GADY: Unsupervised Anomaly Detection on Dynamic Graphs

Anonymous Author(s)

1 Reproduce our experiments

Please find the code of all experiments in this anonymous website: https://anonymous.4open.science/r/GADY.

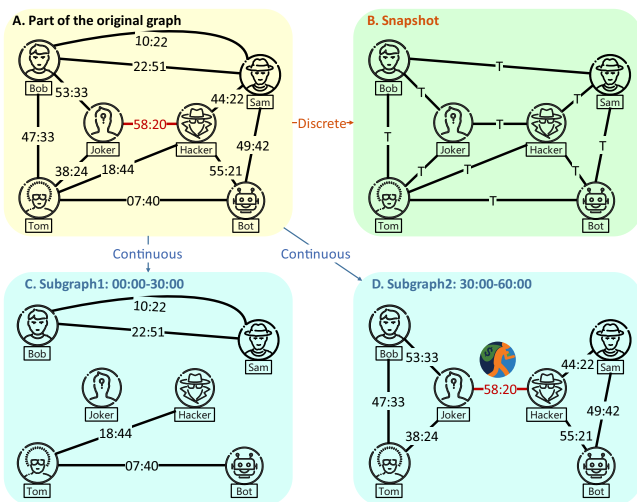

Figure 1: Example of the superiority of anomaly detection in a continuous dynamic graph. When detecting the anomaly edge of 58:20 in the dynamic original graph, discrete dynamic graph methods fall short, while continuous dynamic methods can easily distinguish this anomaly behavior.

2 Time information loss for the discrete dynamic graphs

Existing discrete dynamic graph anomaly detection methods usually first divide the original graph into several snapshots. We illustrate one snapshot in figure 1. Discrete dynamic methods construct the part of the original graph into a snapshot, and repeated edges like 10:22 and 22:51 are transformed into one edge, resulting in time information loss. Also, the time differences within the snapshot are ignored, further losing time information. On the contrary, continuous dynamic graph anomaly detection methods do not ignore any time information; through subgraph1 and subgraph2, the edge at 58:20 can be successfully detected as an anomaly.

3 Baselines

We compared GADY with eight competitive methods, which mainly include four categories. 1. Traditional non-deep learning methods: Node2vec [2] Spectral Clustering [6] DeepWalk [4]; 2. Methods using a combination of deep and traditional learning: NetWalk[7]; 3. Contrast dynamic graph methods: Temporal Graph Networks(TGN) [5]; 4. Anomaly detection using deep learning methods: AddGraph [8] StrGNN [1] TADDY [3]. A detailed introduction is shown as follows.

- **Node2vec** [2] is a social network collected from a forum of the University of California, Irvine, where each node represents a student and an edge represents a message sent.
- **Spectral Clustering** [6] is a social network collected from a forum of the University of California, Irvine, where each node represents a student and an edge represents a message sent.
- **DeepWalk** [4] is a social network collected from a forum of the University of California, Irvine, where each node represents a student and an edge represents a message sent.
- **Netwalk** [7] is a classic anomaly detection method on dynamic graphs, which generates node encodings based on random walk and dynamically updates reservoirs to model networks, and finally uses a dynamic clustering model to score edges for anomalies.
- **TGN** [5] is a classic framework for deep learning on dynamic graphs represented as sequences of timed events. It uses memory modules and graph-based operators to model dynamic system patterns and can outperform previous approaches while significantly being more computationally efficient.
- **AddGraph** [8] is an end-to-end dynamic graph anomaly detection model. It uses GCN and GRU attention to capture structural and temporal information, respectively.
- **StrGNN** [1] is an end-to-end model for anomaly detection on dynamic graphs. It first takes h-hop subgraphs for the target edge, then uses the GCN model to model the structural information within the slice, and uses GRU to model the temporal information between different slices.
- **Taddy** [3] is an end-to-end model for anomaly detection on dynamic graphs. It first samples subgraphs and obtains global encodings by diffusion, obtains local encodings based on distance on subgraphs and obtains relative time encodings according to the time difference. Then, it uses transformers to model dynamic graph networks and a scoring module to obtain anomaly scores.

4 Experiment Setup

For GADY(No-GAN), we generate negative samples using the same method with TADDY [3] and use BCEloss. For the parameters setting of the encoder part, we followed the hyperparameters setting discussed in TGN[5]. We also follow the best parameters proposed in the papers for setting for all hyperparameter settings in the baselines. We follow [3] and use a 50%-50% (train-test) temporal split for all datasets. All experiments were completed using the same seed, and we also found that different seeds did not influence our model performance greatly.

References

[1] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. 2021. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *Proceedings of the 30th ACM international conference on Information & Knowledge Management*. 3747–3756.

[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.
[3] Yixin Liu, Shirui Pan, Yu Guang Wang, Fei Xiong, Liang Wang, Qingfeng Chen, and Vincent CS Lee. 2021. Anomaly detection in dynamic graphs via transformer. *IEEE Transactions on Knowledge and Data Engineering* (2021).
[4] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.
[5] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637* (2020).
[6] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17 (2007), 395–416.
[7] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 2018. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2672–2681.
[8] Li Zheng, Zhenpeng Li, Jian Li, Zhao Li, and Jun Gao. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN.. In *IJCAI*, Vol. 3. 7.