Review

Graph neural network for traffic forecasting: A survey

Weiwei Jiang [a,*], Jiayun Luo [b]

[a] *School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, 100876, China*
[b] *School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore*

ARTICLE INFO

ABSTRACT

Traffic forecasting is important for the success of intelligent transportation systems. Deep learning models, including convolution neural networks and recurrent neural networks, have been extensively applied in traffic forecasting problems to model spatial and temporal dependencies. In recent years, to model the graph structures in transportation systems as well as contextual information, graph neural networks have been introduced and have achieved state-of-the-art performance in a series of traffic forecasting problems. In this survey, we review the rapidly growing body of research using different graph neural networks, e.g. graph convolutional and graph attention networks, in various traffic forecasting problems, e.g. road traffic flow and speed forecasting, passenger flow forecasting in urban rail transit systems, and demand forecasting in ride-hailing platforms. We also present a comprehensive list of open data and source codes for each problem and identify future research directions. To the best of our knowledge, this paper is the first comprehensive survey that explores the application of graph neural networks for traffic forecasting problems. We have also created a public GitHub repository where the latest papers, open data, and source codes will be updated.

1. Introduction

Transportation systems are among the most important infrastructure in modern cities, supporting the daily commuting and traveling of millions of people. With rapid urbanization and population growth, transportation systems have become more complex. Modern transportation systems encompass road vehicles, rail transport, and various shared travel modes that have emerged in recent years, including on-line ride-hailing, bike-sharing, and e-scooter sharing. Expanding cities face many transportation-related problems, including air pollution and traffic congestion. Early intervention based on traffic forecasting is seen as the key to improving the efficiency of a transportation system and to alleviate transportation-related problems. In the development and operation of smart cities and intelligent transportation systems (ITSs), traffic states are detected by sensors (e.g. loop detectors) installed on roads, subway and bus system transaction records, traffic surveillance videos, and even smartphone GPS (Global Positioning System) data collected in a crowd-sourced fashion. Traffic forecasting is typically based on consideration of historical traffic state data, together with the external factors which affect traffic states, e.g. weather and holidays.

Both short-term and long-term traffic forecasting problems for various transport modes are considered in the literature. This survey focuses on the data-driven approach, which involves forecasting based on historical data. The traffic forecasting problem is more challenging than other time series forecasting problems because it involves large data volumes with high dimensionality, as well as multiple dynamics including emergency situations, e.g. traffic accidents. The traffic state in a specific location has both spatial dependency, which may not be affected only by nearby areas, and temporal dependency, which may be seasonal. Traditional linear time series models, e.g. auto-regressive and integrated moving average (ARIMA) models, cannot handle such spatiotemporal forecasting problems. Machine learning (ML) and deep learning techniques have been introduced in this area to improve forecasting accuracy, for example, by modeling the whole city as a grid and applying a convolutional neural network (CNN) as demonstrated by Jiang and Zhang (2018). However, the CNN-based approach is not optimal for traffic foresting problems that have a graph-based form, e.g. road networks.

In recent years, graph neural networks (GNNs) have become the frontier of deep learning research, showing state-of-the-art performance in various applications (Wu et al., 2020). GNNs are ideally suited to traffic forecasting problems because of their ability to capture spatial dependency, which is represented using non-Euclidean graph structures. For example, a road network is naturally a graph, with road intersections as the nodes and road connections as the edges. With graphs as the input, several GNN-based models have demonstrated superior performance to previous approaches on tasks including road traffic flow and speed forecasting problems. These include, for example,

---

* Corresponding author.
E-mail addresses: jww@bupt.edu.cn (W. Jiang), luoj0028@e.ntu.edu.sg (J. Luo).

the diffusion convolutional recurrent neural network (DCRNN) (Li, Yu, Shahabi and Liu, 2018) and Graph WaveNet (Wu, Pan, Long, Jiang, & Zhang, 2019) models. The GNN-based approach has also been extended to other transportation modes, utilizing various graph formulations and models.

To the best of the authors' knowledge, this paper presents the first comprehensive literature survey of GNN-related approaches to traffic forecasting problems. While several relevant traffic forecasting surveys exist (Boukerche, Tao, & Sun, 2020; Boukerche & Wang, 2020a; Fan et al., 2020; George & Santra, 2020; Haghighat et al., 2020; Lee, Eo, Jung, Yoon, & Rhee, 2021; Luca, Barlacchi, Lepri, & Pappalardo, 2020; Manibardo, Laña, & Del Ser, 2021; Pavlyuk, 2019; Shi & Yeung, 2018; Tedjopurnomo, Bao, Zheng, Choudhury, & Qin, 2020; Varghese, Chikaraishi, & Urata, 2020; Xie et al., 2020; Ye, Zhao, Ye, & Xu, 2020a; Yin et al., 2021), most of them are not GNN-focused with only one exception (Ye et al., 2020a). For this survey, we reviewed 212 papers published in the years 2018 to 2020. Additionally, because this is a very rapidly developing research field, we also included preprints that have not yet gone through the traditional peer review process (e.g., arXiv papers) to present the latest progress. Based on these studies, we identify the most frequently considered problems, graph formulations, and models. We also investigate and summarize publicly available useful resources, including datasets, software, and open-sourced code, for GNN-based traffic forecasting research and application. Lastly, we identify the challenges and future directions of applying GNNs to the traffic forecasting problem.

Instead of giving a whole picture of traffic forecasting, our aim is to provide a comprehensive summary of GNN-based solutions. This paper is useful for both the new researchers in this field who want to catch up with the progress of applying GNNs and the experienced researchers who are not familiar with these latest graph-based solutions. In addition to this paper, we have created an open GitHub repository on this topic,[1] where relevant content will be updated continuously.

Our contributions are summarized as follows:

(1) *Comprehensive Review*: We present the most comprehensive review of graph-based solutions for traffic forecasting problems in the past three years (2018–2020).

(2) *Resource Collection*: We provide the latest comprehensive list of open datasets and code resources for replication and comparison of GNNs in future work.

(3) *Future Directions*: We discuss several challenges and potential future directions for researchers in this field, when using GNNs for traffic forecasting problems.

The remainder of this paper is organized as follows. In Section 2, we compare our work with other relevant research surveys. In Section 3, we categorize the traffic forecasting problems that are involved with GNN-based models. In Section 4.1, we summarize the graphs and GNNs used in the reviewed studies. In Section 5, we outline the open resources. Finally, in Section 6, we point out challenges and future directions.

2. Related research surveys

In this section, we introduce the most recent relevant research surveys (most of which were published in 2020). The differences between our study and these existing surveys are pointed out when appropriate. We start with the surveys addressing wider ITS topics, followed by those focusing on traffic prediction problems and GNN application in particular.

Besides traffic forecasting, machine learning and deep learning methods have been widely used in ITSs as discussed in Fan et al. (2020), Haghighat et al. (2020) and Luca et al. (2020). In Haghighat et al. (2020), GNNs are only mentioned in the task of traffic characteristics

---

prediction. Among the major milestones of deep-learning driven traffic prediction (summarized in Figure 2 of Fan et al. (2020)), the state-of-the-art models after 2019 are all based on GNNs, indicating that GNNs are indeed the frontier of deep learning-based traffic prediction research.

Roughly speaking, five different types of traffic prediction methods are identified and categorized in previous surveys (George & Santra, 2020; Xie, Li et al., 2020), namely, statistics-based methods, traditional machine learning methods, deep learning-based methods, reinforcement learning-based methods, and transfer learning-based methods. Some comparisons between different categories have been considered, e.g., statistics-based models have better model interpretability, whereas ML-based models are more flexible as discussed in Boukerche et al. (2020). Machine learning models for traffic prediction are further categorized in Boukerche and Wang (2020a), which include the regression model, example-based models (e.g., k-nearest neighbors), kernel-based models (e.g. support vector machine and radial basis function), neural network models, and hybrid models. Deep learning models are further categorized into five different generations in Lee et al. (2021), in which GCNs are classified as the fourth generation and other advanced techniques that have been considered but are not yet widely applied are merged into the fifth generation. These include transfer learning, meta learning, reinforcement learning, and the attention mechanism. Before these advanced techniques become mature in traffic prediction tasks, GNNs remain the state-of-the-art technique.

Some of the relevant surveys only focus on the progress of deep learning-based methods (Tedjopurnomo et al., 2020), while the others prefer to compare them with the statistics-based and machine learning methods (Manibardo et al., 2021; Yin et al., 2021). In Tedjopurnomo et al. (2020), 37 deep neural networks for traffic prediction are reviewed, categorized, and discussed. The authors conclude that encoder–decoder long short term-memory (LSTM) combined with graph-based methods is the state-of-the-art prediction technique. A detailed explanation of various data types and popular deep neural network architectures is also provided, along with challenges and future directions for traffic prediction. Conversely, it is found that deep learning is not always the best modeling technique in practical applications, where linear models and machine learning techniques with less computational complexity can sometimes be preferable (Manibardo et al., 2021).

Additional research surveys consider aspects other than model selection. In Pavlyuk (2019), spatiotemporal feature selection and extraction pre-processing methods, which may also be embedded as internal model processes, are reviewed. A meta-analysis of prediction accuracy when applying deep learning methods to transport studies is given in Varghese et al. (2020). In this study, apart from the models themselves, additional factors including sample size and prediction time horizon are shown to have a significant influence on prediction accuracy.

To the authors' best knowledge, there are no existing surveys focusing on the application of GNNs for traffic forecasting. Graph-based deep learning architectures are reviewed in Ye et al. (2020a), for a series of traffic applications, namely, traffic congestion, travel demand, transportation safety, traffic surveillance, and autonomous driving. Specific and practical guidance for constructing graphs in these applications is provided. The advantages and disadvantages of both GNNs and other deep learning models, e.g. recurrent neural network (RNN), temporal convolutional network (TCN), Seq2Seq, and generative adversarial network (GAN), are examined. While the focus is not limited to traffic prediction problems, the graph construction process is universal in the traffic domain when GNNs are involved.

3. Problems

In this section, we discuss and categorize the different types of traffic forecasting problems considered in the literature. Problems are first categorized by the traffic state to be predicted. Traffic flow, speed,

and demand problems are considered separately while the remaining types are grouped together under "other problems". Then, the problem-types are further broken down into levels according to where the traffic states are defined. These include road-level, region-level, and station-level categories.

Different problem types have different modeling requirements for representing spatial dependency. For the road-level problems, the traffic data are usually collected from sensors, which are associated with specific road segments, or GPS trajectory data, which are also mapped into the road network with map matching techniques. In this case, the road network topology can be seen as the graph to use, which may contain hundreds or thousands of road segments potentially. The spatial dependency may be described by the road network connectivity or spatial proximity. For the station-level problems, the metro or bus station topology can be taken as the graph to use, which may contain tens or hundreds of stations potentially. The spatial dependency may be described by the metro lines or bus routes. For the region-level problem, the regular or irregular regions are used as the nodes in a graph. The spatial dependency between different regions can be extracted from the land use purposes, e.g., from the points-of-interest data.

A full list of the traffic forecasting problems considered in the surveyed studies is shown in Table 1. Instead of giving the whole picture of traffic forecasting research, only those problems with GNN-based solutions in the literature are listed in Table 1.

Generally speaking, traffic forecasting problems are challenging, not only for the complex temporal dependency, but only for the complex spatial dependency. While many solutions have been proposed for dealing with the time dependency, e.g., recurrent neural networks and temporal convolutional networks, the problem to capture and model the spatial dependency has not been fully solved. The spatial dependency, which refers to the complex and nonlinear relationship between the traffic state in one particular location with other locations. This location could be a road intersection, a subway station, or a city region. The spatial dependency may not be local, e.g., the traffic state may not only be affected by nearby areas, but also those which are far away in the spatial range but connected by a fast transportation tool. The graphs are necessary to capture such kind of spatial information as we would discuss in the next section.

Before the usage of graph theories and GNNs, the spatial information is usually extracted by multivariate time series models or CNNs. Within a multivariate time series model, e.g., vector autoregression, the traffic states collected in different locations or regions are combined together as multivariate time series. However, the multivariate time series models can only extract the linear relationship among different states, which is not enough for modeling the complex and nonlinear spatial dependency. CNNs take a step further by modeling the local spatial information, e.g., the whole spatial range is divided into regular grids as the two-dimensional image format and the convolution operation is performed in the neighbor grids. However, the CNN-based approach is bounded to the case of Euclidean structure data, which cannot model the topological structure of the subway network or the road network.

Graph neural networks bring new opportunities for solving traffic forecasting problems, because of their strong learning ability to capture the spatial information hidden in the non-Euclidean structure data, which are frequently seen in the traffic domain. Based on graph theories, both nodes and edges have their own attributes, which can be used further in the convolution or aggregation operations. These attributes describe different traffic states, e.g., volume, speed, lane numbers, road level, etc. For the dynamic spatial dependency, dynamic graphs can be learned from the data automatically. For the case of hierarchical traffic problems, the concepts of super-graphs and sub-graphs can be defined and further used.

3.1. Traffic flow

Traffic flow is defined as the number of vehicles passing through a spatial unit, such as a road segment or traffic sensor point, in a given time period. An accurate traffic flow prediction is beneficial for a variety of applications, e.g., traffic congestion control, traffic light control, vehicular cloud, etc. (Boukerche & Wang, 2020a). For example, traffic light control can reduce vehicle staying time at the road intersections, optimizing the traffic flow, and reducing traffic congestion and vehicle emission.

We consider three levels of traffic flow problems in this survey, namely, road-level flow, region-level flow, and station-level flow.

Road-level flow problems are concerned with traffic volumes on a road and include *road traffic flow, road origin–destination (OD) Flow*, and *intersection traffic throughput*. In road traffic flow problems, the prediction target is the traffic volume that passes a road sensor or a specific location along the road within a certain time period (e.g. five minutes). In the road OD flow problem, the target is the volume between one location (the origin) and another (the destination) at a single point in time. The intersection traffic throughput problem considers the volume of traffic moving through an intersection.

Region-level flow problems consider traffic volume in a region. A city may be divided into regular regions (where the partitioning is grid-based) or irregular regions (e.g. road-based or zip-code-based partitions). These problems are classified by transport mode into *regional taxi flow, regional bike flow, regional ride-hailing flow, regional dockless e-scooter flow, regional OD taxi flow, regional OD bike flow*, and *regional OD ride-hailing flow* problems.

Station-level flow problems relate to the traffic volume measured at a physical station, for example, a subway or bus station. These problems are divided by station type into *station-level subway passenger flow, station-level bus passenger flow, station-level shared vehicle flow, station-level bike flow*, and *station-level railway passenger flow* problems.

Road-level traffic flow problems are further divided into cases of unidirectional and bidirectional traffic flow, whereas region-level and station-level traffic flow problems are further divided into the cases of inflow and outflow, based on different problem formulations.

While traffic sensors have been successfully used, data collection for traffic flow information is still a challenge when considering the high costs in deployment and maintenance of traffic sensors. Another potential approach is using the pervasive mobile and IoT devices, which have a lower cost generally, e.g., GPS sensors. However, challenges still exist when considering the data quality problems frequently seen in GPS data, e.g., missing data caused by unstable communication links.

The traffic light is another source of challenges for various traffic prediction tasks. Short-term traffic flow fluctuation and the spatial relation change between two road segments can be caused by the traffic light. The way of controlling the traffic light may be different in different time periods, causing an inconsistent traffic flow pattern.

3.2. Traffic speed

Traffic speed is another important indicator of traffic state with potential applications in ITS systems, which is defined as the average speed of vehicles passing through a spatial unit in a given time period. The speed value on the urban road can reflect the crowdedness level of road traffic. For example, Google Maps visualizes this crowdedness level from crowd-sourcing data collected from individual mobile devices and in-vehicle sensors. A better traffic speed prediction is also useful for route navigation and estimation-of-arrival applications.

We consider two levels of traffic speed problems in this survey, namely, road-level and region-level problems. We also include travel time and congestion predictions in this category because they are closely correlated to traffic speed. Travel time prediction is useful for passengers to plan their commuting time and for drivers to select fast routes, respectively. Traffic congestion is one of the most important and

Table 1
Traffic forecasting problems in the surveyed studies.

Problem	Relevant Studies
Road traffic flow	Bai, Yao, Li, Wang, and Wang (2020), Boukerche and Wang (2020b), Cao et al. (2020), Chen, Han, Yin and Cao (2020), Chen, Liao, Hou, Wang and Wen (2020), Chen et al. (2020), Fang, Pan, Xiang and Pan (2020), Fang, Zhang, Meng, Xiang, and Pan (2019), Fukuda, Uchida, Fujii, and Yamada (2020), Guo et al. (2020, 2020), Guo, Lin, Feng, Song and Wan (2019), Huang, Huang, Liu, Dai and Kong (2020), Huang, Zhang, Wen and Chen (2020), Kang, Xu, Hu, and Pei (2019), Kong et al. (2020), Li, Wang, Zhang and Wu (2020), Li et al. (2019, 2020), Li and Zhu (2021), Lu, Zhang, Liu and Xiong (2019), Lv et al. (2020), Mallick, Balaprakash, Rask, and Macfarlane (2020), Pan et al. (2019, 2020), Qu, Zhu, Zang, Xu, and Yu (2020), Song, Lin, Guo and Wan (2020), Sun, Li, Lv and Dong (2020), Tang, Sun, Sun, Peng and Gan (2020), Tian, Guo, Ye, and Xu (2020), Wang et al. (2018, 2020), Wang, Xu, Liu, Zhou and Zhao (2020), Wang, Zhang, Wang and Chen (2020), Wei and Sheng (2020), Wei et al. (2019), Wu, Chen and Wan (2018), Xie, Xiong, and Zhu (2020b), Xin, Miao, Zhu, Jin, and Lu (2020), Xu et al. (2020, 2019), Xu, Wei, Peng, Xuan and Guo (2020), Xu, Zheng, Feng and Chen (2020), Yin et al. (2020), Yu, Yin, and Zhu (2018), Yu, Yin and Zhu (2019), Zhang, Chang, Meng, Xiang and Pan (2020), Zhang, Cheng and Ren (2019), Zhang, Cheng, Ren and Xie (2020), Zhang and Guo (2020), Zhang, Huang, Xu and Xia (2020), Zhang, Jin, Chang, Xiang and Pan (2018), Zhang, Liu, Tang and Xiong (2020), Zhang, Lu and Li (2020) and Zheng, Fan, Wang and Qi (2020)
Road OD flow	Ramadan, Elbery, Zorba, and Hassanein (2020) and Xiong, Ozbay, Jin, and Feng (2020)
Intersection traffic throughput	Sánchez et al. (2020)
Regional taxi flow	Chen et al. (2020), Peng et al. (2020), Qiu et al. (2020), Sun et al. (2020), Wang et al. (2018), Wang, Miao, Chen and Huang (2020), Zhou et al. (2020) and Zhou, Shen, and Huang (2019)
Regional bike flow	Sun, Zhang et al. (2020), Wang, Luo et al. (2018), Wang, Miao et al. (2020) and Zhou, Gu et al. (2020)
Regional ride-hailing flow	Zhou et al. (2019)
Regional dockless e-scooter flow	He and Shin (2020a)
Regional OD taxi flow	Wang, Miao et al. (2020) and Yeghikyan, Opolka, Nanni, Lepri, and Liò (2020)
Regional OD bike flow	Wang, Miao et al. (2020)
Regional OD ride-hailing flow	Shi et al. (2020), Wang, Xu, Peng, Xuan and Zhang (2020) and Wang et al. (2019)
Station-level subway passenger flow	Fang, Pan et al. (2020), Fang et al. (2019), Han et al. (2019), Li et al. (2018), Li, Sergin, Yan, Zhang and Tsung (2020), Liu et al. (2020), Ou et al. (2020), Peng et al. (2020), Ren and Xie (2019), Ye, Zhao, Ye, and Xu (2020b), Zhang, Chen, Cui, Guo and Zhu (2020), Zhang, Chen and Guo (2020) and Zhao, Gao, Liu, Zhao and Xu (2020)
Station-level bus passenger flow	Fang, Pan et al. (2020), Fang et al. (2019) and Peng et al. (2020)
Station-level shared vehicle flow	Zhu et al. (2019)
Station-level bike flow	Chai, Wang, and Yang (2018) and He and Shin (2020b)
Station-level railway passenger flow	He, Zhao, Wang, and Tsui (2020)
Road traffic speed	Agafonov (2020), Bai et al. (2021), Bing, Zhifeng, Yangjie, Jinxing, and Zhanwu (2020), Bogaerts, Masegosa, Angarita-Zapata, Onieva, and Hellinckx (2020), Cai, Janowicz, Mai, Yan, and Zhu (2020), Chen et al. (2020), Chen, Chen, Lai et al. (2020), Chen et al. (2020, 2019), Cirstea, Guo, and Yang (2019), Cui, Henrickson, Ke, and Wang (2019), Cui, Ke, Pu, Ma and Wang (2020), Cui, Lin, Pu and Wang (2020), Diao et al. (2019), Feng, Wu, Zhang, and Wu (2020), Fu, Zhou, and Chen (2020), Ge, Li, Liu, and Zhou (2019a, 2019b), Ge, Li, Wang, Chang, and Wu (2020), Guo, Hu, Qian, Liu et al. (2020), Guo, Hu, Qian, Sun et al. (2020), Guo et al. (2020), Guo, Song and Wang (2019), Guo and Yuan (2020), Guopeng, Knoop, and van Lint (2020), James (2019, 2020), Jepsen, Jensen, and Nielsen (2019, 2020), Jia, Wu, and Zhang (2020), Kim, Chung, and Kim (2020), Lee and Rhee (2019, 2022), Lewenfus, Martins, Chatzinotas, and Ottersten (2020), Li, Li, Peng and Tao (2020), Li, Xiong et al. (2020), Li, Yu et al. (2018), Li and Zhu (2021), Liao et al. (2018), Liu, Ong and Chen (2020), Lu, Gan, Jin, Fu and Zhang (2020), Lu et al. (2020), Lu, Lv, Xie, Du and Huang (2019), Lu, Zhang et al. (2019), Lv et al. (2020), Maas and Bloem (2020), Mallick et al. (2020), Mallick, Balaprakash, Rask, and Macfarlane (2021), Opolka et al. (2019), Oreshkin, Amini, Coyle, and Coates (2021), Pan et al. (2019, 2020), Park et al. (2020), Shin and Yoon (2020), Shleifer, McCreery, and Chitters (2019), Song, Ming, Hu, Niu and Gao (2020), Sun et al. (2021), Tang, Sun and Sun (2020), Wang, Guan, Cao, Zhang and Wu (2020), Wei et al. (2019), Wu et al. (2020, 2019), Xie et al. (2020, 2019), Xie, Xiong, and Zhu (2020a), Xu, Wei et al. (2020), Yang, Chen, Zhou, Gao, and Cao (2020), Yin et al. (2020), Yu and Gu (2019), Yu, Lee and Sohn (2020), Yu, Li, Zhang and Zhu (2019), Zhang, Cheng et al. (2020), Zhang, Guan, Cao, Wang and Wu (2019), Zhang, He, Zhang, Lin and Li (2020), Zhang, James and Liu (2019), Zhang, Jin et al. (2018), Zhang, Jin, Yang, Guo and Ma (2020), Zhang, Li, Lin, Wang and He (2019), Zhang et al. (2020), Zhang, Jin et al. (2018), Zhang, Wang, Chen and Cao (2019), Zhang, Zhang and Jin (2020), Zhao, Gao et al. (2020), Zhao et al. (2019), Zhao, Yang, Wang, Wang and Su (2020), Zheng, Fan et al. (2020), Zhou et al. (2020), Zhou, Yang, Zhong, Chen and Zhang (2020) and Zhu et al. (2022, 2021, 2020)
Road travel time	Fang et al. (2020), Guo, Hu, Qian, Liu et al. (2020), Hasanzadeh, Liu, Duffield, and Narayanan (2019), Shao, Wang, Chen, and Zhou (2020) and Shen, Jin, and Hua (2020)
Traffic congestion	Dai, Xu, Gu, Ji, and Liu (2020), Han, Shen, Yang, and Kong (2020), Mohanty and Pozdnukhov (2018), Mohanty, Pozdnukhov, and Cassidy (2020) and Qin, Xu, Kang and Kwan (2020)
Time of arrival	Hong et al. (2020)
Regional OD taxi speed	Hu, Guo, Yang, Jensen, and Chen (2018)
Ride-hailing demand	Bai et al. (2019), Bai, Yao, Kanhere, Wang and Sheng (2019), Geng et al. (2019, 2019), Jin et al. (2020), Jin, Xi, Sha, Feng and Huang (2020), Ke, Feng, Zhu, Yang and Ye (2021), Lee, Jung, Cheon, Kim, and You (2019), Li and Axhausen (2020), Li, Yang, Tang and Xia (2020) and Pian and Wu (2020)
Taxi demand	Bai, Yao, Kanhere, Wang, Liu et al. (2019), Bai, Yao, Kanhere, Wang and Sheng (2019), Chen, Zhao, Wang, Duan and Zhao (2020), Davis, Raina, and Jagannathan (2020), Du et al. (2020), Hu, Yang, Guo, Jensen, and Xiong (2020), Ke et al. (2021), Lee et al. (2019), Li and Moura (2020), Wu, Zhu and Chen (2020), Xu and Li (2019), Ye, Sun, Du, Fu, and Xiong (2021) and Zheng et al. (2020)
Shared vehicle demand	Luo et al. (2020)
Bike demand	Bai, Yao, Kanhere, Wang, Liu et al. (2019), Bai, Yao, Kanhere, Wang and Sheng (2019), Chen, Rossi, Mahadik and Eldardiry (2020), Du et al. (2020), Guo et al. (2019), Kim, Lee, and Sohn (2019), Lee et al. (2019), Lin, He, and Peeta (2018), Qin, Liu, Wu, Tong and Zhao (2020), Wang et al. (2020), Xiao, Wang, Zhang, and Ni (2020), Ye et al. (2021) and Yoshida et al. (2019)

Table 1 (*continued*).

