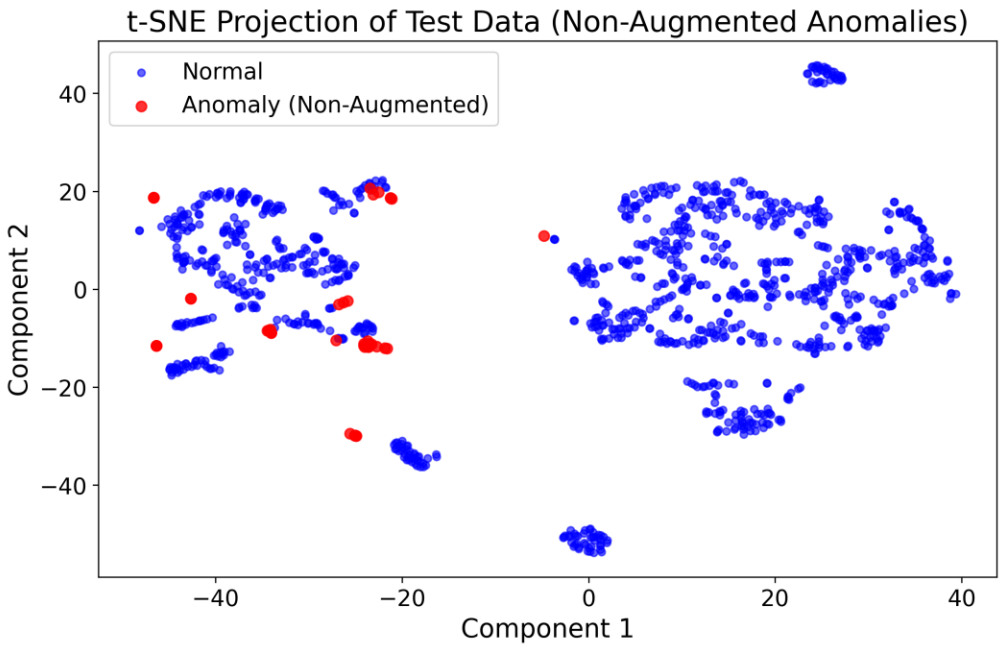
The additional performance degradations detected may be due to data augmentation used to generate more training samples.





Anomaly detection using prediction loss

t-SNE projection shows effective separation of normal and anomalous states.



Discussion & Implications

- Reduced trace size and tracing overhead compared to system-call based methods
- Ideal for real-time large-scale systems:
- Critical path-based approach enhances root cause analysis
- Potential for adaptive tracing and improved tracing efficiency

Conclusions & Future Work

TraceLens provides an efficient and scalable method for early anomaly detection through critical path analysis.

Future Work:

- Exploring other ML models and dataset
- Expand to IoT and resource-constrained environments
- Implement adaptive tracing across layers and nodes