Graph Neural Networks for Pattern Analysis from Time Series

Ayça Avcı, Jeroen de Baat

Abstract— Many real world relations can be expressed as graphs, such as a social network or the state of traffic at a certain time. These relations often involve a spatial component, and are not static but evolve over time. While many deep learning models have been proposed to extract patterns from static graphs, fewer models exist for pattern analysis on spatiotemporal graphs because of the increased complexity data. Yet, the latter has many applications in domains such as behavior prediction, computer vision and robotics. Here, we present related work on the application of neural networks on graphs in general, give an overview of the current state of the field, and provide a comparison and discussion of the described methods.

Index Terms—Artificial Intelligence, Neural Networks, Deep Learning, Graph Neural Network, Time series, Pattern analysis/recognition.

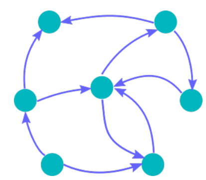
1 Introduction

Pattern analysis is the detection of regularly occurring events in data. There are various methods for this, of which deep learning approaches have gained significant popularity as these have shown to be effective in several disciplines, such as image classification [17] and natural language processing [1]. Recently, there has been much interest in extending pattern analysis to graphs, as many real world relations can be represented as such, for instance, traffic flow [11, 18, 30, 34, 36] and social networks [28, 38]. Applications include the prediction of traffic flow [11, 18, 30, 34, 36], demographic attribute prediction, content recommendation, and targeted advertising [28]. Using deep learning methods on graphs is more challenging due to the increased structural complexity of the data. Even more challenging is the modeling of graphs in time series (i.e. a sequence of graphs which evolve over time), and spatiotemporal graphs (i.e. graphs in times series where the data also has a location component), which is what we will focus on in this paper.

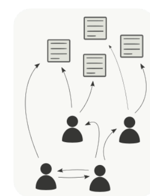
An overview of the current state of the field does, to the best of our knowledge, not exist. In this work, we aim to provide a brief introduction to deep learning and Graph Neural Networks (GNNs), followed by a survey of several graph neural network models for the analysis of time series for pattern recognition. We will focus on how the various approaches work and how these compare to one another, based on their specific applications and effectiveness.

This work is structured as follows: In Section 2, we describe the datasets on which the models are tested. In Section 3, we describe the models themselves, and in particular their networks, data representation methods and goals. The metrics used to compare the models are described in Section 4, and the results obtained form each method are in Section 5. The results are discussed in Section 6, and our findings are concluded in Section 7. First, we will provide definitions and context for the topic at hand.

A graph G is defined as a tuple $G = (V, E)$, with the vector V representing the *vertices* (also called *nodes* or *points*), and the vector E representing the *edges* i.e. the connections between the vertices. A graph can either be directed or undirected, weighted or unweighted, and signed or unsigned. In the context of GNNs, we mainly consider unsigned graphs without self-loops and without multiple edges. Figure 1a shows such a graph. Depending on the data and model, this definition of a graph can be extended. For example, feature vectors can be added for the vertices and edges to express more complex information. Figure 1b shows an example of a more complex graph of a social network, where some vertices represent users and others represent posted messages. Here, an edge between two users could mean that the users know each other, and an edge between a user and a message could mean that the user has posted that message.



(a) A representation of an abstract directed graph, with unlabeled vertices and edges.



(b) A social network represented as a graph.

Fig. 1: Two examples of graphs representations.

Deep Learning is a field within Artificial Intelligence which is defined by the use of artificial neural networks with multiple layers between the input layer and the output layer. These so called *Deep Neural Networks* (DNNs) have been used for a variety of applications with data of different natures, such as image processing [20] and audio signal processing [22].