Traffic accident	Yu et al. (2020), Zhang, Dong, Shang, Zhang and Wang (2020), Zhou, Wang, Xie, Chen and Liu (2020) and Zhou, Wang, Xie, Chen and Zhu (2020)
Traffic anomaly	Liu et al. (2020)
Parking availability	Yang, Ma, Pi, and Qian (2019), Zhang, Liu, Liu, Zhou and Xiong (2020) and Zhang et al. (2020)
Transportation resilience	Wang et al. (2020)
Urban vehicle emission	Xu, Kang, Cao and Li (2020)
Railway delay	Heglund, Taleongpong, Hu, and Tran (2020)
Lane occupancy	Wright, Ehlers, and Horowitz (2019)

urgent transportation problems in cities, which brings significant time loss, air pollution and energy waste. The congestion prediction results can be used to control the road conditions and optimize vehicle flow, e.g., with traffic signal control. In several studies, traffic congestion is judged by a threshold-based speed inference. The specific road-level speed problem categories considered are *road traffic speed, road travel time, traffic congestion,* and *time of arrival* problems; while the region-level speed problem considered is *regional OD taxi speed.*

Traffic speed is concerned in both urban roads and freeways. However, the challenges differ in these two different scenarios. Freeways have a few traffic signals or on/off-ramps, making the prediction easier than the urban case. And the challenge mainly comes from the complex temporal dependency. More complex traffic networks exist in urban roads with more complicated connection patterns and abrupt changes. For example, different road segments may have different speed limit values and the allowed vehicle types. Besides the complex temporal dependency, modeling the spatial dependency becomes a bigger challenge for urban traffic speed forecasting.

3.3. Traffic demand

Traffic demand prediction is a key component for taxi and ride-hailing services to be successful, which benefits these service providers to allocate limited available transportation resources to those urban areas with a higher demand. For passengers, traffic demand prediction encourages the consideration of various transportation forms, e.g., taking the public transit service when taxi or ride-hailing services are in short supply.

Traffic demand refers to the potential demand for travel, which may or may not be fulfilled completely. For example, on an online ride-hailing platform, the ride requests sent by passengers represent the demand, whereas only a subset of these requests may be served depending on the supply of drivers and vehicles, especially during rush hours. Accurate prediction of travel demand is a key element of vehicle scheduling systems (e.g. online ride-hailing or taxi dispatch platforms). However, in some cases, it is difficult to collect the potential travel demand from passengers and a compromise method using transaction records as an indication of the traffic demand is used. In such cases the real demand may be underestimated. Based on transport mode, the traffic demand problems considered include *ride-hailing demand, taxi demand, shared vehicle demand,* and *bike demand.*

3.4. Other problems

In addition to the above three categories of traffic forecasting problems, GNNs are also being applied to the following problems.

Traffic accident and *Traffic anomaly*: the target is to predict the traffic accident number reported to the police system. Traffic anomaly is the major cause of traffic delay and a timely detection and prediction would help the administrators to identify the situation and turn the traffic situation back to normal as quickly as possible. A traffic accident is usually an accident in road traffic involving different vehicles, which may cause significant loss of life and property. The traffic anomaly has a broader definition that deviates from the normal traffic state, e.g., the traffic jam caused by a traffic accident or a public procession.

Parking availability: the target is to predict the availability of vacant parking space for cars in the streets or in a car parking lot.

Urban vehicle emission: while not directly related to traffic states, the prediction of urban vehicle emission is considered in Xu, Kang et al. (2020). Urban vehicle emission refers to the emission produced by motor vehicles, e.g., those use internal combustion engines. Urban vehicle emission is a major source of air pollutants and its amount is affected by different traffic states, e.g., the excess emission would be created in traffic congestion situations.

Railway delay: the delay time of specific routes in the railway system is considered in Heglund et al. (2020).

Lane occupancy: With simulated traffic data, lane occupancy has been measured and predicted (Wright et al., 2019).

4. Graphs and graph neural networks

In this section, we summarize the types of graphs and GNNs used in the surveyed studies, focusing on GNNs that are frequently used for traffic forecasting problems. The contributions of this section include an organized approach for classifying the different traffic graphs based on the domain knowledge, and a summary of the common ways for constructing adjacency matrices, which may not be encountered in other neural networks before and would be very helpful for those who would like to use graph neural networks. The different GNN structures already used for traffic forecasting problems are briefly introduced in this section too. For a wider and deeper discussion of GNNs, refer to Wu, Pan, Chen et al. (2020), Zhang, Cui and Zhu (2020) and Zhou et al. (2020).

4.1. Traffic graphs

4.1.1. Graph construction

A graph is the basic structure used in GNNs. It is defined as $G = (V, E, A)$, where V is the set of vertices or nodes, E is the set of edges between the nodes, and A is the adjacency matrix. Both nodes and edges can be associated with different attributes in different GNN problems. Element a_{ij} of A represents the "edge weight" between nodes i and j. For a binary connection matrix A, $a_{ij} = 1$ if there is an edge between nodes i and j in E, and $a_{ij} = 0$ otherwise. If A is symmetric, the corresponding graph G is defined as undirected. Otherwise, G is directed, when the edge only exists in one direction between a node pair.

For simplicity, we assume that the traffic state is associated with the nodes. The other case with edges can be derived similarly. In practice, the traffic state is collected or aggregated in discrete time steps, e.g. five minutes or one hour, depending on the specific scenario.

For a single time step t, we denote the node feature matrix as $\chi_t \in R^{N \times d}$, where N is the number of nodes and d is the dimension of the node features, i.e., the number of traffic state variables. Now we are ready to give a formal definition of traffic graph.

Definition 4.1 (*Traffic Graph*). A traffic graph (with node features) is defined as a specific type of graph $G = (V, E, A)$, where V is the node set, E is the edge set, and A is the adjacency matrix. For a single time step t, the node feature matrix $\chi_t \in R^{N \times d}$ for G contains specific traffic states, where N is the number of nodes and d is the number of traffic state variables.
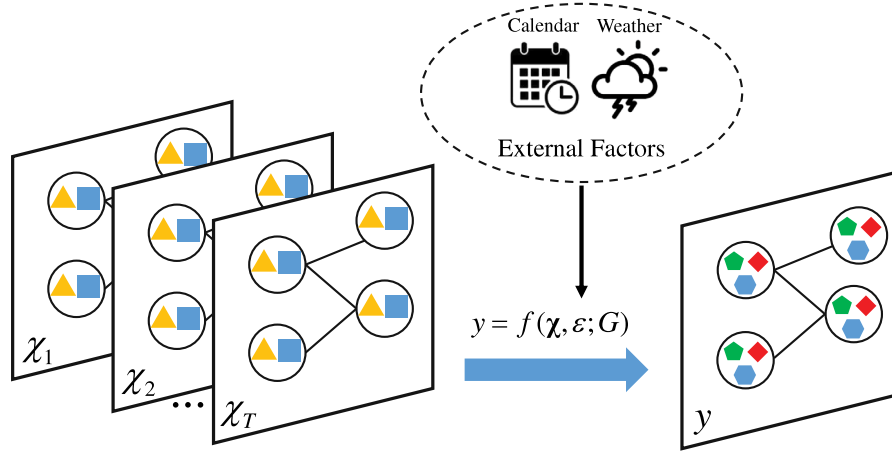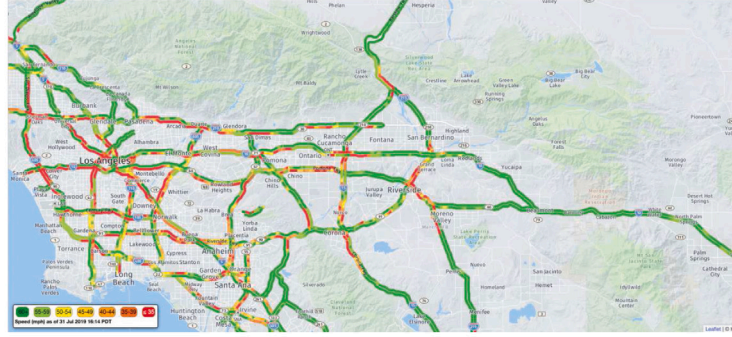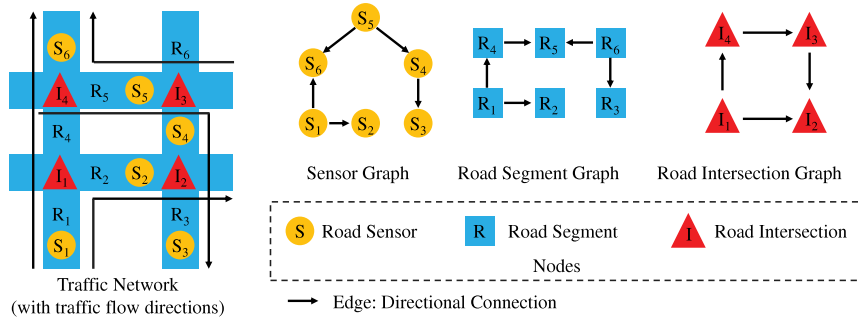
Fig. 1. The single-step graph-based traffic forecasting problem.
Source: Adapted from Ye et al. (2020a) with external factors added.

(a)

(b)

Fig. 2. The real-world case and example of road-level graphs. (a) The road network in the Performance Measurement System (PeMS) where each sensor is a node. Source: http://pems.dot.ca.gov/. (b) The road-level graph examples.
Source: Adapted from Ye et al. (2020a).

Then we give a formal definition of graph-based traffic forecasting problem without leveraging external factors firstly.

Definition 4.2 (*Graph-based Traffic Forecasting*). A graph-based traffic forecasting (without external factors) is defined as follows: find a function f which generates $y = f(\chi; G)$, where y is the traffic state to be predicted, $\chi = \{\chi_1, \chi_2, \ldots, \chi_T\}$ is the historical traffic state defined

on graph G, and T is the number of time steps in the historical window size.

In *single step forecasting*, the traffic state in the next time step only is predicted, whereas in *multiple step forecasting* the traffic state several time steps later is the prediction target. As mentioned in Section 1, traffic states can be highly affected by external factors, e.g. weather and holidays. The forecasting problem formulation, extended to incorporate
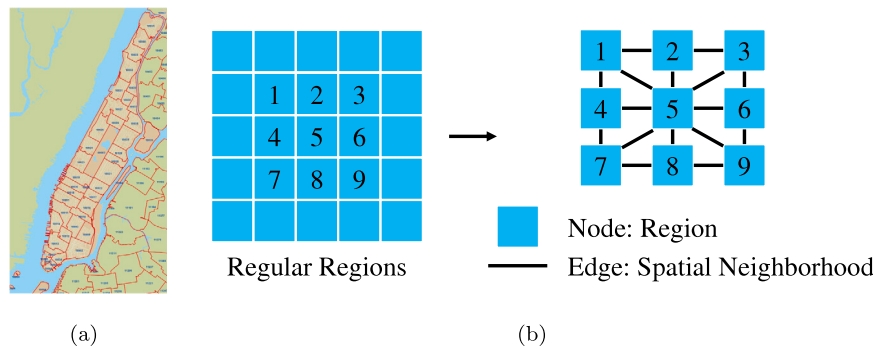
Fig. 3. The real-world case and example of region-level graphs. (a) The zip codes of Manhattan where each zip code zone is a node. Source: https://maps-manhattan.com/manhattan-zip-code-map. (b) The region-level graph example.

these external factors, takes the form $y = f(\chi, \varepsilon; G)$, where ε represents the external factors. Fig. 1 demonstrates the graph-based traffic forecasting problem, where different color patches represent different traffic variables.

Various graph structures are used to model traffic forecasting problems depending on both the forecasting problem-type and the traffic datasets available. These graphs can be pre-defined static graphs, or dynamic graphs continuously learned from the data. The static graphs can be divided into two types, namely, natural graphs and similarity graphs. Natural graphs are based on a real-world transportation system, e.g. the road network or subway system; whereas similarity graphs are based solely on the similarity between different node attributes where nodes may be virtual stations or regions.

We categorize the existing traffic graphs into the same three levels used in Section 3, namely, road-level, region-level and station-level graphs.

Road-level graphs. These include *sensor graphs, road segment graphs, road intersection graphs,* and *road lane graphs.* Sensor graphs are based on traffic sensor data (e.g. the PeMS dataset) where each sensor is a node, and the edges are road connections. The other three graphs are based on road networks with the nodes formed by road segments, road intersections, and road lanes, respectively. The real-world case and example of road-level graphs are shown in Fig. 2. In some cases, road-level graphs are the most suitable format, e.g., when vehicles can move only through pre-defined roads.

Region-level graphs. These include *irregular region graphs, regular region graphs,* and *OD graphs.* In both irregular and regular region graphs the nodes are regions of the city. Regular region graphs, which have grid-based partitioning, are listed separately because of their natural connection to previous widely used grid-based forecasting using CNNs, in which the grids may be seen as image pixels. Irregular region graphs include all other partitioning approaches, e.g. road based, or zip code based (Ke, Qin et al., 2021). In the OD graph, the nodes are origin region–destination region pairs. In these graphs, the edges are usually defined with a spatial neighborhood or other similarities, e.g., functional similarity derived from point-of-interests (PoI) data. The real-world case and example of region-level graphs are shown in Fig. 3.

Station-level graphs. These include *subway station graphs, bus station graphs, bike station graphs, railway station graphs, car-sharing station graphs, parking lot graphs,* and *parking block graphs.* Usually, there are natural links between stations that are used to define the edges, e.g. subway or railway lines, or the road network. The real-world case and example of station-level graphs are shown in Fig. 4.

A full list of the traffic graphs used in the surveyed studies is shown in Table 2. Sensor graphs and road segment graphs are most frequently used because they are compatible with the available public datasets as discussed in Section 5. It is noted that in some studies multiple graphs are used as simultaneous inputs and then fused to improve the forecasting performance (Lv et al., 2020; Zhu et al., 2019).

4.1.2. Adjacency matrix construction

Adjacency matrices are seen as the key to capturing spatial dependency in traffic forecasting (Ye et al., 2020a). While nodes may be fixed by physical constraints, the user typically has control over the design of the adjacency matrix, which can even be dynamically trained from continuously evolving data. We extend the categories of adjacency matrices used in previous studies (Ye et al., 2020a) and divide them into four types, namely, road-based, distance-based, similarity-based, and dynamic matrices.

Road-based Matrix. This type of adjacency matrix relates to the road network and includes connection matrices, transportation connectivity matrices, and direction matrices. A connection matrix is a common way of representing the connectivity between nodes. It has a binary format, with an element value of 1 if connected and 0 otherwise. The transportation connectivity matrix is used where two regions are geographically distant but conveniently reachable by motorway, highway, or subway (Ye et al., 2020a). It also includes cases where the connection is measured by travel time between different nodes, e.g. if a vehicle can travel between two intersections in less than 5 min then there is an edge between the two intersections (Wu, Chen et al., 2018). The less commonly used direction matrix takes the angle between road links into consideration.

Distance-based Matrix. This widely used matrix-type represents the spatial closeness between nodes. It contains two sub-types, namely, neighbor and distance matrices. In neighbor matrices, the element values are determined by whether the two regions share a common boundary (if connected the value is set to 1, generally, or 1/4 for grids, and 0 otherwise). In distance-based matrices, the element values are a function of geometrical distance between nodes. This distance may be calculated in various ways, e.g. the driving distance between two sensors, the shortest path length along the road (Kang et al., 2019; Lee & Rhee, 2022), or the proximity between locations calculated by the random walk with restart (RWR) algorithm (Zhang, Wang et al., 2019). One flaw of distance-based matrices is that the fail to take into account the similarity of traffic states between long-distance nodes, and the constructed adjacency matrix is static in most cases.

Similarity-based Matrix. This type of matrix is divided into two sub-types, namely, traffic pattern and functional similarity matrices. Traffic pattern similarity matrices represent the correlations between traffic states, e.g. similarities of flow patterns, mutual dependencies between different locations, and traffic demand correlation in different regions. Functional similarity matrices represent, for example, the distribution of different types of PoIs in different regions.

Dynamic Matrix. This type of matrix is used when no pre-defined static matrices are used. Many studies have demonstrated the advantages of using dynamic matrices, instead of a pre-defined adjacency matrix, for various traffic forecasting problems.

A full list of the adjacency matrices applied in the surveyed studies is shown in Table 3. Dynamic matrices are listed at the bottom of the table, with no further subdivisions. The connection and distance matrices are the most frequently used types, because of their simple definition and representation of spatial dependency.
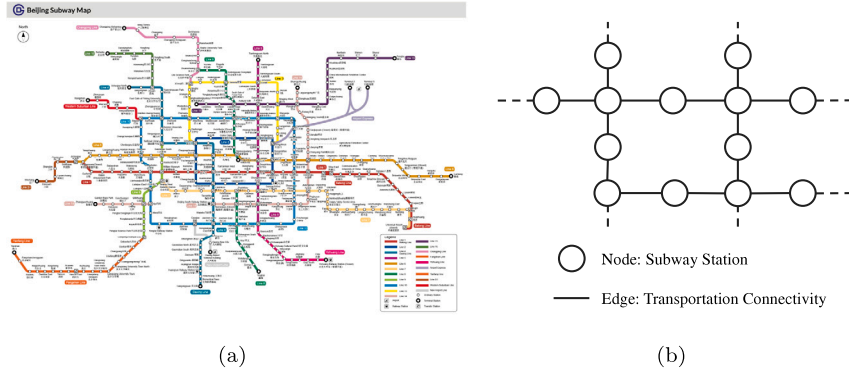
Fig. 4. The real-world case and example of station-level graphs. (a) The Beijing subway system where each subway station is a node. Source: https://www.travelchinaguide.com/cityguides/beijing/transportation/subway.htm. (b) The station-level graph example.

4.2. Graph neural networks

Previous neural networks, e.g. fully-connected neural networks (FNNs), CNNs, and RNNs, could only be applied to Euclidean data (i.e. images, text, and videos). As a type of neural network which directly operates on a graph structure, GNNs have the ability to capture complex relationships between objects and make inferences based on data described by graphs. GNNs have been proven effective in various node-level, edge-level, and graph-level prediction tasks (Jiang, 2022). As mentioned in Section 2, GNNs are currently considered the state-of-the-art techniques for traffic forecasting problems. GNNs can be roughly divided into four types, namely, recurrent GNNs, convolutional GNNs, graph autoencoders, and spatiotemporal GNNs (Wu, Pan, Chen et al., 2020). Because traffic forecasting is a spatiotemporal problem, the GNNs used in this field can all be categorized as the spatiotemporal GNNs. However, certain components of the other types of GNNs have also been applied in the surveyed traffic forecasting studies.

To give the mathematical formulation of GCN, we further introduce some notations. Give a graph $G = (V, E, A)$, $\mathcal{N}(v_i)$ is defined as the neighbor node set of a single node v_i. $\mathbf{D}$ is defined as the degree matrix, of which each element is $\mathbf{D}_{ii} = \|\mathcal{N}(v_i)\|$. $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is defined as the Laplacian matrix of an undirected graph and $\tilde{\mathbf{L}} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is defined as the normalized Laplacian matrix, where $\mathbf{I}_N$ is the identity matrix with size N. Without considering the time step index, the node feature matrix of a graph is simplified as $\mathbf{X} \in R^{N \times d}$, where N is the node number and d is the dimension of the node feature vector as before. The basic notations used in this survey is summarized in Table 4.

When extending the convolution operation from Euclidean data to non-Euclidean data, the basic idea of GNNs is to learn a function mapping for a node to aggregate its own features and the features of its neighbors to generate a new representation. GCNs are spectral-based convolutional GNNs, in which the graph convolutions are defined by introducing filters from graph signal processing in the spectral domain, e.g., the Fourier domain. The graph Fourier transform is firstly used to transform the graph signal to the spectral domain and the inverse graph Fourier transform is further used to transform the result after the convolution operation back. Several spectral-based GCNs are introduced in the literature. Spectral convoluted neural networking (Bruna, Zaremba, Szlam, & LeCun, 2014) assumes that the filter is a set of learnable parameters and considers graph signals with multiple channels. GNN (Henaff, Bruna, & LeCun, 2015) introduces a parameterization with smooth coefficients and makes the spectral filters spatially localized. Chebyshev's spectral CNN (ChebNet) (Defferrard, Bresson, & Vandergheynst, 2016) leverages a truncated expansion in terms of Chebyshev polynomials up to Kth order to approximate the diagonal matrix.

GCN (Kipf & Welling, 2017) is a first-order approximation of ChebNet, which approximates the filter using the Chebyshev polynomials

of the diagonal matrix of eigenvalues. To avoid overfitting, $K = 1$ is used in GCN. Formally, the graph convolution operation $*G$ in GCN is defined as follows:

$$\mathbf{X}_{*G} = \mathbf{W}(\mathbf{I}_N + \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}})\mathbf{X} \tag{1}$$

where $\mathbf{W}$ is a learnable weight matrix, i.e., the model parameters. While in practice, the graph convolution operation is further developed in order to alleviate the potential gradient explosion problem as follows:

$$\mathbf{X}_{*G} = \mathbf{W}(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}})\mathbf{X} \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$.

The alternative approach is spatial-based convolutional GNNs, in which the graph convolutions are defined by information propagation. Diffusion graph convolution (DGC) (Atwood & Towsley, 2016), message passing neural network (MPNN) (Gilmer, Schoenholz, Riley, Vinyals, & Dahl, 2017), GraphSAGE (Hamilton, Ying, & Leskovec, 2017), and graph attention network (GAT) (Veličković et al., 2018) all follow this approach. The graph convolution is modeled as a diffusion process with a transition probability from one node to a neighboring node in DGC. An equilibrium is expected to be obtained after several rounds of information transition. The general framework followed is a message passing network, which models the graph convolutions as an information-passing process from one node to another connected node directly. To alleviate the computation problems caused by a large number of neighbors, sampling is used to obtain a fixed number of neighbors in GraphSAGE. Lastly, without using a predetermined adjacency matrix, the attention mechanism is used to learn the relative weights between two connected nodes in GAT.

