Dartmouth College
Dartmouth Digital Commons

Computer Science Senior Theses | Computer Science

Spring 5-31-2023

GeoRCF-GN: Geography-Aware State Prediction in Dynamic Networks

Barkin Cavdaroglu
Dartmouth College, barkin.cavdaroglu.gr@dartmouth.edu

Follow this and additional works at: https://digitalcommons.dartmouth.edu/cs_senior_theses

Part of the Computer Sciences Commons

Recommended Citation

Cavdaroglu, Barkin, "GeoRCF-GN: Geography-Aware State Prediction in Dynamic Networks" (2023). *Computer Science Senior Theses*. 26.
https://digitalcommons.dartmouth.edu/cs_senior_theses/26

This Thesis (Undergraduate) is brought to you for free and open access by the Computer Science at Dartmouth Digital Commons. It has been accepted for inclusion in Computer Science Senior Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact dartmouthdigitalcommons@groups.dartmouth.edu.

GeoRCF-GN: Geography-Aware State Prediction in Dynamic Networks

Barkin C. [*1]

Abstract

In this paper, we formulate a geography-aware model for predictions over dynamic networks using Graph Neural Network (GNN) and Random Convolutional Feature (RCF) extractor. We take traffic and mobile ad-hoc communication networks as our main case studies, and study their special properties to design an architecture that makes use of information about both the graph topology and the geography that the nodes are located in, as geographical factors are crucial in determining future radio signal quality and traffic sensor readings.

1. Introduction

In recent years, there has been a surge of interest in applying machine learning techniques to graph structured data, enabling the extraction of valuable insights from complex relational structures (Zhou et al., 2020). Graphs provide a powerful framework for representing and analyzing relationships between entities, and their applications span across diverse domains such as social networks, biological systems, transportation networks, and more. Within this evolving landscape, the study of temporal graphs has emerged as a captivating area of research due to its applicability in problems that are temporal in nature, such as communication networks, traffic networks, and social networks. Temporal graphs capture the evolution of relationships over time, making them particularly relevant for modeling dynamic phenomena. In this paper, we study a neural network architecture designed to predict temporal graph properties. We will especially focus on temporal graphs that model scenarios such as traffic and communication networks. There has been a great interest specifically in using machine learning models for traffic flow forecasting, as traffic congestion is a challenging problem with significant consequences. On the other hand, not much research has been done on using machine learning models on communication networks.

First-response teams operating in offline disaster scenarios typically utilize radio devices to communicate with one another due to the absence or unreliability of internet and phone signal access. Nevertheless, radio devices come with their own limitations, and responders in the field frequently act ”blindly,” without cognizance of how their trajectories and rescue plans may impact the reliability of the communication network. Quality of radio signal or probability that one radio device will be able to communicate with another is influenced by many factors, some of which rescuers will not have any control over, e.g. atmospheric conditions and obstructions in the general area (Whitehouse et al., 2007; Lymberopoulos et al., 2006; Luomala & Hakala, 2015). However, in most cases, responders can coordinate actions that in tandem keep the connectivity of communication network intact. Figure 1 conveys how one such coordination

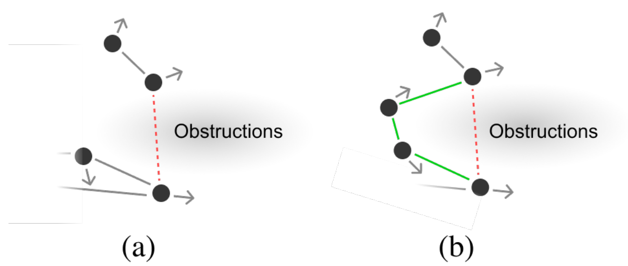

Figure 1. Various tasks to train GNNs for. Figure (a) is node-level task, (b) is edge-level task, and (c) is graph level task.

can prevent network disconnection. Figure 1(a) shows that if responders follow the trajectories they are on, indicated as arrows, the network will disconnect, leaving two responders in the dark. Fig 1(b), on the other hand, shows an example of a coordination around the obstructions that would preserve the network connectivity. The importance of connection between all responders becomes especially vital in dangerous scenarios such as natural disasters. Even though network reliability is an active research area, most studies do not take geographical factor into consideration and instead assume that network failures are random and independent (Liu et al., 2011), which, as our example suggests, is not always the case, and that network reliability is somewhat dependent on the environmental factors that responders operate in. Furthermore, they assume a static network and focus on proactive measures rather than both proactive and reactive ones. This is also relevant to traffic flow forecasting. Having knowledge of the environmental factors that would be practically impossible to manually collect is vital for making the most accurate predictions.