Extensive work has been done on the use of DNNs on non-spatiotemporal graphs [39], with applications ranging from social networks [28, 38] to biology networks. Zhang et al. [39] have identified several challenges in using DNNs to model graphs in particular:

- Graphs often have irregular structures which makes the application of common operations in neural networks, such as convolution, more difficult.
- The heterogeneity and diversity of graphs: they can be heterogeneous or homogeneous, weighted or unweighted, and signed or unsigned. In addition to this, the learning in graphs can be node-focused (e.g. node classification, node prediction, link prediction) or graph-focused (i.e. graph classification, graph generation), depending on the nature of the data and the application. Consequently, there is no general modeling approach for all graphs. Instead, each type of graph and application requires its own approach.
- Graphs can be very large, requiring the algorithms operating on them to be very efficient.
- Graphs often represent data from disciplines such as biology, chemistry, and the social sciences. Understanding the nature of

- *Ayça Avcı, MSc, is a student Computing Science at the University of Groningen, E-mail: a.avci@student.rug.nl.*
- *Jeroen de Baat, MSc, is a student Computing Science at the University of Groningen, E-mail: j.de.baat@student.rug.nl.*

Manuscript received 24 February 2021; posted online 15 April 2021. For information on obtaining reprints of this article, please contact the CS Student Colloquium of the University of Groningen, faculty of Science and Engineering.

Data set	Size	Min	Max	Mean	SD
Japan-Prefectures	47×348	0	26635	655	1711
US-Regions	10×785	0	16526	1009	1351
US-States	49×360	0	9716	223	428

Table 1: "Dataset statistics in terms of min, max, mean, and standard deviation (SD) of patient counts; dataset size means the number of locations multiplied by number of weeks" [8].

the data is often essential to modeling, making the modeling process require interdisciplinary knowledge.

Despite these challenges, many models have been proposed, categorized by Zhang et al. [39] into Graph Recurrent Neural Networks (GRNNs) [24], Graph Convolutional Networks (GCNs) such as [2], Graph Autoencoders (GAEs) such as [27], Graph Reinforcement Learning, and Graph Adversarial Networks. For each of these models, variants exist depending on the nature of the data and modeling task.

As indicated previously, many real world relations — which can be represented using a graph — change over time. The change in relations can therefore be represented as a *sequence* of graphs, also known as *temporal* graphs. The patterns in these graphs may not only extend to the static nodes and edges themselves, but also to their relation to the temporal dimension. Therefore, specialized models are required to perform pattern analysis on these graphs.

2 Datasets

In this section, we describe the datasets used by the models surveyed.

2.1 CalendarGNN: Calender Graph Neural Networks

The CalendarGNN model has been trained and tested using large-scale user behavior logs which have been collected from two real portal websites. The data contains articles and news updates on several topics. The two spatiotemporal datasets are created as $B^{(w1)}$ and $B^{(w2)}$ [28]. These datasets provide spatiotemporal behavior logs of the browsing behavior of users from these two websites, both ranging from January 1, 2018 to June 30, 2018. Each dataset is filtered to 10000 users, who have most clicks, after users have been anonymized [28].

The 3 user attributes used for prediction tasks are as follows:

- Gender: The user's binary gender, where the gender is {"f", "m"}, "f" represents female, and "m" represents male.
- Income: The user's categorical income level, where the income is in $\{0, 1, \ldots, 9\}$. Larger values represent a higher annual income and 0 represents unknown.
- Age: The user's age according to their registered birthday.

2.2 Cola-GNN: Cross-location Attention based Graph Neural Networks

Deng et al. [8] made use of the following datasets as shown in Table 1 for their experiments: The Infectious Disease Weekly Report (IDWR) in Japan to obtain Japan-prefectures data, the Center for Disease Control (CDC) in the United States to obtain influenza data about the US-states, and the ILINet portion of the United States Department of Health and Human Services (US-HHS) to obtain data about the US-region.

2.3 Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks

Kapoor et al. [15] took advantage of the following datasets for the modeling: The New York Times (NYT) COVID-19 dataset for common node features such as day, past cases, and past deaths; the Google COVID-19 Aggregated Mobility Research Dataset to obtain inter-county flows and intra-county flows to establish the graph neural network; and the Google Community Mobility Reports to obtain summarized mobility trends at places which are aggregated at the county level.

