A Survey of Spatio-Temporal Graph Neural Networks for Traffic Prediction

Karan Samel
ksamel@gatech.edu
Georgia Institute of Technology

Constantine Dovrolis
constantine@gatech.edu
Georgia Institute of Technology

ABSTRACT

Deep learning is a growing field due to ongoing collection of data and the increase in compute power. Many of these data collections are represented as graphical structures in comparison to the more common tabular data structures. There has been a recent interest in the deep learning community to leverage these graph structures within models to encode spatial dependencies for better modeling performance. These graph representations often have a temporal component as well that are modeled by recurrent deep learning techniques to create spatio-temporal graph neural networks (STGNNs). This application is being applied for traffic speed prediction, where sensors in a grid are measured over time. We compare different STGNNs tested on traffic prediction to determine the current advances in the field. We conclude by discussing the growing applications of STGNNs in other pertinent fields.

1 INTRODUCTION

Neural network architectures efficiently encode non-linear patterns in large scale datasets into compressed representations for prediction. Most of these algorithms are applied on grid-like samples, such as a vectorized table or an image. While many data sources and prediction problems can be fulfilled in this manner, recent research has been dedicated to applying these techniques to graphs. Graphs do not follow this rigid structure as nodes are connected to a variable number of edges within the graph. This has led to development of geometric deep learning techniques over graphs and manifolds [1] to handle this variable structure, such as graph neural networks (GNNs).

Within the works on GNNs, there are multiple directions of research [7]. One task is to create an embedding representation for a node within a graph with respect to the features of other nodes. This task is further used to create embeddings for an entire graph, which is useful when comparing entire networks structures with each other, such as comparing the similarity of molecules. Such tasks can be done without any prediction objective, as these methods only rely on the underlying structures between nodes and their corresponding subgraphs.

These representations can be used for prediction as well. For a node, given information regarding its neighboring nodes, we may be interested on predicting a value for that node. An application of this is to detect if a user in a social network is actually a bot. Similarly we want to discover hidden relationships between nodes and infer edges. Edge inference is commonly used to generate recommendations to a user given an database of items connected to other users.

Some of these graphical data sources are dynamic, where the values of the edges or nodes change over time. A prominent example is traffic data, which are composed of traffic sensors arranged in a graph-like structure and capture the speed of traffic over time. This information is critical in determining congestion, which is a factor in routing a driver to their destination. Due to the availability and size of these datasets, STGNN variations have been employed to predict these speed values over time. There are review papers [19, 23] that cover these traffic STGNN models, but we review and compare four of these STGNN variations in more detail. We follow this by discussing how these STGNN techniques are applied to other problems as well.

2 TRAFFIC SPEED PREDICTION METHODS

Traffic speed prediction has been a crucial problem before STGNN methods and have been modeled in various ways :

- Historic Average: Takes the average values of that sensor over periods of time.
- Ridge Regression: Models the historical values with linear regression with a regularizer term.
- Auto-regressive (AR) models: These models similarly look at historical sensor values with the forecasting errors made. Commonly used methods are Spatiotemporal AR [14].
- Long Short-Term Memory (LSTM) [8]: Deep learning models that sequentially process historic sensor data to fit future sensor values.

In general these models lack local contextual information of the sensor network or fail to fit non-stationary distributions over time. STGNN models attempt to address these issues with spatial and temporal encoding mechanisms.

2.1 Diffusion Convolutional Recurrent Neural Network (DCRNN)

An initial approach to view traffic flow is the diffusion of vehicles throughout a system of traffic sensors. This diffusion is non-trivial since the traffic dynamics vary from location to location. Simply linking the nearest sensor nodes may not be correct as these edges are directional with respect to traffic. To model this diffusion process across the sensor nodes, DCRNN [12] performs a random walk from each node, keeping track of nodes encountered. This approximates the likelihood that vehicles entering one node will pass through another as no road connection information was provided. This is done in a bi-directional fashion to capture traffic relationships in either direction. Using this sampling method, embeddings for the nodes that are likely to be connected are trained to have similar representations for that time step.

