Deep Learning Based Feature Engineering for Discovering Spatio-Temporal Dependency in Traffic Flow Forecasting

by

Hongfan Mu

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master of Computer Science(Applied Artificial Intelligence)

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Examining Committee

The following served on the Examining Committee for this thesis.

Internal Member(s): Hussein Al Osman.
Associate Professor, School of Electrical Engineering and Computer Science
University of Ottawa

Richard Yu
Professor, Department of Systems & Computer Engineering
Carleton University

Supervisor: Azzedine Boukerche
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University of Ottawa regulations concerning plagiarism, including those regarding consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

Intelligent transportation systems (ITS) have garnered considerable attention for providing efficient traffic management solutions. Traffic flow forecasting is a crucial component of it which serves as the foundation for various state-of-the-art deep learning approaches. Initially, researchers recognized that significant temporal changes from traffic flow data for modelling. However, as researchers delved deeper into the underlying correlations within traffic flow data, they discovered that spatial information from the road network also plays a crucial role in accurate forecasting. Consequently, deep learning methods that incorporate Spatio-temporal representation have been employed to address traffic flow forecasting.

Although recent solutions to this problem are impressive, it is essential to discuss the reasoning behind the architecture of the model. The expression of each feature relies on selecting appropriate models for feature extraction and designing architectures that minimize information loss during modeling.

In this thesis, the work focuses on graph-based Spatio-temporal feature engineering. The experiments are divided into two parts: 1). explores the efficient architecture for expressing spatial-temporal information by considering both different sequential modelling approaches. 2). Based on the result obtained, the second experiment focuses on multi-scale modelling to capture informative Spatio-temporal feature. We propose a model that incorporates sequential modeling and captures multi-scale Spatiotemporal semantics by employing residual connections in different hierarchy. We validate our model using three datasets, each containing varying information for extraction. Taking into account the dataset characteristics and the model structure, our model outperforms the baselines and state-of-the-art models.

The experimental results indicate that the performance of sequential modeling and multi-scale semantics, combined with thoughtful model design, significantly contribute to the overall forecasting performance. Furthermore, our work serves as inspiration for expressive data mining methods that rely on appropriate feature extraction models and architecture design, taking into consideration the information content within the dataset.

Acknowledgements

It has been a tough time during the pandemic for students like me to continue their research, whether due to physical or mental conditions. However, just as Nietzsche, my favourite philosopher, said: "What does not kill me, makes me stronger." So first of all, I want to thank the current times we live in, as they have given me no choice but to push through, making me the stronger person I am today.

Next, I sincerely thank my supervisor, Prof. Azzedine Boukerche, for his patience and support during my whole graduate study. He is always there and welcomes me every time I need a helping hand. I needed help from him to complete my research. I also want to thank Dr. Noura Aljeri, who encouraged me when I doubted myself and almost gave up; she affirmed the value of my work, helped me pace my research, and encouraged me to complete all my work. I also genuinely appreciate the people I worked with during my graduate period - Dr. Mohammad Alnabhan, Dr. Yazan Otoum, and Prof. Paula Branco - who encouraged me. They affirmed my working ability when we worked together in the courses ITI1120, CSI3131, and CSI2132, which made me feel like "Yes, I can" when I felt upset and doubted my ability. I sincerely appreciate my undergraduate supervisor, Prof. Jiang Zhong, at Chongqing University; with his inspiration, I found my interest in Artificial Intelligence and continued further study in this area.

My most enormous thanks go to my parents, the best parents in the world! They support, believe in, guide, and comfort me whenever I feel like life is saying NO to me. My family protects me, makes me feel safe, and gives me the confidence to overcome every difficulty. I feel fortunate to have parents like them!

Also, big thanks to my friends, who gathered when they thought they needed to be there, shared ups and downs, comforted me and accompanied me. My big sister Qiyue Wu, every moment she was just there and helped me no matter what, in everyday life and at work. My good friends, Haoru Xing, and Xia Wu, Yechen Shen, Jinwei Li, make me laugh every time I spend time with them. Further, thank you to all the friends that I have not mentioned. Thanks for showing up in my life.

I hope everyone I love is also beloved, especially my puppy.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Traffic management is a ubiquitous and global issue that continues to grow in importance due to the increase of vehicles on the road every year. More and more, road users can potentially bring continuous traffic congestion every minute, which can decrease the utilization of traffic efficiency. Unpredictable traffic conditions can reduce traffic management efficiency and deteriorate passengers' travel experience [16]. Hence, an efficient and accurate traffic management method is required to solve such problems. In modern society, intelligent transportation systems (ITS) aim to establish reliable connections between roads, vehicles, and people, leading drivers to travel safely and freely on the road, which could improve traffic efficiency [142]. Proposing suitable ITS solutions is also the basis for deploying the Internet of Things (IoT) in future transportation systems.

Traffic flow forecasting is of great impotence to ITS. The reason for it is that the accurate prediction of traffic conditions is regarded as the base of a successful ITS [21]. Since state-of-the-art traffic flow forecasting methods can contribute to reducing the possibility of potential traffic accidents on the road, attaining reasonable and significant utilization of traffic road networks leads to alleviating urban traffic congestion efficiently. Besides, common applications in the traffic field, such as real-time traffic signal control [39,105,118, 120], traffic demand [110,119], route guidance [55,62,79], and automatic navigation [3,7] are all based on the guidance of accurate traffic flow prediction, which shows the significance of reliable and proper traffic flow prediction.

The traffic flow prediction problem generally predicts future traffic conditions based on historical traffic flow observations and current real-time traffic information. In the early days, traffic flow forecasting mainly focused on extracting the temporal features from observed data [63]. It is because traffic conditions fluctuate over time. As research in this area progressed, it was found that the topological structure of road networks also contributed to predictions. As a result, research has an increased focus on finding traffic forecasting solutions based on the combination of spatial and temporal dependencies mined from traffic flow data.

Progress in this area has shown that spatial information contributes to traffic flow forecasting, as the target road sections are effected by local neighbours [12]. Further, it was later discovered that spatial dependency also contains information on spatial dynamics that refers to changing traffic flow affected by nearby regions. For example, the traffic volume of an upstream section would vary because it gets directly affected by downstream sections, which also shows the same effect in turn [27]. To better model the complicated conditions of traffic flow data, researchers have started to consider solutions from three perspectives: spatial dependencies, temporal dependencies, and external influences [124]. However, due to the limited datasets available to account for external factors (weather conditions, car accidents, sports events, etc.), most models mainly focus on capturing the spatial and temporal correlations, with less consideration for the external factors. Given this, we note that particular attention needs to be given to the hidden relationship behind the Spatio-temporal dependency that has been revealed to contribute to the final direction.

Since the solution to traffic flow prediction requires processing the complex and non-linear traffic conditions from real-time data [81], Vlahogianni et al. [96] claimed that data-driven methods, representing the decisions on traffic management that are made based on data analysis, can effectively predict future traffic states. Data-driven methods, specifically machine learning methods, such as K-nearest neighbour (KNN) [112] and support vector machine (SVM) [51] [77] [35], have been introduced to handle the complex and non-linear data in traffic flow forecasting. However, most machine learning models require manual feature selection before modeling real traffic conditions. In other words, it is essential to manually determine the specific features used in machine learning models in advance. In machine learning modeling, if many features are accepted for prediction, the real-time performance of the final result will be affected due to the large number of parameters generated, which leads to a time-consuming problem. Further, noisy data may be introduced into the model. Generally, machine learning approaches employ correlation analysis to support discovering which kinds of features should be selected as inputs. Thus, high-quality inputs can contribute to more accurate predictions. However, Oswald et al. [76] found that non-parametric methods, referring to the models without parametric presets, can provide better results compared to parametric machine learning models because of their performance in capturing the non-linearity feature of traffic flow data. Notably, the proliferation of advanced transport detection systems has caused massive quantities of traffic data to be generated every minute [44]. It is problematic for machine learning approaches, which are known to struggle when solving such "big data" problems (in an efficient and timely manner) due to their computation requirements. Conversely, the outstanding performance of feature learning and end-to-end modeling on mining information from big data has led to deep learning techniques being considered for addressing the traffic flow forecasting problem.

As a basic definition of the problem, from the temporal perspective, traffic flow data is in a time-varying sequence, where the current traffic state is related to the next state and affected by previous states. Hence, the question of why current research has developed solutions focused researchers on mining potential characteristics from sequential traffic flow data [59]. The typical temporal-based models are initially designed for modeling time-series data to handle any sequential characteristics. Popular models, such as the SVM, recurrent neural network (RNN), and its varied models (Long Short-Term Memory Networks (LSTM), Gated Recurrent Unit(GRU), etc.), variant CNN, and transformer are adopted for temporal dependency extraction. Compared to the RNN model with regular data-driven machine learning methods (i.e., BPNN and SVM), Ma et al. [72], proved to be the more suitable NN-based models regarding effectiveness and efficiency.

From the spatial perspective, the discovery of the utilization of spatial information is based on data generated from the topological traffic road network. Thus, the traffic flow data initially has rich information in the temporal dimension and the spatial domain [18, 29, 63]. The models that better perform in describing spatial correlations in traffic networks are the NN-based models, such as convolutional neural networks (CNN) [], graph Neural Networks (GNN), and their corresponding variants. These all show improvements in capturing the spatial dependency of traffic flow data [116].

In other words, deep learning-based models have shown influential temporal and spatial feature extraction performance when solving traffic flow prediction problems.