2.4 Traffic Flow Prediction via Spatial Temporal Graph Neural Network

Wang et al. [30] tested the framework on two real-world traffic datasets as seen Table 2:

Dataset	Sensors	Length	Unit	Size
META-LA	207	4 month	5 min	34,272
PEMS-BAY	325	6 month	5 min	52,116

Table 2: "Dataset statistics in terms of sensors, length, unit and size [30].

- METR-LA: A traffic dataset which is centered around the LA county road network [14] and includes high-resolution spatio-temporal transportation data. Loop-detectors in the network supply traffic speed or volume data as well.
- PEMS-BAY: A traffic dataset which is supplied by the California Department of Transportation (Caltrans) Performance Measurement System (PeMS) [4] to measure traffic in the Bay Area.

3 GNN Methods

In this section we will describe the networks of several spatiotemporal GNN models.

3.1 CalendarGNN: Calender Graph Neural Networks

Wang et al. have developed the CalendarGNN model [28] which predicts specific user features (e.g. binary gender, income and age) based on spatiotemporal behavior data. Various methods exist [7, 13, 16, 19, 26] which aim to predict features using merely temporal (sequential) data, however, the authors state that user behavior often follows a spatiotemporal pattern which can be modeled and used to generate more accurate predictions than the existing methods. The prediction of user behavior has applications in content recommendation and targeted advertising.

3.1.1 The CalendarGNN network

The CalendarGNN model takes as input a network consisting of three sections: locations, timestamps, and items (e.g. reading a news article, clicking on a website, posting a message on social media). Depending on the type of data, the raw features are transformed using a Multilayer Perceptron (i.e. a variation on the original perceptron [23] which uses multiple layers) or Bidirectional Long Short-Term Memory [25] encoder into dense hidden representations. These representations are subsequently concatenated into a vector. The set of all item and location vectors and then embedded in their respective layers.

The item embeddings are then aggregated together with the temporal data, resulting in session embeddings, which in turn are aggregated into separate embeddings for the hour, week, and weekday time units. The session embeddings are also combined with the location embeddings. The embeddings per time unit and location are then aggregated into patterns embeddings, which are fused (by concatenation) into a final user embedding. This user embedding is the input for a dense layer resulting in a prediction. Note that the authors chose hour, week, and weekday time units based on their data and application, however other time units can be used as well.

The aggregation layers use the Gated Recurrent Unit (GRU) [5] for the aggregation function, and a non-linear activation function, e.g. ReLU [21]. The temporal aggregation layer partitions the continuous timestamps of the sessions into discrete time units, aggregate sessions of the same time unit, and aggregate time unit embeddings into a temporal pattern embedding. For the mathematical definitions of each operation, please consult the original paper.

The CalendarGNN model has two limitations. First, the different time units (i.e. week, hour, weekday) are all considered equally important in the aggregation step into patterns, while this may not reflect the input data. Second, the spatial and temporal data are processed separately, while the relation between the two should be captured by the model as well to achieve true modeling of spatiotemporal patterns. To achieve this, the authors propose a variation on the CalendarGNN model called CalendarGNN-Atnn [28]. This model is identical to CalendarGNN, except that it uses pattern definitions which allow for location-time interactions.

3.2 Cola-GNN: Cross-location Attention based Graph Neural Networks

The forecasting of influenza-like illnesses (ILI) is of high importance to epidemiologists in terms of resource allocation and the intervention of outbreaks. Deng et al. [8] focused on the long-term forecasting of ILI by using influenza surveillance data from several locations. It was difficult to accurately forecast long-term epidemics due to limited accountability of short-term input data and the change in the influence of other locations on any location. Deng et al. [8] aimed to construct a long-term prediction of the spread of ILI by accounting for a limited time range of data with deep spatial representations inside a graph propagated model. They implemented a graph neural network framework to model epidemic propagation at the level of the population. Furthermore, they explored capturing sequential dependencies in local time-series data through recurrent neural networks; and identify short- and long-term patterns through dilated temporal convolutions [8].

