An Empirical Study of Link Prediction in Continuous Dynamic Graphs

Amr Abouelkhair [1]

Abstract

This paper reviews various graph representation learning approaches, with a focus on link prediction in continuous dynamic graphs. We examine the design of several methodologies, including Dynamic GNNs, Temporal GNNs, and hybrid models such as GC-LSTM, comparing their performance across diverse real-world datasets like Wikipedia and Reddit. Our study emphasizes the importance of considering temporal dynamics to improve prediction accuracy in evolving networks. We demonstrate through our results that temporal embeddings significantly out perform the heuristics.

1. Introduction

Graph representation learning has been a hot topic the past few years, and justifiably so. (Rossi et al., 2020) It has revolutionized the analysis of graph structured data. Consequently, it has enabled breakthroughs in fields like cyber-security (Han et al., 2020; Rehman et al., 2024), social sciences(Frasca et al., 2020; Monti et al., 2019), and biology (Zitnik et al., 2018; Veselkov et al., 2019), setting a new state of the art. By encoding the information regarding a node, an edge, or a graph into a latent vector space, it enabled models to form a deeper understanding of the data being analyzed and to draw better relationships between data points. (Gao et al., 2023) These models demonstrate better performance in tasks like node classification or link prediction. At the heart of these breakthroughs, we find that Deep graph learning and Auto Encoders (AEs), in particular, set the bench mark, primarily due to their strength in capturing the intricate intra-graph patterns and reducing dimensionality. (Zhang et al., 2020)

However, despite robust capabilities of AEs and the major efforts focusing on handling static graphs, most real-world networks and graphs are inherently dynamic, defined by their evolving relationships and transient interactions.(Rossi et al., 2020) When these temporal dynamics are ignored, we find that static models perform sub-optimally and produce incorrect and unreliable predictions. The importance of temporal information is highlighted in fields like cyber-security, where just the duration of an action can differentiate between benign and malicious activity. (Han et al., 2020; Rehman et al., 2024)

Recently, there have been multiple attempts to develop models designed specifically to handle dynamic graphs. Temporal Graph Networks (TGN)(Rossi et al., 2020) and Dynamic Graph Neural Networks (DGNNs)(Mei & Zhao, 2024; Gao et al., 2023) represent the leading efforts in this space. (Skarding et al., 2022) There have also been efforts in combining the strength of GCNs in creating node representations with the strength of Recurrent Neural Networks (RNNs) in capturing temporal dynamics with GC-LSTM being the most notable example.(Chen et al., 2018)

In this work, we present an empirical evaluation of the leading efforts in each of these categories. We also compare the performance of these models against more traditional heuristic methods.[1] We leverage multiple real-world data sets from social networks in an attempt to present a comprehensive evaluation of dynamic link prediction in different contexts. (Leskovec & Krevl, 2014)

2. Background

This section covers the background of graph representation learning, dynamic graphs, how they differ from static graphs, and previous attempts at analyzing them.

2.1. Graph Representation Learning

With the evolution of both deep learning and learning on graph structured data, we have seen them intersect and result in what is called graph representation learning. In essence, graph representation learning is a way to convert the structural information of a graph into a format that allows us to leverage some of the deep learning methods. The main goal of graph representation learning is capture as much of the complexities, and intricate relationships between the nodes in the graph, while enabling deep learning models to conduct tasks like classification and prediction on that data.(Zhang et al., 2020)

[1]Cheriton School of Computer Science, University of Waterloo, Waterloo, ON, Canada. Correspondence to: Amr Abouelkhair <aabouelk@uwaterloo.edu>.

[1]Code is available at https://github.com/abouelkhair5/DGEval

Graph representation learning leverages techniques like GCNs and AEs to generate an embedding for each node. An embedding is essentially a feature vector, and we try to capture as much information about the node and its surrounding structure into that embedding. This can also be done for the edges or for the whole graph. (Gao et al., 2023) These embeddings can then be passed through a different deep learning model, and the theory is that the more information the embeddings can capture the more accurate the prediction would be when done on them.

2.2. Dynamic Graphs

Unlike static graphs, dynamic graphs encapsulate changes over time in their topology or attributes. The authors of (Skarding et al., 2022) formally define dynamic graphs as

Definition 2.1. A Dynamic Graph is a graph $G = (V, E)$ where $V = \{(v, t_x, t_e)\}$ with v being a vertex in the graph and t_s, t_e are respectively the start and end timestamps for the existence of the vertex (with $t_s \leq t_e$). $E = \{(u, v, t_s, t_e)\}$, with $u, v \in V$ and t_s, t_e are respectively the start and end timestamps for the existence of the edge with (with $t_s \leq t_e$).