This diffusion can be modeled at different time steps and fed into a Gated Recurrent Unit (GRU) network [3], which similar to the LSTM encodes the entire sequence of historical sensor signals. This temporal representations are then fed into a Sequence to Sequence architecture [16]. Here the representations are first encoded with diffusion GRU blocks to provide a salient representation of the spatio-temporal data. A decoder, with diffusion GRU units as well, is trained to break down these encoded representations to predict the sensor network at the next time slice.

2.2 Gated Attention Networks (GaAN)

The GaAN architecture [22] uses graph aggregation [6] to build the sensor node embeddings as opposed to the random walks performed by DCRNN. A graph aggregation on a node is an estimation of that node value through the aggregation of its neighbors. This can be done by simply summing or averaging the neighborhood node's features. However due to varying importance of neighboring nodes, additional steps are taken to weight a neighboring node's influence on a particular node.

This is handled by an attention layer which is trained to weight the effects of neighboring nodes on a particular node [18]. Multiple attention models can be computed, called attention heads, which apply different weights based on the subspace of information each attention model may capture [17]. The effect of the heads is applied uniformly so GaAN introduces a methodology to control the effects of all these attention heads with gates. This is done by adding another layer above the attention heads to compute the importance across the heads when training.

Using this methodology, graph aggregation is improved due to the tuning of neighboring node importance. This can be seen as targeting the neighboring sensors versus sampling them as done in DCRNN. This spatial information per sensor is combined for each time step and fed into a GRU to construct a Graph GRU (GGRU). This is similarly fed into an encoder decoder network to predict the traffic speed for the following time steps.

2.3 Spatiotemporal multi-graph convolution network (ST-MGCN)

Constructing spatial features between intermediate nodes only encompasses a part of the true underlying spatial correlations. ST-MGCN [5] leverages global correlations in traffic networks through multiple sub graphs of the original graph. Graph aggregation can be performed across graphs using these global correlations to best encode these representations. This starts by defining multiple factors that influence between a node and any other node in the graph:

- Neighborhood: The adjacent nodes to a particular node, which is used for standard graph aggregation.
- Functional Similarity: Determining which regions of the network have similar functions, such as cities or parks.
- Transportation Connectivity: A mapping of the physical road connections between sensor nodes.

Each of these factors have a role of how traffic flows and are used to define the spatial graph aggregation to generate a robust representation over the entire network.

When predicting the traffic in a certain region, the first step involves re-weighting the previous time steps using a gating mechanism. This is done by concatenating the historical information for that region and related regions over time. The graph aggregation over this joint representation is computed and its attention is used to re-weight the samples at the previous time steps to highlight the relevant time samples. These weighted time steps are then fed into a recurrent neural network (RNN) [13] trained to predict the next time step for all nodes.

2.4 Spatio-Temporal Graph Convolutional Networks (STGCN)

The previous methods discussed used spatial estimation components in combination with a recurrent network, GRUs or RNNs, to encode traffic spatio-temporal components. STGCN [21] takes a different approach for the temporal encoding by running a 1-D convolution over the sensor nodes. This involves taking a fixed number of sensor time steps and projecting them into a lower dimensional space. Unlike typical convolutional kernels, these convolutions leverage gated linear units (GLU) [4] to only convolve over the previous timesteps and have a gating mechanism to determine which kernels are useful, completing the temporal gated-conv process.

Given temporal features for each sensor, graph aggregation is then performed to capture the neighboring spatial features. This representation is further fed into another temporal gated-conv process to further compress the temporal dimension.

These ST-Conv blocks composed of temporal, spatial, then temporal aggregation can be stacked until the final prediction layer which predicts the sensor speeds at the next time step. This architecture leads to a computational performance improvement over the DCRNN method and trains 10-15 times faster due to the completely convolutional structure, which are not restricted to recurrent computation.

3 METHOD COMPARISONS

The methods we reviewed had different approaches to 1) encode spatial information of traffic data and 2) represent this spatial information over time. These relations are summarized as follows:

Model	Spatial Structure	Temporal Structure
DCRNN	random walk	GRU
GaAN	gated graph aggregation	GRU
ST-MGCN	graph aggregation with objectives: • neighborhood • functional • transportation	RNN
STGCN	graph aggregation	1-D CNN with GLU

Each method used different approaches to solve these two tasks and have corresponding accuracy and efficiency trade-offs. We see for the order DCRNN, STGCN, GaAN and ST-MGCN, the spatial structure composition becomes increasingly advanced. DCRNN samples to estimate its spatial representation, while STGCN and GaAN build thier representations from their neighbors directly. ST-MGCN is able to use the extra contextual information to best encode the underlying spatial representation of the sensor nodes.