The framework, as shown in Figure 2, is comprised of: location-wise interactions (node attributes) to be caught through location-aware attention, short-term and long-term local temporal dependencies (node attributes) to be caught through dilated convolution layer, and the combination of the temporal features and the location-aware attentions through global graph message passing to make forecasts on newly learned hidden location embeddings [8].

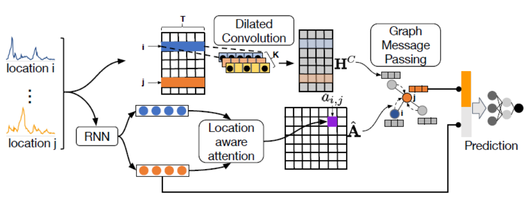


Fig. 2: The Cola-GNN framework [8].

3.2.1 Direct spatio influence learning

Deng et al. [8] constructed a dynamic model to assess the impact ILI in one location can have on the ILI of another location. They first utilized a Recurrent Neural Network (RNN) to learn the hidden states of each location from a certain period. Then, using the data from the RNN, they defined a general attention coefficient to measure to what extend two locations impact each other [8]. Finally, they include the spatial distance between the two locations in their calculations. The feature fusion gate is dynamically learned, and then models the influence that two locations have on each other by weighing the geographic and historic information [8].

3.2.2 Multi-scale dilated convolution

Convolutional Neural Networks (CNN) have proven very accurate in determining grid data, sequence data, and other local patterns. Deng et al. [8] aimed to use CNN for graph message passing. Hence, they adopted a multi-scale dilated convolutional module consisting of multiple parallel convolutional layers with different dilation rates, but the same filter and stride size. They then used multiple filters to produce different filter vectors [8]. They proceeded to concatenate these filter vectors to get the final convolution output. This output encodes local patterns into short-term and long-term trends [8].

3.2.3 Graph message passing

Using Graph Neural Networks (GNN), they designed a flu propagation model. They modeled ILI propagation among different locations, where each location is a node in a graph [8]. In their calculations, the dilated convolved features are used instead of the original time series since multiple levels of granularity can be captured in the hidden temporal features. Using the RNN hidden states and the graph features, they send their combination to the output layer to obtain their prediction [8].

3.3 Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks

During the COVID-19 pandemic, being able to accurately forecast caseload is highly necessary for numerous reasons, such as controlling outbreaks. Currently, two approaches of modeling COVID-19 outbreak are most commonly used: the mechanistic approach, and the time series learning approach [15]. Both approaches usually only depend on information from a single location or nearby locations where a pattern emerged, in forecasting for that location. Utilizing the widespread use of GPS-enabled mobile devices, Kapoor et al. [15] believe that they can build a better model by using more accurate real-time data and developing an approach that combines both the above approaches. They proposed a spatio-temporal graph neural network that uses precise mobility data to forecast daily new COVID-19 cases. The key discernment of the GNN-model is that the input node's signal transformation can be associated with the information propagation of a node's neighbors. This serves to better notify the future hidden state of the original input. The messaging framework designed by Gilmer et al. [10] is a great example of this. The messages are first propagated at the neighboring nodes and then aggregated to obtain new representations [15].

3.3.1 Modeling the COVID-19 graph

Multiple time-series sequences are most often used in the modeling of infectious diseases. However, this method does not take the human mobility across locations into account. Kapoor et al. [15] created a graph with different edge types to model both spatial and temporal dependencies. The edges in the spatial domain represent inter-location movement and are weighted by normalizing the mobility flow against the intra-flow. In Figure 3, edges in the temporal domain represent connections to the past days [15].

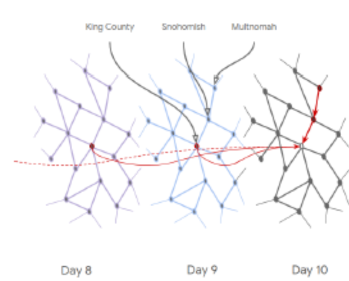


Fig. 3: "A slice of the COVID-19 graph showing spatial and temporal edges (highlighted in red) across three days." [15].