MPNN uses message passing functions to unify different spatial-based variants. MPNN operates in two stages, namely, a message passing phase and a readout phase. The message passing phase is defined as follows:

$$\mathbf{m}_{v_i}^{(t)} = \sum_{v_j \in \mathcal{N}(v_i)} \mathcal{M}^{(t)}(\mathbf{X}_i^{(t-1)}, \mathbf{X}_j^{(t-1)}, \mathbf{e}_{ij}) \tag{3}$$

where $\mathbf{m}_{v_i}^{(t)}$ is the message aggregated from the neighbors of node v_i, $\mathcal{M}^{(t)}(\cdot)$ is the aggregation function in the tth iteration, $\mathbf{X}_i^{(t)}$ is the hidden state of node v_i in the tth iteration, and $\mathbf{e}_{ij}$ is the edge feature vector between node v_i and node v_j.

The readout phase is defined as follows:

$$\mathbf{X}_i^{(t)} = \mathcal{U}^{(t)}(\mathbf{X}_i^{(t-1)}, \mathbf{m}_{v_i}^{(t)}) \tag{4}$$

where $\mathcal{U}^{(t)}(\cdot)$ is the readout function in the tth iteration.

In GAT (Veličković et al., 2018), the attention mechanism (Vaswani et al., 2017) is incorporated into the propagation step and the multi-head attention mechanism is further utilized with the aim of stabilizing the learning process. The specific operation is defined as follows:

$$\mathbf{X}_i^{(t)} = \|_k \sigma(\sum_{j \in \mathcal{N}(v_i)} \alpha^k(\mathbf{X}_i^{(t-1)}, \mathbf{X}_j^{(t-1)})\mathbf{W}^{(t-1)}\mathbf{X}_j^{(t-1)}) \tag{5}$$

Table 2
Traffic graphs in the surveyed studies.

Graph	Node	Edge	Relevant studies
Sensor graph	Traffic sensors	Road links	Bai et al. (2020), Boukerche and Wang (2020b), Cai et al. (2020), Cao et al. (2020), Chen, Chen et al. (2020), Chen, Chen, Lai et al. (2020), Chen, Chen, Xie et al. (2020), Chen, Han et al. (2020), Chen et al. (2019), Chen, Liao et al. (2020), Chen, Zhang et al. (2020), Cirstea et al. (2019), Cui et al. (2019), Cui, Ke et al. (2020), Cui, Lin et al. (2020), Feng et al. (2020), Fukuda et al. (2020), Ge et al. (2019a, 2019b, 2020), Guo, Lin et al. (2019), Guo and Yuan (2020), Huang, Huang et al. (2020), Huang, Zhang et al. (2020), Jia et al. (2020), Kang et al. (2019), Kong et al. (2020), Lewenfus et al. (2020), Li, Li et al. (2020), Li, Wang et al. (2020), Li, Xiong et al. (2019, 2020), Li, Yu et al. (2018), Li and Zhu (2021), Lu, Zhang et al. (2019), Lv et al. (2020), Maas and Bloem (2020), Mallick et al. (2020, 2021), Opolka et al. (2019), Oreshkin et al. (2021), Pan et al. (2019, 2020), Park et al. (2020), Shleifer et al. (2019), Song, Lin et al. (2020), Song, Ming et al. (2020), Sun, Li et al. (2020), Sun et al. (2021), Tang, Sun and Sun (2020), Tang, Sun, Sun, Peng et al. (2020), Tian et al. (2020), Wang, Guan et al. (2020), Wang, Ma et al. (2020), Wang, Peng et al. (2020), Wang, Zhang et al. (2020), Wei and Sheng (2020), Wu, Pan, Long et al. (2020), Wu et al. (2019), Xie et al. (2020a, 2020b), Xin et al. (2020), Xiong et al. (2020), Xu, Dai et al. (2020), Xu, Wei et al. (2020), Xu, Zheng et al. (2020), Yang et al. (2020), Yin et al. (2020), Yu, Li et al. (2019), Yu et al. (2018), Yu, Yin et al. (2019), Zhang, Chang et al. (2020), Zhang, Cheng et al. (2020), Zhang and Guo (2020), Zhang, James et al. (2019), Zhang, Liu et al. (2020), Zhang, Shi et al. (2018), Zhang, Zhang et al. (2020), Zhao et al. (2019), Zhao, Yang et al. (2020), Zheng, Fan et al. (2020), Zhou, Yang, Zhang et al. (2020) and Zhou, Yang, Zhong et al. (2020)
Road segment graph	Road segments	Road intersections	Agafonov (2020), Bai et al. (2021), Chen, Zhao et al. (2020), Cui et al. (2019), Cui, Ke et al. (2020), Cui, Lin et al. (2020), Dai et al. (2020), Diao et al. (2019), Fang, Huang et al. (2020), Fu et al. (2020), Guo, Hu, Qian, Liu et al. (2020), Guo, Hu, Qian, Sun et al. (2020), Guo, Hu, Qian, Sun, Gao et al. (2020), Guo, Song et al. (2019), Guopeng et al. (2020), Han et al. (2020), Hasanzadeh et al. (2019), Hong et al. (2020), James (2019), Jepsen et al. (2019, 2020), Kim et al. (2020), Lee and Rhee (2019, 2022), Liao et al. (2018), Liu, Ong et al. (2020), Lu, Gan et al. (2020), Lu, Lv et al. (2020, 2019), Lv et al. (2020), Pan et al. (2019), Qu et al. (2020), Ramadan et al. (2020), Shin and Yoon (2020), Wang, Chen et al. (2018), Xie, Guo et al. (2020), Xie et al. (2019), Yu, Du et al. (2020), Yu and Gu (2019), Yu, Lee et al. (2020), Zhang, Chang et al. (2020), Zhang, Cheng et al. (2019, 2020), Zhang, Guan et al. (2019), Zhang, He et al. (2020), Zhang, Jin et al. (2018, 2019), Zhang, Liu et al. (2020), Zhang, Lu et al. (2020), Zhang, Wang et al. (2019), Zhao, Gao et al. (2020), Zhao et al. (2019) and Zhu et al. (2022, 2021, 2020)
Road intersection graph	Road intersections	Road segments	Bogaerts et al. (2020), Fang et al. (2019), James (2020), Lu, Lv et al. (2019), Qin, Xu et al. (2020), Sánchez et al. (2020), Shao et al. (2020), Wei et al. (2019), Wu, Chen et al. (2018), Xu et al. (2019), Zhang, Chang et al. (2020), Zhang, Jin et al. (2018, 2019) and Zhang, Li et al. (2019)
Road lane graph	Road lanes	Road line adjacency	Wright et al. (2019)
Irregular region graph	Irregular regions	Regional adjacency or virtual edges	Bai, Yao, Kanhere, Wang, Liu et al. (2019), Bai, Yao, Kanhere, Wang and Sheng (2019), Bing et al. (2020), Chen, Chen, Lai et al. (2020), Davis et al. (2020), Du et al. (2020), Hu et al. (2018, 2020), Ke, Feng et al. (2021), Li and Axhausen (2020), Li and Moura (2020), Liu, Zhao et al. (2020), Mohanty and Pozdnukhov (2018), Mohanty et al. (2020), Sun, Zhang et al. (2020), Ye et al. (2021), Zhang, Dong et al. (2020), Zheng, Hu et al. (2020) and Zhou, Gu et al. (2020)
Regular region graph	Regular regions	Regional adjacency or virtual edges	Davis et al. (2020), Geng, Li et al. (2019), Geng, Wu et al. (2019), He and Shin (2020a), Jin, Cui et al. (2020), Jin, Xi et al. (2020), Lee et al. (2019), Li, Yang et al. (2020), Pan et al. (2020), Pian and Wu (2020), Qiu et al. (2020), Shen et al. (2020), Shi et al. (2020), Wang, Luo et al. (2018), Wang, Miao et al. (2020), Wang, Xu et al. (2020), Wang et al. (2019), Wu, Zhu et al. (2020), Xu, Kang et al. (2020), Xu and Li (2019), Yeghikyan et al. (2020), Zhang, Huang et al. (2020), Zhou et al. (2019), Zhou, Wang, Xie, Chen and Liu (2020) and Zhou, Wang, Xie, Chen and Zhu (2020)
OD graph	OD pair	Virtual edges	Ke, Qin et al. (2021) and Wang, Xu et al. (2020)
Subway station graph	Subway stations	Subway lines	Fang, Pan et al. (2020), Fang et al. (2019), Han et al. (2019), Li, Peng et al. (2018), Li, Sergin et al. (2020), Liu, Chen et al. (2020), Ou et al. (2020), Ren and Xie (2019), Ye et al. (2020b), Zhang, Chen et al. (2020), Zhang, Chen and Guo (2020) and Zhao, Gao et al. (2020)
Bus station graph	Bus stations	Bus lines	Fang, Pan et al. (2020) and Fang et al. (2019)
Bike station graph	Bike stations	Road links	Chai et al. (2018), Chen, Rossi et al. (2020), Du et al. (2020), Guo, Jiang et al. (2019), He and Shin (2020b), Kim et al. (2019), Lin et al. (2018), Qin, Liu et al. (2020), Wang, Guo et al. (2020), Xiao et al. (2020) and Yoshida et al. (2019)
Railway station graph	Railway stations	Railway lines	He et al. (2020) and Heglund et al. (2020)
Car-sharing station graph	Car-sharing stations	Road links	Luo et al. (2020) and Zhu et al. (2019)
Parking lot graph	Parking lots	Road links	Zhang, Liu, Liu, Zhou and Xiong (2020) and Zhang, Liu, Liu, Zhou, Xu et al. (2020)
Parking block graph	Parking blocks	Road links	Yang et al. (2019)

Table 3
Adjacency matrices in the surveyed studies.

Adjacency matrix	Formula	Relevant studies
Connection matrix	$a_{ij} = 1$ when nodes i and j are connected and $a_{ij} = 0$ otherwise	Agafonov (2020), Bai et al. (2021), Bing et al. (2020), Cao et al. (2020), Chen, Chen et al. (2020), Chen, Zhang et al. (2020), Chen, Zhao et al. (2020), Cui et al. (2019), Cui, Ke et al. (2020), Cui, Lin et al. (2020), Davis et al. (2020), Fang, Huang et al. (2020), Guo, Hu, Qian, Liu et al. (2020), Guo, Song et al. (2019), Guo and Yuan (2020), He et al. (2020), Heglund et al. (2020), Hong et al. (2020), Huang, Zhang et al. (2020), James (2019), Li, Sergin et al. (2020), Liu, Chen et al. (2020), Liu, Ong et al. (2020), Liu, Zhao et al. (2020), Lu, Gan et al. (2020), Lu, Lv et al. (2019), Opolka et al. (2019), Ou et al. (2020), Qin, Xu et al. (2020), Qu et al. (2020), Sánchez et al. (2020), Shao et al. (2020), Shen et al. (2020), Song, Lin et al. (2020), Sun et al. (2021), Wang, Chen et al. (2018), Wang, Guo et al. (2020), Wang, Xu et al. (2020, 2020), Wei et al. (2019), Wu, Pan, Long et al. (2020), Xie, Guo et al. (2020), Xie et al. (2019), Xiong et al. (2020), Xu et al. (2019), Xu and Li (2019), Xu, Wei et al. (2020), Yin et al. (2020), Yu, Du et al. (2020), Yu and Gu (2019), Yu, Yin et al. (2019), Zhang, Chen et al. (2020), Zhang, Chen and Guo (2020), Zhang, Guan et al. (2019), Zhang and Guo (2020), Zhang, He et al. (2020), Zhang, Jin et al. (2018, 2019), Zhang, Li et al. (2019), Zhang, Liu, Liu, Zhou and Xiong (2020), Zhang, Liu, Liu, Zhou, Xu et al. (2020), Zhang, Liu et al. (2020), Zhang, Lu et al. (2020), Zhao et al. (2019), Zhou, Wang, Xie, Chen and Liu (2020) and Zhu et al. (2022, 2021, 2020)
Transportation connectivity matrix	$a_{ij} = 1$ when one can travel from node i to node j and $a_{ij} = 0$ otherwise	Geng, Li et al. (2019), Geng, Wu et al. (2019), Luo et al. (2020), Lv et al. (2020), Pan et al. (2019, 2020), Wright et al. (2019), Wu, Chen et al. (2018) and Ye et al. (2020b)
Direction matrix	$a_{ij} =$ the angle between two road segments	Lee and Rhee (2019, 2022) and Shin and Yoon (2020)
Neighbor matrix	$a_{ij} = 1$ when nodes i and j are neighbors and $a_{ij} = 0$ otherwise	Geng, Wu et al. (2019), Hu et al. (2018, 2020), Ke, Feng et al. (2021), Ke, Qin et al. (2021), Lee et al. (2019), Shi et al. (2020), Wang, Luo et al. (2018), Wang et al. (2019), Yeghikyan et al. (2020), Yoshida et al. (2019) and Zheng, Hu et al. (2020)
Distance matrix	$a_{ij} = d_{ij}$ and d_{ij} is some distance between nodes i and j	Bogaerts et al. (2020), Boukerche and Wang (2020b), Cai et al. (2020), Chai et al. (2018), Chen, Chen, Lai et al. (2020), Chen, Chen, Xie et al. (2020), Chen et al. (2019), Chen, Liao et al. (2020), Chen, Rossi et al. (2020), Chen, Zhang et al. (2020), Feng et al. (2020), Ge et al. (2019a, 2019b, 2020), Geng, Li et al. (2019), Guo, Hu, Qian, Sun, Gao et al. (2020), Guo, Jiang et al. (2019), He and Shin (2020a, 2020b), Huang, Huang et al. (2020), Jia et al. (2020), Jin, Cui et al. (2020), Jin, Xi et al. (2020), Kang et al. (2019), Ke, Feng et al. (2021), Ke, Qin et al. (2021), Kim et al. (2020, 2019), Lee and Rhee (2019, 2022), Li and Axhausen (2020), Li, Li et al. (2020), Li, Wang et al. (2020), Li, Yang et al. (2020), Li, Yu et al. (2018), Li and Zhu (2021), Lin et al. (2018), Lu, Zhang et al. (2019), Luo et al. (2020), Mallick et al. (2020, 2021), Mohanty and Pozdnukhov (2018), Mohanty et al. (2020), Pan et al. (2019, 2020), Ren and Xie (2019), Shin and Yoon (2020), Shleifer et al. (2019), Song, Ming et al. (2020), Sun, Li et al. (2020), Tian et al. (2020), Wang, Guan et al. (2020), Wang, Ma et al. (2020), Wang, Peng et al. (2020), Wang, Zhang et al. (2020), Wei and Sheng (2020), Xiao et al. (2020), Xie et al. (2020a, 2020b), Xin et al. (2020), Xu, Dai et al. (2020), Xu, Kang et al. (2020), Xu, Zheng et al. (2020), Yang et al. (2020, 2019), Yu et al. (2018), Zhang, Shi et al. (2018), Zhang, Wang et al. (2019), Zhang, Zhang et al. (2020), Zhao, Yang et al. (2020), Zheng, Fan et al. (2020), Zhou, Gu et al. (2020), Zhou, Yang, Zhang et al. (2020), Zhou, Yang, Zhong et al. (2020) and Zhu et al. (2019)
Traffic pattern similarity matrix	$a_{ij} =$ the correlation coefficient of historical traffic states of nodes i and j	Bai, Yao, Kanhere, Wang, Liu et al. (2019), Bai, Yao, Kanhere, Wang and Sheng (2019), Chai et al. (2018), Chen, Rossi et al. (2020), Dai et al. (2020), Han et al. (2020, 2019), He and Shin (2020a, 2020b), Jin, Cui et al. (2020), Jin, Xi et al. (2020), Ke, Qin et al. (2021), Kim et al. (2019), Lewenfus et al. (2020), Li and Axhausen (2020), Li, Yang et al. (2020), Li and Zhu (2021), Lin et al. (2018), Liu, Chen et al. (2020), Lu, Gan et al. (2020), Lv et al. (2020), Ren and Xie (2019), Sun, Zhang et al. (2020), Wang, Guo et al. (2020), Wang, Miao et al. (2020), Xu, Zheng et al. (2020), Yoshida et al. (2019), Zhou, Gu et al. (2020) and Zhou, Wang, Xie, Chen and Zhu (2020)
Functional similarity matrix	$a_{ij} =$ the correlation coefficient of POI distributions in regions i and j	Ge et al. (2019a, 2019b, 2020), Geng, Li et al. (2019), Geng, Wu et al. (2019), He and Shin (2020a), Jin, Xi et al. (2020), Ke, Qin et al. (2021), Luo et al. (2020), Lv et al. (2020), Shi et al. (2020), Zhang, Dong et al. (2020) and Zhu et al. (2019)
Dynamic matrix	N/A	Bai et al. (2020), Chen, Han et al. (2020), Diao et al. (2019), Du et al. (2020), Fang et al. (2019), Fu et al. (2020), Guo, Hu, Qian, Sun et al. (2020), Guo, Lin et al. (2019), Kong et al. (2020), Li and Axhausen (2020), Li and Moura (2020), Li, Peng et al. (2018), Li, Xiong et al. (2019, 2020), Maas and Bloem (2020), Peng et al. (2020), Shi et al. (2020), Tang, Sun and Sun (2020), Tang, Sun, Sun, Peng et al. (2020), Wu et al. (2019), Wu, Zhu et al. (2020), Ye et al. (2021), Yu, Lee et al. (2020), Zhang, Chang et al. (2020), Zhang, Cheng et al. (2019), Zhang, Huang et al. (2020), Zhang, James et al. (2019) and Zhou et al. (2019)

where ‖ is the concatenation operation, σ is the activation method, $\alpha^k(\cdot)$ is the kth attention mechanism.

A general spatiotemporal GNN structure is shown in Fig. 5(b), in which GCN is used to capture the spatial dependency and 1D-CNN is used to capture the temporal dependency. Both GCN and 1D-CNN components can be replaced with other structures for other spatiotemporal GNNs. A multilayer perceptron (MLP) component is used to generate the desired output. As for comparison, a two-layer GCN is also shown in Fig. 5(a), in which only the spatial dependency is concerned.

Spatiotemporal GNNs can be further categorized based on the approach used to capture the temporal dependency in particular. Most of the relevant studies in the literature can be split into two types, namely,

RNN-based and CNN-based spatiotemporal GNNs (Wu, Pan, Chen et al., 2020). The RNN-based approach is used in Bai, Yao, Kanhere, Wang, Liu et al. (2019), Bai et al. (2020, 2021), Bogaerts et al. (2020), Boukerche and Wang (2020b), Chai et al. (2018), Chen, Chen et al. (2020), Chen et al. (2019), Chen, Liao et al. (2020), Chen, Rossi et al. (2020), Chen, Zhao et al. (2020), Cirstea et al. (2019), Cui et al. (2019), Cui, Ke et al. (2020), Davis et al. (2020), Du et al. (2020), Fang, Pan et al. (2020), Fu et al. (2020), Fukuda et al. (2020), Geng, Li et al. (2019), Guo, Hu, Qian, Liu et al. (2020), Guo, Hu, Qian, Sun et al. (2020), Guo, Jiang et al. (2019), Guo, Song et al. (2019), Guopeng et al. (2020), He et al. (2020), Hu et al. (2018, 2020), Huang, Huang et al. (2020), Huang, Zhang et al. (2020), Jin, Cui et al. (2020), Kang et al. (2019), Ke, Qin et al. (2021), Lewenfus et al. (2020), Li, Xiong et al.
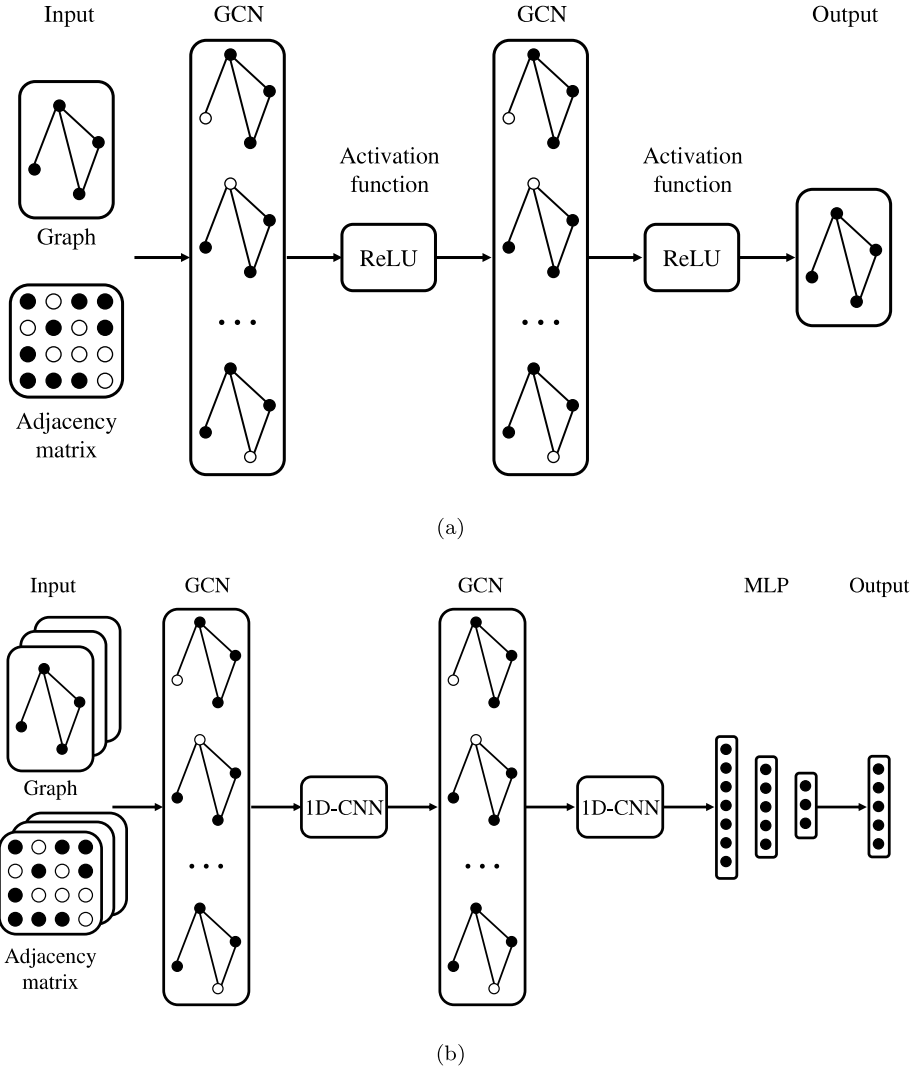
(a)

(b)

Fig. 5. A comparison between a two-layer GCN model and a typical spatiotemporal GNN structure (1D-CNN+GCN as an example). Adapted from Wang, Kong, Huang, and Xiao (2021). (a) a two-layer GCN model; (b) a typical spatiotemporal GNN structure.

Table 4
Basic notations used in this study.