To establish a basis for our research, we reviewed several surveys which provide an introduction to deep learning techniques applied to modeling temporal dependencies of traffic data [13, 15], how to describe spatial correlations in traffic networks and mine Spatio-temporal relationships in the traffic domain [26, 56, 56, 59, 91, 109, 116]. However, the common points of all the surveys - introducing the work on traffic flow prediction problems from the techniques perspective, solely providing the differences of deep neural network architectures, and how each model is used for the proposed solutions - ignored the various dependencies of the problem and technologies for solving the dependencies in detail. For example, the exemplary work of Do et al. [26] presented the different types of neural-based networks applied to short-term traffic state prediction. Their work focused on giving an overview of differentiating the characteristics of neural networks that can be utilized for traffic prediction; however, it needed more discussion on how to depict the spatial and temporal dependencies that determine traffic flow forecasting. Thus, it is worth exploring the specific dependencies arising in real scenarios that determine traffic conditions and how deep learning models or, more specifically, the data-driven methods based on the neural network models applied to extract the corresponding dependency in traffic flow prediction, that also contribute to Spatio-temporal modeling.

In fact, for the study of traffic flow forecasting, we address the importance of understanding the problem first and then modeling the influential predictors that contribute to the problem. We believe that the key to obtaining the appropriate solution is correctly understanding the problem and analyzing the dependencies affecting the traffic flow rather than considering the advanced models to fix the problem while ignoring the accurate dependency modeling. Hence, we believe analyzing the spatial and temporal dependencies will contribute to the traffic flow forecasting problem instead of just applying the multiple

NN-based models to traffic flow forecasting.

Overall, most of the work we reviewed primarily discussed solutions to the problem in the traffic domain without discussing the common relations between one another. They need a comprehensive review of building a deep learning architecture that considers the perspective from both the spatial and temporal dependency in the traffic domain.

1.2 Objectives

The main objectives of our paper are given below:

- Survey the solutions employing the graph neural network model and its variants to describe the different spatial dependencies under the subcategories of, for example, spatial locality, spatial heterogeneity, etc., in traffic flow prediction; and temporal methods for extracting temporal information from traffic flow data.

- Compare the performance of various spatial models based on graph neural networks (GCN and its variants, GAT, etc.) and temporal models (such as RNN and its variants and CNN) to extract the spatial and temporal features from the same data set. We focus on the performance of the models under the evaluation of MAE, MAPE, and RMSE (the models we found consistently considered in other work), with an accuracy of R^2 (a variance that we found most reviewed work omitted).

- Explore the contribution of the Spatio-temporal dependency to the problem and how to extract it, compared to the sole spatial or temporal information.

- Discuss the effects of different sequential modeling on the informative expression of Spatio-temporal features. Moreover, discuss the impacts of constructing the different architectures of the model that enhance the feature representation, such as multi-scale feature fusion, used to enhance information gained on different levels and scales.

- Propose a new solution using the GNN-based model combined with the temporal models that perform better when compared to current state-of-the-art models.

1.3 Contributions

To sum up, the main contributions of our work are:

- To our knowledge, being the first to provide an investigation of mining the complex subcategory of spatial and temporal information that can be mined from traffic flow data. Current research in this domain demonstrates the work for solutions in traffic flow forecasting by only introducing the different techniques involved with the problem, while less work is in describing the explicit dependencies that affect the issue. Nevertheless, we consider it essential to understand the traffic flow forecasting problem with detailed dependencies and provide solutions based on the dependencies, such as the periodicity in the temporal domain, the geographical scale in the spatial domain, etc. We systematically outline the classification of the different dependencies in temporal, spatial, and Spatio-temporal aspects that researchers pay attention to addressing.

- We present the change of the problem definition in traffic flow forecasting, from mining solely temporal information to considering both Spatio-temporal features, in hopes of inspiring future research and potential solutions in this area. We also show the trend of corresponding techniques researchers considered to solve the problem. For example, the trend to spatial dependency extraction was from CNN for extracting grid-structured data to GNNs that capture graph topology.

- We propose a new end-to-end spatial-temporal model that focuses on utilizing multi-scale temporal and spatial information to improve the semantic expression of Spatial-temporal features. The multi-scale Spatial-temporal dependency is extracted from the different layers of the hierarchical architecture in the vertical perspective and the different sizes in the horizontal perspective, thus enabling effective time-series prediction for traffic flow forecasting with the informative Spatio-temporal features gathered from traffic data, especially for long term prediction. We explore how each spatial and temporal component and the detailed residual parts can contribute to the prediction with an ablation study and analysis of the structure on constructing the spatial and temporal components. We discover the impact of the scale size on spatial information extraction on the traffic map and the number of temporal layers exploiting the hidden temporal dynamics.

- We conduct experiments on mining the different amounts of Spatio-temporal information contained within the different lengths of the historical input when designing the model. We discuss the relationship between the input sequence, the historical horizon, and the information contained.
- We highlight the future directions that should receive further attention in the traffic flow forecasting area. We pointed out the potential improvement in the reconstruction of graph structure and the consideration of the multi-source spatial and temporal dependency. We also discuss whether the weights should be considered in applying Spatio-temporal dependency and the representation learning of the feature to express richer information.

1.4 Thesis outline

The rest of the paper is organized as follows:

Chapter 2: we first define the traffic flow forecasting problem, considering time-series information with and without spatial information. Then we propose a new taxonomy, derived from current state-of-the-art research, that details the subcategories of spatial and temporal dependencies relevant to the solution of traffic flow forecasting. As part of our taxonomy, we present suitable methods for depicting fine-grained representation based on corresponding neural network-based models.

Chapter 3 elaborates on the detailed subcategory of spatial and temporal dependency. We categorize and summarize the comprehensive review of the different categories under the temporal, spatial, and Spatio-temporal dependency to show the detailed classification of the subcategory dependency. We also discuss the deep learning techniques under these specific dependencies.

Chapter 4 provides the perspectives on how the current works consider the feature engineering for the solutions. Because the recent work adopts multiple temporal and spatial dependencies in the prediction instead of the single dependency considered, feature engineering from different perspectives can help to contribute to the model performance. Researchers proposed the solution by figuring out the subcategory of the dependencies that domain for the prediction and tried to apply expressive feature engineering to enhance the related information. From the recent research, we learned that the researchers trended to discuss the embedding methods to obtain richer feature information which helps to express the Spatio-temporal information of the data better. Most of the paper rebuilt the input considering the multi-dimensions of the raw data to represent the spatial and temporal correlations. Previous work preferred directly accepting raw data as the input, while current methods include the pre-defined matrix, multilayer perceptron, etc. Ire considered reconstructing the inputs, which helped to receive good results.

Chapter 5 provides the methodologies for this thesis work: exploratory data analysis for exploring the dataset as showed in Chapter 6; spatial component is showed to extract spatial information used in this thesis work; temporal component is for the temporal extraction; Sequential modeling and parallel modeling provide the view on building architecture that usually is considered for the Spatio-temporal modeling; feature fusion discuss the when fusing the information from the different level.

We conduct two experiments in this paper. For the first experiment (Chapter 6), we focus on how different modeling architecture affects Spatial-temporal information expression and its importance compared to the sole spatial or temporal model in traffic flow data. We introduce the different architectures for building spatial, temporal, and spatial-temporal dependency models in the traffic flow prediction problem. Furthermore, we show the two sequential architectures that combine spatial graph and temporal representations and how they deliver powerful representations for embedding Spatio-temporal information of traffic flow data. We consider the solution with Graph Convolutional Neural Network (GCN) for spatial information representation, Gated Recurrent Unit (GRU) for temporal information construction, and two sequential feature extractions for Spatial-temporal dependency. Our experiments show that the models that capture comprehensive Spatio-temporal information with sequential feature extraction outperform the sole spatial and temporal models. Further, we find that the spatial-then-temporal structure outperforms the temporal-then-spatial structure in traffic flow data. Moreover, we find that if a dataset is not big enough, the performance of extracting Spatio-temporal dependency may not be obvious, indicating the need for the fine-grained extraction of spatial-temporal dependency to mine the deeper information, which leads to the research in Chapter 7.

After exploring the first experiment, the second (Chapter 7) aims to extract Spatio-temporal information with spatial-then-temporal sequential modeling at a more fine-grained level. We introduce a multi-scale Spatio-temporal model for traffic flow prediction problems based on the Graph Convolutional Neural Network (GCN) and Convolutional Neural Network (CNN) for spatial and temporal feature extraction. Furthermore, the multi-scale Spatio-temporal semantics has been extracted based on the hierarchical architecture. Experiments show that the model captures comprehensive Spatio-temporal correlations with multi-scale semantics and outperforms the features extracted from the single domains and non-multi-scale models. We conduct an ablation study and a study on the different architectural structures, which reveal the importance of the model architecture for Spatio-temporal feature extraction, particularly for long-term predictions. To explain the effectiveness of our proposed ideas, two real-world datasets (PeMS04 and METR-LA) are evaluated. The size of the datasets differs based on the specific requirements necessary to evaluate each experiment.

In Chapter 8, we further discuss future directions for traffic flow forecasting, such as the consideration of information expression given the graph reconstruction, the trade-off for utilization of the temporal information and spatial information, the potential of techniques for traffic flow prediction that introduce multi-model spatial and temporal correlations, and the more expressive representation learning methods of Spatio-temporal feature vectors.

Chapter 2

Problem Statement

2.1 Problem definition and solution perspectives

2.1.1 Problems defined without spatial dependency

In this chapter, we present the problem definition based on the time-series problem to introduce spatial dependency into modeling. In table 2.3 and 2.2, we demonstrate the corresponding techniques in temporal and spatial dependency modeling.

Table 2.1: Summary of models that describe temporal correlations behind the traffic flow data.