3.3.2 Skip-Connections Model

Concerning graph convolutions, Kapoor et al. [15] integrated skip-connections between layers in the spectral graph convolution model designed by Kipf and Welling [16] to avoid diluting the self-node future state, represented in Figure 4. A learned embedding from the temporal node features is concatenated to the output of each layer [15].

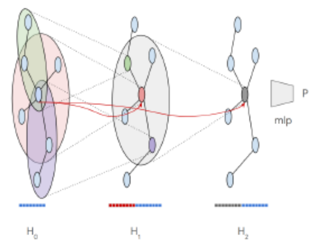


Fig. 4: 2-hop Skip-Connection model [15].

3.4 Traffic Flow Prediction via Spatial Temporal Graph Neural Network

The analysis and predictability of dynamic traffic conditions are of key importance in the planning and construction of roads and future city expansion. The problem lies in the increasing difficulty of traffic flow predictability [30]. This is due to the volatility of vehicle flow in the temporal dimension in the short-term, as well as the complex relationship between the vehicles and the roads in the spatial dimension. The inclusion of road crossings and vehicle lanes, which come with increased complexity, further decreases the predictability of traffic. A time series can be implemented on a road network to represent traffic data, where the spatial proximity of separate roads can be used to connect them [30].

Wang et al. [30] propose a new Graph Neural Network layer with a position-wise attention mechanism such that the traffic flow from connected roads can be better aggregated. Local and global temporal dependence is captured using the combination of a recurrent network and a Transformer layer. A new Spatial-temporal Graph Neural network (STGNN) is specifically designed to model series data with topological and temporal dependency. This new framework is finally tested on real traffic datasets to obtain results about short-term traffic speed predictions [30]. The experiments prove the proposed model is significantly better than other previously used methods. They plan to predict future traffic flow by utilizing historical traffic flow data. This data can be represented on a traffic network of connected traffic sensor nodes with proximity weighted edges [30].

Figure 5 illustrates the proposed spatial-temporal Graph Neural Network framework. There are three main components to the framework: Spatial Graph Neural Network (S-GNN) [30] layers that use the traffic network to represent the spatial relations between different roads, the Gated Recurrent Unit (GRU) layer which serves to represent the temporal relation sequentially, and the Transformer layer which serves to directly represent the long-term temporal dependence on the sequence. The S-GNN layer models the spatial relation between the nodes. As in Figure 5, the S-GNN layer is applied to both the input and the hidden representations of the GRU. Although they represent different perspectives, both the GRU layer and Transformer layer represent the temporal dependency of each node individually [30].

4 Evaluation Metrics

In the experiments, the following evaluation metrics are used to assess the performance of the methods:

- *Pearson's correlation coefficient (PCC)*: Measures the linear dependence between two variables [8].

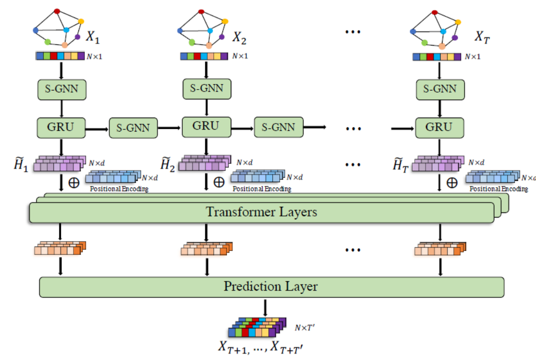


Fig. 5: Spatial Temporal Graph Neural Network Framework [30].

$$PCC = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

- *Root Mean Squared Error (RMSE)*: Measures the difference between two values (true values and the predicted values) after the projection of normalized values to the real range [8].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{2}$$

- *Root Mean Squared Logarithmic Error (RMSLE)*: Measures the difference between the logarithms of two values (the true values and the predicted values) after the projection of normalized values to the real ones [9].

