

# TeleGraph: A Benchmark Dataset for Hierarchical Link Prediction

Min Zhou  
Huawei Noah's Ark Lab  
Shenzhen, China  
zhoumin27@huawei.com

Menglin Yang  
The Chinese University of Hong Kong  
Hong Kong, China  
mlyang@cse.cuhk.edu.hk

Bisheng Li  
Fudan University  
Shanghai, China  
bsli20@fudan.edu.cn

Lujia Pan  
Huawei Noah's Ark Lab  
Shenzhen, China  
panlujia@huawei.com

## ABSTRACT

Link prediction is a key problem for network-structured data, attracting considerable research efforts owing to its diverse applications. The current link prediction methods focus on general networks and are overly dependent on either the closed triangular structure of networks or node attributes. Their performance on sparse or highly hierarchical networks has not been well studied. On the other hand, the available tree-like benchmark datasets are either simulated, with limited node information, or small in scale. To bridge this gap, we present a new benchmark dataset *TeleGraph*, a highly sparse and hierarchical telecommunication network associated with rich node attributes, for assessing and fostering the link inference techniques. Our empirical results suggest that most of the algorithms fail to produce satisfactory performance on a nearly tree-like dataset, which calls for special attention when designing or deploying the link prediction algorithm in practice.

### ACM Reference Format:

Min Zhou, Bisheng Li, Menglin Yang, and Lujia Pan. 2022. TeleGraph: A Benchmark Dataset for Hierarchical Link Prediction. In *GLB 2022@WebConf '22: The ACM Web Conference 2022, April 25–29, 2022, Lyons*. ACM, New York, NY, USA, 6 pages.

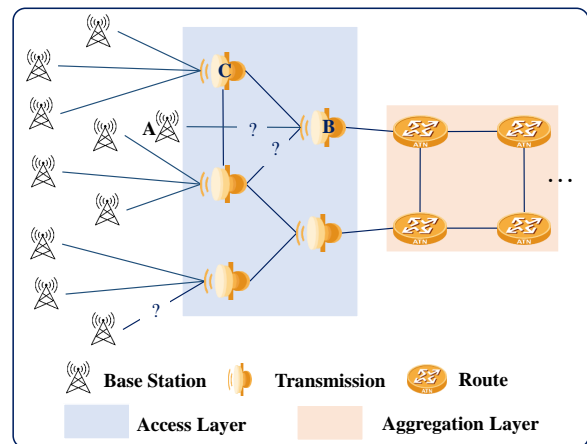
## 1 INTRODUCTION

Link prediction aims to predict whether two nodes in a network are likely to have a link [12, 21] via the partially available topology or/and node attributes, attracting considerable research efforts owing to its diverse applications. According to the techniques involved, the current link prediction methods can be categorized into three classes: heuristic methods, embedding methods, and Graph Neural Network (GNN)-based methods. Heuristic methods infer the likelihood of links via handcrafted similarity measures regarding the structure information [2, 12]. Embedding methods learn free-parameter node embeddings in a transductive manner to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

WebConf '22, April 25–29, 2022, Lyons

© 2022 Association for Computing Machinery.



**Figure 1: An illustration of the access network which is the most basic facility in the telecommunication network. It is responsible for connecting subscribers to the immediate service provider via base stations, transmission, and route. The overall topology is tree-like with a distinct hierarchical layout, but some local areas are grid or circular. Recovering the missing topology between the devices enables the efficient fault management [6], which is of importance to ensure the stability and reliability of the network.**

infer the probability if a node pair being connected [7, 11]. GNN-based methods formulate the link prediction as the binary classification problem in which the explicit node feature could be incorporated [9, 24, 26]. These techniques are typically evaluated on a limited number of regular benchmark datasets such as collaboration network [19] or citation networks [8]. However, many real-world scenarios, such as protein interaction networks [20], knowledge graphs [16], disease spreading network [4], transport networks [1], and social networks [3], are usually sparse or hierarchical in nature, which calls for special attentions [3, 15, 22]. On the other hand, the current tree-like benchmark datasets are either simulated, small in scale, or with limited node attributes, which incurs substantial affect on the progress of related studies.