Spatial / Temporal	Model Category	Gradient Vanishing	Long-distance information	Time cost	Reference
Temporal	RNN	Yes	No	Gradient vanishing; Not able to capture the long-distance time-series information	[143]
	LSTM	No, the gated mechanism to avoid gradient vanishing	Yes, adding cell state to store the long distance information	Yes, the massive parameters	[23] [107] [33] [126]
	GRU	No, gated mechanism to avoid gradient vanishing	YES	No, fewer parameters	[131] [54] [141] [133] [90]
	CNN	No	Yes, by stacking layers	No, the parallel computing for the architecture	[42] [89] [121]
	TCN	No, residual connections	Yes, dilated convolution		[135] [58] [6] [92]

2.1.2 Problems defined without spatial dependency

The traffic flow forecasting problem can generally be defined as a time series problem since it aims to predict future traffic indicators over the time step. As given the historical traffic indicators denoted as $X = [x^{t-T+1}, \ldots, x^t]$ over the past T time slots, the future traffic predictors over the next T' time slices is represented as $Y = [y^{t+1}, \ldots, y^{t+T'}]$. Given the relevant historically observed data till time slice t, the $t+T'$ th-step traffic flow forecasting problem can be stated as

$$Y = F(X),$$

$$[y^{t+1}, \ldots, y^{t+T'}] = F([x^{t-T+1}, \ldots, x^t]),$$

where

$$T' = \begin{cases} 1, & one - step\ prediction \\ n, & multi - step\ prediction \end{cases}$$

Some statistical theories, including the history average ARMA model, Kalman filtering model, linear regression, and non-parametric regression [36], are the first ones that were introduced into traffic flow forecasting. They are relatively simple due to the assumption of the same patterns and characteristics of the future conditions as that of the historical flow data [85].

However, as a nonlinear time-varying system, the performance of the real-world traffic conditions is significantly affected by the complex and irregular previous states on the road. Besides, the variables of statistical approaches are manually selected, relying on domain knowledge. Moreover, because of the out-of-dated equipment used for collecting the traffic data on the road, the usable observations in the traffic domain are limited; only part of the traffic conditions can be reflected due to bias. Therefore, classic mathematical models are adequate for simple tasks and limited datasets. However, due to complicated realistic conditions and requirements for the robustness and accuracy of the approaches, a more reliable approach is expected to be explored.

The machine learning technique is regarded as the new solution for modeling the time-series traffic data after classic statistic approaches. Compared with the statistical methods, Yang et al. [112] demonstrated the impressive ability of data-driven strategies to model non-linear correlations. It implicates that the potentiality of data-driven approaches, such as K-nearest neighbor (KNN) [51], support vector machine (SVM) [77] [35] [76], random forest, and NN-based models, on the complex traffic conditions. It has been demonstrated that machine learning algorithms outperform in capturing non-linearity, high-dimensional changes due to their capacity to mine information and understand the underlying relationship between historical traffic data and future data. Besides, machine learning models, such as SVM [74], obtain higher accuracy under a limited dataset.

However, realistic problems should be considered by applying the machine learning technique to the traffic flow prediction solution—first, feature quality. Since machine learning models require manual feature selection before putting in models, it can lead to unpredictable errors if the features are inappropriate. Second, time cost. Due to its structure, machine learning methods can not utilize traffic data efficiently. Furthermore, since the advanced intelligent transport equipment on the road, and the various data sources, such as video image processors, radar sensors, Global Positioning System (GPS), cellular phones, social media, etc. [85] generated in the explosion, dealing with "Big Data" efficiently is still a problem of machine learning methods [142]. Hence, the introduction of NN-based models in the traffic flow prediction area is expected because of their powerful ability to derive time series patterns from large amounts of data.

RNN Multiple NN-based networks, such as the classic recurrent neural network (RNN), LSTM, and GRU have been adapted to the traffic prediction task because of their capability of modeling the basic temporal dependencies as sequentiality of the traffic flow data.

In the recurrent neural network (RNN), the input of an RNN layer contains information (X) under the current time step t and a hidden state (h) that stores the historical information corresponding to the previous sequence in $t-1$ time steps. Hence, RNN is better to describe the handling of the explicit temporal order from a consecutive traffic record. As we assume the historical indicators over T time slices in sequence as $X = [x^{t-T+1}, x^{t-T+2}, ..., x^t]$, we can have the hidden state for next step as

$$h^t = \sigma(U \cdot x^t + W \cdot h^{t-1} + b),$$

where U, W, b are the learnt weights and bias, and σ is the active function. If we want

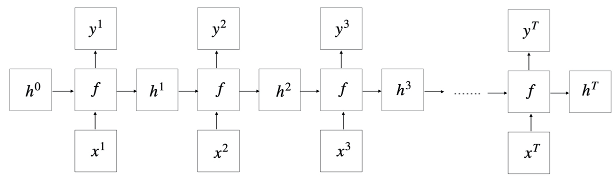


Figure 2.1: The structure of RNN model

to obtain the output of the current time step, we can have

$$y^t = \sigma(V \cdot h^t + c),$$

where V is the weight to be learnt and c is the bias. The structure of RNN is shown in Fig.2.1. We can see that the RNN network has shown better performance than regression models and other machine learning time series models in expressing the context of observations over time [23].

However, as we can see from Fig.2.1, in RNN, a neuron can only accept the information from the previous time step. In other words, the distance of the time steps between the current state and the corresponding time step should not be far. We have to notice that the time-series forecasting in the real-world traffic flow prediction owns the expression on long-term temporal dependency. For example, the traffic flow consistently repeating patterns over time shows the characteristic of periodicity, thus leading to the fact that the model is expected in learning the information over a long distance under certain conditions. Besides, the existing vanilla RNN faces the problem of vanishing gradient - the amount of the extracted information may decrease exponentially as the distance of time steps increases due to the error of the obtained training parameters.

LSTM The RNN variant, long-term short-term memory (LSTM) network, was proposed [49] to avoid the inherent deficiencies of RNN because: first, it was developed to deal with the vanishing gradient and explosion gradient of the classic RNN in the back-propagation process. Besides, RNN only considers the state at the most recent moment, while LSTM adds a filter function to the past state on the basis of RNN, thus preserving the more long-term sequential information which has more influence on the current moment, instead of simply choosing the most recent state. What's more, LSTM employs the sigmoid function as a gate to filter the state or input to control the state of the transmission information, enabling better performance. In a sense, LSTM can store the memory from a longer distance, remembering information within a long time and overlooking unimportant information. It performs better to process time-series data of traffic data in the long distance [136]. The basic architecture of LSTM is similar to RNN while introducing a cell state to store the historical information and adapting forget gate to control the past information from the previous unit, which can be seen from Fig. 2.2. The architecture of LSTM contains three gates to process the time-series data: forget gate, input gate, and output gate.

The output of the forget gate is f^t, which represents the probability of forgetting the previous hidden state:

$$f^t = \sigma(U_f \cdot x^t + W_f \cdot h^{t-1} + b_f)$$

, where the parameters are similar to that in RNN. The active function σ is usually $softmax$, thus, leading to the output of the forget gate between $[0, 1]$

The input gate has two components:

$$\begin{aligned} i^t &= \sigma(U_i \cdot x^t + W_i \cdot h^{t-1} + b_i) \\ a^t &= tanh(U_a \cdot x^t + W_a \cdot h^{t-1} + b_a), \end{aligned} \tag{2.1}$$

where $U_i, W_i, b_i, U_a, W_a, b_a$ are the weights and bias.

The cell state is updated with the obtained output from forget gate and input gate as:

$$C^t = C^{t-1} \odot f^t + i^t \odot a^t,$$

where $\odot$ is the hadamard product.

Hence, we can have the output of the current time step as:

$$\begin{aligned} o^t &= \sigma(U_o \cdot x^t + W_o \cdot h^{t-1} + b_o) \\ h^t &= o^t \odot tanh(C^t), \end{aligned} \tag{2.2}$$

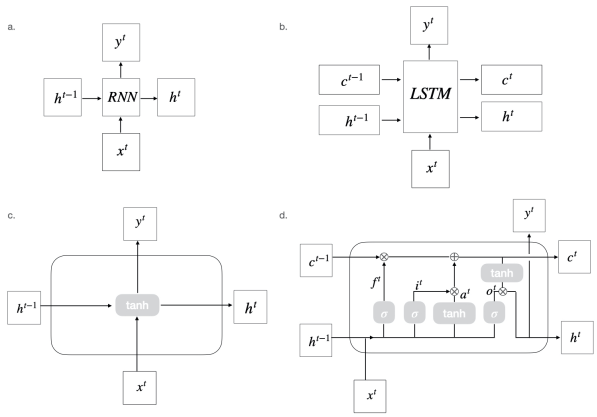


Figure 2.2: The structure comparison of RNN and LSTM model. Both of RNN and LSTM have hidden cells to retain the information from previous intervals where the LSTM introduces a cell state to remove or add information to the memory.

GRU Another variant of RNN is GRU, which is proposed with a simple but more efficient structure than both LSTM and RNN. It has fewer parameters but owns the same impact as LSTM in temporal modellling [116]. GRU owns a similar simple structure as RNN but with different operations inside the GRU unit: two gates cell - reset gate and update gate. The reset gate controls the importance of the hidden state(0/1), and the update gate decides if the previous cell state should be updated with the current hidden state or not.

$$\begin{aligned} r^t &= \sigma(U_r \cdot x^t + W_r \cdot h^{t-1} + b_r) \\ z^t &= \sigma(U_z \cdot x^t + W_z \cdot h^{t-1} + b_z) \\ o^t &= \sigma(U_o \cdot x^t + W_o \cdot h^{t-1} + b_o) \\ h^t &= z^t \odot o^t + (1 - z^t) \odot h^{t-1}, \end{aligned} \tag{2.3}$$

We can see that the difference between standard LSTM and GRU is not so obvious, so the choice of a standard LSTM or GRU depends on the specific task.