This temporal component can significantly affect the interpretation and analysis of the graph. For instance, in social networks, the formation and dissolution of connections reflect evolving relationships, in infrastructure networks, changes can represent disruptions or growths in capacity, and in cyber-security a series of requests conducted over days could be normal while the same series conducted over a few hours could be a Denial of Service (DOS) attack. Static graph models fail to capture these dynamics, treating relationships as if they are constant over time, which can lead to inaccurate predictions and analyses (Rossi et al., 2020).
Dynamic graphs can be further categorized into discrete and continuous graphs based on their temporal granularity. On one end of the spectrum, there are static graphs with no temporal information at all and therefore are very coarse. On the other end we have continuous dynamic graphs where individual edges and nodes have their respective start and end timestamps. In the middle of the spectrum lies discrete graph where the graph is split into static snapshots the describes the total state of the graph at a specific timestamp.

2.3. Previous Approaches at Analyzing Dynamic Graphs

Early attempts at handling dynamic graphs tries to split them into a series of static graphs. However, that results in a total loss of the temporal dynamics and the information associated with it when learning. Therefore it results in a poor prediction performance.

3. Existing Methods

In this section we will cover an overview of some of the most notable approaches that tackled link prediction in dynamic networks.

3.1. Temporal Graph Networks

Temporal Graph Networks (TGNs) are a class of graph neural networks designed specifically to handle the dynamic nature of graph data that changes over time. These networks extend traditional graph neural network architectures by incorporating mechanisms that can process and learn from the temporal aspects of graph data, such as the formation or dissolution of edges, and changes in node attributes over time.
The core innovation of TGNs lies in their ability to maintain a memory component that captures the state of the graph at various points in time. This memory is updated continuously as new interactions occur, allowing the model to learn temporal dependencies and patterns within the graph data. For instance, in a social network, a TGN can learn the evolving relationships and interactions among users, which are crucial for tasks like recommending friends or content.(Rossi et al., 2020)
TGNs are also unique in their ability to handle continuous graphs without any pre-processing. On the other hand, other models convert continuous models to either discrete or static models prior to analyzing them.(Skarding et al., 2022)
Essentially, TGN is an autoencoder that performs temporal embedding. It leverages its memory module that captures the historic state for each node to generate an embedding. What is unique about this class of models is that the embedding would change for the same node as more time passes and more interactions happen that impact the node in question. This has resulted in exceptional performance in tasks like detecting malicious activities on a cloud based host. (Cheng et al., 2024)

3.2. Temporal Graph Attention

Temporal Graph Attention (TGAT) are similar to TGNs in the fact that they are designed with the dynamic nature of graph data in mind. It introduces a self attention layer in order to capture the temporal dynamics of node interactions. This allows the model to weigh the importance of a node's historical interaction based on their timing resulting in a temporal embeddings that contains information about the history of the node. They incorporate the timing of interaction in a fashion similar to positional encoders in transformers.
Further more, the temporal information is also considered when calculating the attention score. This basically allows

the model to assign a relevance score to the interaction based on their timing and age. This allows this model to generalize to unseen nodes and different temporal contexts.
Similarly to TGN, albeit with a different executio, TGATs generate temporal embeddings that represent nodes and capture their history allowing for the use of other deep models as decoder. Furthermore, TGATs can also process continuous dynamic graphs without the need for further preprocessing or splitting the data into snapshots.

3.3. GC-LSTM

Graph Convolutional Long Short-Term Memory (GC-LSTM) is a hybrid neural network model that combines spectral graph convolutional networks (GCNs) with long short-term memory (LSTM) units to effectively process graph-structured data that evolves over time. This model leverages the strength of GCNs to capture spatial relationships within the graph by encoding node features into a low-dimensional space that preserves the topology and node connectivity. Concurrently, it integrates LSTM units to model temporal dependencies, making it particularly suited for dynamic graphs where the graph structure and node interactions change over time. The GC-LSTM model is designed to learn both spatial and temporal features simultaneously, enabling it to maintain a continuous state that evolves as new data is processed (Chen et al., 2018).
In practical applications, the GC-LSTM model can handle a variety of tasks, such as dynamic link prediction, node classification, and graph forecasting, where both the historical context and current state of the graph are crucial for accurate predictions. The graph convolution layer first processes the node features to capture the structural context, which are then fed into the LSTM layer to incorporate temporal dynamics. This sequence ensures that every snapshot of the graph contributes to understanding the overall evolution of the graph structure. By maintaining a memory of past graph states through the LSTM's recurrent connections, GC-LSTM can predict future states of the graph or the behavior of its elements with a high degree of accuracy.
However, unlike TGNs, GC-LSTM handle discrete dynamic graphs and not continuous graphs. Therefore, in order to adapt continuous graphs to be processed using GC-LSTM, they have to be split into batches or snapshots. This compromises some of the granularity in comparison to TGN.