To bridge the gap and further foster the link prediction research, we publicly release *TeleGraph*: a medium-sized (41,143 nodes) undirected heterogeneous network (3 classes) with multiple informative node attributes (240 types). Telecommunication networks are keys

to supporting personal communications as well as those of businesses and other organizations in the modern era. To ensure the stability and reliability of the large telecommunication network, a crucial task is efficient fault management, which requires the precise device connections enabling the engineers to analyze, locate, and recover the faults [6]. An illustration of the access layer of a telecommunication network is given in Figure 1, where the overall topology is tree-like but some local areas are grid or circular, and some connections between devices are missing. As a real-world dataset associated with complex topology and multiple node attributes and whose properties are shared by many engineered transport networks [1], we believe *TeleGraph* creates exciting opportunities for assessing and gestating both link inference techniques and node embedding procedures.

**Main contributions.** The major results and contributions presented in our work can be summed up as follows:

- We release *TeleGraph*, a highly hierarchical and sparse dataset which can be used to benchmark the link prediction algorithms.
- We conduct a descriptive analysis of the dataset and discuss the particular modeling challenges that the dataset poses.
- We assess the performance of existing link prediction algorithms regarding to both AUC and AP, provide insights and prescriptive guidance for industrial settings from our observations.

## 2 PRELIMINARY

### 2.1 Notations and Problem Definition

Consider an undirected graph  $G = (V, E, X)$  with node set  $V$ , the edge set  $E$ , and node features  $X$ , respectively. We denote  $A \in \{0, 1\}^{N \times N}$  as the adjacency matrix of  $G$ , i.e., the  $(i, j)$ -th entry in  $A$  is 1 if and only if there is an edge in  $E$  between  $v_i$  and  $v_j$  ( $v \in V$ ). Given network structure or/and node features, link prediction methods aim to preserve/learn the similarity measures of the node pair to infer missing links or detect the spurious links.

According to the techniques involved, the current Link prediction algorithms can be generally classified into three main paradigms: heuristic methods, embedding methods, and GNN-based methods. Heuristic methods compute some heuristic node similarity scores as the likelihood of links [12, 13], which are simple yet effective but not applicable for diverse scenarios. Embedding methods learn node embedding based on connections between nodes and compute similarity scores [7, 14]. GNN-based methods infer the existence of links by employing graph neural networks. More specifically, Graph Auto-Encoder(GAE) or Variational GAE [9] learn node representations through an auto-encoder framework where various GNN architectures have been utilized as the encoders. SEAL [26, 27] reformulated the link prediction task as the subgraph binary classification which shows apparent advantages over GAEs in most scenarios. In Section 4, we will carry on extensive experiments to evaluate the performance of diverse link prediction methods.

### 2.2 Evaluation Metrics

To test the algorithms' performance, the edge set  $E$  is randomly divided into the training, validation, and test set. Two standard metrics: area under the receiver operating characteristic curve

(AUC) [28] and Average Precision (AP) [25] are widely applied to evaluate the link prediction measures or algorithms.

(i) **AUC:** Given the ranking of the non-observed links, the AUC value can be interpreted as the probability that a randomly chosen missing link is given a higher score than a randomly chosen non-existent link. At each time we randomly pick a missing link and a non-existent link to compare their scores, if among  $n$  independent comparisons, there are  $n_1$  times the missing link having a higher score and  $n_2$  times have the same score, the AUC value is given as:

$$\text{AUC} = \frac{n_1 + 0.5n_2}{n}. \quad (1)$$

(ii) **AP:** Given the ranking of the non-observed links, the precision is defined as the ratio of relevant items selected to the number of items selected. AP measures the weighted mean of precision achieved at each threshold regard the precision-recall curve, with the increase in recall used as the weight:

$$\text{AP} = \sum_n (R_n - R_{n-1})P_n, \quad (2)$$

where  $R_n$  and  $P_n$  are the recall and precision at the  $n$ th threshold.

