SLADE: Detecting Dynamic Anomalies in Edge Streams without Labels via Self-Supervised Learning

Jongha Lee — KAIST AI — jhsk777@kaist.ac.kr

Sunwoo Kim — KAIST AI — kswoo97@kaist.ac.kr

Kijung Shin — KAIST AI — kijungs@kaist.ac.kr

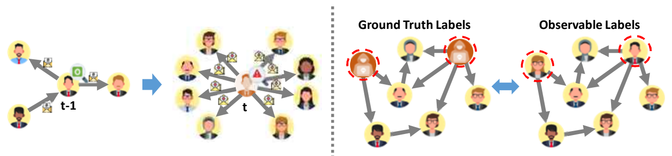
KDD2024 BARCELONA, SPAIN

Summary

- **Motivation**
  - To address 3 challenges (time delay in detection, dynamically changing states, lack of anomaly labels) that arise when detecting anomalies in real-world graphs.
- **Proposed Method: SLADE**
  - We propose SLADE for rapid detection of dynamic anomalies in edge streams, without relying on labels.
  - SLADE trains neural network models to perform two self-supervised tasks (temporal contrast, memory generation) that align with our assumed normal patterns.
- **Contribution**
  - **Unsupervised**: SLADE can detect dynamic anomalies without label supervision.
  - **Effective**: SLADE outperforms (supervised) baselines in dynamic anomaly detection.
  - **Constant Inference Speed**: SLADE requires a constant inference time per edge.

Introduction

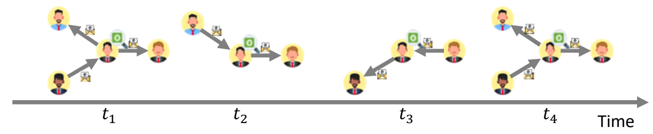
- **Challenges in Detecting Anomalies in Real-world Graphs**
  - **Time Delay in Detection**: Time delay in the detection of anomalies can increase harm to benign users.
  - **Dynamically Changing States**: A user behave normally during one time period but abnormally during another time period.
  - **Lack of Anomaly Labels**: Many neural network-based methods rely on label supervision for detecting complex anomalies, but labeled anomalies are often unavailable.



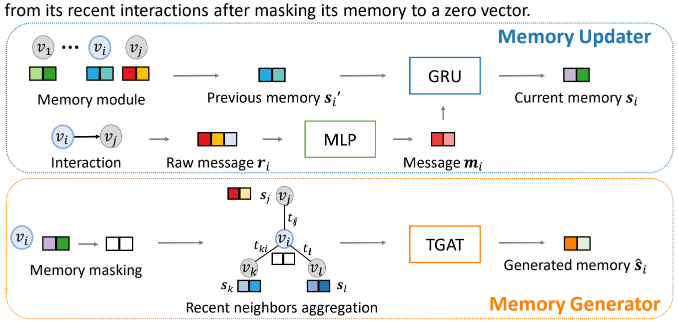
- **Continuous Time Dynamic Graph**
  - **Definition**: A stream of temporal edges with timestamps $G = (\delta_1, \delta_2, \ldots)$, where each temporal edge $\delta_n = (v_i, v_j, t_n)$ arriving at time t_n is directional from the source node v_i to the destination node v_j.
  - **Why CTDGs?**: CTDG-based methods process each new edge, whenever it arrives, with minimal detection time delay, compared to static-graph or DTDG-based ones.
- **Problem Description**
  - **Goal:** We aim to accurately classify the current dynamic status of actor node of each temporal edge in a CTDG, which is either normal or abnormal.
  - **Focuses**: (a) instantly detecting anomalies as they occur, (b) adapting to dynamic changing states, and (c) handling the scarcity of dynamic anomaly labels.