The DCRNN, GaAN, and ST-MGCN further encode this spatial representations at each time slice through recurrent models. Due to this procedure, these methods typically take longer to train as the spatial representation has to be fed into the recurrent model sequentially. This gives an Advantage to STCCN in terms of performance due to the scalability of convolution operations.

The predictive performance of each models varied by application. When predicting the traffic speed in METR-LA [9], a dataset of highway sensors in Los Angeles, GaAN is able to outperform DCRNN. Similarly in PeMSD7 [2], collected from California highway sensors, STGCN outperforms DCRNN as well due to its simpler architecture. For Ride-share demand forecasting ST-MGCN proves to improve over the STGCN approach due to the extra graph aggregation objectives. We surmise that various architectures need to be tested and iterated on based on the specific dataset and objective.

4 OTHER STGNN APPLICATIONS

STGNNs have been applied to traffic prediction problems due their graphical nature and abundance of time slices. These similar architectures have been leveraged in other domains as well.

In RE-NET [11], multi-relational knowledge graphs over time are used for event prediction. There are works in human pose classification and forecasting as well. This is done by rolling out a spatial graph through time as done in Structural-RNN [10], and by running graph aggregation across time slices in ST-GCN [20] (A different architecture from the STGCN described earlier). In the healthcare domain, medication for a patient are predicted by G-BERT [15] by their historical medications and diagnoses, which are represented in hierarchical fashions.

As STGNN methods become more efficient in encoding spatial and temporal structures, more of such applications can be pursued that have less data compared to traffic data sets.

5 CONCLUSION

The advancements of deep learning have led to the development of graph inference methods. These methods range from predicting a node to embedding entire graphs by constructing a spatial representation of graph nodes. When graph data is presented sequentially, these spatial representations are extended into the temporal space with STGNN architectures. Due to their graphical structure and abundance of data points, we focus on traffic prediction problems. We review four architectures and discuss their spatial structures, temporal structures, and compute trade-offs. We conclude by discussing how STGNNs are being used to solve other problems as well.

REFERENCES

[1] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 34, 4 (2017), 18–42.

[2] Chao Chen, Karl Petty, Alexander Skabardonis, Pravin Varaiya, and Zhanfeng Jia. 2001. Freeway performance measurement system: mining loop detector data. *Transportation Research Record* 1748, 1 (2001), 96–102.

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[4] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 933–941.

[5] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[7] William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. arXiv:cs.SI/1709.05584

[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. 2014. Big data and its technical challenges. *Commun. ACM* 57, 7 (2014), 86–94.

[10] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. 2016. Structural-RNN: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5308–5317.

[11] Woojeong Jin, Changlin Zhang, Pedro Szekely, and Xiang Ren. 2019. Recurrent Event Network for Reasoning over Temporal Knowledge Graphs. *arXiv preprint arXiv:1904.05530* (2019).

[12] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[13] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

[14] R. Kelley Pace, Ronald Barry, John M. Clapp, and Mauricio Rodriquez. 1998. Spatiotemporal Autoregressive Models of Neighborhood Effects. *The Journal of Real Estate Finance and Economics* 17, 1 (01 Jul 1998), 15–33. https://doi.org/10.1023/A:1007799028599

[15] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019. Pre-training of Graph Augmented Transformers for Medication Recommendation. *arXiv preprint arXiv:1906.00346* (2019).

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[18] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[19] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2019. A Comprehensive Survey on Graph Neural Networks. arXiv:cs.LG/1901.00596

[20] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[21] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 3634–3640.

[22] Jiani Zhang, Xingjian Shi, Junyuan Xie, Hao Ma, Irwin King, and Dit-Yan Yeung. 2018. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *arXiv preprint arXiv:1803.07294* (2018).

[23] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. arXiv:cs.LG/1812.08434