## 3 THE TELECOMMUNICATION DATASET

Telecommunication networks are keys to support personal communications as well as those of businesses and other organizations in the modern era. To ensure the stability and reliability of the large telecommunication network, a crucial task is efficient fault management, which requires the precise device connections enabling the engineers to analyze, locate, and recover the faults. Typically, the network typologies are recovered by the Link Layer Discovery Protocol (LLDP) via analyzing the device configuration files recorded on the network management systems (NMSs). However, as networks become increasingly complex, carriers need to maintain devices from multiple vendors and diverse standards (i.e., 2/3/4/5G), and data from different NMSs may not be associated or updated timely. It makes device connections are hard to be inferred via manually configuration file collection and protocol analysis. On the other hand, the alarm logs which describe the devices' status and latent connections are readily and easily to access. Analogous to a disease propagation network, alarms of the telecommunication network show strong temporal and spatial correlations that a fault occurring on one device triggers alarms of its own and also lead to alarms reported on connected devices. The incomplete topology also provides supportive information to infer the missing connections.

### 3.1 Dataset Description

The *TeleGraph* is an access layer of a metropolitan telecommunication network, which contains 41,143 vertices (devices), categorized into 3 types: routers, transmission (e.g., microwave), and base station (e.g. NodeB). The raw data includes the paths log and the alarms log, in which the paths log gives how information transits through the network. A path is an ordered list and a device may belong to multiple paths. The alarm logs tell when and where the alarms happened. Table 1 and Table 2 depict the examples of path and alarm log, respectively. The collected alarm logs range from 12<sup>th</sup> to 16<sup>th</sup> April 2019, with more than six million alarms categorized into 240 types. By combing paths and associating the alarms with

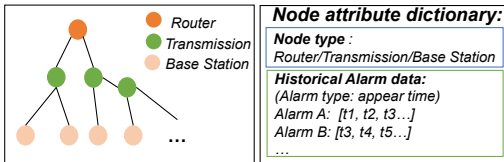
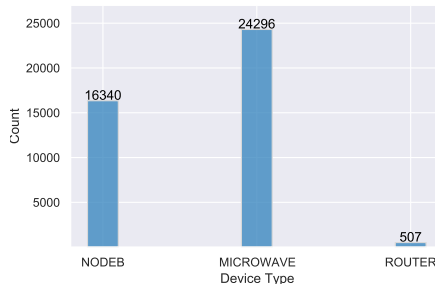
the corresponding devices, we got an attributed graph in which the attributes information of each device (i.e., node) corresponds to alarm logs with each alarm type and the list of alarm occurrences of each type, sorted by time, as illustrated in Fig. 2. Readers may refer to the work [6] for more details on the preprocessing.

**Table 1: Example of paths log**

Path Id	Device Name	Device Type	Path Hop
1	Device 1	ROUTER	0
1	Device 2	MICROWAVE	1
1	Device 3	NODEB	2
2	Device 1	ROUTER	0
2	Device 4	MICROWAVE	1
2	Device 5	MICROWAVE	2
2	Device 6	NODEB	3
...	...	...	...

**Table 2: Example of alarms log**

Alarm Name	Device Type	Alarm Source	Occurrence Time
Alarm 1	NODEB	Device 3	2019-04-12 10:40:23
Alarm 2	MICROWAVE	Device 2	2019-04-12 10:40:24
Alarm 3	MICROWAVE	Device 2	2019-04-12 10:40:26
Alarm 4	ROUTE	Device 1	2019-04-12 10:40:51
...	...	...	...

**Figure 2: The network topology (left) with alarm attributes (right) of the TeleGraph dataset.****Figure 3: Number of different device types**

We first analyze the number of each type of device, which is given in Figure 3. As we can see, the transmission device (i.e., MICROWAVE) and base station (i.e., NODEB) account for most and the hub device (i.e. router) has a much small number. We further investigate the edges, which are summarized in Table 3. In general, a telecommunication network is bi-directed. In the downlink mode (e.g. a

user receives the information from the website), the data comes from the core network flowing from routers to the base station (to terminal devices)<sup>1</sup> via the microwaves, which makes the network a tree-like structure. However, the MICROWAVE may connect to the same type on the other paths, leading to the network in some areas grid or circular.