CNN In time series models, recursive neural networks (RNNs) are a traditional approach that gathers global information sequentially through recursion without the use of parallel computation. Sequential data may also be modeled using CNN using a two-dimensional "block" (m*n matrix). It is possible to think about time series data as a

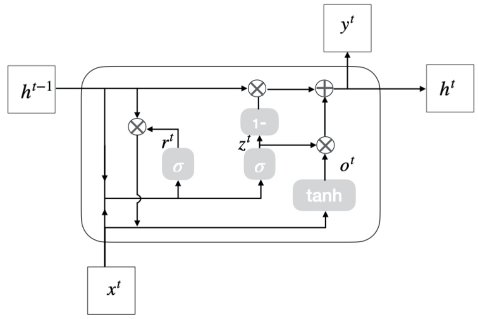


Figure 2.3: The cell structure of GRU.

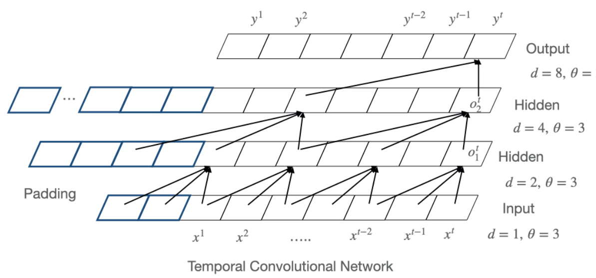


Figure 2.4: The structure of TCN.

one-dimensional object (1*n vector). A sufficiently large receptive field may be attained using CNN's multi-layer network structure, which requires an excessive amount of time because of the several layers.

TCN However, due to the advantage of processing in large-scale parallel structure and increasing the speed of network training at the same time, Duranton et al. [135] adapted the temporal convolutional networks(TCN) model to predict traffic flow. The TCN model is based on the CNN model and made the improvements in causal convolution, dilated convolution, and residual connections [129]. The results showed that TCN outperformed the latest LSTM and typical GRU models. Additionally, TCN avoids the issues of gradient dispersion and gradient explosion in RNN since it is more adaptable in processing the various lengths of historical information [58] [6]. The example is shown in Fig. 2.4.

Assume we have the time series data as $X = [x^{t-T+1}, x^{t-T+2}, ..., x^t]$ to make the prediction in $Y = [y^{t-T+1}, y^{t-T+2}, ..., y^t]$, with the filter as $F_f = (f_1, f_2, f_3)$ since $\theta = 3$. Hence, the hidden state in the first hidden layer at time step t with causal convolution is

$$\begin{aligned} o_1^t = F_f \odot X = \sum_{k=1}^{K} f_k x_{t-K+k} \\ = f_1 x_{t-2} + f_2 x_{t-1} + f_3 x_t. \end{aligned} \tag{2.4}$$

Since the size of the convolution kernel decides the amount of the information extracted, to increase the receptive field and avoid losing information, TCN introduces dilated convolution into the model. The dilated convolution is done by inserting holes into the "block" to expand the receptive field. It introduces the dilation rate that refers to the number of kernel intervals (the dilatation rate in the standard CNN is 1). The advantage of the dilated convolution is to expand the receptive field without missing information in no pooling operation as that in CNN. As the dilation rate in the first hidden layer is 2, the dilation convolution in the second hidden layer is:

$$\begin{aligned} o_2^t = F_f \odot o_1^t = \sum_{k=1}^{K} f_k o_{1_{t-(K-k)d}} \\ = f_1 o_{1_{t-2d}} + f_2 o_{1_{t-d}} + f_3 o_{1_t}, \end{aligned} \tag{2.5}$$

the receptive field of which is $(K-1)d+1$. But, the convolution operation solely focuses on capturing the local information, which requires the stack of layers for a larger receptive field to capture richer global information. The residual connections are introduced into the model to solve the degradation problem. The degradation problem is, due to the deeper network enlarging the receptive field, the increasing number of network layers causes the accuracy of the training set tends to be stable or even drop. The mathematical explanation for it is the solution space is more complicated, thus leading to the stochastic gradient descent method failing to receive the global optimization but being stuck in the local dilemma. The deeper the network, the more abstract the features and the more semantic information obtained; the residual connections of TCN refuse the problems that existed in the deeper layers, such as gradient dispersion, gradient explosion, and even performance degrades [47].

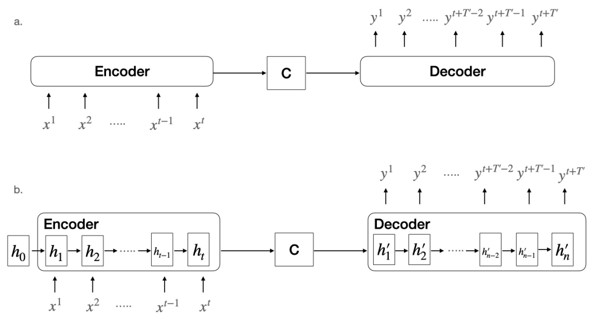


Figure 2.5: The structure of Seq2Seq.

Besides the models, we also want to present the model architecture for time series forecasting: Seq2Seq and transformer.

Seq2Seq Seq2Seq is a typical architecture under the Encoder-Decoder structure, seen in Fig. 2.5. The RNN requires the input and output same length, but seq2seq provides the solution with the unequal length of the input and output. c is the latent vector that contains the transformed information of the input and can be converted to the output sequence, which is:

$$\begin{aligned} h_t &= f(x_t, h_{t-1}) \\ c &= q(h_1, h_2, ..., h_t) \\ h'_n &= g(c, h'_1, h'_2, ..., h'_{n-1}) \\ y_n &= G(h'_n) \end{aligned} \tag{2.6}$$

There are also drawbacks of the seq2seq when applying to time-series data:1. Because the length of context x is pre-defined and fixed, the information can get lost when compressing; 2. The impact of each input to the target y_i is the same, while it is different in reality. For example, the condition in the past time steps differently affects the next, as the influence of the near ones outweighs the distant ones. [4] employed the attention mechanism to address the different importance of the target on Seq2Seq. It is to learn the attention coefficients of each input from the sequence, and then merge that according to the importance.

Compared with the original Seq2Seq model, every h is calculated based on the same context c, attention mechanism generates a different context c_n at each time step to solve

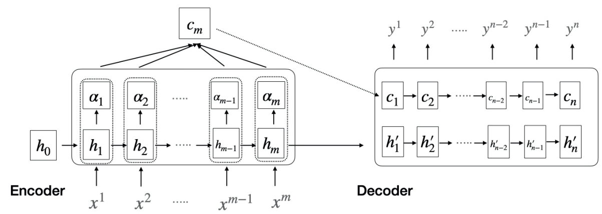


Figure 2.6: The structure of Seq2Seq with attention mechanism.

this problem(Fig. 2.7).

$$c_n = \sum_{i=1}^{m} \alpha_{ni} h_i, \tag{2.7}$$

where α_{ni} is used to measure the influence of the h'_n in the decoder to the hidden state h_i in encoder at the time step i. The weight $\alpha_n i$ is:

$$\alpha_{ni} = \frac{exp(score(h'_n, h_i))}{\sum_{k=1}^{m} exp(score(h'_n, h_k))}, \tag{2.8}$$

where $score(h'_n, h_i)$ is to calculate the similarity of the hidden state h'_n in the decoder and h_i in the encoder. Finally, we can have the output of the Seq2Seq model with attention mechanism in:

$$\begin{aligned} h'_n &= g(c_n, h'_1, h'_2, ..., h'_{n-1}) \\ y_n &= G(h'_n) \end{aligned} \tag{2.9}$$

But Seq2Seq only pays attention to the relationship between the input and the output, ignoring the position information. It means the position information in the time series data cannot be captured by Seq2Seq.

Transformer Google proposed an architecture named "Transformer" [93] provides another method of thinking for time-series modeling based on the Encoder-Decoder structure. Transformer relies on the attention mechanism to obtain global information in one step. Before, researchers usually captured the global feature by applying CNN in multiple layers. Transformer accepts the position-embedding and self-attention mechanism to obtain the hidden states instead of the recursion structure in RNN-related models. The application

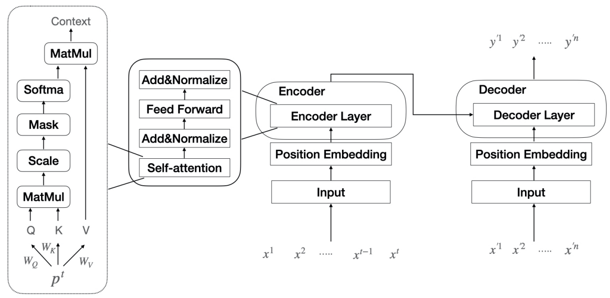


Figure 2.7: The structure of Transformer.

of the attention mechanism brought the transformer the powerful capability for time series forecasting [61], which lies in the simultaneous improvements in modeling long-term and short-term temporal correlations for time series data based on the multi-head attention structure.