Various machine learning algorithms have been proposed in the field of communication networks and traffic flow forecasting. Adaptive channel equalization is one such example, where (Kumar et al., 1998; Zhang & Zhang, 2007; Erdogmus et al., 2001) propose various machine learning and deep learning methods to reduce potential information loss as much as possible due to dense and simultaneous communication. In similar vein, there could exist several types of interference in the same environment, therefore, it could be useful to discover the existence and types of such interference to decrease packet loss and transmission latency. Schmidt et al. use deep CNN based model to classify various types of interference (2017), whereas Grimaldi et al. use SVM-based methods for the same task (2017). In (Li et al., 2010), authors use reinforcement learning techniques to find an adaptive rate control strategy to optimize link layer performance and minimize power consumption. However, there hasn't been as much research activity that deals with the problem of prediction in MANETs. As for traffic flow forecasting, Li et al. (2017) make use of diffusion convolution instead of more commonly used spectral convolution method, and Zheng et al. (2021) use attention based convolutional LSTM model with even better performance than the previous state-of-the-art. However, to our knowledge, there is no research on the subject that also factors in the environmental factors.

In this paper, we propose an architecture that uses Random Convolutional Feature (RCF) extractor to capture geographical information from satellite images and GNN to capture information about graph topology and node and edge features. We will be using a subset of METR-LA[1] for training.

2. Preliminaries

Generally, a graph can be defined as a tuple $G = (V, E)$ where V is the set of nodes and E is the set of edges. In some cases such as ours, we might want to attribute some features to nodes, e.g. node degrees, and edges, e.g. distance between two nodes. Using graph neural networks (GNNs) (Scarselli et al., 2008), we can make inferences on nodes, edges, and whole graphs such as node classification (Kipf & Welling, 2016), link prediction (Zhang & Chen, 2018; Long et al., 2022; Li et al., 2014) and graph classification (Zhang et al., 2018; Lee et al., 2018; Gao & Ji, 2019).

2.1. Graph Neural Networks

The discovery of GNNs were motivated in part due to the incompatibility of graph data with classical deep learning models, such as convolutional neural networks (CNNs) as they strictly require grid-structured inputs. This follows from the unordered property of V, the set of nodes in a graph. In other words, any learnable function f that operates on graphs must be permutation equivariant to the order of its input, since a graph G with adjacency matrix A can be represented by multiple different adjacency matrices $\{P_1 A, P_2 A, ..., P_n A\}$ where P_i is any permutation matrix on A. This means that $f(A) = f(P_i A)$ for $i \in [1, n]$.

[1]METR-LA can be found here.

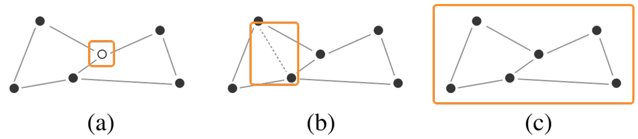

Figure 2. Various tasks to train GNNs for. Figure (a) is node-level task, (b) is edge-level task, and (c) is graph level task.

2.1.1. Graph Convolutional Network (GCN)

There are various GNN models. However, it is usually expected that any GNN model contains 2 fundamental operations. AGGREGATE collects information from node u's neighborhood, $\mathcal{N}_u$, and UPDATE combines this aggregated information into a hidden embedding for u. One such model that has been increasingly popular is GCN model

$$H^{l+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^l W^l) \tag{1}$$

where $\tilde{A} = A + I$, $\tilde{D}$ is degree matrix of $\tilde{A}$, H^l is hidden node embeddings and W^l is a trainable weight matrix for current layer l (Kipf & Welling, 2016). This operation is first-order approximation of localized spectral filters on graphs. In essence, GCN is performing aggregation of features per node u *normalized* by its degree and that of its neighbor as $\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}$.