**Table 3: Number of edges among different devices**

Source device	Target device	Number of Edges
ROUTER	ROUTER	0
ROUTER	MICROWAVE	2,001
ROUTER	NODEB	22
MICROWAVE	MICROWAVE	23,282
MICROWAVE	NODEB	16,119
NODEB	NODEB	0
Total		41,424

### 3.2 Exploratory Analysis

We further present some characteristics on the topology of *TeleGraph* and the statistics are summarized in Table 4, where density [23] and hyperbolicity [4] measure the sparsity and “tree-likeness” of a given network, respectively. Specifically, the density is 0 for a graph without edges and 1 for a complete graph, and the hyperbolicity value of approximately zero means a high tree-likeness. As anticipated, *TeleGraph*, which has 41,143 nodes and 41,424 edges, is highly tree-like and sparse as it has very small values on both density and hyperbolicity. We then visualize the degree distribution, as given in Figure 4. As expected, the power-law phenomenon is very obvious with most of the nodes having a degree less than three and a very few hub nodes (i.e. routers) with distinctly large degree values. Though the dataset is nearly a tree, a few loops exist with the largest ones with 13 nodes and the smallest one having only 3 nodes. We further analyze the distribution of the loop size which is further given in Figure 5. As observed, most of the cycles involve only 3 or 4 nodes and a small proportion of the cycles have more than 5 nodes.

**Table 4: Statistics of TeleGraph**

Statistics	#Nodes	#Edges	#Cycles	Density $\rho$	Hyperbolicity $\delta$
Value	41143	41424	684	0.000049	0

\* The smaller  $\rho$  means the dataset is more sparse.

\* The smaller  $\delta$  indicates the dataset has a more evident tree-like structure.

In summary, the *TeleGraph* is a highly sparse and hierarchical network with rich node attributes, which is then a suitable benchmark to assess and foster link prediction techniques. However, inferring the missing links in a highly sparse and tree-like graph is challenging. As mentioned in work [15], most of the widely used heuristics measures overly count on the closed triangle structures and perform poorly in a tree-like or highly sparse network. On the other hand, most of the GNN-based link prediction algorithms are feature-centric, and how to utilize and encode the node attributes is also not easy.

<sup>1</sup>For the up-link mode, the information flows from the base station to microwave and then router to core network

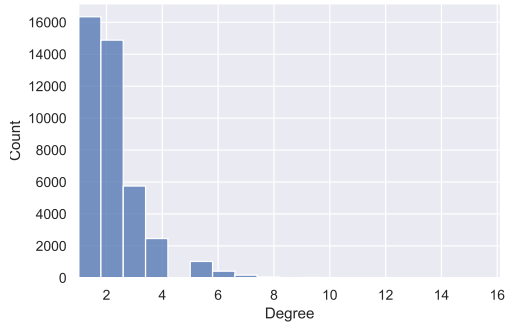


Figure 4: Degree distribution of TeleGraph, which is asymptotic

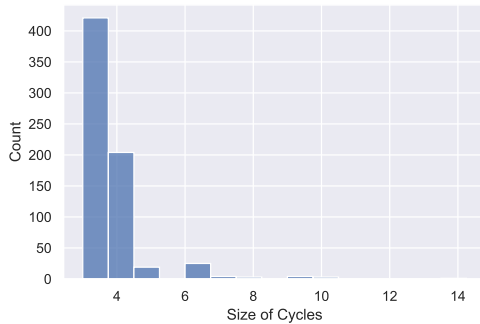


Figure 5: Node number distribution of the cycles

## 4 BENCHMARK EXPERIMENTS

In this section, we present the benchmark experiments and discuss the observations.

