Microservice Anomaly Detection and Root Cause Analysis Using Temporal Graph Networks with Adaptive Memory Protection

2410014 Tomoya Ueno

Microservice architecture has become a standard design pattern in modern large-scale distributed systems. Hundreds to thousands of independent services collaborate, forming chains of Remote Procedure Calls (RPCs) where a single user request propagates across multiple services. While this architecture provides development flexibility and scalability, it significantly increases the complexity of operational monitoring. Local failures can rapidly cascade throughout the entire system. For example, a memory leak or CPU anomaly in one service can cause latency increases in dependent upstream services, ultimately leading to degraded user experience or system downtime. In such environments, it is crucial not only to detect anomalies quickly but also to identify the true root cause.

Traditional threshold-based monitoring and univariate time series analysis cannot account for complex inter-service dependencies and tend to generate many false positives. This leads to alert fatigue among operators and makes rapid identification of the true root cause difficult. To address this challenge, Graph Neural Networks (GNNs) that explicitly handle service dependencies (topology) have gained attention. In particular, Temporal Graph Networks (TGN), equipped with memory mechanisms that store long-term state for each node, possess an ideal model structure for learning patterns of normal service behavior.

In this study, we identified the **memory contamination problem** as a critical challenge when applying TGN to anomaly detection. The original TGN was designed for dynamic graph learning tasks such as link prediction and node classification, unconditionally incorporating all input events into GRU memory. In the context of anomaly detection, this design causes anomalous data to accumulate in memory, corrupting the normal profile. Furthermore, in microservices, partial anomalies frequently occur where only some metrics are anomalous while others remain normal, such as CPU being anomalous while latency is normal.

Existing defense methods against memory contamination have notable limitations. Hard threshold filtering methods such as T-Shield and MemStream use binary decisions (reject/accept) based on anomaly scores, causing abrupt transitions near the threshold, lacking smooth control. Furthermore, these methods process entire data points uniformly, unable to support independent control for each metric group as required in microservice environments.

To address these challenges, we propose **ADA-TGN (Temporal Graph Networks with Adaptive Memory Protection)**. Unlike existing methods, ADA-TGN achieves continuous gating through Sigmoid-based soft transitions rather than hard thresholds, ensuring stability in boundary regions. The mechanism operates at the metric group level,

enabling selective protection during partial anomalies. Additionally, as an inference-time mechanism, it can be retrofitted to trained models and adapted to changing anomaly patterns by recalculating statistics from new normal period data.

The contributions of this study are threefold. First, we identified the memory contamination problem in TGN and clarified the fundamental challenge in memory-based anomaly detection. Second, we introduced a **Dual-Stage Soft-Gating mechanism** to achieve adaptive memory protection that selectively suppresses memory updates during anomalies. Third, we enabled independent control of memory updates at the metric group level (Latency, Memory, CPU, Network), allowing continued memory updates for normal metric groups even during partial anomalies.

The Soft-Gating mechanism of ADA-TGN operates in two stages. In the first stage (GRU Input Soft-Gating), input signals to the GRU are attenuated according to the anomaly degree, suppressing the inflow of anomalous data. In the second stage (Memory Soft-Gating), the memory state after GRU update is interpolated with the previous memory, retaining the previous memory during anomalies. This two-stage protection effectively blocks the influence of anomalous data on memory. For anomaly degree estimation, reconstruction errors for each metric group are used. This design enables suppressing memory updates for anomalous metric groups while continuing memory updates for normal metric groups even during partial anomalies.

For feature extraction, 12-dimensional features across 4 metric groups are extracted from microservice metrics data. These features are standardized using P99 scaling and Asinh transformation on a per-service basis. Inter-service communication relationships are represented as graph structure and utilized for neighborhood information aggregation through Graph Attention.

In evaluation experiments, we used the RE2-OB dataset from the RCAEval benchmark, comprising 60 experiments (5 services $\times$ 4 fault types $\times$ 3 replications). For each experiment, an independent model was trained using only normal period data to construct the normal profile.

Through ablation studies comparing ADA-TGN with Vanilla-TGN (without Soft-Gating), we quantitatively confirmed the memory protection effect. In an experiment where Latency and Memory anomalies were prominent, we measured the deviation of memory states from the normal region during anomaly periods. In Vanilla-TGN, memory deviation reached 3-4 times the normal period baseline across all metric groups, indicating severe memory contamination. In contrast, ADA-TGN maintained memory deviation within normal levels (0.82-0.92 times) for these groups, confirming the memory protection effect of Soft-Gating. This difference directly translates to reconstruction stability: Vanilla-TGN produces unstable reconstructions from contaminated memory, while ADA-TGN maintains stable normal-pattern reconstructions even during anomaly periods.

For anomaly detection performance, the method combining ADA-TGN with dynamic

SPOT (dSPOT) achieved Precision 0.882, Recall 1.000, and F1 score 0.937. All 60 experiments were successfully detected (zero false negatives), with false positives occurring in 8 experiments. These false positives were caused by sudden load fluctuations during normal periods that exceeded dSPOT's adaptive threshold before it could adjust.

For root cause analysis performance, ADA-TGN achieved Avg@5 of 0.880, outperforming BARO's 0.745 by 18.1% when evaluated on the same dataset. Performance varied by fault type: CPU and Memory anomalies achieved exceptionally high accuracy (Avg@5: 0.960 and 0.987, respectively) because these anomalies are directly observable as resource consumption at the root cause service. In contrast, DELAY anomalies showed lower performance (Avg@5: 0.720) due to the cascade effect, where latency increases propagate and amplify along service dependencies, causing upstream services to exhibit higher anomaly scores than the actual root cause.

This study presents a solution to the memory contamination problem that was unavoidable in TGN-based anomaly detection, and demonstrates improved accuracy in anomaly detection and root cause analysis for microservice environments. The concept of memory protection through the Soft-Gating mechanism represents a general framework of adaptive state update control based on reconstruction errors, suggesting potential applicability to other memory-based time series models.