2.1.2. Graph Attention Network (GAT)

Another hugely successful variant of GNNs is graph attention network, which incorporates attention weights to aggregator function. Specifically, hidden embedding of node u is calculated as

$$\alpha_{uv} = \frac{\exp(\sigma_n(a(\mathbf{W}h_u \mathbf{W}h_v)))}{\sum_{w \in \mathcal{N}_u} \exp(\sigma_n(a(\mathbf{W}h_u \mathbf{W}h_w)))} \tag{2}$$

$$h_u = \sigma_a(\sum_{v \in \mathcal{N}_u} \alpha_{uv} \mathbf{W} h_v) \tag{3}$$

where α_{uv} can be thought of as node v's *influence* on node u, $\mathbf{W}$ is weight matrix that parameterize shared attentional mechanism a, and σ_n and σ_a are non-linearity functions such as LeakyReLU (Veličković et al., 2017).

2.1.3. Set Aggregator

There is a growing body of work on set aggregator functions to improve the performance of GNNs. Some notable ones include DeepSets (Zaheer et al., 2017) and JanossyPooling (Murphy et al., 2018), which inspired our aggregator function explained in Section 3. Specifically, DeepSets solves the problem of permutation-equivariance required by aggregator function by transforming each element x in a set S using a function of choice, adding up all transformations of elements and processing the output using another function of choice. Since x is expected to be a vector, we can use any function we want, including any deep network. JanossyPooling takes a different approach, and provides two ways we can employ to improve set aggregator function: (1) We generate all permutations of a given set, apply a function of choice to each one of permutations, and average the results, or (2) we find a canonical ordering on set S, and apply a function of choice on the ordered input. However, it is expected that this ordering is not random and makes use of properties inherent to elements in the set.

2.2. Temporal Network State Prediction

Recently, learning over temporal graphs (or dynamic networks) has become a major research topic as it can help us better understand how graph data modeling complex relations changes over time. Earlier works applied matrix factorization or other types of aggregations to capture the temporal dimension (Dunlavy et al., 2011; Yu et al., 2017). While more recent works turned to deep neural networks for better performance. For example, Singer et al. (2019) proposed an alignment method over different snapshots and used an LSTM layer to learn the evolution of each node over time. da Xu et al. (2020) extended Graph Attention Networks with a temporal dimension when aggregating the neighbors. This was further extended in (Singer et al., 2022) by tBDFS which added a new layer that is responsible for capturing Depth-First Search patterns in a temporal graph. Moreover, Spatial-Temporal Graph Neural Networks (STGNNs) are being used to model the complex spatial-temporal correlations such as in traffic, social network, and pedestrian trajectories data (Wang et al., 2020; Peng et al., 2020; Min et al., 2021; Zhou et al., 2021). STGNNs usually model the traffic system as a diffusion process (Li et al., 2017) and combine diffusion convolution and sequential models such as GRU and LSTM. (Lei et al., 2019) has presented the novel temporal link prediction model GCN-GAN to tackle the dynamic link prediction problem in network systems. The authors combine GCN, LSTM and GAN to effectively deal with temporal link prediction tasks in weighted dynamic networks. They tested their model with four datasets of different network systems and demonstrated strong performance against other six competitors. In (Bonner et al., 2019), authors introduced the Temporal Neighbourhood Aggregation (TNA) model for temporal link prediction task. Their novel TNA block uses GCN for spatial and GRU for temporal learning before concatenating the outputs to pass into a linear layer for spatio-temporal learning. Additionally, they use Variational Autoencoder for generating the future adjacency matrix. Their model outperformed other popular ones such as GCN-GAN, LSTM and GraphSAGE. Shao et al. (2022a) propose a model named FastSTLSG for the same problem but directly apply to MANETs by using chaotic time theory for *slicing* sequence of networks before passing them into a variant of GCN. They generate the future adjacency matrix using Least Squares GAN.