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) - \log(\hat{y}_i + 1))^2} \tag{3}$$

- *Mean Absolute Error (MAE)*: Measures the absolute difference of two continuous variables [30].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4}$$

- *Mean Absolute Percentage Error (MAPE)*: Measures the absolute difference divided by true value of two continuous variables (the true value and the predicted value) [30].

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}| \tag{5}$$

- *R-squared (R^2)*: Represents the proportion of the variance for a dependent variable which is explained by an independent variable [6].

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{6}$$

Here, we list the surveyed models and the metrics used:

- CalendarGNN: Calender Graph Neural Network: *PCC, RMSE, MAE, R^2*.
- Cola-GNN: Cross-location Attention based Graph Neural Networks: *PCC, RMSE, MAE.*

- Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks: *PCC, RMSE, RMSLE.*
- Traffic Flow Prediction via Spatial Temporal Graph Neural Network: *MAE, MAPE.*

5 Findings

5.1 CalendarGNN: Calender Graph Neural Networks

The CalendarGNN and CalendarGNN-Attn models have been tested using large-scale user behavior logs from two real portal websites providing news updates and articles on various topics. The models were used to predict the user's binary gender, income and age. Compared against various baseline methods as well as logistic/linear regression (LR), LearnSuc [29], and SR-GNN [31], the proposed models outperform all other models based on nearly all different metrics used.

5.2 Cola-GNN: Cross-location Attention based Graph Neural Networks

The results of their methods are evaluated in terms of the Root Mean Square Error (RMSE) and Pearson's Correlation (PCC) metrics [8]. Leadtime represents the number of weeks that is predicted by the model in advance. They also showcase a relative performance advantage of their method to the best baseline model. The variance in their data is the cause of the large differences in RMSE values across the different datasets. Considering a relatively small leadtime window, their proposed method performs the best and the most stable for all the datasets. This is also true for most datasets if they consider a long lead time window [8]. The performance results of Vector Autoregression (VAR) and RNN suggest the necessity to control model complexity when only insufficient data is available, as well as that long-term ILI forecasts require a better design to capture the spatial and temporal dependencies. All methods perform relatively equally well when a short leadtime window is used, but the simpler methods quickly degenerate into severe inaccuracy as the leadtime window is increased [8].

5.3 Examining COVID-19 Forecasting using Spatio-Temporal Graph Neural Networks

Kapoor et al. [15] represent the forecasting performance of the spatio-temporal GNN in comparison to some baseline models. The Root Mean Squared Log Error (RMSLE) and Pearson Correlation (PCC) [15] evaluation metrics are represented for the predicted caseload and the case deltas. The results are that the graph neural network surpasses the baselines and obtains the best score on nearly every evaluation metric. Furthermore, inserting further mobility data improves performance for all the deep models, yet weakens the performance of the Autoregressive Integrated Moving Average (ARIMA) baseline [15]. It is understood this to be due to ARIMA assuming fixed dynamics and a linear dependence on county-level mobility.

5.4 Traffic Flow Prediction via Spatial Temporal Graph Neural Network

Wang et al. [30] measure the forecasting performance of different methods using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) as metrics [30]. Since diverse traffic conditions can be expected from different areas, an absolute error could be useful to indicate where the model is overfitting to relatively simple samples. Meanwhile, in volatile areas, a square error is more scrutinizing and can therefore give better performance under complex situations [30].

A time-scale forecasting of 15, 30, and 60 minutes was averaged for the results. The STGNN proposed framework is superior to all other methods across all timescales, errors, and datasets. The superiority of STGNN was more profound in PEMS-BAY than on METR-LA [30]. Two variations of STGNN were tested as well; one without gated recurrent units and the other without transformer layers. These are also mostly superior to all the baseline methods across all timescales, errors, and datasets, and yet were still inferior to the STGNN with all the components present. This indicates how all components discussed are key to optimal results but are still significant even without GRU and Transformer [30].