Symbol	Description
G	Graph
V	Node set
E	Edge set
A	Adjacency matrix
χ_t or $\mathbf{X}$	Node feature matrix w/o time step index t
N	Node number
d	Node feature dimension
$\mathcal{N}(v_i)$	Neighbor node set of a single node v_i
$\mathbf{D}$	Degree matrix
$\mathbf{L}$	Laplacian matrix
$\bar{\mathbf{L}}$	Normalized Laplacian matrix
$\mathbf{I}_N$	Identity matrix with size N

(2019, 2020), Li, Yang et al. (2020), Li, Yu et al. (2018), Li and Zhu (2021), Liao et al. (2018), Lin et al. (2018), Liu, Chen et al. (2020), Liu, Zhao et al. (2020), Lu, Lv et al. (2020, 2019), Lu, Zhang et al. (2019), Luo et al. (2020), Lv et al. (2020), Mallick et al. (2020, 2021), Mohanty and Pozdnukhov (2018), Mohanty et al. (2020), Pan et al. (2019, 2020), Peng et al. (2020), Pian and Wu (2020), Qiu et al. (2020), Qu et al. (2020), Ramadan et al. (2020), Shao et al. (2020), Shen et al. (2020), Shi et al. (2020), Sun et al. (2021), Wang, Chen et al. (2018), Wang, Guo et al. (2020), Wang, Luo et al. (2018), Wang, Ma et al. (2020), Wang, Miao et al. (2020), Wang, Peng et al. (2020), Wang, Xu et al. (2020), Wang et al. (2019), Wei and Sheng (2020), Wright et al. (2019), Wu, Chen et al. (2018), Wu, Zhu et al. (2020), Xiao et al. (2020), Xie, Guo et al. (2020), Xie et al. (2019), Xin et al. (2020), Xu et al. (2019), Xu and Li (2019), Yang et al. (2019), Ye et al. (2021, 2020b), Yin et al. (2020), Yu, Yin et al. (2019), Zhang, Chen et al. (2020), Zhang, Cheng et al. (2020), Zhang, Guan et al. (2019), Zhang and Guo (2020), Zhang, James et al. (2019), Zhang, Li et al. (2019), Zhang, Liu, Liu, Zhou and Xiong (2020), Zhang, Liu, Liu, Zhou, Xu et al. (2020), Zhang, Lu et al. (2020), Zhang, Shi et al. (2018), Zhang, Wang et al. (2019), Zhao et al. (2019), Zhao, Yang et al. (2020), Zhou, Gu et al. (2020), Zhou et al. (2019), Zhou, Wang, Xie, Chen and Zhu (2020), Zhou, Yang, Zhang et al. (2020) and Zhu et al. (2022, 2019, 2021, 2020); while the CNN-based approach is used in Chen, Chen, Lai et al. (2020), Chen, Han et al. (2020), Dai et al. (2020), Diao et al. (2019), Fang et al. (2019), Feng et al. (2020), Guo, Lin et al. (2019), Guo and Yuan (2020), Heglund et al. (2020), Hong et al. (2020), Kong et al. (2020), Lee and Rhee (2019, 2022), Li, Li et al. (2020), Li, Wang et al. (2020), Lu, Gan et al. (2020), Maas and Bloem (2020), Ou et al. (2020), Song, Ming et al. (2020), Sun, Li et al. (2020), Tang, Sun and Sun (2020), Tang, Sun, Sun, Peng et al. (2020), Tian et al. (2020), Wang, Guan et al. (2020), Wang, Zhang et al. (2020), Wu, Pan, Long et al. (2020), Wu et al. (2019), Xie

et al. (2020a), Xu, Dai et al. (2020), Xu, Kang et al. (2020), Yu, Du et al. (2020), Yu et al. (2018), Zhang, Chang et al. (2020), Zhang, Chen and Guo (2020), Zhang, He et al. (2020), Zhang, Zhang et al. (2020), Zhao, Gao et al. (2020), Zheng, Hu et al. (2020) and Zhou, Wang, Xie, Chen and Liu (2020).

With the recent expansion of relevant studies, we add two sub-types of spatiotemporal GNNs in this survey, namely, *attention-based* and *FNN-based*. Attention mechanism is firstly proposed to memorize long source sentences in neural machine translation (Vaswani et al., 2017). Then it is used for temporal forecasting problems. As a special case, Transformer is built entirely upon attention mechanisms, which makes it possible to access any part of a sequence regardless of its distance to the target (Cai et al., 2020; Jin, Xi et al., 2020; Li & Moura, 2020; Xie et al., 2020b). The attention-based approaches are used in Bai, Yao, Kanhere, Wang and Sheng (2019), Cai et al. (2020), Chen, Chen, Xie et al. (2020), Fang, Huang et al. (2020), Jin, Xi et al. (2020), Li and Moura (2020), Park et al. (2020), Wang, Xu et al. (2020), Xie et al. (2020b), Zhang, Dong et al. (2020), Zhang, Liu et al. (2020), Zheng, Fan et al. (2020) and Zhou, Yang, Zhong et al. (2020), while the simpler FNN-based approach is used in Agafonov (2020), Cao et al. (2020), Chen, Zhang et al. (2020), Ge et al. (2019a, 2019b, 2020), Geng, Wu et al. (2019), Guo, Hu, Qian, Sun, Gao et al. (2020), Han et al. (2019), He and Shin (2020a, 2020b), Kim et al. (2019), Li, Peng et al. (2018), Qin, Liu et al. (2020), Ren and Xie (2019), Song, Lin et al. (2020), Sun, Zhang et al. (2020), Wei et al. (2019), Yeghikyan et al. (2020), Yu, Lee et al. (2020), Yu, Li et al. (2019), Zhang, Huang et al. (2020) and Zhang, Jin et al. (2018, 2019). Apart from using neural networks to capture temporal dependency, other techniques that have also been combined with GNNs include autoregression (Lee et al., 2019), Markov processes (Cui, Lin et al., 2020), and Kalman filters (Xiong et al., 2020).

Among different approaches for temporal modeling, RNNs suffer from time-consuming iterations and gradient vanishing or explosion problem with long sequences. CNNs demonstrate their superiority in terms of simple structure, parallel computing and stable gradients. As for the traffic problems, the spatial and temporal dependencies are closely intertwined in reality. For example, it is argued that the historical observations in different locations at different times have varying impacts on central region in the future (Guo, Lin et al., 2019). Some efforts are put to jointly modeling the potential interaction between spatial and temporal features and one promising direction is the incorporate of the graph convolution operations into RNNs to capture spatial–temporal correlations (Chen, Chen, Xie et al., 2020; Chen et al., 2019; Guo, Hu, Qian, Liu et al., 2020; Liu, Chen et al., 2020; Yu, Yin et al., 2019; Zhou et al., 2019). For example, the localized spatio-temporal correlation information is extracted simultaneously with the adjacency matrix of localized spatio-temporal graph in Song, Lin et al. (2020), in which a localized spatio-temporal graph that includes both temporal and spatial attributes is constructed first and a spatial-based GCN method is applied then.

Of the additional GNN components adopted in the surveyed studies, convolutional GNNs are the most popular, while recurrent GNN (Scarselli, Gori, Tsoi, Hagenbuchner, & Monfardini, 2008) and Graph Auto-Encoder (GAE) (Kipf & Welling, 2016) are used less frequently. We further categorize convolutional GNNs into the following five types: (1) GCN (Kipf & Welling, 2017), (2) DGC (Atwood & Towsley, 2016), (3) MPNN (Gilmer et al., 2017), (4) GraphSAGE (Hamilton et al., 2017), and (5) GAT (Veličković et al., 2018). These relevant graph neural networks are listed chronologically in Fig. 6. While different GNNs can be used for traffic forecasting, a general design pipeline is proposed in Zhou, Cui et al. (2020) and suggested for future studies as follows:

1. Find graph structure. As discussed in Section 4, different traffic graphs are available.
2. Specify graph type and scale. The graphs can be further classified into different types if needed, e.g., directed/undirected graphs, homogeneous/heterogeneous graphs, static/dynamic graphs. For most cases in traffic forecasting, the graphs of the same type are used in a single study. As for the graph scale, the graphs in the traffic domain are not as large as those for the social networks or academic networks with millions of nodes and edges.
3. Design loss function. The training setting usually follows the supervised approach, which means the GNN-based models are firstly trained on a training set with labels and then evaluated on a test set. The forecasting task is usually designed as the node-level regression problem. Based on these considerations, the proper loss function and evaluation metrics can be chosen, e.g., root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE).
4. Build model using computational modules. The GNNs discussed in this section are exactly those which have already been used as computational modules to build forecasting models in the surveyed studies.

A full list of the GNN components used in the surveyed studies is shown in Table 5.

Recurrent GNNs are developed in the early stage of GNN-relevant studies and less used for traffic forecasting after the appearance of more efficient architectures such as GCNs and GATs. As indicated by the name, recurrent GNNs are inspired by recurrent neural networks. Instead of updating the internal states along a temporal sequence as the normal recurrent neural networks, recurrent GNNs update the node representation in a recurrent neural approach, in which the message exchange with the neighbor nodes terminates when an equilibrium state is reached (Wu, Pan, Chen et al., 2020). Only a few relevant studies for traffic forecasting are based on recurrent GNNs. Considering that propagation patterns in the transportation system are time-variant, graph recurrent neural network (GRNN) (Wang, Chen et al., 2018) only uses one time-step propagation to capture the dynamic relations without involving a too long execution process to achieve the equilibrium state. Graph structured recurrent neural network (GSRNN) (Wang, Luo et al., 2018) extends the structural-RNN (SRNN) architecture to the directed graph case and uses a weighted sum pooling for the edge inputs. GSRNN also subsamples the nodes to maintain scalability when dealing with large graphs. Graph LSTM (GLSTM) (Lu, Lv et al., 2020, 2019) uses a temporal directed attributed graph to model the complex traffic flow and implement several variants of LSTMs with a graph neural block under the encoder–decoder framework to model spatial–temporal dependencies.

The advantage of recurrent GNNs over normal RNNs is the incorporation of information exchange with neighbor nodes, which helps to improve the traffic forecasting performance of a single node, compared with the case without using neighbors. However, the efficiency and scalability of recurrent GNNs are inferior to follow-up GCNs and GATs, making them less preferable for traffic forecasting in more recent studies.

Similar to recurrent GNNs, GAEs can be used to build graph embeddings, which are further processed by recurrent neural networks or other modules. GAEs are defined as the deep neural architectures that map nodes into a latent feature space and decode graph information from latent representations (Wu, Pan, Chen et al., 2020). Graph embedding is firstly used in spatio-temporal deep graph infomax (TDGI) (Opolka et al., 2019), in which the spatial information of the traffic graph is successfully encoded and more accurate predictions are achieved by an LSTM Seq2Seq model with the encoded features. In some cases, the traffic graphs can be sparse and DeepWalk is proposed to learn robust representations in graph embedding recurrent neural network (GERNN) (Xu et al., 2019) and graph embedding-generative adversarial network (GE-GAN) (Xu, Wei et al., 2020). GERNN further applies LSTM to learn dynamic temporal dependencies from the learned representations. GE-GAN further leverages the Wasserstein GAN to learn the road traffic state data distribution and generate road traffic state data. Instead of using DeepWalk, a graph auto-encoder is used
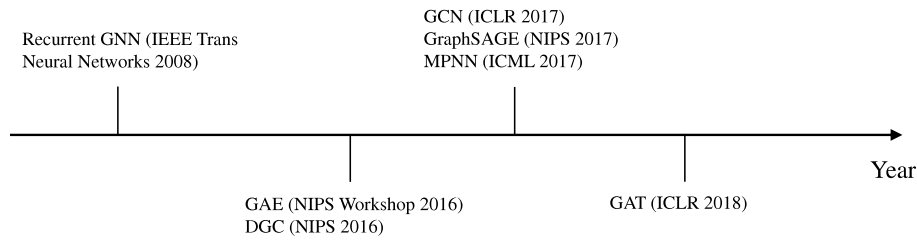
Fig. 6. The relevant graph neural networks in this survey.

Table 5
GNNs in the surveyed studies.

GNN	Relevant studies
Recurrent GNN	Lu, Lv et al. (2020, 2019), Wang, Chen et al. (2018) and Wang, Luo et al. (2018)
GAE	Opolka et al. (2019), Shen et al. (2020), Xu et al. (2019) and Xu, Wei et al. (2020)
GCN	Agafonov (2020), Bai, Yao, Kanhere, Wang, Liu et al. (2019), Bai, Yao, Kanhere, Wang and Sheng (2019), Bai et al. (2020, 2021), Bing et al. (2020), Bogaerts et al. (2020), Boukerche and Wang (2020b), Cai et al. (2020), Cao et al. (2020), Chai et al. (2018), Chen, Chen et al. (2020), Chen, Chen, Xie et al. (2020), Chen, Liao et al. (2020), Chen, Rossi et al. (2020), Chen, Zhang et al. (2020), Chen, Zhao et al. (2020), Cui et al. (2019), Cui, Ke et al. (2020), Cui, Lin et al. (2020), Dai et al. (2020), Davis et al. (2020), Diao et al. (2019), Du et al. (2020), Fang, Pan et al. (2020), Fang et al. (2019), Feng et al. (2020), Fu et al. (2020), Ge et al. (2019a, 2019b, 2020), Geng, Li et al. (2019), Geng, Wu et al. (2019), Guo, Hu, Qian, Liu et al. (2020), Guo, Hu, Qian, Sun et al. (2020), Guo, Hu, Qian, Sun, Gao et al. (2020), Guo, Jiang et al. (2019), Guo, Lin et al. (2019), Guo, Song et al. (2019), Guopeng et al. (2020), Han et al. (2020, 2019), He and Shin (2020a), He et al. (2020), Heglund et al. (2020), Hong et al. (2020), Hu et al. (2018, 2020), Huang, Zhang et al. (2020), James (2019, 2020), Jepsen et al. (2019, 2020), Jia et al. (2020), Jin, Cui et al. (2020), Ke, Feng et al. (2021), Ke, Qin et al. (2021), Kim et al. (2019), Lee and Rhee (2019, 2022), Lewenfus et al. (2020), Li and Axhausen (2020), Li, Li et al. (2020), Li and Moura (2020), Li, Peng et al. (2018), Li, Wang et al. (2020), Li, Xiong et al. (2019, 2020), Li, Yang et al. (2020), Li and Zhu (2021), Liao et al. (2018), Lin et al. (2018), Liu, Chen et al. (2020), Liu, Zhao et al. (2020), Lu, Gan et al. (2020), Lu, Zhang et al. (2019), Luo et al. (2020), Lv et al. (2020), Maas and Bloem (2020), Mohanty and Pozdnukhov (2018), Mohanty et al. (2020), Peng et al. (2020), Qin, Liu et al. (2020), Qin, Xu et al. (2020), Qiu et al. (2020), Qu et al. (2020), Ramadan et al. (2020), Ren and Xie (2019), Shao et al. (2020), Shi et al. (2020), Song, Lin et al. (2020), Song, Ming et al. (2020), Sun, Li et al. (2020), Sun et al. (2021), Sun, Zhang et al. (2020), Tang, Sun and Sun (2020), Tang, Sun, Sun, Peng et al. (2020), Wang, Guo et al. (2020), Wang, Ma et al. (2020), Wang, Miao et al. (2020), Wang, Xu et al. (2020, 2020), Wang, Zhang et al. (2020), Wu, Pan, Long et al. (2020), Wu et al. (2019), Xiao et al. (2020), Xie, Guo et al. (2020), Xin et al. (2020), Xiong et al. (2020), Xu, Dai et al. (2020), Xu, Kang et al. (2020), Xu, Zheng et al. (2020), Yang et al. (2019), Ye et al. (2021, 2020b), Yeghikyan et al. (2020), Yoshida et al. (2019), Yu, Du et al. (2020), Yu and Gu (2019), Yu, Lee et al. (2020), Yu, Li et al. (2019), Yu et al. (2018), Yu, Yin et al. (2019), Zhang, Chang et al. (2020), Zhang, Chen et al. (2020), Zhang, Chen and Guo (2020), Zhang, Cheng et al. (2019, 2020), Zhang, Dong et al. (2020), Zhang, Guan et al. (2019), Zhang, Jin et al. (2018, 2019), Zhang, Li et al. (2019), Zhang, Liu, Liu, Zhou and Xiong (2020), Zhang, Liu, Liu, Zhou, Xu et al. (2020), Zhang, Liu et al. (2020), Zhang, Lu et al. (2020), Zhang, Shi et al. (2018), Zhang, Wang et al. (2019), Zhao, Gao et al. (2020), Zhao et al. (2019), Zhao, Yang et al. (2020), Zheng, Hu et al. (2020), Zhou, Gu et al. (2020), Zhou et al. (2019), Zhou, Wang, Xie, Chen and Liu (2020), Zhou, Wang, Xie, Chen and Zhu (2020) and Zhu et al. (2022, 2019, 2021, 2020)
DGC	Chen, Han et al. (2020), Chen et al. (2019), Fukuda et al. (2020), Kim et al. (2020), Li, Yu et al. (2018), Mallick et al. (2020, 2021), Ou et al. (2020), Wang, Guan et al. (2020), Wang, Peng et al. (2020), Xie et al. (2020a), Zhou, Yang, Zhang et al. (2020) and Zhou, Yang, Zhong et al. (2020)
MPNN	Wang et al. (2019), Wei et al. (2019) and Xu, Dai et al. (2020)
GraphSAGE	Liu, Ong et al. (2020)
GAT	Cirstea et al. (2019), Fang, Huang et al. (2020), Guo and Yuan (2020), He and Shin (2020b), Huang, Huang et al. (2020), Jin, Xi et al. (2020), Kang et al. (2019), Kong et al. (2020), Pan et al. (2019, 2020), Park et al. (2020), Pian and Wu (2020), Song, Ming et al. (2020), Tang, Sun and Sun (2020), Tang, Sun, Sun, Peng et al. (2020), Tian et al. (2020), Wei and Sheng (2020), Wright et al. (2019), Wu, Chen et al. (2018), Wu, Zhu et al. (2020), Xie et al. (2020b), Xu and Li (2019), Yang et al. (2020), Yin et al. (2020), Zhang and Guo (2020), Zhang, He et al. (2020), Zhang, Huang et al. (2020), Zhang, James et al. (2019), Zhang, Zhang et al. (2020) and Zheng, Fan et al. (2020)

in TTPNet (Shen et al., 2020) to learn low-dimensional representation vectors of grids in the road network with the goal of preserving both the local and global structures, which are further processed by LSTMs for prediction.

The advantages of GAEs and graph embeddings are their abilities of generating low-dimensional representations, which are preferable in a large-scale and sparse traffic graph and help to ease the computation burdens. However, some spatial information is inevitably lost in the embedding process, which decreases the overall forecasting accuracy.

DGC is a classical spatial-based graph convolution which regards graph convolution as a diffusion process and has been applied for traffic forecasting earlier than other GCN variants. The most famous one is diffusion convolutional recurrent neural network (DCRNN) (Li, Yu et al., 2018), which uses diffusion graph convolutional networks and RNNs to learn the representations of spatial dependencies and temporal relations. DCRNN was originally proposed for traffic speed forecasting and is now widely used as a baseline. To create the traffic graph, the adjacency matrix is defined as the thresholded pairwise road network

distances. Compared with other graph convolutional models that can only operate on undirected graphs, e.g., ChebNet, DCRNN introduces the diffusion convolution (DC) operation for directed graph and is more suitable for transportation scenarios, which is defined as follows:

$$\mathbf{X}_{*DC} = \sum_{k=0}^{K-1} (\theta_{k,1}(D_O^{-1}A)^k) + (\theta_{k,2}(D_I^{-1}A^T)^k)\mathbf{X} \tag{6}$$

where $\mathbf{X} \in R^{N \times d}$ is the node feature matrix, A is the adjacency matrix, D_O and D_I are diagonal out-degree and in-degree matrices, $\theta_{k,1}$ and $\theta_{k,2}$ are model parameters, K is the number of diffusion steps. By defining and using out-degree and in-degree matrices, DCRNN models the bidirectional diffusion process to capture the influence of both upstream and downstream traffic. While DCRNN is a strong baseline, it is not suitable or desirable for the undirected graph cases. Then DCRNN is extended with a stronger learning ability in graph GRU in Zhang, Shi et al. (2018), in which a unified method for constructing an RNN based on an arbitrary graph convolution operator is proposed, instead of the single RNN model used in DCRNN.

The advantages of DGCs include the universality as a preprocessing step that can be applied to any graph and the effectiveness of capturing spatial dependencies in traffic domains. The disadvantages include the high computation and memory costs when the diffusion information from all nodes is considered in a large-scale graph.

Currently, the most widely used GNN format for traffic forecasting is the GCN, as indicated by the literature count in Table 5. GCNs overcome the disadvantages of DGCs by considering only the local neighbor nodes. However, the contribution weights from the local neighbor nodes to the central node are explicitly predefined, making them less flexible compared with GATs.

Many classical baselines as well as the state-of-the-art models all belong to the GCN category. Spatio-temporal graph convolutional network (STGCN) (Yu et al., 2018) stacks multiple spatio-temporal convolution blocks and each block concatenate two temporal convolution and one graph convolution layer. ChebNet is chosen as the graph convolution operator in STGCN, after a comparison with its first-order approximation. The usage of temporal convolution layers instead of RNNs for temporal modeling accelerates the training phase of STGCN. Attention based Spatio-temporal graph convolutional network (AST-GCN) (Guo, Lin et al., 2019) further introduces two attention layers in STGCN to capture the dynamic correlations in spatial dimension and temporal dimension, respectively.

Graph WaveNet (Wu et al., 2019) constructs a self-adaptive matrix to uncover unseen graph structures automatically from the data and WaveNet, which is based on causal convolutions, is used to learn temporal relations. However, the self-adaptive matrix in Graph WaveNet is fixed after training, which is unable to be adjusted dynamically with the data characteristics.

Adaptive graph convolutional recurrent network (AGCRN) (Bai et al., 2020) enhances the normal GCN with two adaptive modules. The first is a node adaptive parameter learning module, which factorizes the parameters and generates node-specific parameters from the shared weights and bias pools to capture node-specific patterns. And the other is a data adaptive graph generation module to infer the inter-dependencies among different traffic series automatically.

Spatial–temporal fusion graph neural network (STFGNN) (Li & Zhu, 2021) generates a novel temporal graph to model the spatial–temporal correlation that is beyond the representation ability of spatial graphs. A novel fusion operation is further incorporated into STFGNN to learn the hidden dependencies from both spatial and temporal graphs. STFGNN is also capable of handling long sequences by stacking the fusion graph and gated convolution modules.

Coupled layer-wise convolutional recurrent neural network (CCRNN) (Ye et al., 2021) proposes a novel graph convolution module named coupled layer-wise graph convolution (CGC), based on self-learned adjacency matrices varying from layer to layer. A layer-wise coupling mechanism is further proposed to use the hidden correlations

of topological structures and bridges adjacency matrices in different layers. The gated recurrent unit is used to make the final predictions, based on the spatial hidden states obtained by aggregating multi-level demand dependency.

Geographic and long-term temporal graph convolutional recurrent neural network (GLT-GCRNN) (Sun et al., 2021) learns the rich interactions between roads sharing similar geographic or long-term temporal patterns, which are often neglected in existing studies that only consider spatially-near nodes. A novel geographic and long-term temporal (GLT) graph construction method is designed with the consideration for the geographic and long-term temporal similarity. Then GCN is adopted to capture the spatial correlations of one link and its similar ones in both geographical and long-term temporal aspects more effectively.

MPNN follows and develops the idea of message exchange from recurrent GNNs, but with the spatial graph convolution operations. The message passing mechanism is used to realize a recursive interactions in DualGraph (Wei et al., 2019), which capture the nearby information flow between nodes and edges in the proposed DualMap block. Compared with the approach of only considering propagation among nodes, the dual graph approach of using both node and edge features generates a better forecasting performance. Spatial transformer is later proposed as a general MPNN in Spatial–Temporal Transformer Network (STTN) (Xu, Dai et al., 2020), which dynamically models directed spatial dependencies and captures real-time conditions and directions of traffic flows. MPNNs have the advantage of building a general mathematical formula for different GCN variants. However, it is found that MPNNs are incapable of distinguishing different graph structures based on the graph embedding, which is less preferable for traffic forecasting problems when graph structures are essential. Another flaw of MPNNs is the concern of a potential high computation cost.

GraphSAGE follows the idea of node feature encoding from GAEs, but with the graph convolution layers, which helps to boost the effectiveness of graph embeddings. GraphSAGE is used to generate the node representations from the local neighborhood in Liu, Ong et al. (2020) and then traffic speed is predicted based on the learned representations. Compared with GCNs and MPNNs, GraphSAGE has the advantages of a higher training efficiency with a batch-training algorithm and a less possibility of the memory overflow problem with a fixed sample size. Another advantage of GraphSAGE is the tolerance for the presence of missing data since GraphSAGE does not require all nodes in the graph to be present during the training of embeddings. However, GraphSAGE suffers from the similar information loss problem with GAEs and GEs when used for traffic forecasting. The contributions from neighboring nodes to the central node is identical in GraphSAGE, which also makes it less effective for forecasting, when compared with GCNs which explicitly assigns a predetermined weight.