3. Methods

Our data consists of timelines of snapshots of graph taken at specific intervals. A snapshot refers to node and edge information of a graph at a specific time. We represent each graph as $G_t = (V_t, E_t, \vec{V}_t, \vec{E}_t, \vec{g}_t)$. Set V_t provides indexing i over $\vec{V}_t$ and E_t provides indexing $k = \{i, j\}$ over $\vec{E}_t$. The remaining $\vec{V}_t = \{\vec{h}_i \,|\, \vec{h}_i \in \mathbb{R}^{F_{\text{node}}} \; \forall i \in V_t\}$, $\vec{E}_t = \{\vec{e}_k \,|\, \vec{e}_k \in \mathbb{R}^{F_{\text{edge}}} \; \forall k \in E_t\}$, and $\vec{g}_t \in \mathbb{R}^{F_{\text{global}}}$ with F_{node}, F_{edge}, F_{global} representing node, edge, and global feature dimensions, are collections of features for nodes, edges, and graph for graph at time i, respectively. Our objective therefore is

$$f(G_0, G_1, ..., G_t) = Z \tag{4}$$

where Z can be anything from adjacency matrix at time $t+1$ to set of node properties at time $t+1$. Function f, which can be seen as composition of many functions, is what we want to learn using samples of sequence $G_0, G_1, ..., G_t$.

3.1. Map Feature Extraction

As our problem is geographical in nature, that is, each of our node has GPS coordinates associated with them, we will use sattelite images bounded by the coordinates of the endpoints of each edge using Google Maps Static API. However, the image returned by the Google Maps Static API will have redundant information. The image in Fig. 3 is what the API gives us when we use the GPS coordinates of our two connected nodes. But we only care about the portion of the image that is between these two points. There are various ways we can crop the original image. For example, if we are dealing with a communication network, then we would use Fresnel Zone to create a mask. In wireless communication, when an electromagnetic wave travels between a transmitter and a receiver, it encounters various obstacles such as buildings, trees, and terrain features, as discussed in previous section extensively. These obstacles can cause the wave to diffract, scatter, and interfere with the direct line of sight path. The Fresnel zone helps to quantify the effects of diffraction and interference caused by these obstacles. The radius of the first Fresnel Zone is calculated with

$$F_1[m] = 8.656\sqrt{\frac{D[km]}{f[Ghz]}} \tag{5}$$

where $D[km]$ is distance between two points in kilometers and $f[Ghz]$ is frequency of transmitted signal in gigahertz. For traffic flow prediction task, a simpler ellipse with semi-minor $\frac{D[km]}{2}$ and semi-major $D[km]$ should be sufficient. After this calculation, it is relatively simple to mask the image using the elliptic region.

Depending on the task at hand, we have two methods we can use to extract features from these images. We can either adopt a multitask learning approach such as the one described in (Kendall et al., 2018), or integrate the image and graph learning into a single process by fusing the features extracted from map images with the initial edge features such as distance between the endpoints. For traffic flow prediction, we have chosen to go with the latter approach, because the dataset we are using, METR-LA, has stationary nodes and fixed topology, therefore we have the same map images between epochs, which makes it redundant to separate the graph and image learning.

Figure 3. Image of map that contains the endpoints of an edge

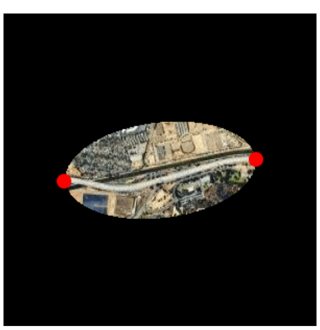

Figure 4. Masked image of the map that contains the endpoints of an edge

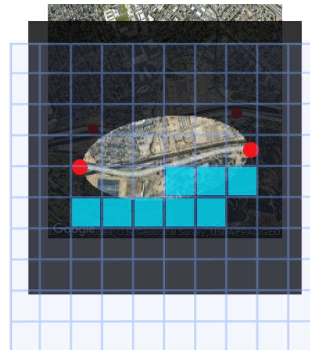

Figure 5. Feature extraction from the masked map

3.2. GN Block

We make use of the generalized functional model Graph Network (GN) for relational reasoning over our data (Battaglia et al., 2018). As illustrated in Figure 2, our model has three sub-modules for edge, node, and graph level processing. For communication networks, edge-level properties such as signal strength would be affected by node- and graph-level properties and not vice versa as is the assumption in (Battaglia et al., 2018), because it is not necessarily the edge-level property that determines the position of a highly mobile node in a communication network, but the opposite. However, in traffic flow prediction, the distance between two traffic sensors can in fact affect how much traffic could flow from one end to another. Furthermore, graph-level properties can also affect node and edge-level properties. In a communication network, as discussed in Introduction, factors such as atmospheric conditions and temperature can affect the signal strength and in some cases the position of a node. However, as we are using METR-LA, this will not be necessary for us. With these in mind, we have chosen to use the order of standard GN.