The attention mechanism behind the transformer is similar to that in Seq2Seq. We assume the time-series input $X = [x^{t-T+1}, \ldots, x^t]$ over the past T time slots to predict $Y = [y^{t+1}, \ldots, y^{t+T'}]$. The input layer converts the data to the required vector, and the position-embedding layer then encodes the sequential information of it with the sin-cos functions as $[p^{t-T+1}, \ldots, p^t]$. In the architecture shown as Fig. 2.7, the self-attention mechanism mentions the key(K), value(V), and query(Q) that are transformed by the matrices W_k, W_v, W_q which are calculated with the vector V. The idea of K, Q, V is from the retrieval system as querying the target Q from the pairs (K, V). Hence, the attention weights:

$$\begin{aligned} \alpha_{ni} &= softmax(\frac{QK^T}{\sqrt{d_k}}) \\ &= \sum_{i=1}^{t} exp(\frac{< q_n, k_i >}{\sqrt{d_k}}) \end{aligned} \tag{2.10}$$

The calculation of the align score of q_n and all the k can be as obtaining the similarity of the record at time step n with all the records from the selected time period. The $\sqrt{d_k}$ is to keep the gradient stable. Using the probability obtained by the softmax function to calculate the weighted average of current V at all times, thus, the output is the vector that contains the global information at all time steps according to the Self-Attention. So the results

Table 2.2: Summary of Spatial models

Spatial / Temporal	Model Category		Data structure	Aggregater	Reference
Spatial	CNN		Grid	Convolutional Aggregator	[114]
	GNN		Graph	Massage passing	[104]
	GCN	Non-spectral	Graph	Convolutional Aggregator	[19]
		Spectral	Graph	Spectral filters	[42] [131]
	GAT		Graph	Attention Aggregator	[60] [111] [89]

of the self-attention layer are $s_n = \sum_{i=1}^{t} \alpha_{ni} v_i$ The multi-head attention is to help the network capture richer features/information under the subspaces, which can be compared to applying the multiple convolutional kernels in CNN. It works through repeating the self-attention in h times, as:

$$Multihead(Q, K, V) = Contact(head_1, head_2, ..., head_h), \tag{2.11}$$

where each head is the output of the self-attention.

Wang et al. [102] fed the extracted spatial and temporal correlations to the transformer architecture to obtain the long-term temporal features of the traffic flow data. A residual connection and layer normalization are also applied in the sub-layer to enhance the features.

2.1.3 Problems defined with spatial dependency

With further research, the spatial information on the traffic road network leads to the success of the work in traffic flow prediction [63]. Hence, how to mine the non-linear spatial and temporal relationships behind the traffic data is a challenge faced. Since the deep learning methods contributes to mining the non-linear information efficiently, many deep-learning-based solutions are applied in the traffic domain, such as traffic flow forecasting, accident detection, and abnormal detection, to extract the spatial dependency. Given the previous definition, which only considers the temporal information in traffic data, the solution here that considers spatial context can be represented as S, at time slice t as:

$$Y = f(X, S) = f([x^{t-T+1:t}], S).$$

CNNBefore, to depict the topology of the traffic network, the road segments were regarded as the 2D structured data consisting of latitude and longitude, which was inspired by applying CNN to exploit the Euclidean traffic flow data. Zheng et al. explored the traffic flow graph structure by dividing the route into more fine-grained sections and adapting the CNN on the pre-processing matrix [138]. CNN has the sub-sampling(max-pooling) layers specially designed to reduce the traffic map into the sub-grid structure. And it can learn the characteristics of the relations within the grid parts from the locality, and then combine the extracted local context into a high-level representation. CNN receives success in capturing local spatial dependency.

Du et al. discussed the impact of the locality on the traffic flow forecasting problem and proposed a hybrid model combining LSTM and CNN. It considers the performance of CNN to handle the local information in the sub-grid structure [28]. This model employed the LSTM to capture the long-term temporal information, and CNN for the local dependency. The extracted information was fused before feeding into the model. And the results showed that the consideration of the locality feature could contribute to the results in satisfactory accuracy and effectiveness even under complex non-linear urban traffic conditions.

In general, Convolutional Neural Network (CNN) models the locality by decomposing the traffic network into grids [114] but ignoring the topological structure graph-wide of the transportation network because of the complicated spatial correlations in reality. In other words, Euclidean structure can not fully describe spatial correspondences.

For example, two road sections are very close in Euclidean space, but the relations of a pair of road links may be in opposite directions due to the topology. So, it is hard to model the completely different traffic flow patterns under CNN. This means that the spatial structure in traffic is non-Euclidean and directional which requires the models to be able to capture the non-Euclidean topological information. To discover and make use of the spatial dependence of traffic flow data for better prediction, a more appropriate method that can express the traffic network mathematically into a graph is required, thus leading to the application of graph-based deep learning methods in traffic flow prediction.

Compared with Euclidean-structured data, such as images, graph-structured data is more complicated when modeling. First, during each time slice, the size of the non-euclidean graph data is not fixed for every input, which directly makes it difficult to the application of CNN for graph data. For example, in image processing, we have a fixed size of input images. It is easy to manually predefine the size for every input because the neighbors of a target node are at a fixed distance. In other words, when you find the center node, the neighboring nodes are all found. But in handling graph-structured data, it is hard to define a fixed input size since every node on a different number of neighbors and multiple distances. And, since the graph-structured data are not in strict order, which means it can not be as the grid-structured data to have the input in order when adapting the CNN. In terms of a mathematical expression, compared with the Euclidean structure data as grid-structured image data, the dimension of the feature matrix of each block in the graph is not uniform. Because the unified operators cannot be used to organize data, CNN is no longer able to directly perform operations like convolution and pooling on the graph-structured data. This has prompted discussions about how to handle properties like sparse connectivity, weight sharing, and feature extraction in handling graph-structured data.

Since the traffic network is usually graph-structured, the transition between the traffic states of the target node in the whole network can be defined as a graph Markov process. In order to handle missing values in the obtained raw data while anticipating short-term traffic, Williams et al. introduced the graph Markov network (GMN), particularly in the context of edge computing and online learning [106].

At the end of 2018, scientists from companies and institutions such as DeepMind, Google Brain, MIT, and the University of Edinburgh jointly proposed the concept of a graph neural network [8]. Based on Graph Neural Network (GNN), the traffic network can be regarded as a natural graph, where each observation is a node and the connection is the edge. The definitions of a graph are as follows: $G = (V, E)$

GNN An unweighted target traffic graph can be constructed as $G = (V, E)$, where, V is a set of nodes on the traffic graph, and E is a set of edges. $V = \{v_1, v_2, \ldots, v_N\}$ refers to N nodes on the graph, and each node represents an observation. The edge E refers to the connection between two nodes, which is represented by the adjacency matrix $A \in R^{N\times N}$, containing the topological information of the road network. If the targets are connected to the adjacent ones, the element of the matrix is 1, otherwise, it will be 0. For each node

i, it has its own characteristics x_i, which can be represented in a feature matrix X_{N*D}. For the feature matrix X_{N*D}, N represents the number of nodes, and D is denoted as the number of features of each node and can also be regarded as the dimension of the feature vector. The problem definition with spatial correlations can be updated as:

$$\begin{aligned} Y &= f(X, G(V, E)) \\ &= f([X^{t-T+1:t}], G(V, E)) \end{aligned} \tag{2.12}$$

Besides the feature matrix, an adjacency matrix plays an important role in accurate spatial modeling. It is because the expression of the node feature highly relies on the information from the neighbors. The adjacency matrix is the key to describing the spatial relations of the nodes in the graph [109]. In the traffic domain, because of the different assumptions in the specific scenario, the adjacency matrix is classified as a fixed matrix and dynamic matrix with the spatial structure whether changing to the evolving over time [116].

Given the previous achievement in traffic flow prediction, to better describe the topological structure of the traffic graph, many researchers tend to extend generalizing the convolution of CNN in 2-dimensional data to the graph-structured data, which leads to the introduction of the convolution operation applied on graph neural network(GNN).

In image processing, the convolution operation on the 2-d data is to extract features from an input image, which preserves the hidden relationship between pixels. The core technique of graph convolution network(GCN) is also the convolution operation, which is the same as CNN. Hence, the characteristics of CNN on the structural level are also of great significance to GCN [139]: (1) A graph is neural in sparse connectivity. (2) The time cost of the neural network can be reduced through weight sharing. (3) Multiple layers denote the features extracted at a different level. However, compared with CNN, a drawback of GCN is also obvious - it is difficult to define the local kernels and pooling operations in a graph directly because of the different number of neighbor nodes around each target, which leads to the discussion of the methods on how to aggregate information to a central node in a graph.

The GCN is typically classified as spatial GCN and spectral GCN. The spatial GCN works on incorporating the properties of a node based on its k local neighbors by directly multiplying the adjacency matrix to extract the features where the spectral version transfers the adjacency matrix to the Laplacian matrix to return the Fourier basis for the graph to capture the features. The mainstream of the GCN in the traffic domain are spectral graph convolution and diffusion graph convolution [116], the variant of vanilla spatial GCN. Essentially, all kinds of spatial graph-based neural network models are differentiated on aggregating approaches within each layer, while the spectral graph-based neural network models are on the choice of the filter g_θ [109].

The update operation for each layer is to update the states of the nodes in the current layer to the next status, which, in the graph-based neural network, can be written as a nonlinear function [139]:

$$H^{l+1} = f(H^l, A). \tag{2.13}$$

A is the adjacent matrix of the traffic graph. H^l represents the information of all nodes in layer l that should be updated, and H^{l+1} is the updated layer. f is the update function. It could be the GRU function in Gated Graph Neural Network(GGNN) [66], message function in Message Passing Neural Network(MPNN [38]) or activation function in GraphSage [45] as the model required.

In the spectral domain, the basic idea of convolution operation is to first convert the signal from the spatial domain to the spectral domain by the Fourier Transform as $f(t)->$ $\hat{f(t)} = U^T f(t)$, and multiply it with the convolution kernel $g_\theta(\lambda)$ in the spectral domain. And then transformed the outcome back to the spatial domain through the Inverse Fourier Transform. This is because the convolution operation in the spatial domain is the same as the multiple operations in the spectral domain. The extracted feature in the spectral domain is $\hat{y} = \hat{f(t)} \cdot g_\theta$. Since we are working on the dependency in the spatial domain, the extracted feature should be converted by the Inverse Fourier Transform as $y = U\hat{y}$, where U is the eigenvalue that is related to the Laplacian matrix of a graph. The place matrix is an important matrix used in graph theory. In a graph $G = (V, E)$, the normalized Laplacian matrix of a graph is defined as $L = D - A$, where D is the degree matrix of the graph, and A is the adjacency matrix of the graph. Hence, can have

$$H^l = \sigma(LH^{l-1}W^l) \tag{2.14}$$

to describe the update operation in the spectral domain, where W^l is the weighted parameter matrix of the lth layer, and σ is a nonlinear activation function, such as ReLU. The Laplacian matrix L is directly multiplied by the feature matrix H. But there are two problems with this operation: it ignores the impact of the node itself. We can have $L^{sym} = D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}} = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}} = I_a + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, which introduces the Symmetric normalized Laplacian to solve the self-transmission problem.

$$
\begin{aligned}
H^{l+1} &= \sigma(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}H^lW^l) \\
&= \sigma(D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}}H^{l-1}W^l) \\
&= \sigma((I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}})H^{l-1}W^l)
\end{aligned}
\tag{2.15}
$$

is proposed to solve the mentioned problem.