While GCNs are dominant in the traffic forecasting area, we also notice a growing trend in the use of GATs. There are already some studies which apply GATs for traffic forecasting and more are expected in the near future. Spatial–temporal dynamic graph attention network (STDGAT) (Pian & Wu, 2020) is proposed for the regional ride-hailing demand prediction problem, in which the adaptive importance allocation for different neighbor regions are extracted and described by different pair-wise correlations. A novel time-specific commuting-based graph attention mode is designed to construct a dynamic graph structure, so that the dynamic time-specific spatial relationships throughout the timeline can be captured efficiently. Instead of using geographical connections, the commuting relationships between regions are used in the proposed graph attention mode. Spatial–temporal convolutional graph attention network (ST-CGA) (Zhang, Huang et al., 2020) jointly models the multi-granularity temporal dynamics and global inter-region traffic dependencies with the integrative framework of the self-attention module and graph attention subnet. ST-CGA adopts a multi-view collaboration framework and embeds multi-level temporal signals into the channel-aware spatial relation encoder, with a

channel-aware recalibration residual network. Spatial–temporal graph attention network (STGAT) (Kong et al., 2020) proposes a dual path architecture to capture both potential and existing spatial dependencies with the graph attention mechanism and a gated fusion mechanism is designed to combine the dual path outputs. STGAT further stacks the gated temporal convolution layers for capturing the long temporal dependency.

The self-attention mechanism is applied in several relevant studies. Self-attention based spatio-temporal graph neural network (SAST-GNN) (Xie et al., 2020b) utilizes a self-attention mechanism to extract features from the temporal dimension and the spatial dimension simultaneously, by adding channels(or heads) and residual blocks. The self-attention based module can efficiently filter noise from nodes with lower relationships and aggregate the higher quality information. ST-GRAT (Park et al., 2020) contains both the spatial attention and the temporal attention, in which the graph attention module dynamically adjusts the spatial correlation based on road states and the temporal attention module captures traffic speed changes. ST-GRAT further proposes a self-attention module with the sentinel vectors to make the decisions between retrieving new features from spatially correlated nodes or preserving existing features with an improved interpretable ability.

The multi-head attention mechanism is also used for traffic forecasting. Graph multi-attention network (GMAN) (Zheng, Fan et al., 2020) adopts the multi-head approach to compute the attention score, when building the spatio-temporal embeddings. Graph attention temporal convolutional network (GATCN) (Zhang, He et al., 2020) captures the complex spatio-temporal coupling effects within the traffic network with the multi-head self-attention mechanism. Spatial–temporal multi-head graph attention network (ST-MGAT) (Tian et al., 2020) contains a multi-head graph attention network module for capturing the dynamic spatial relations between nodes and a temporal convolution module based on the dilated convolution structure and gate mechanisms for capturing the dynamic time correlations.

GATs further improve over GraphSAGE and GCN by implicitly capturing the contribution weight from neighboring nodes via an end-to-end numeral network architecture, so that more important nodes receive large weights (Wu, Pan, Chen et al., 2020). Another advantage is that different attention mechanisms, e.g., the multi-head attention, can be further used to increase the expressive capability of GATs. However, the disadvantage of GATs is that only one-hop neighborhood nodes are considered without using the whole graph structure. This is not desirable for traffic forecasting problems, when a strong spatial dependency happens between two areas with a long distance.

5. Open data and source codes

In this section, we summarize the open data and source code used in the surveyed papers. These open data are suitable for GNN-related studies with graph structures discussed in Section 4, which can be used to formulate different forecasting problems in Section 3. We also list the GNN-related code resources for those who want to replicate the previous GNN-based solutions as baselines in the follow-up studies.

5.1. Open data

We categorize the data used in the surveyed studies into three major types, namely, graph-related data, historical traffic data, and external data. Graph-related data refer to those data which exhibit a graph structure in the traffic domain, i.e., transportation network data. Historical traffic data refer to those data which record the historical traffic states, usually in different locations and time points. We further categorize the historical traffic data into sub-types as follows. External data refer to the factors that would affect the traffic states, i.e., weather data and calendar data. Some of these data can be used in the graph-based

modeling directly, while the others may require some pre-processing steps before being Incorporated into GNN-based models.

Transportation Network Data. These data represent the underlying transportation infrastructure, e.g., road, subway, and bus networks. They can be obtained from government transportation departments or extracted from online map services, e.g., OpenStreetMap. Based on their topology structure, these data can be used to build the graphs directly, e.g., the road segments or the stations are nodes and the road intersections or subway links are the edges. While this modeling approach is straightforward, the disadvantage is that only static graphs can be built from transportation network data.

Traffic Sensor Data. Traffic sensors, e.g. loop detectors, are installed on roads to collect traffic information, e.g., traffic volume or speed. This type of data is widely used for traffic prediction, especially road traffic flow and speed prediction problems. For graph-based modeling, each sensor can be used as a node, with road connections as the edges. One advantage of using traffic sensor data for graph-based modeling is that the captured traffic information can be used directly as the node attributes, with little pre-processing overhead. One exception is that the sensors are prone to hardware faults, which causes the missing data or data noise problems and requires corresponding pre-processing techniques, e.g., data imputation and denoising methods. Another disadvantage of using traffic sensor data for graph-based modeling is that the traffic sensors can only be installed in a limited number of locations for a series of reasons, e.g., installation cost. With this constraint, only the part of the road networks with traffic sensors can be incorporated into a graph, while the uncovered areas are neglected.

GPS Trajectory Data. Different types of vehicles (e.g. taxis, buses, online ride-hailing vehicles, and shared bikes) can be equipped with GPS receivers, which record GPS coordinates in 2–60 s intervals. The trajectory data calculated from these GPS coordinate samples can be matched to road networks and further used to derive traffic flow or speed. The advantage of using GPS trajectory data for graph-based modeling is both the low expense to collect GPS data with smartphones and the wider coverage with the massive number of vehicles, compared with traffic sensor data. However, GPS trajectory data contain no direct traffic information, which can be derived with corresponding definitions though. The data quality problems also remain with GPS trajectory data and more pre-processing steps are required, e.g., map matching.

Location-based Service Data. GPS function is also embedded in smartphones, which can be used to collect various types of location-related data, e.g., check-in data, point-of-interest data, and route navigation application data. The pros and cons of using location-based service data are similar with GPS trajectory data. And the difference is that location-based service data are often collected in a crowd-sourced approach, with more data providers but potentially a lower data quality.

Trip Record Data. These include departure and arrival dates/times, departure and arrival locations, and other trip information. Traffic speed and demand can derived from trip record data from various sources, e.g., taxis, ride-hailing services, buses, bikes, or even dock-less e-scooters used in He and Shin (2020a). These data can be collected in public transportation systems with mature methods, for example, by AFC (Automatic Fare Collection) in the subway and bus systems. Trip record data have the advantage of being capable of constructing multiple graph-based problems, e.g., station-level traffic flow and demand problems. They are also easier to collect in existing public transportation systems.

Traffic Report Data. This type of data is often used for abnormal cases, e.g., anomaly report data used in Liu, Zhao et al. (2020) and traffic accident report data used in Zhang, Dong et al. (2020), Zhou, Wang, Xie, Chen and Liu (2020) and Zhou, Wang, Xie, Chen and Zhu (2020). Traffic report data are less used in graph-based modeling because of their sparsity in both spatial and temporal dimensions, compared with trip record data.

Multimedia Data. This type of data can be used as an additional input to deep learning models or for verifying the traffic status indicated by other data sources. Multimedia data used in the surveyed studies include the Baidu street-view images used in Qin, Xu et al. (2020) for traffic congestion, as well as satellite imagery data (Zhang, Dong et al., 2020), and video surveillance data (Shao et al., 2020). Multimedia data are also less seen in graph-based modeling because of their higher requirement for data collection, transmission and storage, compared with traffic sensor data with similar functionalities. It is also more difficult to extract precise traffic information, e.g., vehicle counts, from images or videos through image processing and object detection techniques.

Simulated Traffic Data. In addition to observed real-world datasets, microscopic traffic simulators are also used to build virtual training and testing datasets for deep learning models. Examples in the surveyed studies include the MATES Simulator used in Fukuda et al. (2020) and INTEGRATION software used in Ramadan et al. (2020). With many real-world datasets available, simulated traffic data are rarely used in GNN-based and more broader ML-based traffic forecasting studies. Traffic simulations have the potential of modeling unseen graphs though, e.g., evaluating a planned road topology.

Weather Data. Traffic states are highly affected by the meteorological factors including temperature, humidity, precipitation, barometer pressure, and wind strength.

Calendar Data. This includes the information on weekends and holidays. Because traffic patterns vary significantly between weekdays and weekends/holidays, some studies consider these two cases separately. Both weather and calendar data have been proven useful for traffic forecasting in the literature and should not be neglected in graph-based modeling as external factors.

While present road network and weather data can be easily found on the Internet, it is much more difficult to source historical traffic data, both due to data privacy concerns and the transmission and storage requirements of large data volumes. In Table 6 we present a list of the open data resources used in the surveyed studies. Most of these open data are already cleaned or preprocessed and can be readily used for benchmarking and comparing the performance of different models in future work.

5.1.1. Traffic sensor data

The relevant open traffic sensor data are listed as follows.

METR-LA[2]: This dataset contains traffic speed and volume collected from the highway of the Los Angeles County road network, with 207 loop detectors. The samples are aggregated in 5-minute intervals. The most frequently referenced time period for this dataset is from March 1st to June 30th, 2012.

Performance Measurement System (PeMS) Data[3]: This dataset contains raw detector data from over 18,000 vehicle detector stations on the freeway system spanning all major metropolitan areas of California from 2001 to 2019, collected with various sensors including inductive loops, side-fire radar, and magnetometers. The samples are captured every 30 s and aggregated in 5-minute intervals. Each data sample contains a timestamp, station ID, district, freeway ID, direction of travel, total flow, and average speed. Different subsets of PeMS data have been used in previous studies, for example:

- PeMS-BAY[4]: This subset contains data from 325 sensors in the Bay Area from January 1st to June 30th, 2017.
- PeMSD3: This subset uses 358 sensors in the North Central Area. The frequently referenced time period for this dataset is September 1st to November 30th, 2018.

- PeMSD4: This subset uses 307 sensors in the San Francisco Bay Area. The frequently referenced time period for this dataset is January 1st to February 28th, 2018.
- PeMSD7: This subset uses 883 sensors in the Los Angeles Area. The frequently referenced time period for this dataset is May to June, 2012.
- PeMSD8: This subset uses 170 sensors in the San Bernardino Area. The frequently referenced time period for this dataset is July to August, 2016.

Seattle Loop[5]: This dataset was collected by inductive loop detectors deployed on four connected freeways (I-5, I-405, I-90, and SR-520) in the Seattle area, from January 1st to 31st, 2015. It contains the traffic speed data from 323 detectors. The samples are aggregated in 5-minute intervals.

5.1.2. Taxi data

The open taxi datasets used in the surveyed studies are listed as follows.

T-drive (Yuan et al., 2010): This dataset contains a large number of taxicab trajectories collected by 30,000 taxis in Beijing from February 1st to June 2nd, 2015.

SHSpeed (Shanghai Traffic Speed) (Wang, Chen et al., 2018)[6]: This dataset contains 10-minute traffic speed data, derived from raw taxi trajectory data, collected from 1 to 30 April 2015, for 156 urban road segments in the central area of Shanghai, China.

TaxiBJ (Zhang, Zheng, & Qi, 2017): This dataset contains inflow and outflow data derived from GPS data in more than 34,000 taxicabs in Beijing from four time intervals: (1) July 1st to October 30th, 2013; (2) March 1st to June 30th, 2014; (3) March 1st to June 30th, 2015; and (4) November 1st, 2015 to April 10th, 2016. The Beijing city map is divided into 32×32 grids and the time interval of the flow data is 30 min.

TaxiSZ (Zhao et al., 2019)[7]: This dataset is derived from taxi trajectories in Shenzhen from January 1st to 31st, 2015. It contains the traffic speed on 156 major roads of the Luohu District every 15 min.

TaxiCD[8]: This dataset contains 1.4 billion GPS records from 14,864 taxis collected from August 3rd to 30th, 2014 in Chengdu, China. Each GPS record consists of a taxi ID, latitude, longitude, an indicator of whether the taxi is occupied, and a timestamp.

TaxiNYC[9]: The taxi trip records in New York starting from 2009, in both yellow and green taxis. Each trip record contains pick-up and drop-off dates/times, pick-up and drop-off locations, trip distances, itemized fares, rate types, payment types, and driver-reported passenger counts.

5.1.3. Ride-hailing data

The open ride-hailing data used in the surveyed studies are listed as follows.

UberNYC[10]: This dataset comes from Uber, which is one of the largest online ride-hailing companies in the USA, and is provided by the NYC Taxi & Limousine Commission (TLC). It contains data from over 4.5 million Uber pickups in New York City from April to September 2014, and 14.3 million more Uber pickups from January to June 2015.

Didi GAIA Open Data[11]: This open data plan is supported by Didi Chuxing, which is one of the largest online ride-hailing companies in China.

Table 6
Open data for traffic prediction problems.

Dataset Name	Relevant Studies
METR-LA	Bai et al. (2021), Boukerche and Wang (2020b), Cai et al. (2020), Cao et al. (2020), Chen, Chen, Lai et al. (2020), Chen, Chen, Xie et al. (2020), Chen et al. (2019), Cirstea et al. (2019), Cui, Lin et al. (2020), Feng et al. (2020), Jia et al. (2020), Li, Yu et al. (2018), Li and Zhu (2021), Lu, Zhang et al. (2019), Opolka et al. (2019), Oreshkin et al. (2021), Pan et al. (2019, 2020), Park et al. (2020), Shleifer et al. (2019), Song, Ming et al. (2020), Tian et al. (2020), Wang, Guan et al. (2020), Wang, Ma et al. (2020), Wu, Pan, Long et al. (2020), Wu et al. (2019), Xie et al. (2020a), Xu, Wei et al. (2020), Yang et al. (2020), Yu, Yin et al. (2019), Zhang, Chang et al. (2020), Zhang and Guo (2020), Zhang, Shi et al. (2018), Zhang, Zhang et al. (2020), Zhou, Yang, Zhang et al. (2020) and Zhou, Yang, Zhong et al. (2020)
PeMS all	Mallick et al. (2020, 2021)
PeMS-BAY	Boukerche and Wang (2020b), Cai et al. (2020), Cao et al. (2020), Chen, Chen, Xie et al. (2020), Chen et al. (2019), Cui, Lin et al. (2020), Feng et al. (2020), Guo and Yuan (2020), Li, Xiong et al. (2020), Li, Yu et al. (2018), Li and Zhu (2021), Oreshkin et al. (2021), Pan et al. (2019, 2020), Park et al. (2020), Shleifer et al. (2019), Song, Ming et al. (2020), Tian et al. (2020), Wang, Guan et al. (2020), Wang, Ma et al. (2020), Wu, Pan, Long et al. (2020), Wu et al. (2019), Xie et al. (2020a, 2020b), Xu, Dai et al. (2020), Yu, Li et al. (2019), Zhang and Guo (2020), Zhang, Zhang et al. (2020), Zheng, Fan et al. (2020), Zhou, Yang, Zhang et al. (2020) and Zhou, Yang, Zhong et al. (2020)
PeMSD3	Cao et al. (2020), Chen, Zhang et al. (2020), Li and Zhu (2021), Song, Lin et al. (2020) and Wang, Zhang et al. (2020)
PeMSD4	Bai et al. (2020), Cao et al. (2020), Chen, Han et al. (2020), Chen, Liao et al. (2020), Ge et al. (2019a, 2019b, 2020), Guo, Hu, Qian, Sun et al. (2020), Guo, Lin et al. (2019), Huang, Huang et al. (2020), Huang, Zhang et al. (2020), Li, Wang et al. (2020), Li, Xiong et al. (2019), Li and Zhu (2021), Song, Lin et al. (2020), Tang, Sun, Sun, Peng et al. (2020), Wang, Zhang et al. (2020), Wei and Sheng (2020), Xin et al. (2020), Xu, Zheng et al. (2020), Yin et al. (2020), Zhang, Chang et al. (2020), Zhang, Liu et al. (2020) and Zhao, Yang et al. (2020)
PeMSD7	Cao et al. (2020), Chen, Zhang et al. (2020), Ge et al. (2019a, 2019b, 2020), Huang, Huang et al. (2020), Li and Zhu (2021), Song, Lin et al. (2020), Sun, Li et al. (2020), Tang, Sun, Sun, Peng et al. (2020), Wang, Zhang et al. (2020), Xie et al. (2020b), Xin et al. (2020), Xu, Dai et al. (2020), Yu, Li et al. (2019), Yu et al. (2018), Yu, Yin et al. (2019), Zhang, Cheng et al. (2020), Zhang, James et al. (2019) and Zhao, Yang et al. (2020)
PeMSD8	Bai et al. (2020), Cao et al. (2020), Chen, Han et al. (2020), Guo, Hu, Qian, Sun et al. (2020), Guo, Lin et al. (2019), Huang, Huang et al. (2020), Li, Wang et al. (2020), Song, Lin et al. (2020), Wang, Zhang et al. (2020), Wei and Sheng (2020), Yin et al. (2020), Zhang, Liu et al. (2020) and Li and Zhu (2021)
Seattle Loop	Cui et al. (2019), Cui, Ke et al. (2020), Lewenfus et al. (2020) and Sun et al. (2021)
T-Drive	Pan et al. (2019, 2020)
SHSpeed	Guo, Song et al. (2019), Wang, Chen et al. (2018) and Zhang, Cheng et al. (2020)
TaxiBJ	Bai, Yao, Kanhere, Wang and Sheng (2019), Wang, Luo et al. (2018) and Zhang, Huang et al. (2020)
TaxiSZ	Bai et al. (2021) and Zhao et al. (2019)
TaxiCD	Hu et al. (2018, 2020)
TaxiNYC	Davis et al. (2020), Du et al. (2020), Hu et al. (2018), Jin, Xi et al. (2020), Li and Axhausen (2020), Li and Moura (2020), Sun, Zhang et al. (2020), Xu and Li (2019), Ye et al. (2021), Zhang, Huang et al. (2020), Zheng, Hu et al. (2020), Zhou, Wang, Xie, Chen and Zhu (2020) and Zhou et al. (2019)
UberNYC	Jin, Xi et al. (2020) and Ke, Feng et al. (2021)
DiDiChengdu	Bogaerts et al. (2020), Li, Yang et al. (2020), Qu et al. (2020), Wang, Xu et al. (2020, 2020), Zhang, Cheng et al. (2019) and Zhou et al. (2019)
DiDiTTIChengdu	Lu, Gan et al. (2020)
DiDiXi'an	Bogaerts et al. (2020) and Qu et al. (2020)
DiDiHaikou	Pian and Wu (2020) and Jin, Cui et al. (2020)
BikeDC	Sun, Zhang et al. (2020) and Wang, Guo et al. (2020)
BikeNYC	Bai, Yao, Kanhere, Wang and Sheng (2019), Chai et al. (2018), Du et al. (2020), Guo, Jiang et al. (2019), He and Shin (2020b), Lee et al. (2019), Lin et al. (2018), Sun, Zhang et al. (2020), Wang, Guo et al. (2020), Wang, Luo et al. (2018), Ye et al. (2021) and Zhang, Huang et al. (2020)
BikeChicago	Chai et al. (2018)
SHMetro	Liu, Chen et al. (2020)
HZMetro	Liu, Chen et al. (2020)

- **DiDiChengdu**: This dataset contains the trajectories of DiDi Express and DiDi Premier drivers within Chengdu, China. The data contains trips from October to November 2016.
- **DiDiTTIChengdu**: This dataset represents the DiDi Travel Time Index Data in Chengdu, China in the year of 2018, which contains the average speed of major roads every 10 min.
- **DiDiXi'an**: This dataset contains the trajectories of DiDi Express and DiDi Premier drivers within Xi'an, China. The data contains trips from October to November 2016.
- **DiDiHaikou**: The dataset contains DiDi Express and DiDi Premier orders from May 1st to October 31st, 2017 in the city of Haikou, China, including the coordinates of origins and destinations, pickup and drop-off timestamps, as well as other information.

5.1.4. Bike data

The open bike data used in the surveyed studies are listed as follows.

BikeNYC[12]: This dataset is from the NYC Bike System, which contains 416 stations. The frequently referenced time period for this dataset is from 1st July, 2013 to 31th December, 2016.

BikeDC[13]: This dataset is from the Washington D.C. Bike System, which contains 472 stations. Each record contains trip duration, start and end station IDs, and start and end times.

BikeChicago[14]: This dataset is from the Divvy System Data in Chicago, from 2015 to 2020.

5.1.5. Subway data

The subway data referenced in the surveyed studies are listed as follows.

SHMetro (*Liu, Chen et al., 2020*)[15]: This dataset is derived from 811.8 million transaction records of the Shanghai metro system collected from July 1st to September 30th, 2016. It contains 288 metro stations and 958 physical edges. The inflow and outflow of each station are provided in 15 min intervals.

HZMetro (*Liu, Chen et al., 2020*)[16]: This dataset is similar to SHMetro, from the metro system in Hangzhou, China, in January 2019. It contains 80 metro stations and 248 physical edges, and the aggregation time length is also 15 min.

5.2. Open source codes

Several open source frameworks for implementing general deep learning models, most of which are built with the Python programming language, can be accessed online, e.g. TensorFlow,[17] Keras,[18] PyTorch,[19] and MXNet.[20] Additional Python libraries designed for implementing GNNs are available. These include DGL,[21] pytorch_geometric,[22] and Graph Nets.[23]

Many authors have also released open-source implementations of their proposed models. The open source projects for traffic flow, traffic speed, traffic demand, and other problems are summarized in Tables 7, 8, 9, and 10, respectively. In these open source projects, TensorFlow and PyTorch are the two frameworks that are used most frequently.

---

12 https://www.citibikenyc.com/system-data.
13 https://www.capitalbikeshare.com/system-data.
14 https://www.divvybikes.com/system-data.
15 Download link: https://github.com/ivechan/PVCGN.
16 Download link: https://github.com/ivechan/PVCGN.
17 https://www.tensorflow.org/.
18 https://keras.io/.
19 https://pytorch.org/.
20 https://mxnet.apache.org/.
21 https://www.dgl.ai/.
22 https://pytorch-geometric.readthedocs.io/.
23 https://github.com/deepmind/graph_nets.

5.3. State-of-the-art performance

It is known that different works use different datasets and it is very hard to assess the relative performance of different state-of-the-art models (Tedjopurnomo et al., 2020). Even for those studies using the same dataset, different subsets may be used. Different preprocessing techniques, e.g., the missing data imputation method, and different evaluation settings, e.g., the training/validation/test subset split ratio, also cause incomparable results. Considering these difficulties, we only summarize those comparable results for the most frequently used datasets from the surveyed studies in this part.

Some commonly used evaluation metrics, namely, RMSE, MAE and MAPE, are defined as follows:

- $\text{RMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2};$
- $\text{MAE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{M} \sum_{i=1}^{M} |y_i - \hat{y}_i|;$
- $\text{MAPE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{M} \sum_{i=1}^{M} \frac{|y_i - \hat{y}_i|}{y_i};$

where $\mathbf{y}$ denotes the true values, $\hat{\mathbf{y}}$ denotes the predicted values, and M is the number of values to predict. A lower RMSE or MAE value indicates a better prediction performance. The summary for the state-of-the-art performance is shown in Table 11, with all or some of the above evaluation metrics and best values in bold. The default prediction time period is 60 min in Table 11 unless otherwise specified. Some classical baselines are also listed for comparison if available, e.g., DCRNN (Li, Yu et al., 2018), STGCN (Yu et al., 2018) and Graph WaveNet (Wu et al., 2019). Interested readers are recommended to check the experimental details in relevant studies.