We use random convolution feature (RCF) extractor (Rolf et al., 2021) from TorchGeo library (Stewart et al., 2022) with a linear layer head to extract features from our images and MLP with a single hidden layer to embed initial edge and graph features.

$$\vec{e}_{im} = RCF(e_{im}) \tag{6}$$

$$\vec{e}_i^{\,'} = \mathbf{MLP}_e(\vec{e}_i) \tag{7}$$

$$\vec{e}_i^{\,f} = tanh(\delta^e([\mathbf{W^e_e}\vec{e}_i^{\,'}, \mathbf{W^e_{im}}\vec{e}_{im}, \mathbf{W^e_{g_t}}\vec{g}_t, \mathbf{W^e_{g_{t-1}}}\vec{G}_{t-1}])) \tag{8}$$

Here, $\vec{G}_{t-1}$ is the learned hidden embedding of the graph at time $t-1$. For $t = 0$ we can initialize $\vec{G}_{t-1}$ to $\tilde{\mathbf{0}}$.

For node features, we employ weighted multi-head attention where the final node embedding is based on weighted sum of each head's *opinionatedness* in reference to prediction task, where we measure opinionatedness using variance of the attention weights it assigns to each node. This is similar to "confidence" of a head as described in (Voita et al., 2019), but instead of using maximum weight, we define opinionatedness with variance of weights. We start by calculating normalized attention coefficients $\varkappa_{ij}^n$ per node i, which is node j's influence on node i.

$$\varkappa_{ij}^n = \frac{\exp(\sigma(\delta_n([\mathbf{W_n^n}\vec{h_i}, \mathbf{W_n^n}\vec{h_j}])))}{\sum_{k\in\mathcal{N}_i^n}\exp(\sigma(\delta_n([\mathbf{W_n^n}\vec{h_i}, \mathbf{W_n^n}\vec{h_k}]))} \tag{9}$$

We then apply our weighted multi-head mechanism with mean as our aggregator function, and concatenate $\varkappa_{ij}^n$ and $\varkappa_{ik}^e$ to get the final node embedding

$$\varkappa_{\mathcal{N}_i^n}^* = \sigma\left(\sum_{q=1}^{H} \tilde{Var}(\{\varkappa_{ij}^n, \forall j \in \mathcal{N}_i^n\}) \cdot \mathrm{H}_q^n(i)\right) \tag{10}$$

$$\vec{h}_i = \left(\frac{1}{|\mathcal{N}_i^n|}\sum_{j\in\mathcal{N}_i^n} \varkappa_{ij}^n \mathbf{W_n^n}\vec{h_j}\right) \tag{11}$$

$\tilde{Var}$ is normalized variance of the given set of attention weights. Pruning on the number of heads can be done following the work in (Voita et al., 2019).

We proceed by updating the aggregated node embeddings.