3.4. Heuristics

While Heuristics do not really account for temporal dynamics, they surprisingly perform very well in link prediction tasks, even on dynamic graphs. (Skarding et al., 2022) These methods are rather simple when compared to GNNs. However, it could be argued that their simplicity is one of their strength.
Most of the link prediction heuristic rely on a the main assumption that nodes are more likely to form nodes if they have common neighbors. They leverage this idea to give a similarity scores to node pairs to rank the likelihood of links forming. The simplest of these heuristics is the Common Neighbor (CN) heuristic.(Liben-Nowell & Kleinberg, 2003) Which defines the similarity score as:

$$|\Gamma(u) \cap \Gamma(v)| \tag{1}$$

here $\Gamma(u)$ represents the neighbors of node u. Another slightly more sophisticated heuristic, albeit based on the same idea, is Jaccard Coefficient. This defines the similarity score between two nodes as:

$$\frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{2}$$

in a similar fashion to CN. There is a rich array of literature discussing different variations of those heuristics.
Another more modern class of heuristics consider the shortest path between two nodes. Examples include Newton heuristic (Wahid-Ul-Ashraf et al., 2018) which combines the shortest path between two nodes and degree centrality to compute a similarity score, as well as common neighbor centrality based parameterized algorithm (CCPA) (Ahmad et al., 2020) which combines common neighbors with shortest paths approaches.

4. Methodology

In this section we will cover our experiment setup. We focus only on methods that tackle continuous dynamic graphs, which are the temporal embedding methods. We also pick one heuristic of each class to enrich our evaluation further. For each of the temporal embedding methods we repeat the evaluation 10 times and report the mean and the standard deviation. For the heuristic methods we only report one result as they are deterministic.

4.1. Evaluation Metrics

In order to evaluate our models fairly we mainly consider mean average precision (mAP) and the area under the characteristic curve (AUC). We follow a similar approach as the authors of (Skarding et al., 2022) as we agree with the points they present regarding these metrics as the task of link prediction in dynamic graphs is extremely unbalanced with orders of magnitudes more non-links vs links in each time step. The mean average precision is equivalte to the area under the precision-recall curve and the area under the ROC curve is a common method to evaluate models performing link prediction.

4.2. Implementation

In this section we describe how we implemented each of the methods we evaluated. For all the methods below we write the experiment code in python and we leverage PyTorch and PyTorch Geometric(Fey & Lenssen, 2019) to keep our implementation as close to the authors intention as possible.

4.2.1. TGN

In order, to implement TGN (Rossi et al., 2020) we draw heavy inspirations from the examples provided by PyTorch Geometrics (PyG)[2]. Despite the evaluation setup being slightly different than the original design the authors proposed[3] we prefer using the examples provided by the PyG team. We are further reassured by the fact that the PyG team mentioned that they have discussed with the original author and found this evaluation setup to be more realistic.

4.2.2. TGAT

We do not re-implement this appraoch. However, we repurpose the implementation originally published by the author for evaluating the same dataset.[4] After multiple results, we produce similar results to what the authors report in their work.(Xu et al., 2020)

4.2.3. Heuristics

We leverage the BenchDGCN repo [5] created by the authors of (Skarding et al., 2022). The evaluation framework implements multiple heuristics with the aid of the networkx python library. We focus on the results of the Jaccard and newton heuristics. This repo also results in very similar results to what was reported in the paper. We only run the heuristic once as they are deterministic.

4.3. Datasets Used

4.3.1. Wikipedia

This is a continuous and bipartite dataset. This represent Wikipedia editing history with the nodes being either a Wikipedia user of a Wikipedia Page. An edge represents a user editing a page thus it is bipartite. With 9227 nodes and 39804 edges it is a rather sparse network.(Leskovec & Krevl, 2014)

4.3.2. Reddit

This is also a continuous and bipartite dataset. This dataset represents posting on reddit with the nodes being either a reddit user or a subreddit. An edge represents the user posting on the subreddit. Since the edges are all timestamped we have as much granularity as possible. (Leskovec & Krevl, 2014) This is one of the largest available datasets. Despite having a similar number of nodes to the Wikipedia dataset with 10984 nodes it has much more edges with 307593 edges. (Skarding et al., 2022)

Table 1. Results of different models on different datasets.

Method	Wikipedia	Reddit
TGN AP	95.83± 0.09	98.05± 0.09
TGN AUC	95.65± 0.09	98.02± 0.08
TGAT AP	98.18± 0.2	96.7± 0.2
TGAT AUC	98.10± 0.2	96.7± 0.2
Jaccard AP	35.6	2.3
Jaccard AUC	99.9	99.7
Newton AP	16.9	6.2
Newton AUC	99.9	81.0

5. Discussion

The results of our evaluation are presented in Table 1.