Since the relevant studies of applying GNNs for traffic forecasting are growing everyday, the results listed in this part are not guaranteed to be the latest ones and the readers are recommended to follow our Github repository to track latest results.

6. Challenges and future directions

In this section, we discuss general challenges for traffic prediction problems as well as specific new challenges when GNNs are involved. While GNNs achieve a better forecasting performance, they are not the panacea. Some existing challenges from the border topic of traffic forecasting remain unsolved in current graph-based studies. Based on these challenges, we discuss possible future directions as well as early attempts in these directions. Some of these future directions are inspired from the border traffic forecasting research and remain insightful for the graph-based modeling approach. We would also highlight the special opportunities with GNNs.

6.1. Challenges

6.1.1. Heterogeneous data

Traffic prediction problems involve both spatiotemporal data and external factors, e.g., weather and calendar information. Heterogeneous data fusion is a challenge that is not limited to the traffic domain. GNNs have enabled significant progress by taking the underlying graph structures into consideration. However, some challenges remain; for example, geographically close nodes may not be the most influential, both for CNN-based and GNN-based approaches. Another special challenge for GNNs is that the underlying graph information may not be correct or up to date. For example, the road topology data of OpenStreetMap, an online map services, are collected in a crowd-sourced approach, which may be inaccurate or lagged behind the real road network. The spatial dependency relationship extracted by GNNs with these inaccurate data may decrease the forecasting accuracy.

Data quality concerns present an additional challenge with problems such as missing data, sparse data and noise potentially compromising forecasting results. Most of the surveyed models are only evaluated with processed high-quality datasets. A few studies do, however, take

Table 7

Open source projects for traffic flow related problems.

Article	Year	Framework	Problem	Link
Zheng, Fan et al. (2020)	2020	TensorFlow	Road traffic flow, Road traffic speed	https://github.com/zhengchuanpan/GMAN
Bai et al. (2020)	2020	PyTorch	Road traffic flow	https://github.com/LeiBAI/AGCRN
Song, Lin et al. (2020)	2020	MXNet	Road traffic flow	https://github.com/wanhuaiyu/STSGCN
Tang, Sun, Sun, Peng et al. (2020)	2020	TensorFlow	Road traffic flow	https://github.com/sam101340/GAGCN-BC-20200720
Wang, Zhang et al. (2020)	2020	MXNet, PyTorch	Road traffic flow	https://github.com/zkx741481546/Auto-STGCN
Guo, Hu, Qian, Sun et al. (2020)	2020	PyTorch	Road traffic flow, Road traffic speed	https://github.com/guokan987/DGCN
Li and Zhu (2021)	2020	MXNet	Road traffic flow, Road traffic speed	https://github.com/MengzhangLI/STFGNN
Tian et al. (2020)	2020	PyTorch, DGL	Road traffic flow	https://github.com/Kelang-Tian/ST-MGAT
Xiong et al. (2020)	2020	TensorFlow	Road OD Flow	https://github.com/alzmxx/OD_Prediction
Peng et al. (2020)	2020	Keras	Road station-level subway passenger flow, Station-level bus passenger flow, Regional taxi flow	https://github.com/RingBDStack/GCNN-In-Traffic
Qiu et al. (2020)	2020	Pytorch	Regional taxi flow	https://github.com/Stanislas0/ToGCN-V2X
Yeghikyan et al. (2020)	2020	PyTorch	Regional OD taxi flow	https://github.com/FelixOpolka/Mobility-Flows-Neural-Networks
Zhang, Chen et al. (2020)	2020	Keras	Station-level subway passenger flow	https://github.com/JinleiZhangBJTU/ResNet-LSTM-GCN
Zhang, Chen and Guo (2020)	2020	Keras	Station-level subway passenger flow	https://github.com/JinleiZhangBJTU/Conv-GCN
Liu, Chen et al. (2020)	2020	PyTorch	Station-level subway passenger flow	https://github.com/ivechan/PVCGN
Ye et al. (2020b)	2020	Keras	Station-level subway passenger flow	https://github.com/start2020/Multi-STGCnet
Pan et al. (2019)	2019	MXNet, DGL	Road traffic flow, Road traffic speed	https://github.com/panzheyi/ST-MetaNet
Guo, Lin et al. (2019)	2019	MXNet	Road traffic flow	https://github.com/wanhuaiyu/ASTGCN
Guo, Lin et al. (2019)	2019	PyTorch	Road traffic flow	https://github.com/wanhuaiyu/ASTGCN-r-pytorch
Wang, Chen et al. (2018)	2018	PyTorch	Road traffic flow	https://github.com/xxArbiter/grnn
Yu et al. (2018)	2018	TensorFlow	Road traffic flow	https://github.com/VeritasYin/STGCN_IJCAI-18
Li, Peng et al. (2018)	2018	Keras	Station-level subway passenger flow	https://github.com/RingBDStack/GCNN-In-Traffic
Chai et al. (2018)	2018	TensorFlow	Bike flow	https://github.com/Di-Chai/GraphCNN-Bike

Table 8

Open source projects for traffic speed related problems.

Article	Year	Framework	Problem	Link
Zhang, Huang et al. (2020)	2020	Keras	Road traffic speed	https://github.com/jillbetty001/ST-CGA
Bai et al. (2021)	2020	TensorFlow	Road traffic speed	https://github.com/lehaifeng/T-GCN/tree/master/A3T
Yang et al. (2020)	2020	TensorFlow	Road traffic speed	https://github.com/fanyang01/relational-ssm
Wu, Pan, Long et al. (2020)	2020	PyTorch	Road traffic speed	https://github.com/nnzhan/MTGNN
Mallick et al. (2021)	2020	TensorFlow	Road traffic speed	https://github.com/tanwimallick/TL-DCRNN
Chen, Chen et al. (2020)	2020	PyTorch	Road traffic speed	https://github.com/Fanglanc/DKFN
Lu, Gan et al. (2020)	2020	PyTorch	Road traffic speed	https://github.com/RobinLu1209/STAG-GCN
Guopeng et al. (2020)	2020	TensorFlow, Keras	Road traffic speed	https://github.com/RomainLITUD/DGCN_traffic_forecasting
Shen et al. (2020)	2020	PyTorch	Road travel time	https://github.com/YibinShen/TTPNet
Hong et al. (2020)	2020	TensorFlow	Time of arrival	https://github.com/didi/heteta
Wu et al. (2019)	2019	PyTorch	Road traffic speed	https://github.com/nnzhan/Graph-WaveNet
Shleifer et al. (2019)	2019	PyTorch	Road traffic speed	https://github.com/sshleifer/Graph-WaveNet
Zhao et al. (2019)	2019	TensorFlow	Road traffic speed	https://github.com/lehaifeng/T-GCN
Cui et al. (2019)	2019	TensorFlow	Road traffic speed	https://github.com/zhiyongc/Graph_Convolutional_LSTM
Jepsen et al. (2019, 2020)	2019	MXNet	Road traffic speed	https://github.com/TobiasSkovgaardJepsen/relational-fusion-networks
Li, Yu et al. (2018)	2018	TensorFlow	Road traffic speed	https://github.com/liyaguang/DCRNN
Li, Yu et al. (2018)	2018	PyTorch	Road traffic speed	https://github.com/chnsh/DCRNN_PyTorch
Zhang, Shi et al. (2018)	2018	MXNet	Road traffic speed	https://github.com/jennyzhang0215/GaAN
Liao et al. (2018)	2018	TensorFlow	Road traffic speed	https://github.com/JingqingZ/BaiduTraffic
Mohanty and Pozdnukhov (2018) and Mohanty et al. (2020)	2018	TensorFlow	Traffic congestion	https://github.com/sudatta0993/Dynamic-Congestion-Prediction

data quality related problems into consideration, e.g., using the Kalman filter to deal with the sensor data bias and noise (Chen, Chen et al., 2020), infilling missing data with moving average filters (Hasanzadeh et al., 2019) or linear interpolation (Agafonov, 2020; Chen, Chen et al., 2020). Missing data problem could be more common in GNNs, with the potential missing phenomena happening with historical traffic data or

Table 9

Open source projects for traffic demand related problems.

Article	Year	Framework	Problem	Link
Hu et al. (2020)	2020	TensorFlow	Taxi demand	https://github.com/hujilin1229/od-pred
Davis et al. (2020)	2020	TensorFlow, PyTorch	Taxi demand	https://github.com/NDavisK/Grids-versus-Graphs
Ye et al. (2021)	2020	PyTorch	Taxi demand, Bike demand	https://github.com/Essaim/CGCDemandPrediction
Lee et al. (2019)	2019	TensorFlow, Keras	Ride-hailing demand, Bike demand, Taxi demand	https://github.com/LeeDoYup/TGGNet-keras
Ke, Qin et al. (2021)	2019	Keras	Taxi demand	https://github.com/kejintao/ST-ED-RMGC

Table 10

Open source projects for other problems.

Article	Year	Framework	Problem	Link
Zhou, Wang, Xie, Chen and Liu (2020)	2020	TensorFlow	Traffic accident	https://github.com/zzyy0929/AAAI2020-RiskOracle/
Yu, Du et al. (2020)	2020	PyTorch, DGL	Traffic accident	https://github.com/yule-BUAA/DSTGCN
Zhang, Liu, Liu, Zhou and Xiong (2020)	2020	PyTorch, DGL	Parking availability	https://github.com/Vvrep/SHARE-parking_availability_prediction-Pytorch
Wang, Peng et al. (2020)	2020	TensorFlow	Transportation resilience	https://github.com/Charles117/resilience_shenzhen
Wright et al. (2019)	2019	TensorFlow, Keras	Lane occupancy	https://github.com/mawright/trafficgraphnn

Table 11

State-of-the-art performance for traffic prediction problems.

Dataset	RMSE	MAE	MAPE	Relevant studies
METR-LA	7.59	3.60	10.5%	DCRNN
	7.40	3.55	10.0%	ST-UNet (Yu, Yin et al., 2019)
	7.37	3.53	10.0%	Graph WaveNet
	7.20	3.30	9.7%	SLCNN (Zhang, Chang et al., 2020)
	6.68	3.28	9.08%	Traffic transformer (Cai et al., 2020)
	6.40	**3.18**	**8.81%**	STFGNN (Li & Zhu, 2021)
PeMS-BAY	4.74	2.07	4.9%	DCRNN
	4.53	2.03	4.8%	SLCNN (Zhang, Chang et al., 2020)
	4.52	1.95	4.63%	Graph WaveNet
	4.32	1.86	4.31%	GMAN (Zheng, Fan et al., 2020)
	4.36	1.77	4.29%	Traffic transformer (Cai et al., 2020)
	3.74	**1.66**	**3.77%**	STFGNN (Li & Zhu, 2021)
PeMSD3	30.31	18.18	18.91%	DCRNN
	30.12	17.49	17.15%	STGCN
	32.94	17.48	16.78%	Graph WaveNet
	28.34	**16.77**	**16.30%**	STFGNN (Li & Zhu, 2021)
PeMSD4	39.70	25.45	17.29%	Graph WaveNet
	34.89	21.16	13.83%	STGCN
	33.44	21.22	14.17%	DCRNN
	32.26	**19.83**	**12.97%**	AGCRN (Bai et al., 2020)
	31.88	**19.83**	13.02%	STFGNN (Li & Zhu, 2021)
PeMSD7	42.78	26.85	12.12%	Graph WaveNet
	38.78	25.38	11.08%	STGCN
	38.58	25.30	11.66%	DCRNN
	35.80	**22.07**	**9.21%**	STFGNN (Li & Zhu, 2021)
PeMSD8	31.05	19.13	12.68%	Graph WaveNet
	27.09	17.50	11.29%	STGCN
	26.36	16.82	10.92%	DCRNN
	26.22	16.64	10.60%	STFGNN (Li & Zhu, 2021)
	25.22	**15.95**	**10.09%**	AGCRN (Bai et al., 2020)
Seattle Loop	8.22	4.64	11.18%	DCRNN
	3.59	**2.45**	**5.90%**	GLT-GCRNN (Sun et al., 2021)
TaxiSZ	4.76	3.38	N/A	Graph WaveNet
	4.64	3.31	N/A	DCRNN
	4.13	2.79	N/A	T-GCN (Zhao et al., 2019)
	4.13	2.76	N/A	STGCN
	4.10	2.77	N/A	AST-GCN (Zhu et al., 2021)
	3.97	**2.74**	N/A	A3T-GCN (Bai et al., 2021)
TaxiNYC (30 min)	22.65	18.46	N/A	STGCN
	14.79	8.43	N/A	DCRNN
	13.07	8.10	N/A	Graph WaveNet
	9.56	**5.50**	N/A	CCRNN (Ye et al., 2021)
BikeNYC (30 min)	3.60	2.76	N/A	STGCN
	3.29	1.99	N/A	Graph WaveNet
	3.21	1.90	N/A	DCRNN
	2.84	**1.74**	N/A	CCRNN (Ye et al., 2021)

underlying graph information, e.g., GCNs are proposed to fill data gaps in missing OD flow problems (Yao et al., 2020).

Traffic anomalies (e.g., congestion) are an important external factor that may affect prediction accuracy and it has been proven that under congested traffic conditions a deep neural network may not perform as well as under normal traffic conditions (Mena-Oreja & Gozalvez, 2020). However, it remains a challenge to collect enough anomaly data to train deep learning models (including GNNs) in both normal and anomalous situations. The same concern applies for social events, public holidays, etc.

Challenges also exist for data privacy in the transportation domain. As discussed in Section 5.1, many open data are collected from individual mobile devices in a crowd sourcing approach. The data administrator must guarantee the privacy of individuals who contribute their personal traffic data, as the basis for encouraging a further contribution. Different techniques may be used, e.g., privacy-preserving data publishing techniques and privacy-aware data structures without personal identities.

6.1.2. Multi-task performance

For the public service operation of ITSs, a multi-task framework is necessary to incorporate all the traffic information and predict the demand of multiple transportation modes simultaneously. For example, knowledge adaption is proposed to adapt the relevant knowledge from an information-intensive source to information-sparse sources for demand prediction (Li, Bai, Liu, Yao and Waller, 2020). Related challenges lie in data format incompatibilities as well as the inherent differences in spatial or temporal patterns. While some of the surveyed models can be used for multiple tasks, e.g., traffic flow and traffic speed prediction on the same road segment, most can only be trained for a single task at one time.

Multi-task forecasting is a bigger challenge in graph-based modeling because different tasks may use different graph structures, e.g., road-level and station-level problems use different graphs and thus are difficult to be solved with a single GNN model. Some efforts that have been made in GNN-based models for multi-task prediction include taxi departure flow and arrival flow (Chen, Zhao et al., 2020), region-flow and transition-flow (Wang, Xu et al., 2020), crowd flows, and OD of the flows (Wang, Miao et al., 2020). However, most of the existing attempts are based on the same graph with multiple outputs generated by feed forward layers. Nonetheless, GNN-based multi-task prediction for different types of traffic forecasting problems is a research direction requiring significant further development, especially those requiring multiple graph structures.

6.1.3. Practical implementation

A number of challenges prevent the practical implementation of the models developed in the surveyed studies in city-scale ITSs.

First, there is significant bias introduced by the small amount of data considered in the existing GNN-based studies which, in most cases, spans less than one year. The proposed solutions are therefore not necessarily applicable to different time periods or different places. If longer traffic data are to be used in GNNs, the corresponding change of the underlying traffic infrastructures should be recorded and updated, which increases both the expense and difficulty of the associated data collection process in practice.

A second challenge is the computation scalability of GNNs. To avoid the huge computation requirements of the large-scale real-world traffic network graphs, only a subset of the nodes and edges are typically considered. For example, most studies only use a subset of the PeMS dataset when considering the road traffic flow or speed problems. Their results can therefore only be applied to the selected subsets. Graph partitioning and parallel computing infrastructures have been proposed for solving this problem. The traffic speed and flow of the entire PeMS dataset with 11,160 traffic sensor locations are predicted simultaneously in Mallick et al. (2020), using a graph-partitioning method that

decomposes a large highway network into smaller networks and trains a single DCRNN model on a cluster with graphics processing units (GPUs). However, increased modeling power can only improve the state-of-the-art results with narrow performance margins, compared to statistical and machine learning models with less complex structures and computational requirements.

A third challenge is presented by changes in the transportation networks and infrastructure, which are essential to build the graphs in GNNs. The real-world network graphs change when road segments or bus lines are added or removed. Points-of-interest in a city also change when new facilities are built. Static graph formulations are not enough for handling these situations. Some efforts have been made to solve this problem with promising results. For example, a dynamic Laplacian matrix estimator is proposed to find the change of Laplacian matrix, according to changes in spatial dependencies hidden in the traffic data (Diao et al., 2019), and a Data Adaptive Graph Generation (DAGG) module is proposed to infer the inter-dependencies between different traffic series automatically, without using pre-defined graphs based on spatial connections (Bai et al., 2020).

6.1.4. Model interpretation

The challenge of model interpretation is a point of criticism for all "black-box" machine learning or deep learning models, and traffic forecasting tasks are no exception (Barredo-Arrieta, Laña, & Del Ser, 2019; Wu, Tan, Qin, Ran and Jiang, 2018). While there have been remarkable progresses for visualizing and explaining other deep neural network structures, e.g., CNNs, the development of post-processing techniques to explain the predictions made by GNNs is still in an early phase (Baldassarre & Azizpour, 2019; Pope, Kolouri, Rostami, Martin, & Hoffmann, 2019; Ying, Bourgeois, You, Zitnik, & Leskovec, 2019) and the application of these techniques to the traffic forecasting domain has not yet been addressed.

Compared with other similar forecasting problems in other domains, lack of model interpretation may be a more severe problem in the transportation domain, when complex data types and representations of heterogeneous traffic data make it more challenging to design an interpretable deep learning model, compared with other data formats, e.g., images and text. While some efforts have been made to incorporate the state space model to increase the model interpretation for traffic forecasting (Li, Yan, Yang and Jin, 2019), this problem has not fully solved, especially for GNN-based models.

6.2. Future directions

6.2.1. Centralized data repository

A centralized data repository for GNN-based traffic forecasting resources would facilitate objective comparison of the performance of different models and be an invaluable contribution to the field. This future direction is proposed for the challenge of heterogeneous data as well as the data quality problem. Another unique feature of this repository could be the inclusion of graph-related data, which have not be provided directly in previous traffic forecasting studies.

Some criteria for building such data repositories, e.g. a unified data format, tracking of dataset versions, public code and ranked results, and sufficient record lengths (longer than a year ideally), have been discussed in previous surveys (Manibardo et al., 2021). Compiling a centralized and standardized data repository is particularly challenging for GNN-based models where natural graphs are collected and stored in a variety of data formats (e.g. Esri Shapefile and OSM XML used by Openstreetmap are used for digital maps in the GIS community) and various different similarity graphs can be constructed from the same traffic data in different models.

Some previous attempts in this direction have been made in the machine learning community, e.g. setting benchmarks for several traffic

prediction tasks in Papers With Code,[24] and in data science competitions, e.g., the Traffic4cast competition series.[25] However, the realization of a centralized data repository remains an open challenge.

A centralized data repository is also the basis for benchmarking traffic prediction, which is previously discussed in Section 5.3. With more and more GNN-based models being proposed, it becomes even more difficult to compare different models and validate the effectiveness of new traffic forecasting methods without a considerable effort, when a standardized benchmark dataset and consistent experimental settings have not been established yet. The most close one is the PeMS dataset, but it covers the road-level case only and more efforts are still needed, especially for the remaining cases.

6.2.2. Traffic graph design

While various graphs have been constructed in the surveyed studies as discussed in Section 4.1 and have been proven successful to some extent, most of them are natural graphs based on a real-world transportation system, e.g. the road network or subway system, as the current development status. And most of the graphs used are static, instead of dynamic ones. One specific direction that is not fully considered before is the design of transportation knowledge graph. As an important tool for knowledge integration, knowledge graph is a complex relational network that consists of concepts, entities, entity relations and attributes (Yin et al., 2021). The transportation knowledge graph helps to leverage the traffic semantic information to improve the forecasting performance. And the challenge is to extract the hidden transportation domain knowledge from multi-source and heterogeneous traffic data.

6.2.3. Combination with other techniques

GNNs may be combined with other advanced techniques to overcome some of their inherent challenges and achieve better performance.

Data Augmentation. Data augmentation has been proven effective for boosting the performance of deep learning models, e.g. in image classification tasks and time series prediction tasks. Data augmentation is proposed for the challenge of the possible forecasting bias introduced by the small amount of available data. However, due to the complex structure of graphs, it is more challenging to apply data augmentation techniques to GNNs. Recently, data augmentation for GNNs has proven helpful in semi-supervised node classification tasks (Zhao et al., 2021). However, it remains a question whether data augmentation may be effective in traffic forecasting GNN applications.

Transfer Learning. Transfer learning utilizes knowledge or models trained for one task to solve related tasks, especially those with limited data. In the image classification field, pre-trained deep learning models from the ImageNet or MS COCO datasets are widely used in other problems. In traffic prediction problems, where a lack of historical data is a frequent problem, transfer learning is a possible solution. For GNNs, transfer learning can be used from a graph with more historical traffic data for the model training process to another graph with less available data. Transfer learning can also be used for the challenge caused by the changes in the transportation networks and infrastructure, when new stations or regions have not accumulated enough historical traffic data to train a GNN model. A novel transfer learning approach for DCRNN is proposed in Mallick et al. (2021), so that a model trained on data-rich regions of highway network can be used to predict traffic on unseen regions of the highway network. The authors demonstrated the efficacy of model transferability between the San Francisco and Los Angeles regions using different parts of the California road network from the PeMS.

Meta-learning. Meta-learning, or learning how to learn, has recently become a potential learning paradigm that can absorb information from a task and effectively generalize it to an unseen task. Meta-learning is proposed for the challenge of GNN-based multi-task prediction, especially those involving multiple graphs. There are different types of meta learning methods and some of them are combined with graph structures for describing relationships between tasks or data samples (Liu, Zhou, Long, Jiang, & Zhang, 2019; Satorras & Estrach, 2018). Based on a deep meta learning method called network weight generation, ST-MetaNet$^+$ is proposed in Pan et al. (2020), which leverages the meta knowledge extracted from geo-graph attributes and dynamic traffic context learned from traffic states to generate the parameter weights in graph attention networks and RNNs, so that the inherent relationships between diverse types of spatiotemporal correlations and geo-graph attributes can be captured.

Generative Adversarial Network (GAN) (Goodfellow et al., 2014). GAN is a machine learning framework that has two components, namely, a generator, which learns to generate plausible data, and a discriminator, which learns to distinguish the generator's fake data from real data. After training to a state of Nash equilibrium, the generator may generate undistinguished data, which helps to expand the training data size for many problems, including GNN-based traffic forecasting. GAN is proposed for the challenges caused by the small data amount used in previous studies or the changes in the transportation networks and infrastructure when not enough historical traffic data are available. In Xu, Wei et al. (2020), the road network is used directly as the graph, in which the nodes are road state detectors and the edges are built based on their adjacent links. DeepWalk is used to embed the graph and the road traffic state sensor information is transferred into a low-dimensional space. Then, the Wasserstein GAN (WGAN) (Arjovsky, Chintala, & Bottou, 2017) is used to train the traffic state data distribution and generate predicted results. Both public traffic flow (i.e., Caltrans PeMSD7) and traffic speed (i.e., METR-LA) datasets are used for evaluation, and the results demonstrate the effectiveness of the GAN-based solution when used in graph-based modeling.

Automated Machine Learning (AutoML). The application of machine learning requires considerable manual intervention in various aspects of the process, including feature extraction, model selection, and parameter adjustment. AutoML automatically learns the important steps related to features, models, optimization, and evaluation, so that machine learning models can be applied without manual intervention. AutoML would help to improve the implementation of machine learning models, including GNNs. AutoML is proposed for the challenge for computational requirements in graph-based modeling, in which case the hyper parameter tuning for GNNs can be more efficient with state-of-the-art AutoML techniques. An early attempt to combine AutoML with GNNs for traffic prediction problems is an Auto-STGCN algorithm, proposed in Wang, Zhang et al. (2020). This algorithm searches the parameter space for STGCN models quickly based on reinforcement learning and generates optimal models automatically for specific scenarios.