$$\vec{h}_i^* = \mathbf{MLP}_n([\vec{e}_i^f, \vec{h}_i]) \tag{12}$$

To update global graph embedding, we apply aggregation over all node and edge embeddings. We have several ways to do this: (a) we can separately pool nodes' and edges' embeddings and concatenate them with $\vec{g}$ before passing them to our readout function, as shown in (Hamrick et al., 2018), or (b) we can apply a similar canonical ordering to nodes or edges, train a neural function on the ordered sequence, and then perform (a). We choose to go with (b) since it allows us to exploit some important properties about our graph that we discussed previously. The method we are using is a variant of Janossy pooling described in (Murphy et al., 2018). However, instead of using all permutations of the node set as our space, we first order nodes according to their diffusion centrality defined as $DC(\mathbf{g}; q, T) := \mathbf{H}(\mathbf{g}; q, T) \cdot \mathbf{1} = (\sum_{t=1}^{T}(q\mathbf{g})^t) \cdot \mathbf{1}$ where $DC(\mathbf{g}; q, T)_i$ is the expected total number of times that message originating from i is received by any of the nodes of the network during a T-period time interval (Banerjee et al., 2014; 2013), and then for each node $i \in V$, we let π_i be sequence of nodes $j \in V$ ordered by $\sum_{(k,l)\in\psi_{ij}} \eta_l^k$, where ψ_{ij} is the edge pair sequence of shortest path between i and j, and Π the set of all such sequences. Then, we globally aggregate node embeddings as

$$\vec{V}' = \mathbf{MEAN}(\{\mathbf{GRU}([\pi_i]) \mid \pi_i \in \Pi\}) \tag{13}$$

Table 1. Root Mean-Squared Error with Horizon 3

MODEL	RMSE
DCRNN(Li et al., 2018)	5.17
STEP(Shao et al., 2022b)	4.98
MegaCRN(Jiang et al., 2022)	4.94
D2STGNN(Shao et al., 2022c)	4.88
GeoRCF-GN	4.552

We choose to use GRU for our function since the ordering is based on the propagation of information spreading out from node i. Based on experiments, this could be changed to RNN or LSTM. We proceed by globally aggregating the edge embeddings and pass our results to readout function. A similar ordering could be found for edges using *current-flow betweenness centrality*. However, to keep things simple, we decided to use MEAN function over edge embedding set $\vec{E}$.

$$\vec{E}_t' = \mathbf{MEAN}(\vec{E}) \tag{14}$$

$$\vec{G}_t = \mathrm{READOUT}([\vec{E}', \vec{V}', \vec{G}_{t-1}]) \tag{15}$$

where $\vec{E}$ and $\vec{V}$ are embedding sets of nodes and edges, respectively. For READOUT function, we can use both standard and neural functions. Recent work by Buterez et al. (2022) demonstrated that neural readouts performed better in majority of the datasets they used with competitive results in the rest without any hyper-parameter tuning or cross-validation. however, we chose to use mean to keep the number of parameters of our model more manageable.

4. Results

We have not performed hyperparameter tuning. We leave this for our future studies as we try to build a MANET dataset. Furthermore, we could potentially increase our accuracy by using better CNN models than RCF extractor, which we couldn't do because the addition of 1722 images per sample added significant memory overheads that our personal machines were not able to handle. We trained our model for 4 epochs using weight decay of 0.0005 and learning rate of 0.001. Furthermore, we used 4 attention heads and 1-hop neighborhood aggregation.

5. Discussion

We faced several difficulties such as lack of publicly available datasets for mobile communication networks that includes GPS coordinates and links. Although our objective was to show the impact of geo-aware models to predict the next state of graphs, we find it difficult to draw a conclusion with certainty here whether such models could outperform models that are solely based on graph topology, since the nodes in our dataset are stationary, and edges are static.

This meant that we would be using the same 1722 images throughout the training process. Furthermore, many of the assumptions we utilized in this paper are based on special properties of communication networks, and not traffic networks. However, the preliminary results of our experiment are encouraging and show the promise of using geographical features in tasks such as traffic flow prediction.

References

Banerjee, A., G a, A., Duflo, E., and Jackson, M. The diffusion of microfinance. *Science (New York, N.Y.)*, 341: 1236498, 07 2013. doi: 10.1126/science.1236498.

Banerjee, A., Chandrasekhar, A. G., Duflo, E., and Jackson, M. O. Gossip: Identifying central individuals in a social network. Working Paper 20422, National Bureau of Economic Research, August 2014. URL http://www.nber.org/papers/w20422.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, H. F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K. R., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M. M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

Bonner, S., Abarghouei, A. A., Jackson, P. T. G., Brennan, J., Kureshi, I., Theodoropoulos, G., McGough, A. S., and Obara, B. Temporal neighbourhood aggregation: Predicting future links in temporal graphs via recurrent variational graph convolutions. *CoRR*, abs/1908.08402, 2019. URL http://arxiv.org/abs/1908.08402.