But we take the original definition as an example. Since $L = U\Lambda U^T$, U is the eigenvalue and Λ is the eigenvector, we can expend the formulation:

$$
y = U\hat{y} = U\hat{f(t)} = Ug_\theta U^T f(t) \tag{2.16}
$$

The hidden state of a node refers to the information of a node we mentioned above, for example, defined as $h_v^{t+1} = f(x_v, x_c, h_n^t, x_n)$, which represents the hidden state of node v in the layer $t+1$ is updated by the aggregation of the embedded node feature, the edge feature, the hidden state of the neighbors, and the edge feature of the neighbors. The specific node information depends on the specific requirement. Because the node information and the Laplacian matrix can be obtained directly from a graph, the difference of the spectral GCN lies in the filter g_θ.

For example, a filter in spectral GCN is Chebyshev polynomials [24]. The ChebNet method believes that the value of the convolution kernel in the spectral domain is a function related to the eigenvalue of a graph, which can be used to approximate the Chebyshev polynomials. Since $g_\theta = \sum_{k=0}^{K} \theta_k T_k(\Lambda)$, the function can be as :

$$\begin{aligned} y &= U g_\theta U^T f(t) \\ &= U \sum_{k=0}^{K} \theta_k T_k(\Lambda) U^T f(t) \\ &= \sum_{k=0}^{K} \theta_k T_k(U(\Lambda)U^T) f(t) \\ &= \sum_{k=0}^{K} \theta_k T_k(L) f(t) \end{aligned} \tag{2.17}$$

Because the T_k is the Chebyshev polynomials, θ_k is the coefficient to be learned. The [57] is based on the Chebyshev polynomials, which can be regarded as a further simplification of ChebNet. It only considers the 1st-order Chebyshev polynomial and each convolution kernel has only one parameter. However, there is also a drawback of the spectral GCN, the definition of the Laplacian matrix limited the graph in an undirected structure. And it assumes the static graph under the situation, while in practice the graph always time-varying, especially in the traffic domain.

In the spatial domain, it can be seen as two steps: 1). aggregation, which refers to how the center node collects the information from the neighbors, as $h_v^{l+1} = f(h_u^{l+1})$, where u are the neighbors of node v; 2). update operation, which represents how the nodes update their states to the next level based on the information aggregated, as $h^{l+1} = g(h^l)$. Before the aggregation, the nodes on the graph get the feature representation after the feature embedding. The aggregating approach designed in the spatial GCN can be regarded as a message-passing process. The central node, in the aggregation process, will exchange the information with that in the nearby k-hops, and update itself until the equilibrium of all the nodes on the graph is reached. The aggregation process aims to get the new representation for each node on the map by considering the information of the target itself and its neighbors. For example, one of the simplest convolutional operations is to add the hidden states of all neighbors to that of the center node. As in spatial GCN, it is similar to CNN to extract the spatial features from the topological graph directly. The information updated in aggregating could be the feature of the target node n, the information of its neighbors, the edge feature between the node n and its connected neighbors, the hidden state of the neighbors, and the previous hidden state of the node n.

Some of the work considers the spatial graph-based neural network in their model [19] [64] [134]. However, because the traffic graph, in reality, is huge, which could cause a time-consuming problem when computing the multiplication of the adjacency matrix, it is not common to employ the vanilla spatial GCN in the large traffic graph in the graph-level solution. Unlike spatial GCN, which directly aggregates the information to get the new representations for each node one by one and then updates the states of all nodes in the layer, the spectral graph-based neural network applied the Laplacian matrix to get the signal in the spectral domain from the spatial domain and finally inverse it back to the spatial domain. It is because the convolution operation in the spatial domain is the same as that in the spectral domain, which can make the feature extraction more simple and less time-consuming. They are applied in the node-level solution because of the state h, the stacking of k layers of spatial GCN represent the k-hops of the target node when it comes to discussing the impact of the multi-hops on the central node, the spatial GCN has been considered to be adapted [90] [98].

But some restrictions in the GCN model when applied to traffic flow forecasting problems:

- The traffic flow dynamically changes all the time. But GCN can not reflect the temporal dynamics of the traffic flow since the model construction relies on the static property of the graph.

- The GCNs capture the spatial information from the neighbors in the same impacts, which cannot reflect the different neighbors to the central node. For example, nodes on the busy commuter line weigh more heavily than those that are not on the line. And, in a road section, the inflow and outflow of the same node could be on the contrary in the morning rush hour and night rush hour. Hence, it is necessary to describe the different importance of the neighbors to the central nodes from the spatial respective.

- The aggregation approach of GCN depends on the graph, or, the matrix generated from the traffic network. Hence, it can limit the generalization from one region to the others.

GAT Graph attention neural network has been introduced to solve the above problems based on the attention mechanism [94]. It considers the weighted summation of the features among neighboring nodes. The weights completely depend on the neighboring nodes, independent of the graph structure. Attention mechanism has been developed in improving the performance of the models discoing the spatio-temporal relation in network [100].

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}$$

where α_{ij} is the obtained attention coefficient of node i connected with node j among all its connections.

Both GCN and GAT focus on aggregating the features of neighboring nodes to the central node and learning new feature expressions of the node based on the local stationery of the graph. The core difference between GCN and GAT is that GAT employs a single attention mechanism to aggregate the information with different weights. We can notice a normalized constant in GCN that is designed based on the graph structure. And because so, it is also why the generalization ability of the GCN model is unsatisfactory. In essence, GAT replaces the normalizing constant in GCN by using the attention weights to contribute to aggregating the neighboring node feature. It achieves to assign different weights to different neighbor nodes due to its impact on the target. To a certain extent, GAT will be powerful because the feature correlation between vertex is better described in the model without the effect of the graph structure.

As the original GAT adapted the single attention mechanism, Gated Attention Network (GaAN) Zhang et al. utilized the multi-head attention mechanism with the self-attention mechanism to gather information from different heads [123]. The idea behind the multi-head attention mechanism is that each attention head only focuses on one subspace of the input sequence, and is independent of each other. It is to obtain much richer information on features in the subspace.

Table 2.3: Categorization of the subcategory of the temporal/spatial dependency.

Spatial / Temporal	Dependency subcategory	Subcategory description	Model category
Temporal	Short-term temporal correlations	Short-term prediction period, usually between 5-15 minutes	RNN/LSTM/GRU/TCN
	Long-term temporal correlations	Long-term prediction period, usually longer than 30 minutes; periodicity as daily temporal pattern, weekly temporal pattern	LSTM/GRU/CNN/TCN/Transformer
	Temporal dynamics	Shifting temporal information periodically with neighbouring time intervals	Attention mechanism/ Dynamic Time Warping (DTW)
Spatial	Spatial static information	Topology information	CNN/GNN/GCN
	Spatial dynamics	the impacts of neighbouring nodes	Attention mechanism/
	Geographical scale	the impact of a larger scale on the centre node	Attention mechanism/ GCN/CNN
	Regional Spatiality		Attention mechanism

2.2 Summary

In this chapter, we went through the generation of our problem definition, solely considering the temporal dependency part of the overall Spatio-temporal dependency. We also provided related techniques that may be applied to solve the corresponding problems, from temporal models such as RNN and its variants, CNN, TCN, Transformer, etc., to spatial models such as GNN and its variants. We also provided a discussion on the differences between these models when applied as techniques to our problem definition.

Chapter 3

Sub-categories of dependencies and related work

There has been much success in applying deep learning techniques to traffic flow forecasting problems. The key to the performance of traffic prediction models relies on the accurate representation of traffic features. However, before modeling features, it is also essential to analyze the details of the features that may contribute to a solution. Researchers now tend to discover multiple expressions across the relationships between spatial, temporal, and Spatio-temporal dependency in the traffic domain, allowing for results that better match real-life situations. In this section, we will review the different perspectives of the spatial and temporal dependencies from current state-of-the-art research for solving traffic flow forecasting problems and then discuss models introduced in such research.

3.1 Temporal dependency

Traffic flow data is naturally within the temporal domain, which, in other words, means the previous states of the traffic flow may directly lead to changes in future traffic states. Initially, researchers in this field developed solutions that primarily focused on depicting the temporal correlations behind the data that directly contribute to traffic flow prediction. Temporal dependency can be generally divided into two types: short-term temporal dependency and long-term temporal dependency. The short-term temporal correlation reflects how the traffic state of the previous time step may directly influence the state in the subsequent time slots (within a 5-15 minute period). In contrast, the long-term temporal correlation shows the periodicity of the traffic states in a much more extended period, e.g., a day or a week. With additional research into long-term temporal dependency, some work also showed the importance of mining the relationship between the current and non-adjacent time steps.