Bayesian Network. Most of the existing studies aim for deterministic models that make mean predictions. However, some traffic applications rely on uncertainty estimates for the future situations. To tackle this gap, the Bayesian network, which is a type of probabilistic graphical model using Bayesian inference for probability computations, is a promising solution. The combination of GNNs with Bayesian networks is proposed for the challenge of GNN model interpretation. With probabilistic predictions, uncertainty estimates are generated for the future situations, especially the chance of extreme traffic states. A similar alternative is Quantile Regression, which estimates the quantile function of a distribution at chosen points, combined with Graph WaveNet for uncertainty estimates (Maas & Bloem, 2020).

6.2.4. Applications in real-world ITS systems

Last but not the least, most of the surveyed GNN-based studies are only based on the simulations with historical traffic data, without being validated or deployed in real-world ITS systems. However, there are a number of potential applications, especially for GNN-based models with the better forecasting performance. To name a few potential cases, the GNN-based forecasting model can be used for traffic light control in signalized intersections, when each intersection is modeled as a node in the graph and the corresponding traffic flow forecasting result can be used to design the traffic light control strategy. Another example is the application in map service and navigation applications, in which each road segment is modeled as a node in the graph and the corresponding traffic speed and travel time forecasting result can be used to calculate the estimated time of arrival. A third example is the application in online ride-hailing service providers, e.g., Uber and Lyft, in which each region is modeled as a node and the corresponding ride-hailing demand forecasting can be used to design a more profitable vehicle dispatching and scheduling system. Inspired by these potential application scenarios, there are a lot of potential research opportunities for researchers from both the academia and the industry.

7. Conclusion

In this paper, a comprehensive review of the application of GNNs for traffic forecasting is presented. Three levels of traffic problems and graphs are summarized, namely, road-level, region-level and station-level. The usages of recurrent GNNs, convolutional GNNs and graph autoencoders are discussed. We also give the latest collection of open dataset and code resource for this topic. Challenges and future directions are further pointed out for the follow-up research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agafonov, A. (2020). Traffic flow prediction using graph convolution neural networks. In *2020 10th international conference on information science and technology (ICIST)* (pp. 91–95). IEEE.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv: 1701.07875.

Atwood, J., & Towsley, D. (2016). Diffusion-convolutional neural networks. In *NIPS*.

Bai, L., Yao, L., Kanhere, S. S., Wang, X., Liu, W., & Yang, Z. (2019). Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 2293–2296).

Bai, L., Yao, L., Kanhere, S. S., Wang, X., & Sheng, Q. Z. (2019). Stg2seq: spatial-temporal graph to sequence model for multi-step passenger demand forecasting. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 1981–1987). AAAI Press.

Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in neural information processing systems*.

Bai, J., Zhu, J., Song, Y., Zhao, L., Hou, Z., Du, R., et al. (2021). A3T-GCN: attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, *10*(7), 485.

Baldassarre, F., & Azizpour, H. (2019). Explainability techniques for graph convolutional networks. In *International conference on machine learning (ICML) workshops, 2019 workshop on learning and reasoning with graph-structured representations*.

Barredo-Arrieta, A., Laña, I., & Del Ser, J. (2019). What Lies beneath: A note on the explainability of black-box machine learning models for road traffic forecasting. In *2019 IEEE intelligent transportation systems conference (ITSC)* (pp. 2232–2237). IEEE.

Bing, H., Zhifeng, X., Yangjie, X., Jinxing, H., & Zhanwu, M. (2020). Integrating semantic zoning information with the prediction of road link speed based on taxi GPS data. *Complexity*, *2020*.

Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., & Hellinckx, P. (2020). A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C (Emerging Technologies)*, *112*, 62–77.

Boukerche, A., Tao, Y., & Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Computer Networks*, *182*, Article 107484.

Boukerche, A., & Wang, J. (2020a). Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, *181*, Article 107530.

Boukerche, A., & Wang, J. (2020b). A performance modeling and analysis of a novel vehicular traffic flow prediction system using a hybrid machine learning-based model. *Ad Hoc Networks*.

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). Spectral networks and deep locally connected networks on graphs. In *2nd international conference on learning representations, ICLR 2014*.

Cai, L., Janowicz, K., Mai, G., Yan, B., & Zhu, R. (2020). Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., et al. (2020). Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, *33*.

Chai, D., Wang, L., & Yang, Q. (2018). Bike flow prediction with multi-graph convolutional networks. In *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 397–400).

Chen, F., Chen, Z., Biswas, S., Lei, S., Ramakrishnan, N., & Lu, C.-T. (2020). Graph convolutional networks with Kalman filtering for traffic prediction. In *Proceedings of the 28th international conference on advances in geographic information systems* (pp. 135–138).

Chen, K., Chen, F., Lai, B., Jin, Z., Liu, Y., Li, K., et al. (2020). Dynamic spatio-temporal graph-based CNNs for traffic flow prediction. *IEEE Access*, *8*, 185136–185145.

Chen, W., Chen, L., Xie, Y., Cao, W., Gao, Y., & Feng, X. (2020). Multi-range attentive bicomponent graph convolutional network for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*.

Chen, L., Han, K., Yin, Q., & Cao, Z. (2020). GDCRN: Global diffusion convolutional residual network for traffic flow prediction. In *International conference on knowledge science, engineering and management* (pp. 438–449). Springer.

Chen, C., Li, K., Teo, S. G., Zou, X., Wang, K., Wang, J., et al. (2019). Gated residual recurrent graph neural networks for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 485–492).

Chen, J., Liao, S., Hou, J., Wang, K., & Wen, J. (2020). GST-GCN: A geographic-semantic-temporal graph convolutional network for context-aware traffic flow prediction on graph sequences. In *2020 IEEE international conference on systems, man, and cybernetics (SMC)* (pp. 1604–1609). IEEE.

Chen, H., Rossi, R. A., Mahadik, K., & Eldardiry, H. (2020). A context integrated relational spatio-temporal model for demand and supply forecasting. arXiv preprint arXiv:2009.12469.

Chen, X., Zhang, Y., Du, L., Fang, Z., Ren, Y., Bian, K., et al. (2020). TSSRGCN: Temporal spectral spatial retrieval graph convolutional network for traffic flow forecasting. In *2020 IEEE international conference on data mining (ICDM)*. IEEE.

Chen, Z., Zhao, B., Wang, Y., Duan, Z., & Zhao, X. (2020). Multitask learning and GCN-based taxi demand prediction for a traffic road network. *Sensors*, *20*(13), 3776.

Cirstea, R.-G., Guo, C., & Yang, B. (2019). Graph attention recurrent neural networks for correlated time series forecasting. In *MileTS19@KDD*.

Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*.

Cui, Z., Ke, R., Pu, Z., Ma, X., & Wang, Y. (2020). Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. *Transportation Research Part C (Emerging Technologies)*, *115*, Article 102620.

Cui, Z., Lin, L., Pu, Z., & Wang, Y. (2020). Graph Markov network for traffic forecasting with missing data. *Transportation Research Part C (Emerging Technologies)*, *117*, Article 102671. http://dx.doi.org/10.1016/j.trc.2020.102671, URL: http://www.sciencedirect.com/science/article/pii/S0968090X20305866.

Dai, R., Xu, S., Gu, Q., Ji, C., & Liu, K. (2020). Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In *KDD '20, Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 3074–3082). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3394486.3403358.

Davis, N., Raina, G., & Jagannathan, K. (2020). Grids versus graphs: Partitioning space for improved taxi demand-supply forecasts. *IEEE Transactions on Intelligent Transportation Systems*.

Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th international conference on neural information processing systems* (pp. 3844–3852).

Diao, Z., Wang, X., Zhang, D., Liu, Y., Xie, K., & He, S. (2019). Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 890–897).

Du, B., Hu, X., Sun, L., Liu, J., Qiao, Y., & Lv, W. (2020). Traffic demand prediction based on dynamic transition convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*.

Fan, X., Xiang, C., Gong, L., He, X., Qu, Y., Amirgholipour, S., et al. (2020). Deep learning for intelligent traffic sensing and prediction: recent advances and future challenges. *CCF Transactions on Pervasive Computing and Interaction*, 1–21.

Fang, X., Huang, J., Wang, F., Zeng, L., Liang, H., & Wang, H. (2020). Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps. In *KDD '20, Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2697–2705). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3394486.3403320.

Fang, S., Pan, X., Xiang, S., & Pan, C. (2020). Meta-MSNet: Meta-learning based multi-source data fusion for traffic flow prediction. *IEEE Signal Processing Letters*.

Fang, S., Zhang, Q., Meng, G., Xiang, S., & Pan, C. (2019). GSTNet: Global spatial-temporal network for traffic flow prediction. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI-19* (pp. 2286–2293). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2019/317.

Feng, D., Wu, Z., Zhang, J., & Wu, Z. (2020). Dynamic global-local spatial-temporal network for traffic speed prediction. *IEEE Access, 8*, 209296–209307.

Fu, J., Zhou, W., & Chen, Z. (2020). Bayesian spatio-temporal graph convolutional network for traffic forecasting. arXiv preprint arXiv:2010.07498.

Fukuda, S., Uchida, H., Fujii, H., & Yamada, T. (2020). Short-term prediction of traffic flow under incident conditions using graph convolutional recurrent neural network and traffic simulation. *IET Intelligent Transport Systems*.

Ge, L., Li, H., Liu, J., & Zhou, A. (2019a). Temporal graph convolutional networks for traffic speed prediction considering external factors. In *2019 20th IEEE international conference on mobile data management (MDM)* (pp. 234–242). IEEE.

Ge, L., Li, H., Liu, J., & Zhou, A. (2019b). Traffic speed prediction with missing data based on TGCN. In *2019 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 522–529). IEEE.

Ge, L., Li, S., Wang, Y., Chang, F., & Wu, K. (2020). Global spatial-temporal graph convolutional network for urban traffic speed prediction. *Applied Sciences, 10*(4), 1509.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., et al. (2019). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 3656–3663).

Geng, X., Wu, X., Zhang, L., Yang, Q., Liu, Y., & Ye, J. (2019). Multi-modal graph interaction for multi-graph convolution network in urban spatiotemporal forecasting. arXiv preprint arXiv:1905.11395.

George, S., & Santra, A. K. (2020). Traffic prediction using multifaceted techniques: A survey. *Wireless Personal Communications, 115*(2), 1047–1106.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning* (pp. 1263–1272). PMLR.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems, 27*, 2672–2680.

Guo, K., Hu, Y., Qian, Z., Liu, H., Zhang, K., Sun, Y., et al. (2020). Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Guo, K., Hu, Y., Qian, Z., Sun, Y., Gao, J., & Yin, B. (2020). Dynamic graph convolution network for traffic forecasting based on latent network of Laplace matrix estimation. *IEEE Transactions on Intelligent Transportation Systems*.

Guo, K., Hu, Y., Qian, Z. S., Sun, Y., Gao, J., & Yin, B. (2020). An optimized temporal-spatial gated graph convolution network for traffic forecasting. *IEEE Intelligent Transportation Systems Magazine*.

Guo, R., Jiang, Z., Huang, J., Tao, J., Wang, C., Li, J., et al. (2019). BikeNet: Accurate bike demand prediction using graph neural networks for station rebalancing. In *2019 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 686–693). IEEE.

Guo, S., Lin, Y., Feng, N., Song, C., & Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 922–929).

Guo, J., Song, C., & Wang, H. (2019). A multi-step traffic speed forecasting model based on graph convolutional LSTM. In *2019 Chinese automation congress (CAC)* (pp. 2466–2471). IEEE.

Guo, G., & Yuan, W. (2020). Short-term traffic speed forecasting based on graph attention temporal convolutional networks. *Neurocomputing*.

Guopeng, L., Knoop, V. L., & van Lint, H. (2020). Dynamic graph filters networks: A gray-box model for multistep traffic forecasting. In *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)* (pp. 1–6). IEEE.

Haghighat, A. K., Ravichandra-Mouli, V., Chakraborty, P., Esfandiari, Y., Arabi, S., & Sharma, A. (2020). Applications of deep learning in intelligent transportation systems. *Journal of Big Data Analytics in Transportation, 2*(2), 115–145.

Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems* (pp. 1024–1034).

Han, X., Shen, G., Yang, X., & Kong, X. (2020). Congestion recognition for hybrid urban road systems via digraph convolutional network. *Transportation Research Part C (Emerging Technologies), 121*, Article 102877.

Han, Y., Wang, S., Ren, Y., Wang, C., Gao, P., & Chen, G. (2019). Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks. *ISPRS International Journal of Geo-Information, 8*(6), 243.

Hasanzadeh, A., Liu, X., Duffield, N., & Narayanan, K. R. (2019). Piecewise stationary modeling of random processes over graphs with an application to traffic prediction. In *2019 IEEE international conference on big data (Big data)* (pp. 3779–3788). IEEE.

He, S., & Shin, K. G. (2020a). Dynamic flow distribution prediction for urban dockless E-scooter sharing reconfiguration. In *Proceedings of the web conference 2020* (pp. 133–143).

He, S., & Shin, K. G. (2020b). Towards fine-grained flow forecasting: A graph attention approach for bike sharing systems. In *WWW '20, Proceedings of the web conference 2020* (pp. 88–98). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3366423.3380097.

He, Y., Zhao, Y., Wang, H., & Tsui, K. L. (2020). GC-LSTM: A deep spatiotemporal model for passenger flow forecasting of high-speed rail network. In *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)* (pp. 1–6). IEEE.

Heglund, J. S., Taleongpong, P., Hu, S., & Tran, H. T. (2020). Railway delay prediction with spatial-temporal graph convolutional networks. In *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)* (pp. 1–6). IEEE.

Henaff, M., Bruna, J., & LeCun, Y. (2015). Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163.

Hong, H., Lin, Y., Yang, X., Li, Z., Fu, K., Wang, Z., et al. (2020). Heteta: Heterogeneous information network embedding for estimating time of arrival. In *KDD '20, Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2444–2454). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3394486.3403294.

Hu, J., Guo, C., Yang, B., Jensen, C. S., & Chen, L. (2018). Recurrent multi-graph neural networks for travel cost prediction. arXiv preprint arXiv:1811.05157.

Hu, J., Yang, B., Guo, C., Jensen, C. S., & Xiong, H. (2020). Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. In *2020 IEEE 36th international conference on data engineering (ICDE)* (pp. 1417–1428). IEEE.

Huang, R., Huang, C., Liu, Y., Dai, G., & Kong, W. (2020). LSGCN: Long short-term traffic prediction with graph convolutional networks. In C. Bessiere (Ed.), *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI-20* (pp. 2355–2361). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2020/326, Main track.

Huang, Y., Zhang, S., Wen, J., & Chen, X. (2020). Short-term traffic flow prediction based on graph convolutional network embedded LSTM. In *International conference on transportation and development 2020* (pp. 159–168). Reston, VA: American Society of Civil Engineers.

James, J. (2019). Online traffic speed estimation for urban road networks with few data: A transfer learning approach. In *2019 IEEE intelligent transportation systems conference (ITSC)* (pp. 4024–4029). IEEE.

James, J. (2020). Citywide traffic speed prediction: A geometric deep learning approach. *Knowledge-Based Systems*, Article 106592.

Jepsen, T. S., Jensen, C. S., & Nielsen, T. D. (2019). Graph convolutional networks for road networks. In *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems* (pp. 460–463).

Jepsen, T. S., Jensen, C. S., & Nielsen, T. D. (2020). Relational fusion networks: Graph convolutional networks for road networks. *IEEE Transactions on Intelligent Transportation Systems*.

Jia, C., Wu, B., & Zhang, X.-P. (2020). Dynamic spatiotemporal graph neural network with tensor network. arXiv preprint arXiv:2003.08729.

Jiang, W. (2022). Graph-based deep learning for communication networks: A survey. *Computer Communications, 185*, 40–54.

Jiang, W., & Zhang, L. (2018). Geospatial data to images: A deep-learning framework for traffic forecasting. *Tsinghua Science and Technology, 24*(1), 52–64.

Jin, G., Cui, Y., Zeng, L., Tang, H., Feng, Y., & Huang, J. (2020). Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. *Transportation Research Part C (Emerging Technologies), 117*, Article 102665.

Jin, G., Xi, Z., Sha, H., Feng, Y., & Huang, J. (2020). Deep multi-view spatiotemporal virtual graph neural network for significant citywide ride-hailing demand prediction. arXiv preprint arXiv:2007.15189.

Kang, Z., Xu, H., Hu, J., & Pei, X. (2019). Learning dynamic graph embedding for traffic flow forecasting: A graph self-attentive method. In *2019 IEEE intelligent transportation systems conference (ITSC)* (pp. 2570–2576). IEEE.

Ke, J., Feng, S., Zhu, Z., Yang, H., & Ye, J. (2021). Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. *Transportation Research Part C (Emerging Technologies), 127*, Article 103063.

Ke, J., Qin, X., Yang, H., Zheng, Z., Zhu, Z., & Ye, J. (2021). Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network. *Transportation Research Part C (Emerging Technologies), 122*, Article 102858.

Kim, S.-S., Chung, M., & Kim, Y.-K. (2020). Urban traffic prediction using congestion diffusion model. In *2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia)* (pp. 1–4). IEEE.

Kim, T. S., Lee, W. K., & Sohn, S. Y. (2019). Graph convolutional network approach applied to predict hourly bike-sharing demands considering spatial, temporal, and global effects. *PLOS ONE, 14*(9), Article e0220782.

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations (ICLR '17)*.

Kong, X., Xing, W., Wei, X., Bao, P., Zhang, J., & Lu, W. (2020). STGAT: Spatial-temporal graph attention networks for traffic flow forecasting. *IEEE Access*.

Lee, K., Eo, M., Jung, E., Yoon, Y., & Rhee, W. (2021). Short-term traffic prediction with deep neural networks: A survey. *IEEE Access, 9*, 54739–54756.

Lee, D., Jung, S., Cheon, Y., Kim, D., & You, S. (2019). Demand forecasting from spatiotemporal data with graph networks and temporal-guided embedding. arXiv preprint arXiv:1905.10709.

Lee, K., & Rhee, W. (2019). Graph convolutional modules for traffic forecasting. CoRR, abs/1905.12256. URL: http://arxiv.org/abs/1905.12256. arXiv:1905.12256.

Lee, K., & Rhee, W. (2022). DDP-GCN: Multi-graph convolutional network for spatiotemporal traffic forecasting. *Transportation Research Part C (Emerging Technologies), 134*, Article 103466.

Lewenfus, G., Martins, W. A., Chatzinotas, S., & Ottersten, B. (2020). Joint forecasting and interpolation of time-varying graph signals using deep learning. *IEEE Transactions on Signal and Information Processing over Networks*.

Li, A., & Axhausen, K. W. (2020). Short-term traffic demand prediction using graph convolutional neural networks. *AGILE: GIScience Series, 1*, 1–14.

Li, C., Bai, L., Liu, W., Yao, L., & Waller, S. T. (2020). Knowledge adaption for demand prediction based on multi-task memory neural network. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 715–724).

Li, Z., Li, L., Peng, Y., & Tao, X. (2020). A two-stream graph convolutional neural network for dynamic traffic flow forecasting. In *2020 IEEE 32nd international conference on tools with artificial intelligence (ICTAI)* (pp. 355–362). IEEE.

Li, Y., & Moura, J. M. (2020). Forecaster: A graph transformer for forecasting spatial and time-dependent data. In *Proceedings of the twenty-fourth european conference on artificial intelligence*.

Li, J., Peng, H., Liu, L., Xiong, G., Du, B., Ma, H., et al. (2018). Graph CNNs for urban traffic passenger flows prediction. In *2018 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 29–36). IEEE.

Li, Z., Sergin, N. D., Yan, H., Zhang, C., & Tsung, F. (2020). Tensor completion for weakly-dependent data on graph for metro passenger flow prediction. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*.

Li, W., Wang, X., Zhang, Y., & Wu, Q. (2020). Traffic flow prediction over muti-sensor data correlation with graph convolution network. *Neurocomputing*.

Li, Z., Xiong, G., Chen, Y., Lv, Y., Hu, B., Zhu, F., et al. (2019). A hybrid deep learning approach with GCN and LSTM for traffic flow prediction. In *2019 IEEE intelligent transportation systems conference (ITSC)* (pp. 1929–1933). IEEE.

Li, Z., Xiong, G., Tian, Y., Lv, Y., Chen, Y., Hui, P., et al. (2020). A multi-stream feature fusion approach for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Li, L., Yan, J., Yang, X., & Jin, Y. (2019). Learning interpretable deep state space model for probabilistic time series forecasting. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 2901–2908).

Li, W., Yang, X., Tang, X., & Xia, S. (2020). SDCN: Sparsity and diversity driven correlation networks for traffic demand forecasting. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International conference on learning representations (ICLR '18)*.

Li, M., & Zhu, Z. (2021). Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 4189–4196).

Liao, B., Zhang, J., Wu, C., McIlwraith, D., Chen, T., Yang, S., et al. (2018). Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 537–546).

Lin, L., He, Z., & Peeta, S. (2018). Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C (Emerging Technologies), 97*, 258–276.

Liu, L., Chen, J., Wu, H., Zhen, J., Li, G., & Lin, L. (2020). Physical-virtual collaboration modeling for intra-and inter-station metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, J., Ong, G. P., & Chen, X. (2020). GraphSAGE-based traffic speed forecasting for segment network with sparse data. *IEEE Transactions on Intelligent Transportation Systems*.

Liu, R., Zhao, S., Cheng, B., Yang, H., Tang, H., & Yang, F. (2020). ST-MFM: A spatiotemporal multi-modal fusion model for urban anomalies prediction. In *Proceedings of the twenty-fourth european conference on artificial intelligence*.

Liu, L., Zhou, T., Long, G., Jiang, J., & Zhang, C. (2019). Learning to propagate for graph meta-learning. In *Advances in neural information processing systems* (pp. 1039–1050).

Lu, B., Gan, X., Jin, H., Fu, L., & Zhang, H. (2020). Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1025–1034).

Lu, Z., Lv, W., Cao, Y., Xie, Z., Peng, H., & Du, B. (2020). LSTM variants meet graph neural networks for road speed prediction. *Neurocomputing*.

Lu, Z., Lv, W., Xie, Z., Du, B., & Huang, R. (2019). Leveraging graph neural network with LSTM for traffic speed prediction. In *2019 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications, cloud & big data computing, internet of people and smart city innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 74–81). IEEE.

Lu, M., Zhang, K., Liu, H., & Xiong, N. (2019). Graph hierarchical convolutional recurrent neural network (GHCRNN) for vehicle condition prediction. arXiv preprint arXiv:1903.06261.

Luca, M., Barlacchi, G., Lepri, B., & Pappalardo, L. (2020). Deep learning for human mobility: a survey on data and models. arXiv preprint arXiv:2012.02825.

Luo, M., Du, B., Klemmer, K., Zhu, H., Ferhatosmanoglu, H., & Wen, H. (2020). D3P: Data-driven demand prediction for fast expanding electric vehicle sharing systems. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 4*(1), 1–21.

Lv, M., Hong, Z., Chen, L., Chen, T., Zhu, T., & Ji, S. (2020). Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Maas, T., & Bloem, P. (2020). Uncertainty intervals for graph-based spatio-temporal traffic prediction. arXiv preprint arXiv:2012.05207.

Mallick, T., Balaprakash, P., Rask, E., & Macfarlane, J. (2020). Graph-partitioning-based diffusion convolution recurrent neural network for large-scale traffic forecasting. *Transportation Research Record*, Article 0361198120930010.

Mallick, T., Balaprakash, P., Rask, E., & Macfarlane, J. (2021). Transfer learning with graph neural networks for short-term highway traffic forecasting. In *2020 25th international conference on pattern recognition (ICPR)* (pp. 10367–10374). IEEE.

Manibardo, E. L., Laña, I., & Del Ser, J. (2021). Deep learning for road traffic forecasting: Does it make a difference? *IEEE Transactions on Intelligent Transportation Systems*.