Buterez, D., Janet, J. P., Kiddle, S. J., Oglic, D., and Liò, P. Graph neural networks with adaptive readouts, 2022. URL https://arxiv.org/abs/2211.04952.

Dunlavy, D. M., Kolda, T. G., and Acar, E. Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):1–27, 2011.

Erdogmus, D., Rende, D., Principe, J. C., and Wong, T. F. Nonlinear channel equalization using multilayer perceptrons with information-theoretic criterion. In *Neural Networks for Signal Processing XI: Proceedings of the 2001 IEEE Signal Processing Society Workshop (IEEE Cat. No. 01TH8584)*, pp. 443–451. IEEE, 2001.

Gao, H. and Ji, S. Graph u-nets. In *international conference on machine learning*, pp. 2083–2092. PMLR, 2019.

Grimaldi, S., Mahmood, A., and Gidlund, M. An svm-based method for classification of external interference in industrial wireless sensor and actuator networks. *Journal of Sensor and Actuator Networks*, 6(2):9, 2017.

Hamrick, J. B., Allen, K. R., Bapst, V., Zhu, T., McKee, K. R., Tenenbaum, J. B., and Battaglia, P. W. Relational inductive bias for physical construction in humans and machines. *CoRR*, abs/1806.01203, 2018. URL http://arxiv.org/abs/1806.01203.

Jiang, R., Wang, Z., Yong, J., Jeph, P., Chen, Q., Kobayashi, Y., Song, X., Fukushima, S., and Suzumura, T. Spatio-temporal meta-graph learning for traffic forecasting. *arXiv preprint arXiv:2211.14701*, 2022.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, 2018.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

Kumar, P., Saratchandran, P., and Sundararajan, N. Communication channel equalisation using minimal radial basis function neural networks. In *Neural Networks for Signal Processing VIII. Proceedings of the 1998 IEEE Signal Processing Society Workshop (Cat. No.98TH8378)*, pp. 477–485, 1998. doi: 10.1109/NNSP.1998.710678.

Lee, J. B., Rossi, R., and Kong, X. Graph classification using structural attention. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1666–1674, 2018.

Lei, K., Qin, M., Bai, B., Zhang, G., and Yang, M. GCN-GAN: A non-linear temporal link prediction model for weighted dynamic networks. *CoRR*, abs/1901.09165, 2019. URL http://arxiv.org/abs/1901.09165.

Li, X., He, H., and Yao, Y.-D. Reinforcement learning based adaptive rate control for delay-constrained communications over fading channels. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2010.

Li, X., Du, N., Li, H., Li, K., Gao, J., and Zhang, A. A deep learning approach to link prediction in dynamic networks. In *Proceedings of the 2014 SIAM International conference on data mining*, pp. 289–297. SIAM, 2014.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Graph convolutional recurrent neural network: Data-driven traffic forecasting. *CoRR*, abs/1707.01926, 2017. URL http://arxiv.org/abs/1707.01926.

Li, Y., Yu, R., Shahabi, C., and Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

Liu, J., Jiang, X., Nishiyama, H., and Kato, N. Reliability assessment for wireless mesh networks under probabilistic region failure model. *IEEE Transactions on Vehicular Technology*, 60(5):2253–2264, 2011.

Long, Y., Wu, M., Liu, Y., Fang, Y., Kwoh, C. K., Chen, J., Luo, J., and Li, X. Pre-training graph neural networks for link prediction in biomedical networks. *Bioinformatics*, 38(8):2254–2262, 2022.

Luomala, J. and Hakala, I. Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks. In *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pp. 1247–1255. IEEE, 2015.

Lymberopoulos, D., Lindsey, Q., and Savvides, A. An empirical characterization of radio signal strength variability in 3-d ieee 802.15. 4 networks using monopole antennas. In *European Workshop on Wireless Sensor Networks*, pp. 326–341. Springer, 2006.

Min, S., Gao, Z., Peng, J., Wang, L., Qin, K., and Fang, B. Stgsn—a spatial–temporal graph neural network framework for time-evolving social networks. *Knowledge-Based Systems*, 214:106746, 2021.