6 Discussion

In the previous sections, we have described four models for pattern analysis on graphs in time series. Our observations are that this is indeed a relatively new field, with proposed models that are somewhat 'isolated', meaning that each model has a unique approach, application and is trained on a very specific dataset. We do not see any convergence towards one particular approach that performs the best, although we do observe some very general patterns.
We see that the proposed models:

- operate on spatiotemporal graphs,
- use frameworks with multiple sequential processing steps, and
- are composed of existing models, e.g. CNNs, GRUs, MLPs.

It is difficult to directly compare the models as the approaches (and thus the frameworks) are significantly different. Any differences in effectiveness cannot easily be attributed to any specific minor or major difference in the framework. In addition to this, the datasets used in the experiment are all different, as are the goals. Also note that all proposed models substantiate their results by performing an experiment on a very limited number of datasets. Analytical analysis, for example of time and space complexities, of the proposed models was largely absent.

When choosing a model with a particular application in mind, having domain knowledge will be very useful, if not essential, as the proposed models have been developed with a specific goal in mind, and may not perform well under different circumstances.

For more context, a broader look at the field is summarized in Table 3, listing the (abbreviated) model names, network types, datasets used and metrics. We see here, again, that most models use some form of convolution and that there is very little overlap in the datasets used.

7 Conclusion

In this literature review, we discussed how different Graph Neural Networks are used for analyzing time series data. All the proposed models are effective in producing results for the application they were designed for. However, lacking information, little can be said about which model is better in extracting patterns from spatiotemporal graph data in general. We concluded that since each method use different dataset and evaluation metrics, comparison between proposed methods is not trivial.

To reach definite conclusions about the effectiveness and generalization ability of the proposed models, more research is needed in which the models with the same goal are directly compared using the multiple identical datasets. In addition to this, more general variations of the models could be designed with the goal of making them suitable for a broader range of applications. These variations could then be directly compared in the general ability to extract patterns from spatiotemporal graph data, perhaps leading to more fundamentally relevant results. However, the models' respective usefulness related to their specific application is certainly valuable on a practical level.

In their survey of Deep Learning on Graphs, Zhang et al. [39] have categorized four possible future directions, which includes the study of dynamic graphs. The other directions are: new models for unstudied graph structures, compositionality of existing models, and interpretability and robustness. All three can be studied on dynamic graphs as well.

Acknowledgements

The authors wish to thank expert reviewer dr. Estefanía Talavera Martínez and reviewers Nitin Paul and Floris Westerman.

Model name(s)	Network type(s)	Data set(s)	Metric(s)
ASTGCN [12]	Convolutional, attention.	PeMSD4, PeMSD8.	RMSE, MAE.
CalendarGNN, CalendarGNN-Attn [28]	Attention, aggregation.	Self-collected.	Mean accuracy, AUC, F1, MCC, Cohen's kappa, R^2, MAE, RMSE, PCC.
Cola-GNN [8]	Convolutional, recurrent.	Japan-Prefectures, US-States, US-Regions.	RMSE, MAE, PCC, Leadtime.
STGNN [15]	SGC, message passing.	NYT COVID-19, Google COVID-19 AMR, Google CMR.	RMSLE, PCC.
STGNN [30]	Convolutional.	METR-LA, PEMS-BAY.	MAE, MAPE.
DMVST-Net [35]	Convolutional, LSTM.	Self-collected.	RMSE, MAPE.
DCRNN [18]	Convolutional, recurrent.	METR-LA, PEMS-BAY.	RMSE, MAE, MAPE.
WD-GCN, CD-GCN [19]	Convolutional, LSTM.	DBLP (subset), CAD-120.	Monte Carlo Cross-Validation, Wilcoxon test.
Graph WaveNet [32]	Convolutional.	METR-LA, PEMS-BAY.	MAE, RMSE, MAPE.
STDN, STDN-Graph [34]	Convolutional, FGN, PSAM.	STDN: NYC-Taxi, NYC-Bike. STDN-Graph: Self-collected.	RMSE, MAPE.
STGCN [33]	Convolutional.	Deepmind Kinetics human action dataset, OpenPose, NTU-RGB+D.	Accuracy percentage.
STGCN [37]	Convolutional, gated convolutional.	BJER4, PeMSD.	MAE, MAPE, RMSE.
StemGNN [3]	Spectral, spectral convolution.	METR-LA, PEMS-BAY, PEMS07, PEMS03, PEMS04, PEMS08, Solar, Electricity, ECG5000, COVID-19.	MAE, MAPE, RMSE.