Proposed Method: SLADE

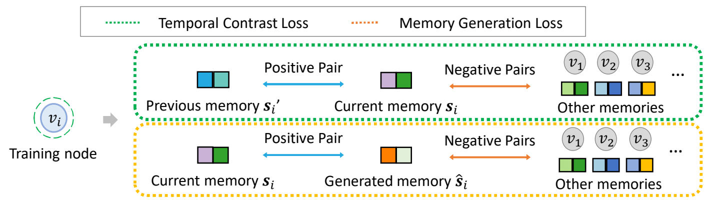
- **Normal Pattern Assumptions**
  - **A1. Stable Long-term Interaction Patterns / A2. Predictability (Restorability) of Patterns**



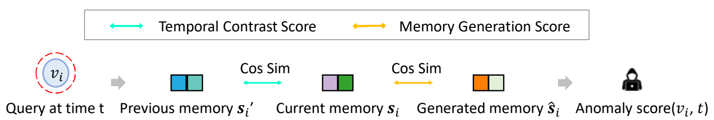
- **Proposed Self-Supervised Tasks**
  - SLADE employs two self-supervised tasks to train its model (i.e., deep neural network)
  - **S1. Temporal Contrast**: This aims to minimize drift in dynamic node representations over short-term periods (related to **A1**).
  - **S2. Memory Generation**: This aims to accurately generate (restore) dynamic node representations based only on recent neighbors (related to **A2**).
- **Core Modules of SLADE**
  - **Memory Modules**: The memory of each node represents its long-term interaction patterns.
  - **Memory Updater**: This neural network captures evolving characteristics of nodes' interaction patterns. It is employed to update the memory.
  - **Memory Generator**: This neural network is used to generate the memory of a target node from its recent interactions after masking its memory to a zero vector.



- **Training Objective**
  - SLADE assumes that training set nodes are normal and trains neural networks by minimizing **Temporal Contrast Loss** and **Memory Generation Loss** to perform well in **S1** and **S2**.



- **Anomaly Scoring**
  - In the test phase, SLADE measures how much each node deviates from **A1** (by **Temporal Contrast Score**) and **A2** (by **Memory Generation Score**) to identify anomalous states.

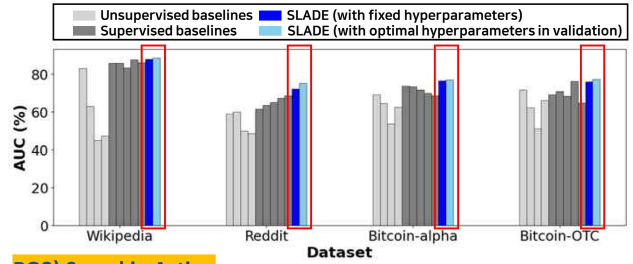


Discussion and Analysis

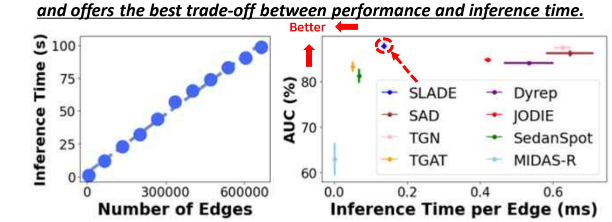
- **Time Complexity Analysis**
  - The detection time complexity in response to a query node in SLADE is $O(kd_s{}^2 + d_s d_m)$, which is constant with respect to the graph size.
    - k is the recent neighbor sample count, while d_s and d_s indicate the dimension of memory vectors and messages respectively.

Experimental Results

- **Research Questions**
  - We review our experiments for answering the following main research questions: RQ1) Accuracy, RQ2) Speed RQ3) Type Analysis.
- **RQ1) Accuracy**
  - ***SLADE outperforms other baselines, including those relying on label supervision, in dynamic anomaly detection on 4 real-world datasets.***



- **RQ2) Speed in Action**
  - ***SLADE maintains a constant inference time per edge regardless of graph size and offers the best trade-off between performance and inference time.***



- **RQ3) Type Analysis in the Absence of Anomaly Labels**
  - **T1. Hijacked Anomalies** (deviated from A1 and A2)
  - **T2. New or Rarely-Interacting Anomalies** (deviated from A1 and A2)
  - **T3. Consistent Anomalies** (deviated from A2)
  - The first dataset involves using hijacked accounts to continuously send spam emails (**T1, T3**), while the second dataset involves using new accounts (**T2, T3**).
  - ***SLADE can detect above mentioned anomaly types without relying on labels.***