3.1.1 Short-term temporal correlations

Traffic flow shows significant short-term temporal relations. The task of the Short-term traffic flow forecasting problem is to predict the changes in traffic flow (flow, speed, etc.) of a road in the next few minutes. For example, because of the current traffic congestion on the road, traffic conditions in the next time slot may likely be the same as the current time step since it is directly affected by the states at the last minute. Such strong short-term temporal dependency is the basic information we use to predict future traffic flow. Since the connection between the units of the recurrent neural network (RNN) can form a directed loop, the RNN, and its variant (LSTM, GRU, etc.) models are the typical choices used to depict time states in the short term. In other words, capturing the short-term temporal correlations from the traffic data by the RNN and its variants is powerful.

Due to the influences on more distant time states that may enhance the patterns of short-term temporal relations, some work considered adopting the Long-Term Short-Term Memory (LSTM) network to replace classic RNN in modeling the short-term temporal dependencies. It is because LSTM could consider longer-period temporal information, while the typical RNN only considers the context at the most recent moment. Based on the structure of RNN, LSTM adds a filter function to the past time states, thus aggregating the distant impacts on more previous time steps to address the short-term correlations.

However, the variant LSTM and GRU models can not overcome the significant drawbacks in RNN-type models - problems of gradient vanishing and the time cost of computing during the training process. Thus, the Temporal convolutional network (TCN) utilizes a combined architecture of dilated convolutions and residual connections to solve the above problems. The typical convolutional action, referring to the back-propagation algorithm, employs the different paths from the temporal direction of the sequence, computing the gradient of the loss function concerning each weight by the chain rule to prevent the above problems.

Although the TCN displays an outstanding architecture, Zhang et al. [125] showed that the number of TCN layers should be further discussed for any short-term flow prediction research considering TCN. On the one hand, the shallow structure of the TCN architecture may make it harder to attain complex temporal relationships, while deeper layers may result in over-fitting. Hence, when considering the short-term temporal correlations in traffic flow forecasting, the methods above may give great hints towards a solution. However, one must still be decided for use only in specific conditions.

3.1.2 Long-term temporal correlations

Besides the direct impact on the traffic condition in the next minutes, traffic flow data usually contains long-term information, such as the fluctuation in an hour, a day, a week, or even seasonal periodic patterns in the long temporal distance. Fig. 3.1(b). Zhang et al. [131] showed that the exogenous information in more detailed segments, such as the time of the day, weekday or weekend, and historical statistic information, would contribute to long-term prediction. However, it does not mean that the longer the prediction interval, the better the performance. Guo et al. [42] showed that the corresponding prediction difficulty gets greater when the prediction has prolonged time intervals. Further, Yao et al. [114] pointed out that the increasing length of each time step may enlarge the risk of gradient vanishing, which leads to the significant weakening effects of periodicity.

Variant LSTM models are naturally considered to capture the long-term temporal features in many works because the memory cell is created to maintain the information over a long distance. However, LSTM and its variants have time-consuming issues because of the longer-period information stored; thus, gradient vanishing and explosion problems in the back-propagation process when training are also the problem. On the other hand, it is a trade-off to choose GRU or LSTM for long-term temporal correlations modeling because

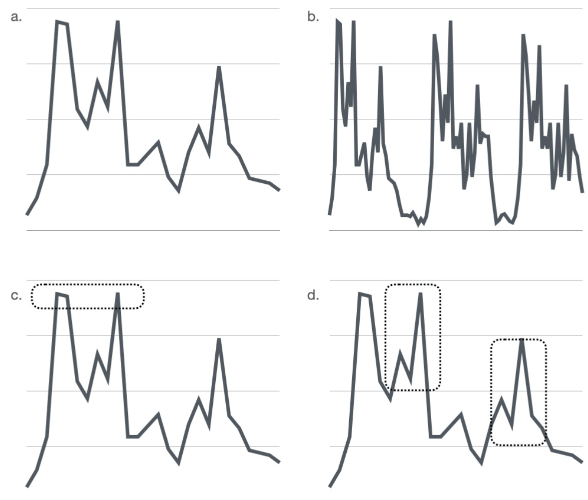


Figure 3.1: Temporal dependency in traffic flow forecasting. a. Short-term dependency in time series. b. Periodic patterns in the long-term dependency. c. Mismatching of the numerical value in time series. d. The local trend of dependency in the long term.

GRU owns a more straightforward structure than LSTM and with lower training time costs, but LSTM can perform better in other cases.

Regardless, the general definition of the long-term temporal dependency requiring inputs from more distant time steps is that traffic flow has periodic characteristics within a specific period. Such periodicity refers to the fluctuation in an hour, a day, a week, or even seasonal periodic patterns in the long temporal distance as from Fig. 3.1(b). Zhang et al. [131] demonstrated that the exogenous information in more detailed segmentation, such as the time of the day, weekday-or-weekend, the historical statistic information, would contribute to the long-term prediction. For example, every weekday rush hour happens in the morning around 9:00 a.m. and afternoon around 5:00 p.m. It is the average busy time on the commuter lines for most commuters every day. Such daily and weekly periodicity protects the periodic temporal information for the traffic flow. It also should be an essential factor for the traffic flow prediction problem since it can reflect the variability of the traffic flow in the long term. [97] considered this situation and separately captured the temporal information as closeness, period, and trend by the same ConvLSTM structures and with different weights.

Wang et al. [99] proposed similar ideas toward capturing multiple temporal components. They proposed a dynamic mechanism to employ time-varying hypergraphs for capturing the hourly, daily, and weekly changes in traveling OD times. The applied hypergraph theory outperformed the current state-of-the-art graph-based and non-graph-based methods in capturing long-term temporal dependency. Besides utilizing these models to extract different components, Wang et al. manually predefined the temporal features from the recent and periodic time steps for temporal modeling. It is another common consideration for periodicity in prediction [98].

In addition, another method for depicting the long-term dependency is to construct a stack of CNN layers in the time domain. The application of CNN in temporal modeling showed that it is possible to utilize the 1D CNN layers for sequential modeling, which suggests the superiority of typical RNNs [42] [86]. It effectively reduces the problems in recurrent neural network architectures, such as vanishing gradient problems, making the model easier to converge and train. The convolution operation merges the temporal information at the neighbouring time slice in the time dimension to extract the long-term temporal dependency, which can be seen in 3.2. After the GCN was applied to aggregate neighbouring information of each node on the graph, Guo et al. [42] stacked the convolution operation in the temporal dimension to collect the context of subsequent time slots. It extracted the hourly, daily, and weekly periodicity from the same network structure that the Saptio-temporal (ST) blocks with an ST attention layer and an ST convolution layer. Further, due to the advantage of capturing the sequentiality of the traffic flow graph by CNN, Yang et al. [111] considered applying the gated dilated CNN to capture the long-term temporal relations for traffic flow forecasting. Because of the dilated convolution application, the gated dilated CNN can also express the ability to capture the long-term temporal dependency in the non-adjacent time steps. They proved that the final traffic flow prediction depends on the temporal patterns from adjacent time steps and non-adjacent time steps, which may also contribute to overall modeling.

Zheng et al. [137] adopted the encoder-decoder structure for traffic flow prediction. They embedded the input by considering the temporal context embedded before encoding traffic features. The long-term relation was modelled by a transform attention layer be-

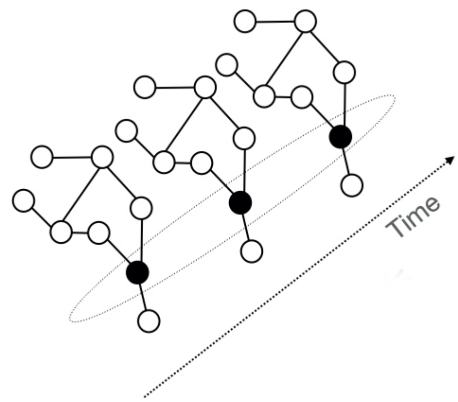


Figure 3.2: Aggregation of information from the neighboring time slices

tween the encoder and decoder that eased the error propagation effect. It preserved the long-term temporal relationship by calculating the relevance of future and historical time steps, thus enhancing long-term forecasting.

Transformer [93] has been considered in long-term dependency extraction because of the self-attention mechanism connecting data at distant positions [65]. Moreover, due to the achievements of Transformer in other fields, such as text processing and machine translation, for capturing long-term features, the Transformer has also been considered in the traffic domain field [17] [108] [43]. Cai et al. [17] pointed out that Li et al. [65] ignored the network-wide information in forecasting with Transformer; they addressed the traffic flow prediction with both the long-term temporal dependency and spatial dependency. They extracted the spatial feature by GNN and then encoded it for the continuity and periodicity of the time-series data information as the global and local temporal feature to the Encoder layer in Transformer. Guo et al. [43] obtained the global receptive fields with the multi-head self-attention mechanism in the Transformer structure. They expanded the self-attention layer based on the causal convolution operation to fulfill the continuous trend with the local temporal context. The local temporal context refers to the local periodicity combined with the global periodicity as the temporal input. They manually defined the input with the global periodic pattern, which was the time segment from the same day in the past w weeks, while the local periodicity is that in the past d consecutive days. Then it concatenated the two pieces of periodic information to the Spatio-temporal encoder-decoder architecture.

3.1.3 Temporal dynamics

In addition to static temporal correlations as the short-term and long-term temporal dependency, the traffic flows also showed dynamic characteristics in the time domain. Yao et al. [114] revealed that traffic flow has obvious periodic tendencies, but it is not as strictly periodic as we think. For example, exact rush hours are not always at the same time every weekday; evening peaks could be busy from 4:00 p.m. to 6:00 p.m. on Monday to Thursday but would vary on Friday from 2:00 p.m. to 4:00 p.m., which is a temporal shifting in the temporal domain. Due to this, the researchers developed the Periodically Shifted Attention Mechanism, based on the LSTM combined with an attention mechanism to capture the periodic temporal shifting. First, they obtained $h_{i,t}^{p,q}$, representing the predicted time t between the time interval q in the region i. The LSTM model is applied to collect the preliminary information within the previous day p. Furthermore, the attention coefficient $\alpha_{i,t}^{p,q}$ is calculated to measure the importance of the time interval q in day p. And finally, the long-term temporal representation $h_{i,t}^{p}$ that includes the shifting periodic information is given as $h_{i,t}^{p} = \sum_{q \in Q} \alpha_{i,t}^{p,q} h_{i,t}^{p,q}$.