**Baselines.** To investigate the link prediction algorithms on the tree-like *TeleGraph* dataset, we experiment with diverse well-known baselines, including heuristic methods: common neighbors (CN) [12], Adamic-Adar (AA) [12], Personalized PageRank (PPR) [18]; embedding-based models: *node2vec* [7]; GAE-based models [9]; and the state-of-the-art link prediction models: NeoGNN [24] and SEAL [27]. Specifically, heuristic methods formulate hand-crafted rules as scoring functions. *node2vec* [7] learns representations on graphs via optimizing a neighborhood preserving objective. For GAE-like baselines, we use different encoders including well-known Euclidean GNN models GCN [10], GAT [17], and pioneering hyperbolic graph neural network HGNN [4]<sup>2</sup>. Their differences are mainly in the mechanism of message passing where GCN aggregates information through structure information while GAT aggregates information through feature correlation. Hyperbolic deep models generalize (graph) neural networks into hyperbolic space, abstracting underlying hierarchical layout in the graph datasets, achieving competing results of tree-like datasets on various tasks. NeoGNN [24] learns the structural embeddings by the overlapped neighborhoods which are further combined with the node representations obtained from the feature-based GNN for link prediction. SEAL [27] has achieved various SOTA results on link prediction tasks whose priority mainly lies in the extraction and encoding of local structural information.

<sup>2</sup>Inner product is employed as a decoder to compute the link existence.

**Hyperparameter settings.** We split the edges to 85%/5%/10% for training, validation and test, respectively. And the random seed is set as 2 in data split. The embedding dimensions are set as 32 as well as the batch size for all models in order to make comparison fair. Each experiment is conducted for 10 times and we report both the mean and standard deviation. All the code is implemented with PyTorch and we use the implementation in PyTorch Geometric [5] for the GNN-related algorithms.

### 4.1 Experiment Results

The *TeleGraph* dataset provides both the device’s connections as well as the non-vectorized node attributes (i.e., alarm occurrence). Graph neural networks are essentially feature-centric, which encourages us to first investigate some preliminary feature engineering schemes. Then we provide more comprehensive comparisons regarding the benchmark methods.

Table 5: Results on validation (denoted as Val) and test set of different feature schema

Encoder	Embeddings	Val AUC	Val AP	Test AUC	Test AP
GCN	one-hot	69.01 ± 0.97	69.79 ± 0.89	68.74 ± 0.77	69.62 ± 0.64
	count	64.52 ± 0.29	65.38 ± 0.32	64.20 ± 0.30	65.38 ± 0.32
	random	51.87 ± 0.65	54.79 ± 0.69	51.87 ± 0.73	54.88 ± 0.72

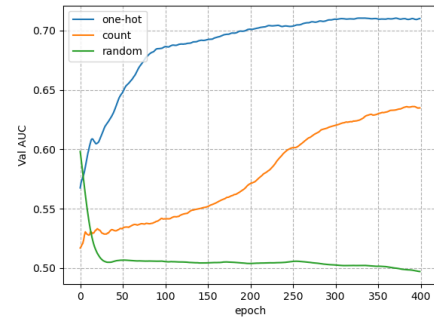


Figure 6: AUC vs Epoch on validation set

**4.1.1 Comparison on different feature schemes.** In this work, we come up with three initiatory embedding initialization schemes. In particular, *one-hot* and *count* assign a  $1 \times 240$  dimensional vector to each node according to whether the alarm appears or not and the number of occurrences, respectively. *random* initializes an input embedding for each node randomly, which is widely adopted in situations where the node attributes are not available. We here employ it to justify the informativeness of the alarm logs. The results are summarized in Table 5<sup>3</sup>. As we can see, *one-hot* produces the best performance with the AUC and AP on the test set both exceeding the second-best *count* by 5%. One the other hand, randomly initiating the features produces the most unsatisfactory results that both the AUC and AP are around 0.5, which is very similar to what chance would predict. As further observed from Figure 6, the AUC

<sup>3</sup>We here only list the results of GCN due to the page limitation, both the results and analysis are consistent to that of GAT and HGNN.

value of *random* even decreases with the training epoch. This is probably because the dataset is very sparse in nature, masking the validation and test edges make the message propagation and aggregation even harder and randomly assigning the features may bring some noise that hinders the learning of models.