Murphy, R. L., Srinivasan, B., Rao, V. A., and Ribeiro, B. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *CoRR*, abs/1811.01900, 2018. URL `http://arxiv.org/abs/1811.01900`.

Peng, H., Wang, H., Du, B., Bhuiyan, M. Z. A., Ma, H., Liu, J., Wang, L., Yang, Z., Du, L., Wang, S., et al. Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting. *Information Sciences*, 521:277–290, 2020.

Rolf, E., Proctor, J., Carleton, T., Bolliger, I., Shankar, V., Ishihara, M., Recht, B., and Hsiang, S. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature Communications*, 12(1):4392, 2021. doi: 10.1038/s41467-021-24638-z. URL `https://doi.org/10.1038/s41467-021-24638-z`.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Schmidt, M., Block, D., and Meier, U. Wireless interference identification with convolutional neural networks. In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, pp. 180–185. IEEE, 2017.

Shao, H., Wang, L., Liu, H., and Zhu, R. A link prediction method for manets based on fast spatio-temporal feature extraction and lsgans. *Scientific Reports*, 12(1):16896, 2022a. doi: 10.1038/s41598-022-20981-3. URL `https://doi.org/10.1038/s41598-022-20981-3`.

Shao, Z., Zhang, Z., Wang, F., and Xu, Y. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pp. 1567–1577. ACM, 2022b.

Shao, Z., Zhang, Z., Wei, W., Wang, F., Xu, Y., Cao, X., and Jensen, C. S. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proc. VLDB Endow.*, 15(11):2733–2746, 2022c.

Singer, U., Guy, I., and Radinsky, K. Node embedding over temporal graphs. *CoRR*, abs/1903.08889, 2019. URL `http://arxiv.org/abs/1903.08889`.

Singer, U., Roitman, H., Guy, I., and Radinsky, K. tbdfs: Temporal graph neural network leveraging dfs, 2022. URL `https://arxiv.org/abs/2206.05692`.

Stewart, A. J., Robinson, C., Corley, I. A., Ortiz, A., Lavista Ferres, J. M., and Banerjee, A. TorchGeo: Deep learning with geospatial data. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, pp. 1–12, Seattle, Washington, 11 2022. Association for Computing Machinery. doi: 10.1145/3557915.3560953. URL `https://dl.acm.org/doi/10.1145/3557915.3560953`.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *CoRR*, abs/1905.09418, 2019. URL `http://arxiv.org/abs/1905.09418`.

Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., and Yu, J. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, pp. 1082–1092, 2020.

Whitehouse, K., Karlof, C., and Culler, D. A practical evaluation of radio signal strength for ranging-based localization. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(1):41–52, 2007.

Xu, D., Ruan, C., Körpeoglu, E., Kumar, S., and Achan, K. Inductive representation learning on temporal graphs. *CoRR*, abs/2002.07962, 2020. URL `https://arxiv.org/abs/2002.07962`.

Yu, W., Aggarwal, C. C., and Wang, W. Temporally factorized network modeling for evolutionary network analysis. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pp. 455–464, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450346757. doi: 10.1145/3018661.3018669. URL https://doi.org/10.1145/3018661.3018669.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. J. Deep sets. *CoRR*, abs/1703.06114, 2017. URL http://arxiv.org/abs/1703.06114.

Zhang, L. and Zhang, X. Mimo channel estimation and equalization using three-layer neural networks with feedback. *Tsinghua Science Technology*, 12(6):658–662, 2007. ISSN 1007-0214. doi: https://doi.org/10.1016/S1007-0214(07)70171-2. URL https://www.sciencedirect.com/science/article/pii/S1007021407701712.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31, 2018.

Zhang, M., Cui, Z., Neumann, M., and Chen, Y. An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zheng, H., Lin, F., Feng, X., and Chen, Y. A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):6910–6920, 2021. doi: 10.1109/TITS.2020.2997352.

Zhou, H., Ren, D., Xia, H., Fan, M., Yang, X., and Huang, H. Ast-gnn: An attention-based spatio-temporal graph neural network for interaction-aware pedestrian trajectory prediction. *Neurocomputing*, 445:298–308, 2021.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001. URL https://www.sciencedirect.com/science/article/pii/S2666651021000012.