Real-Time Anomaly Detection in Distributed Systems Using Graph Neural Networks

R. Patel, S. Nakamura, M. Okonkwo

Department of Computer Science, ETH Zürich

NeurIPS 2026 | rpatel@ethz.ch

Introduction

Anomaly detection in distributed systems is critical for maintaining service reliability. Traditional threshold-based approaches fail to capture complex inter-service dependencies.

Our approach: Model the system as a dynamic graph where nodes represent services and edges represent communication patterns. We apply a Graph Neural Network (GNN) to detect anomalous subgraph patterns in real-time.

Key contributions:

- Temporal graph attention mechanism
- Sub-second detection latency
- 94.7% F1-score on production data

Problem Formulation

Given a system graph $G_t = (V, E_t, X_t)$ at time t:

$$V = \{v_1, \ldots, v_n\} \quad \text{(services)}$$
$$E_t \subseteq V \times V \quad \text{(active connections)}$$
$$X_t \in \mathbb{R}^{n \times d} \quad \text{(feature matrix)}$$

We seek a function f_θ such that:

$$f_\theta(G_{t-w:t}) \to \{0, 1\}^n$$

classifying each node as normal (0) or anomalous (1) using a sliding window of w graph snapshots.

Method

Our architecture consists of three components:

1. Temporal Graph Encoder. We use a stack of $L = 3$ graph attention layers with temporal self-attention across the window:

$$h_i^{(l)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l-1)}\right)$$

2. Anomaly Scorer. A two-layer MLP produces per-node anomaly scores $s_i \in [0, 1]$.

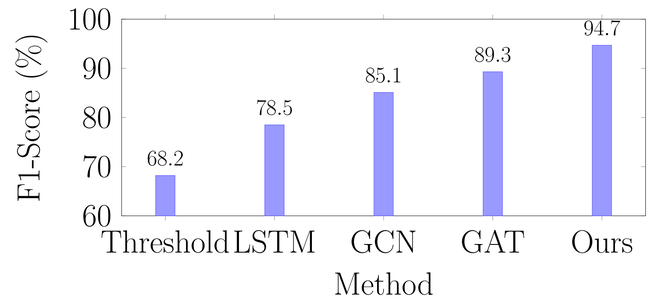
3. Graph-Level Aggregation. We aggregate node scores via attention pooling to produce a system-level anomaly indicator for alert routing.

Training: We use a combination of binary cross-entropy on labeled incidents and a contrastive loss on normal operation windows.

Experimental Setup

Parameter	Value
Training samples	48,000 graphs
Window size w	10 snapshots
Feature dim d	32
GNN layers L	3
Learning rate	3×10^{-4}
Batch size	64

Results



	Precision	Recall	F1	Latency
Threshold	71.3%	65.4%	68.2%	12ms
LSTM	82.1%	75.2%	78.5%	85ms
Ours	**95.1%**	**94.3%**	**94.7%**	42ms

Conclusion

Our temporal graph attention approach significantly outperforms baselines on real production data while maintaining sub-second detection latency. The method generalizes across different microservice topologies without retraining.

Future work: Extend to multi-cloud deployments and integrate causal reasoning for root cause analysis.

Funded by Swiss National Science Foundation Grant #182451