We notice that the temporal encoding methods outperform the heuristics methods by multiple folds when it comes to precision. This can be attributed to the fact that the heuristic completely disregard the temporal dynamics of the dataset in question. The surprising fact, however, is that we find that heuristics produce very high AUC numbers. We theorize that this could be due to high performance at higher confidence thresholds.

One problem that hasn't been tackled by any of these models is considering the next level of complication by evaluation heterogeneous temporal graphs. All of the datasets currently being considered are homogeneous graphs with only one edge type. However there exists numerous datasets in the real world where we see multiple edge types like (Manzoor et al., 2016). The only attempt at tackling this problem that we found was presented by (Fan et al., 2022) which re-adapts the TGN presented in this work to handle heterogeneous graphs. We leave the evaluation of such methods for future works.

6. Conclusion

In this work, we study the problem of link prediction in dynamic graphs. We present a brief survey of the history of this problems and previous attempts to solve it. We show case the current state of the art performance. Finally, we conduct a brief evaluation comparing a couple of the state of the art performers as well as a few of the well known heuristics in the space. We show that the performance of the more complex deep models significantly out performs the

[2]https://github.com/pyg-team/pytorch_geometric

[3]https://github.com/twitter-research/tgn

[4]https://github.com/StatsDLMathsRecomSys/Inductive-representation-learning-on-temporal-graphs/blob/master/README.md

[5]https://github.com/xkcd1838/bench-DGNN/tree/299fee7dda057c6e9479bcb713afbc1a7c31b16b

heuristic methods with multiple folds. Finally we present an argument for where more complex deep models could show an even more significant improvement over other previous approaches.

References

Ahmad, I., Akhtar, M. U., Noor, S., and Shahnaz, A. Missing link prediction using common neighbor and centrality based parameterized algorithm. *Scientific reports*, 10(1): 364, 2020.

Chen, J., Xu, X., Wu, Y., and Zheng, H. GC-LSTM: graph convolution embedded LSTM for dynamic link prediction. *CoRR*, abs/1812.04206, 2018. URL http://arxiv.org/abs/1812.04206.

Cheng, Z., Lv, Q., Liang, J., Wang, Y., Sun, D., Pasquier, T., and Han, X. Kairos: Practical intrusion detection and investigation using whole-system provenance. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2024.

Fan, Y., Ju, M., Zhang, C., and Ye, Y. Heterogeneous temporal graph neural network. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pp. 657–665. SIAM, 2022.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Frasca, F., Rossi, E., Eynard, D., Chamberlain, B., Bronstein, M., and Monti, F. Sign: Scalable inception graph neural networks. *arXiv preprint arXiv:2004.11198*, 2020.

Gao, C., Zhu, J., Zhang, F., Wang, Z., and Li, X. A novel representation learning for dynamic graphs based on graph convolutional networks. *IEEE Transactions on Cybernetics*, 53(6):3599–3612, 2023. doi: 10.1109/TCYB.2022.3159661.

Han, X., Pasquier, T., Bates, A., Mickens, J., and Seltzer, M. Unicorn: Runtime provenance-based detector for advanced persistent threats. *arXiv preprint arXiv:2001.01525*, 2020.

Leskovec, J. and Krevl, A. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Liben-Nowell, D. and Kleinberg, J. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and knowledge management*, pp. 556–559, 2003.

Manzoor, E., Milajerdi, S. M., and Akoglu, L. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1035–1044, 2016.

Mei, P. and Zhao, Y. h. Dynamic network link prediction with node representation learning from graph convolutional networks. *Scientific Reports*, 14(1):538, 2024.

Monti, F., Frasca, F., Eynard, D., Mannion, D., and Bronstein, M. M. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*, 2019.

Rehman, M. U., Ahmadi, H., and Hassan, W. U. Flash: A comprehensive approach to intrusion detection via provenance graph representation learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 139–139. IEEE Computer Society, 2024.

Rossi, E., Chamberlain, B., Frasca, F., Eynard, D., Monti, F., and Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Skarding, J., Hellmich, M., Gabrys, B., and Musial, K. A robust comparative analysis of graph neural networks on dynamic link prediction. *IEEE Access*, 10:64146–64160, 2022. doi: 10.1109/ACCESS.2022.3175981.

Veselkov, K., Gonzalez, G., Aljifri, S., Galea, D., Mirnezami, R., Youssef, J., Bronstein, M., and Laponogov, I. Hyperfoods: Machine intelligent mapping of cancer-beating molecules in foods. *Scientific reports*, 9 (1):9237, 2019.

Wahid-Ul-Ashraf, A., Budka, M., and Musial-Gabrys, K. Newton's gravitational law for link prediction in social networks. In *Complex Networks & Their Applications VI: Proceedings of Complex Networks 2017 (The Sixth International Conference on Complex Networks and Their Applications)*, pp. 93–104. Springer, 2018.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

Zhang, Z., Cui, P., and Zhu, W. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):249–270, 2020.

Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.