Mena-Oreja, J., & Gozalvez, J. (2020). A comprehensive evaluation of deep learning-based techniques for traffic prediction. *IEEE Access, 8*, 91188–91212.

Mohanty, S., & Pozdnukhov, A. (2018). Graph cnn+ lstm framework for dynamic macroscopic traffic congestion prediction. In *International workshop on mining and learning with graphs*.

Mohanty, S., Pozdnukhov, A., & Cassidy, M. (2020). Region-wide congestion prediction and control using deep learning. *Transportation Research Part C (Emerging Technologies), 116*, Article 102624.

Opolka, F. L., Solomon, A., Cangea, C., Veličković, P., Liò, P., & Hjelm, R. D. (2019). Spatio-temporal deep graph infomax. In *Representation learning on graphs and manifolds, ICLR 2019 workshop*.

Oreshkin, B. N., Amini, A., Coyle, L., & Coates, M. (2021). FC-GAGA: Fully connected gated graph architecture for spatio-temporal traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 9233–9241).

Ou, J., Sun, J., Zhu, Y., Jin, H., Liu, Y., Zhang, F., et al. (2020). STP-TrellisNets: Spatial-temporal parallel TrellisNets for metro station passenger flow prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1185–1194).

Pan, Z., Liang, Y., Wang, W., Yu, Y., Zheng, Y., & Zhang, J. (2019). Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1720–1730).

Pan, Z., Zhang, W., Liang, Y., Zhang, W., Yu, Y., Zhang, J., et al. (2020). Spatio-temporal meta learning for urban traffic prediction. *IEEE Transactions on Knowledge and Data Engineering*.

Park, C., Lee, C., Bahng, H., Tae, Y., Jin, S., Kim, K., et al. (2020). ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1215–1224).

Pavlyuk, D. (2019). Feature selection and extraction in spatiotemporal traffic forecasting: a systematic literature review. *European Transport Research Review, 11*(1), 6.

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

Pian, W., & Wu, Y. (2020). Spatial-temporal dynamic graph attention networks for ride-hailing demand prediction. arXiv preprint arXiv:2006.05905.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., & Hoffmann, H. (2019). Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 10772–10781).

Qin, T., Liu, T., Wu, H., Tong, W., & Zhao, S. (2020). RESGCN: Residual graph convolutional network based free dock prediction in bike sharing system. In *2020 21st IEEE international conference on mobile data management (MDM)* (pp. 210–217). IEEE.

Qin, K., Xu, Y., Kang, C., & Kwan, M.-P. (2020). A graph convolutional network model for evaluating potential congestion spots based on local urban built environments. *Transactions in GIS*.

Qiu, H., Zheng, Q., Msahli, M., Memmi, G., Qiu, M., & Lu, J. (2020). Topological graph convolutional network-based urban traffic flow and density prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Qu, Y., Zhu, Y., Zang, T., Xu, Y., & Yu, J. (2020). Modeling local and global flow aggregation for traffic flow forecasting. In *International conference on web information systems engineering* (pp. 414–429). Springer.

Ramadan, A., Elbery, A., Zorba, N., & Hassanein, H. S. (2020). Traffic forecasting using temporal line graph convolutional network: Case study. In *ICC 2020-2020 IEEE international conference on communications (ICC)* (pp. 1–6). IEEE.

Ren, Y., & Xie, K. (2019). Transfer knowledge between sub-regions for traffic prediction using deep learning method. In *International conference on intelligent data engineering and automated learning* (pp. 208–219). Springer.

Sánchez, C. S., Wieder, A., Sottovia, P., Bortoli, S., Baumbach, J., & Axenie, C. (2020). GANNSTER: Graph-augmented neural network spatio-temporal reasoner for traffic forecasting. In *International workshop on advanced analysis and learning on temporal data (AALTD)*. Springer.

Satorras, V. G., & Estrach, J. B. (2018). Few-shot learning with graph neural networks. In *International conference on learning representations*.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80.

Shao, K., Wang, K., Chen, L., & Zhou, Z. (2020). Estimation of urban travel time with sparse traffic surveillance data. In *Proceedings of the 2020 4th high performance computing and cluster technologies conference & 2020 3rd international conference on big data and artificial intelligence* (pp. 218–223).

Shen, Y., Jin, C., & Hua, J. (2020). TTPNet: A neural network for travel time prediction based on tensor decomposition and graph embedding. *IEEE Transactions on Knowledge and Data Engineering*.

Shi, H., Yao, Q., Guo, Q., Li, Y., Zhang, L., Ye, J., et al. (2020). Predicting origin-destination flow via multi-perspective graph convolutional network. In *2020 IEEE 36th international conference on data engineering (ICDE)* (pp. 1818–1821). IEEE.

Shi, X., & Yeung, D.-Y. (2018). Machine learning for spatiotemporal sequence forecasting: A survey. arXiv preprint arXiv:1808.06865.

Shin, Y., & Yoon, Y. (2020). Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*.

Shleifer, S., McCreery, C., & Chitters, V. (2019). Incrementally improving graph WaveNet performance on traffic prediction. arXiv preprint arXiv:1912.07390.

Song, C., Lin, Y., Guo, S., & Wan, H. (2020). Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 914–921).

Song, Q., Ming, R., Hu, J., Niu, H., & Gao, M. (2020). Graph attention convolutional network: Spatiotemporal modeling for urban traffic prediction. In *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)* (pp. 1–6). IEEE.

Sun, X., Li, J., Lv, Z., & Dong, C. (2020). Traffic flow prediction model based on spatio-temporal dilated graph convolution. *KSII Transactions on Internet & Information Systems*, *14*(9).

Sun, Y., Wang, Y., Fu, K., Wang, Z., Zhang, C., & Ye, J. (2021). Constructing geographic and long-term temporal graph for traffic forecasting. In *2020 25th international conference on pattern recognition (ICPR)* (pp. 3483–3490). IEEE.

Sun, J., Zhang, J., Li, Q., Yi, X., Liang, Y., & Zheng, Y. (2020). Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 1.

Tang, C., Sun, J., & Sun, Y. (2020). Dynamic spatial-temporal graph attention graph convolutional network for short-term traffic flow forecasting. In *2020 IEEE international symposium on circuits and systems (ISCAS)* (pp. 1–5). IEEE.

Tang, C., Sun, J., Sun, Y., Peng, M., & Gan, N. (2020). A general traffic flow prediction approach based on spatial-temporal graph attention. *IEEE Access*, *8*, 153731–153741.

Tedjopurnomo, D. A., Bao, Z., Zheng, B., Choudhury, F., & Qin, A. (2020). A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Tian, K., Guo, J., Ye, K., & Xu, C.-Z. (2020). ST-MGAT: Spatial-temporal multi-head graph attention networks for traffic forecasting. In *2020 IEEE 32nd international conference on tools with artificial intelligence (ICTAI)* (pp. 714–721). IEEE.

Varghese, V., Chikaraishi, M., & Urata, J. (2020). Deep learning in transport studies: A meta-analysis on the prediction accuracy. *Journal of Big Data Analytics in Transportation*, 1–22.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998–6008.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations*.

Wang, X., Chen, C., Min, Y., He, J., Yang, B., & Zhang, Y. (2018). Efficient metropolitan traffic prediction based on graph recurrent neural network. arXiv preprint arXiv:1811.00740.

Wang, Y., Guan, X., Cao, J., Zhang, N., & Wu, H. (2020). Forecast network-wide traffic states for multiple steps ahead: A deep learning approach considering dynamic non-local spatial correlation and non-stationary temporal dependency. *Transportation Research Part C (Emerging Technologies)*, *119*, Article 102763. http://dx.doi.org/10.1016/j.trc.2020.102763, URL: http://www.sciencedirect.com/science/article/pii/S0968090X20306756.

Wang, Q., Guo, B., Ouyang, Y., Shu, K., Yu, Z., & Liu, H. (2020). Spatial community-informed evolving graphs for demand prediction. In *Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML-PKDD 2020)*.

Wang, J., Kong, L., Huang, Z., & Xiao, J. (2021). Survey of graph neural network. *Computer Engineering*, *47*(4), 1–12.

Wang, B., Luo, X., Zhang, F., Yuan, B., Bertozzi, A. L., & Brantingham, P. J. (2018). Graph-based deep modeling and real time forecasting of sparse spatio-temporal data. arXiv preprint arXiv:1804.00684.

Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., et al. (2020). Traffic flow prediction via spatial temporal graph neural network. In *WWW '20, Proceedings of the web conference 2020* (pp. 1082–1092). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3366423.3380186.

Wang, S., Miao, H., Chen, H., & Huang, Z. (2020). Multi-task adversarial spatial-temporal networks for crowd flow prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1555–1564).

Wang, H.-W., Peng, Z.-R., Wang, D., Meng, Y., Wu, T., Sun, W., et al. (2020). Evaluation and prediction of transportation resilience under extreme weather events: A diffusion graph convolutional approach. *Transportation Research Part C (Emerging Technologies)*, *115*, Article 102619.

Wang, F., Xu, J., Liu, C., Zhou, R., & Zhao, P. (2020). MTGCN: A multitask deep learning model for traffic flow prediction. In *International conference on database systems for advanced applications* (pp. 435–451). Springer.

Wang, Y., Xu, D., Peng, P., Xuan, Q., & Zhang, G. (2020). An urban commuters' OD hybrid prediction method based on big GPS data. *Chaos. An Interdisciplinary Journal of Nonlinear Science*, *30*(9), Article 093128.

Wang, Y., Yin, H., Chen, H., Wo, T., Xu, J., & Zheng, K. (2019). Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 1227–1235).

Wang, C., Zhang, K., Wang, H., & Chen, B. (2020). Auto-STGCN: Autonomous spatial-temporal graph convolutional network search based on reinforcement learning and existing research results. arXiv preprint arXiv:2010.07474.

Wei, C., & Sheng, J. (2020). Spatial-temporal graph attention networks for traffic flow forecasting. In *IOP conference series: Earth and environmental science, Vol. 587*. IOP Publishing, Article 012065.

Wei, L., Yu, Z., Jin, Z., Xie, L., Huang, J., Cai, D., et al. (2019). Dual graph for traffic forecasting. *IEEE Access*.

Wright, M. A., Ehlers, S. F., & Horowitz, R. (2019). Neural-attention-based deep learning architectures for modeling traffic dynamics on lane graphs. In *2019 IEEE intelligent transportation systems conference (ITSC)* (pp. 3898–3905). IEEE.

Wu, T., Chen, F., & Wan, Y. (2018). Graph attention LSTM network: A new model for traffic flow forecasting. In *2018 5th international conference on information science and control engineering (ICISCE)* (pp. 241–245). IEEE.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., & Zhang, C. (2020). Connecting the dots: Multivariate time series forecasting with graph neural networks. In *KDD '20, Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 753–763). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3394486.3403118.

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI-19* (pp. 1907–1913). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2019/264.

Wu, Y., Tan, H., Qin, L., Ran, B., & Jiang, Z. (2018). A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C (Emerging Technologies)*, *90*, 166–180.

Wu, M., Zhu, C., & Chen, L. (2020). Multi-task spatial-temporal graph attention network for taxi demand prediction. In *Proceedings of the 2020 5th international conference on mathematics and artificial intelligence* (pp. 224–228).

Xiao, G., Wang, R., Zhang, C., & Ni, A. (2020). Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks. *Multimedia Tools and Applications*, 1–19.

Xie, Q., Guo, T., Chen, Y., Xiao, Y., Wang, X., & Zhao, B. Y. (2020). Deep graph convolutional networks for incident-driven traffic speed prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1665–1674).

Xie, P., Li, T., Liu, J., Du, S., Yang, X., & Zhang, J. (2020). Urban flow prediction from spatiotemporal data using machine learning: A survey. *Information Fusion*, *59*, 1–12.

Xie, Z., Lv, W., Huang, S., Lu, Z., Du, B., & Huang, R. (2019). Sequential graph neural network for urban road traffic speed prediction. *IEEE Access*.

Xie, Y., Xiong, Y., & Zhu, Y. (2020a). ISTD-GCN: Iterative spatial-temporal diffusion graph convolutional network for traffic speed forecasting. arXiv preprint arXiv:2008.03970.

Xie, Y., Xiong, Y., & Zhu, Y. (2020b). SAST-GNN: A self-attention based spatio-temporal graph neural network for traffic prediction. In *International conference on database systems for advanced applications* (pp. 707–714). Springer.

Xin, Y., Miao, D., Zhu, M., Jin, C., & Lu, X. (2020). InterNet: Multistep traffic forecasting by interacting spatial and temporal features. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 3477–3480).

Xiong, X., Ozbay, K., Jin, L., & Feng, C. (2020). Dynamic origin–destination matrix prediction with line graph neural networks and Kalman filter. *Transportation Research Record*, Article 0361198120919399.

Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., et al. (2020). Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908.

Xu, D., Dai, H., Wang, Y., Peng, P., Xuan, Q., & Guo, H. (2019). Road traffic state prediction based on a graph embedding recurrent neural network under the SCATS. *Chaos. An Interdisciplinary Journal of Nonlinear Science*, *29*(10), Article 103125.

Xu, Z., Kang, Y., Cao, Y., & Li, Z. (2020). Spatiotemporal graph convolution multifusion network for urban vehicle emission prediction. *IEEE Transactions on Neural Networks and Learning Systems*.

Xu, Y., & Li, D. (2019). Incorporating graph attention and recurrent architectures for city-wide taxi demand prediction. *ISPRS International Journal of Geo-Information*, *8*(9), 414.

Xu, D., Wei, C., Peng, P., Xuan, Q., & Guo, H. (2020). GE-GAN: A novel deep learning framework for road traffic state estimation. *Transportation Research Part C (Emerging Technologies)*, *117*, Article 102635.

Xu, X., Zheng, H., Feng, X., & Chen, Y. (2020). Traffic flow forecasting with spatial-temporal graph convolutional networks in edge-computing systems. In *2020 international conference on wireless communications and signal processing (WCSP)* (pp. 251–256). IEEE.

Yang, F., Chen, L., Zhou, F., Gao, Y., & Cao, W. (2020). Relational state-space model for stochastic multi-object systems. In *International conference on learning representations*.

Yang, S., Ma, W., Pi, X., & Qian, S. (2019). A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C (Emerging Technologies)*, *107*, 248–265.

Yao, X., Gao, Y., Zhu, D., Manley, E., Wang, J., & Liu, Y. (2020). Spatial origin-destination flow imputation using graph convolutional networks. *IEEE Transactions on Intelligent Transportation Systems*.

Ye, J., Sun, L., Du, B., Fu, Y., & Xiong, H. (2021). Coupled layer-wise graph convolution for transportation demand prediction. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 4617–4625).

Ye, J., Zhao, J., Ye, K., & Xu, C. (2020a). How to build a graph-based deep learning architecture in traffic domain: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

Ye, J., Zhao, J., Ye, K., & Xu, C. (2020b). Multi-STGCnet: A graph convolution based spatial-temporal framework for subway passenger flow forecasting. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Yeghikyan, G., Opolka, F. L., Nanni, M., Lepri, B., & Liò, P. (2020). Learning mobility flows from urban features with spatial interaction models and neural networks. In *2020 IEEE international conference on smart computing (SMARTCOMP)* (pp. 57–64). IEEE.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B. (2020). Multi-stage attention spatial-temporal graph networks for traffic prediction. *Neurocomputing*.

Yin, X., Wu, G., Wei, J., Shen, Y., Qi, H., & Yin, B. (2021). Deep learning on traffic prediction: Methods, analysis and future directions. *IEEE Transactions on Intelligent Transportation Systems*.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). Gnnexplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems* (pp. 9244–9255).

Yoshida, A., Yatsuhiro, Y., Hata, N., Higurashi, T., Tateiwa, N., Wakamatsu, T., et al. (2019). Practical end-to-end repositioning algorithm for managing bike-sharing system. In *2019 IEEE international conference on big data (Big data)* (pp. 1251–1258). IEEE.

Yu, L., Du, B., Hu, X., Sun, L., Han, L., & Lv, W. (2020). Deep spatio-temporal graph convolutional network for traffic accident prediction. *Neurocomputing*.

Yu, J. J. Q., & Gu, J. (2019). Real-time traffic speed estimation with graph convolutional generative autoencoder. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3940–3951.

Yu, B., Lee, Y., & Sohn, K. (2020). Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN). *Transportation Research Part C (Emerging Technologies)*, *114*, 189–204.

Yu, B., Li, M., Zhang, J., & Zhu, Z. (2019). 3D graph convolutional networks with temporal graphs: A spatial information free framework for traffic forecasting. arXiv preprint arXiv:1903.00919.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18* (pp. 3634–3640). International Joint Conferences on Artificial Intelligence Organization, http://dx.doi.org/10.24963/ijcai.2018/505.

Yu, B., Yin, H., & Zhu, Z. (2019). ST-UNet: A spatio-temporal U-network for graph-structured time series modeling. arXiv preprint arXiv:1903.05631.

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., et al. (2010). T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems* (pp. 99–108).

Zhang, Q., Chang, J., Meng, G., Xiang, S., & Pan, C. (2020). Spatio-temporal graph structure learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*.

Zhang, J., Chen, F., Cui, Z., Guo, Y., & Zhu, Y. (2020). Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, J., Chen, F., & Guo, Y. (2020). Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit. *IET Intelligent Transport Systems*.

Zhang, Y., Cheng, T., & Ren, Y. (2019). A graph deep learning method for short-term traffic forecasting on large road networks. *Computer-Aided Civil and Infrastructure Engineering*, *34*(10), 877–896.

Zhang, Y., Cheng, T., Ren, Y., & Xie, K. (2020). A novel residual graph convolution deep learning model for short-term network-based traffic forecasting. *International Journal of Geographical Information Science*, *34*(5), 969–995.

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

Zhang, Y., Dong, X., Shang, L., Zhang, D., & Wang, D. (2020). A multi-modal graph neural network approach to traffic risk forecasting in smart urban sensing. In *2020 17th annual IEEE international conference on sensing, communication, and networking (SECON)* (pp. 1–9). IEEE.

Zhang, N., Guan, X., Cao, J., Wang, X., & Wu, H. (2019). A hybrid traffic speed forecasting approach integrating wavelet transform and motif-based graph convolutional recurrent neural network. arXiv preprint arXiv:1904.06656.

Zhang, T., & Guo, G. (2020). Graph attention LSTM: A spatio-temperal approach for traffic flow forecasting. *IEEE Intelligent Transportation Systems Magazine*.

Zhang, K., He, F., Zhang, Z., Lin, X., & Li, M. (2020). Graph attention temporal convolutional network for traffic speed forecasting on road networks. *Transportmetrica B: Transport Dynamics*, 1–19.

Zhang, X., Huang, C., Xu, Y., & Xia, L. (2020). Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1853–1862).

Zhang, C., James, J., & Liu, Y. (2019). Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Access*, *7*, 166246–166256.

Zhang, Q., Jin, Q., Chang, J., Xiang, S., & Pan, C. (2018). Kernel-weighted graph convolutional network: A deep learning approach for traffic forecasting. In *2018 24th international conference on pattern recognition (ICPR)* (pp. 1018–1023). IEEE.

Zhang, T., Jin, J., Yang, H., Guo, H., & Ma, X. (2019). Link speed prediction for signalized urban traffic network using a hybrid deep learning approach. In *2019 IEEE intelligent transportation systems conference (ITSC)* (pp. 2195–2200). IEEE.

Zhang, Z., Li, M., Lin, X., Wang, Y., & He, F. (2019). Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C (Emerging Technologies)*, *105*, 297–322.

Zhang, W., Liu, H., Liu, Y., Zhou, J., & Xiong, H. (2020). Semi-supervised hierarchical recurrent graph neural network for city-wide parking availability prediction. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*.

Zhang, W., Liu, H., Liu, Y., Zhou, J., Xu, T., & Xiong, H. (2020). Semi-supervised city-wide parking availability prediction via hierarchical recurrent graph neural network. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, H., Liu, J., Tang, Y., & Xiong, G. (2020). Attention based graph covolution networks for intelligent traffic flow analysis. In *2020 IEEE 16th international conference on automation science and engineering (CASE)* (pp. 558–563). IEEE.

Zhang, Y., Lu, M., & Li, H. (2020). Urban traffic flow forecast based on FastGCRNN. *Journal of Advanced Transportation*, *2020*.

Zhang, J., Shi, X., Xie, J., Ma, H., King, I., & Yeung, D. Y. (2018). GaAN: Gated attention networks for learning on large and spatiotemporal graphs. In *34th conference on uncertainty in artificial intelligence 2018, UAI 2018*.

Zhang, Y., Wang, S., Chen, B., & Cao, J. (2019). GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction. In *2019 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Zhang, X., Zhang, Z., & Jin, X. (2020). Spatial-temporal graph attention model on traffic forecasting. In *2020 13th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI)* (pp. 999–1003). IEEE.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the thirty-first AAAI conference on artificial intelligence* (pp. 1655–1661).

Zhao, B., Gao, X., Liu, J., Zhao, J., & Xu, C. (2020). Spatiotemporal data fusion in graph convolutional networks for traffic prediction. *IEEE Access*.

Zhao, T., Liu, Y., Neves, L., Woodford, O., Jiang, M., & Shah, N. (2021). Data augmentation for graph neural networks. In *Proceedings of the 30th international joint conference on artificial intelligence*. AAAI Press.

Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., et al. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Zhao, H., Yang, H., Wang, Y., Wang, D., & Su, R. (2020). Attention based graph bi-LSTM networks for traffic forecasting. In *2020 IEEE 23rd international conference on intelligent transportation systems (ITSC)* (pp. 1–6). IEEE.

Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*.

Zheng, B., Hu, Q., Ming, L., Hu, J., Chen, L., Zheng, K., et al. (2020). Spatial-temporal demand forecasting and competitive supply via graph convolutional networks. arXiv preprint arXiv:2009.12157.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open, 1*, 57–81.

Zhou, Q., Gu, J.-J., Ling, C., Li, W.-B., Zhuang, Y., & Wang, J. (2020). Exploiting multiple correlations among Urban regions for crowd flow prediction. *Journal of Computer Science and Technology, 35*, 338–352.

Zhou, X., Shen, Y., & Huang, L. (2019). Revisiting flow information for traffic prediction. arXiv preprint arXiv:1906.00560.

Zhou, Z., Wang, Y., Xie, X., Chen, L., & Liu, H. (2020). RiskOracle: A minute-level citywide traffic accident forecasting framework. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34*.

Zhou, Z., Wang, Y., Xie, X., Chen, L., & Zhu, C. (2020). Foresee urban sparse traffic accidents: A spatiotemporal multi-granularity perspective. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, F., Yang, Q., Zhang, K., Trajcevski, G., Zhong, T., & Khokhar, A. (2020). Reinforced spatio-temporal attentive graph neural networks for traffic forecasting. *IEEE Internet of Things Journal*.

Zhou, F., Yang, Q., Zhong, T., Chen, D., & Zhang, N. (2020). Variational graph neural networks for road traffic prediction in intelligent transportation systems. *IEEE Transactions on Industrial Informatics*.

Zhu, J., Han, X., Deng, H., Tao, C., Zhao, L., Wang, P., et al. (2022). KST-GCN: A knowledge-driven spatial-temporal graph convolutional network for traffic forecasting. *IEEE Transactions on Intelligent Transportation Systems*.

Zhu, H., Luo, Y., Liu, Q., Fan, H., Song, T., Yu, C. W., et al. (2019). Multistep flow prediction on car-sharing systems: A multi-graph convolutional neural network with attention mechanism. *International Journal of Software Engineering and Knowledge Engineering, 29*(11n12), 1727–1740.

Zhu, J., Wang, Q., Tao, C., Deng, H., Zhao, L., & Li, H. (2021). AST-GCN: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access, 9*, 35973–35983.

Zhu, H., Xie, Y., He, W., Sun, C., Zhu, K., Zhou, G., et al. (2020). A novel traffic flow forecasting method based on RNN-GCN and BRB. *Journal of Advanced Transportation, 2020*.