Table 3: Overview of models

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[2] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs, 2014.

[3] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang. Spectral temporal graph neural network for multivariate time-series forecasting. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17766–17778. Curran Associates, Inc., 2020.

[4] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia. Freeway performance measurement system: Mining loop detector data. *Transportation Research Record*, 1748(1):96–102, 2001.

[5] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[6] A. Colin Cameron and F. A. Windmeijer. An r-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2):329–342, 1997.

[7] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering, 2017.

[8] S. Deng, S. Wang, H. Rangwala, L. Wang, and Y. Ning. Cola-gnn: Cross-location attention based graph neural networks for long-term ili prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 245–254, New York, NY, USA, 2020. Association for Computing Machinery.

[9] K. V. Desai and R. Ranjan. Insights from the wikipedia contest (ieee contest for data mining 2011), 2014.

[10] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry, 2017.

[11] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *AAAI*, 2019.

[12] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:922–929, 07 2019.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997.

[14] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, July 2014.

[15] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion. Examining covid-19 forecasting using spatio-temporal graph neural networks, 2020.

[16] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2017.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, May 2017.

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2018.

[19] F. Manessi, A. Rozza, and M. Manzo. Dynamic graph convolutional networks. *Pattern Recognition*, 97:107000, Jan 2020.

[20] S. Mohapatra, T. Swarnkar, and J. Das. 2 - deep convolutional neural network in medical image processing. In V. E. Balas, B. K. Mishra, and R. Kumar, editors, *Handbook of Deep Learning in Biomedical Engineering*, pages 25–60. Academic Press, 2021.

[21] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[22] H. Purwins, B. Li, T. Virtanen, J. Schluter, S.-Y. Chang, and T. Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, May 2019.

[23] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[24] L. Ruiz, F. Gama, and A. Ribeiro. Gated graph recurrent neural networks. *IEEE Transactions on Signal Processing*, 68:6303–6318, 2020.

[25] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[26] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.

[27] R. van den Berg, T. N. Kipf, and M. Welling. Graph convolutional matrix completion, 2017.

[28] D. Wang, M. Jiang, M. Syed, O. Conway, V. Juneja, S. Subramanian, and N. V. Chawla. Calendar graph neural networks for modeling time structures in spatiotemporal user behaviors. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 2581–2589, New York, NY, USA, 2020. Association for Computing Machinery.

[29] D. Wang, M. Jiang, Q. Zeng, Z. Eberhart, and N. V. Chawla. Multi-type itemset embedding for learning behavior success. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2397–2406, 2018.

[30] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, WWW '20, page 1082–1092, New York, NY, USA, 2020. Association for Computing Machinery.

[31] C. Wu and M. Yan. Session-aware information embedding for e-commerce product recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 2379–2382, New York, NY, USA, 2017. Association for Computing Machinery.

[32] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang. Graph wavenet for deep spatial-temporal graph modeling, 2019.

[33] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition, 2018.

[34] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction, 2018.

[35] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li. Deep multi-view spatial-temporal network for taxi demand prediction, 2018.

[36] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, Jul 2018.

[37] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2018.

[38] C. Zang, P. Cui, and C. Faloutsos. Beyond sigmoids: The nettide model for social network growth, and its applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 2015–2024, New York, NY, USA, 2016. Association for Computing Machinery.

[39] Z. Zhang, P. Cui, and W. Zhu. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.