**Table 6: Performance Comparison on *TeleGraph*. Average values with the standard deviations are reported. For both AUC and AP, the higher, the better.**

Method	Val AUC	Val AP	Test AUC	Test AP
CN	51.01± 0.00	51.00± 0.00	51.06± 0.00	51.06± 0.00
AA	51.01± 0.00	50.99± 0.00	51.06± 0.00	51.06± 0.00
PPR	51.76± 0.00	51.60± 0.00	51.75± 0.00	51.83± 0.00
Node2Vec	52.22 ± 0.68	56.34 ± 0.69	51.94 ± 0.51	56.01 ± 0.54
GCN	69.01 ± 0.97	69.79 ± 0.89	68.74 ± 0.77	69.62 ± 0.64
GAT	66.80 ± 0.12	67.78 ± 0.13	67.51 ± 0.12	67.07 ± 0.16
HGCN	67.49 ± 0.36	67.65 ± 0.24	67.46 ± 0.38	67.84 ± 0.27
NeoGNN	63.20 ± 1.00	67.38 ± 0.66	62.45 ± 1.13	67.07 ± 0.70
SEAL	80.74 ± 0.15	80.43 ± 0.23	79.48 ± 0.14	78.82 ± 0.20

**4.1.2 Performance comparison of baseline methods.** The experimental results of baseline methods on *TeleGraph* are further summarized in Table 6<sup>4</sup>. As anticipated, the heuristic methods (i.e., CN, AA, and PPR) which are largely based on the closed triangular structures fail to correctly predict the link existence of the given tree-like dataset. The AUC and AP values on both validation and test are around 0.5, which is very similar to the random guess. On the other hand, the GAE-based methods (i.e., GCN, GAT, and HGCN) and sub-graph classification based GNN model SEAL outperform either the heuristic measures or the graph embedding model (i.e., Node2Vec) by large margins, confirming the GNN-based solutions which formulate the link prediction task as a supervised learning problem are promising. It is noted, HGCN, which has been widely proved to be promising for tree-like dataset, is neck and neck with its Euclidean counterpart GAT. The possible reason is the handcrafted node features fail to well-preserve the information carried on the node attributes, which hinders the representation power of hyperbolic geometry. Though NeoGNN suggests explicitly incorporating the structural information to the feature-GNNs, it still falls into the GAE framework and barely produces satisfactory performance on the given dataset. Last but not least, SEAL, which formulates the link prediction task as a sub-graph classification problem and further explicitly encodes the node position, shows apparent advantages over the others on this highly hierarchical and sparse dataset. In summary, both the structural information and informative attributes are for link prediction. Most of the handcrafted heuristic measures heavily count on the triangle structure or preferential attachment assumption and fail to work in the highly tree-like and sparse dataset. Though the subgraph extraction and encoding scheme (i.e., SEAL) is promising, explicitly encoding and incorporating the semantic information is needed for further performance improvement.

<sup>4</sup>*one-hot* is adopted as the initial embedding schema for message-passing based models (i.e., GAT, GCN, and HGCN).

## 5 CONCLUSION

Link prediction is the problem of detecting the presence of a connection between two entities in a network. Research fields, ranging from network science to machine learning and data mining, have taken a great interest in link prediction task. Given that hierarchical patterns found in many real-world applications while the corresponding research datasets are inadequate, in this work, we present a new real-world dataset *TeleGraph*, which is a medium sized telecommunication network with a rich set of attributes. Our descriptive analysis of the dataset has demonstrated it is highly hierarchical and sparse, which makes the heuristic measures fail to work. We verified this recognition by a series of experiments. Our findings show that most of the available algorithms fail to produce the satisfactory performance on this tree-like dataset except the subgraph-based GNN-models. More specifically, the results of a series heuristic measures are even close to random guesses, which calls for special attention in practice. We also believe that *TeleGraph* can serve as an important benchmark to assess and foster novel link prediction techniques.

## ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their constructive suggestions.

## REFERENCES

- [1] Ashleigh Ballantyne, Nicholas Lawrance, Michael Small, Melinda Hodkiewicz, and David Burton. 2018. Fault prediction and modelling in transport networks. In *2018 IEEE International Symposium on Circuits and Systems*. IEEE, 1–5.
- [2] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.
- [3] Peter S Bearman, James Moody, and Katherine Stovel. 2004. Chains of affection: The structure of adolescent romantic and sexual networks. *American journal of sociology* 110, 1 (2004), 44–91.
- [4] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*. 4869–4880.
- [5] Matthias Fey and Jan E. Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*.
- [6] Philippe Fournier-Viger, Ganghuan He, Min Zhou, Mourad Nouioua, and Jiahong Liu. 2020. Discovering alarm correlation rules for network fault management. In *International Conference on Service-Oriented Computing*. Springer, 228–239.
- [7] 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*.
- [8] Zengfeng Huang, Shengzhong Zhang, Chong Xi, Tang Liu, and Min Zhou. 2021. Scaling up graph neural networks via graph coarsening. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 675–684.
- [9] Thomas N Kipf and Max Welling. 2016. Variational Graph Auto-Encoders. *NIPS Workshop on Bayesian Deep Learning* (2016).
- [10] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.
- [11] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [12] David Liben-Nowell and Jon Kleinberg. 2007. The link-prediction problem for social networks. *Journal of the American society for information science and technology* 58, 7 (2007), 1019–1031.
- [13] Linyuan Lü and Tao Zhou. 2011. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* 390, 6 (2011), 1150–1170.
- [14] 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.
- [15] Ke-ke Shang, Tong-chen Li, Michael Small, David Burton, and Yan Wang. 2019. Link prediction for tree-like networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 29, 6 (2019), 061103.
- [16] Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2018. Poincare Glove: Hyperbolic Word Embeddings. In *International Conference on Learning*

*Representations.*

- [17] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* (2018).
- [18] Hanzhi Wang, Zhewei Wei, Junhao Gan, Sibow Wang, and Zengfeng Huang. 2020. Personalized pagerank to a target node, revisited. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 657–667.
- [19] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413.
- [20] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks?. In *International Conference on Learning Representations*.
- [21] Menglin Yang, Min Zhou, Marcus Kalander, Zengfeng Huang, and Irwin King. 2021. Discrete-time Temporal Network Embedding via Implicit Hierarchical Learning in Hyperbolic Space. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1975–1985.
- [22] Menglin Yang, Min Zhou, Zhihao Li, Jiahong Liu, Lujia Pan, Hui Xiong, and Irwin King. 2022. Hyperbolic Graph Neural Networks: A Review of Methods and Applications. *arXiv preprint arXiv:2202.13852* (2022).
- [23] Menglin Yang, Min Zhou, Jiahong Liu, Defu Lian, and Irwin King. 2022. HRCF: Enhancing Collaborative Filtering via Hyperbolic Geometric Regularization. In *Proceedings of the Web Conference 2022*.
- [24] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J Kim. 2021. Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction. *Advances in Neural Information Processing Systems* 34 (2021).
- [25] Ethan Zhang and Yi Zhang. 2009. *Average Precision*. Springer US, Boston, MA, 192–193.
- [26] Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in Neural Information Processing Systems* 31 (2018), 5165–5175.
- [27] Muhan Zhang, Pan Li, Yinglong Xia, Kai Wang, and Long Jin. 2021. Labeling Trick: A Theory of Using Graph Neural Networks for Multi-Node Representation Learning. *Advances in Neural Information Processing Systems* 34 (2021).
- [28] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting missing links via local information. *The European Physical Journal B* 71, 4 (2009), 623–630.