Li et al. [70] used the Dynamic Time Warping (DTW) method to capture the shifting temporal information. DTW helps find suitable matching pairs with similarities for the consideration of signature time domain features, allowing for calculations of the distance between the similarities of two-time series to obtain the shifting temporal feature along the time axis. Guo et al. [43] employed a traditional self-attention mechanism to match traffic flow with the same numerical value and a CNN layer to capture the temporal context for the local trend.

3.2 Spatial dependency

Because of significant improvements in the performance of graph-structured information in current work, we know that the introduction of graph-based neural networks in traffic

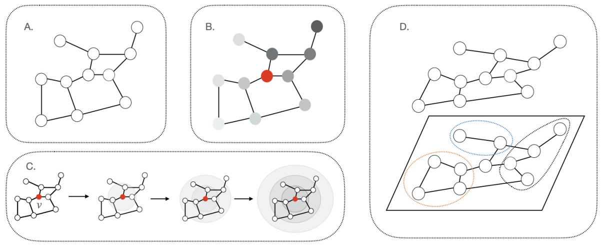


Figure 3.3: The spatial dependency in the traffic flow graph. A. The topology of the graph structure. B. The impact of the neighboring nodes. C. The impact within k-hops. D. The regional similarity.

flow forecasting models outperforms previous non-graph-based network models [84, 111]. Popular spatial models depicting the traffic graph information are GCN. However, the theory backing the GCN [57] shows that existing GCNs and their variants only consider the static topological structure of graphs. In contrast, the traffic road graph in the real world is dynamic and highly non-linear. The expected graph-based spatial models should discuss the local impact of the nearby nodes and the multiple spatial dependencies in reality. For example, the spread of the influence is not isotropic for each node on the traffic graph. The geographical location of the road sections and the functional role of road segments will both significantly impact the final traffic flow prediction. Thus, the correlation of the traffic state can not be solely judged by the local spatial proximity. At the beginning of employing the graph-based neural network in traffic flow forecasting, it was essential to select the model to describe best the different topologies of the local road network on the spread of the influence on the traffic conditions. With further study in this area, the discovery of more detailed categories of spatial dependencies should be addressed in future work. It is worth noticing the increase in research discussing the influence of the multiple spatial dependencies, such as the global and local spatial dependency, the static and dynamic spatial dependency, and the different hops in spatial scale.

3.2.1 Spatial static information

Yang et al. [111] illustrated that the topological structure of the traffic graph should be addressed since the graph-based methods outperform the grid-structured methods in prediction, as Fig. 7.1(a). The graph topology reflects the global stationary spatial information on a whole network scale. An $N \times N$ adjacency matrix is represented as the graph structure in the graph convolutional network (GCN), which reflects the connectivity among each node on the traffic road graph. It is obtained directly when the traffic graph is constructed and used throughout the process. When the GCN appeared in the traffic flow prediction field, most works considered the static connective topology information for the prediction and performed well. In the spatial GCN domain, the operations on the graph can be regarded as extracting the spatial dependency from the traffic graph. The convolution for each node can be regarded as an update of the static local information. On the other hand, spectral GCN captures the static global spatial information by the transformation. Based on its theory, the Laplacian matrix is transformed by the adjacency matrix reflecting the connectivity of each node on the graph, used to directly convey the global spatial dependency.

3.2.2 Spatial dynamics information

Models considering the topological information outperformed the previous ones with non-graph dependency. However, the impacts of the weights on neighbouring nodes are not inherently the same, Fig. 7.1(b). Affected by the adjacent segments or the nearby neighbours, the real-time traffic conditions of individual segments change over time. For example, morning traffic congestion will lead to high volumes for the nodes from the residence to the workplace. In contrast, the connected neighbours not on the commute path for congested nodes will not be as crucial to the commuting ones. Since the neighbours do not affect the target node in the exact weights, equal weights of all neighbouring nodes will lose the dynamic spatial information of the target. Hence, the solution should not only pay attention to the pre-designed static adjacency matrix throughout the entire prediction process but to a better expressive representation of the weights of the different nodes. The simple way for modeling the time-varying spatial correlations is via a predefined adjacency matrix with prior knowledge of the traffic graph. In other words, the static properties can represent the components that refer to the dynamic spatial correlations. For example, Fang et al. [33] considered that the static adjacency matrix of the graph updated in each time interval can be used as the dynamic adjacency matrix. Feng et al. [34] constructed an adaptive adjacency matrix predicted by the predefined adjacency matrix based on the dynamic graph learning (DGL) concept. The methods for learning spatial dynamics focus on calculating different weights of the real-time spatial information obtained from the data directly. Different kinds of aggregators are adopted for modeling these locality dynamics. Zhang et al. [123] pointed out that some of the GCN and its variants tried to assign a non-parametric weight, such as graph pooling aggregator, and graph pairwise sum aggregator, to the connected nodes, thus producing dynamic spatial correlations. Nevertheless, the non-parametric aggregators can not differentiate the edge weights between the target and all its neighbours. The attention mechanism is a popular and more effective aggregator to be introduced for specifying a node's weights in a neighbourhood without a prior understanding of the traffic graph topology.

Some work [5, 42, 43], defined a spatial attention mechanism to adaptively adjust the attention scores with the neighbours to capture the dynamics on the graph. Nevertheless, there are also differences among the applications, Guo et al. [42] utilized the attention mechanism to extract the spatial dynamics before the spatial convolutional operations on the graph, while Bai et al. [5] and Guo et al. [43] employed it to obtain the attention coefficients based on extracted spatial information from the traffic graph. The attention mechanism is incorporated in GNNs as graph attention networks (GAT) [94], the weights of which can be directly represented by attention coefficients under the graph structure. GAT uses masked self-attention layers to assign the aggregation weights within the same hop. The application of the standard attention mechanism can be regarded as a single attention head under a multi-head attention mechanism since it expresses the representation under one subspace. The multi-head attention mechanism takes the single-head attention further, which runs through a single attention head several times in parallel, thus, enhancing the ability to explore representation under multiple subspaces. For example, the features of images can be regarded under multiple subspaces since each has different features, e.g., colour space, line space, texture space, etc. Lu et al. [70] aimed to get richer latent information from different representation subspaces. Further, Yang et al. [111] and Zheng et al. [137] extended the self-attention mechanism to the multi-head one to stabilize the learning process.

3.2.3 Geographical scale

Most graph-structured models consider generating the spatial representation given the information from one-hop neighbours, such as vanilla GCN and GAT, since the effects of the information from neighbouring nodes plays an significant role [10, 11, 73, 75]. However, the impact of a larger scale on the centre node could reflect the richer informative spatial dependency because the nodes at k-hops can describe the different influences towards the target segment [111, 132], as seen in Fig. 7.1(c). Given a city-wide traffic network, when traffic congestion happens in a specific road segment, the linked neighbouring roads could be actively affected by this great event in varying degrees. Because the radiation zone could be affected in multi-hops [95], a traffic jam trouble may last for several hours in a specific road segment if it occurs. A more enormous k hop of the node may help capture the broader spatial dependency compared with the one-hop model.

To distinguish the contributions of neighbours in different hops, Wang et al. [98] constructed a stack of k graph convolutional layers to get the information from different neighbouring scales and computed the attention coefficients for each layer. Given that multiple scales of spatial and temporal features affect the final result, Yang et al. [111] adopted MST and MSS sub-blocks to capture the spatial and temporal correlations separately. They fused the correlations from the different spatial and temporal scales when generating the outputs of the spatial-temporal relationships. The various impacts of the ST correlations in multiple scales are based on a weighted sum fusing method, which expresses the ability of the multi-scale ST correlations.

3.2.4 Spatial heterogeneity

Computation time for graph-based solutions is usually high since traffic graphs are generally quite large. The inflow/outflow interactions between adjacent regions may lead to local similarities(Fig. 7.1(d)); hence strong spatial relations might exist in the nearby regions. Zhang et al. [133] designed the transportation neighbourhood adjacency matrix based on the spatial proximity between nodes in the road graph to show the improvement of their model when considering local similarities. However, some work also pointed out that the nodes on the traffic road network could own the similarity no matter the distance between one another. For example, it was found that a road segment may have a very similar impact to another even if it is comparatively distant because of similar road environments [103]. Region-level similarity has become a heated topic in recent years; thus, more detailed solutions discussing region similarity will be shown in the next section.

3.3 The relationship of spatial and temporal dependency

Even though we can see so many surveys in this area have realized the importance of utilizing spatial and temporal correlations to traffic flow forecasting, previous research still needs to include a view of how to describe the spatial and temporal dependency relationship to design the spatio-temporal architecture. In fact, as the similar idea of ensemble methods in machine learning, the architecture of spatio-temporal modelling has shared familiar points with it, which builds the submodels in a parallel and sequential way. The parallel and sequential modelling of spatial and temporal dependency shows the inspiration of the solution to traffic flow forecasting.

After figuring out the spatial and temporal dependency, we further categorize the architecture of capturing spatio-temporal dependency when considered in modelling, which can be divided into parallel and sequential modelling.

3.3.0.1 Sequential modelling in spatio-temporal architecture

The basic submodels are integrated sequentially in ensemble modelling, and the same is in spatio-temporal modelling. The principle of sequential modelling is exploiting the submodels' dependencies by assigning hidden relationships that we can not discover by eyes; the overall prediction performance can be improved when compared by sole modelling. First,

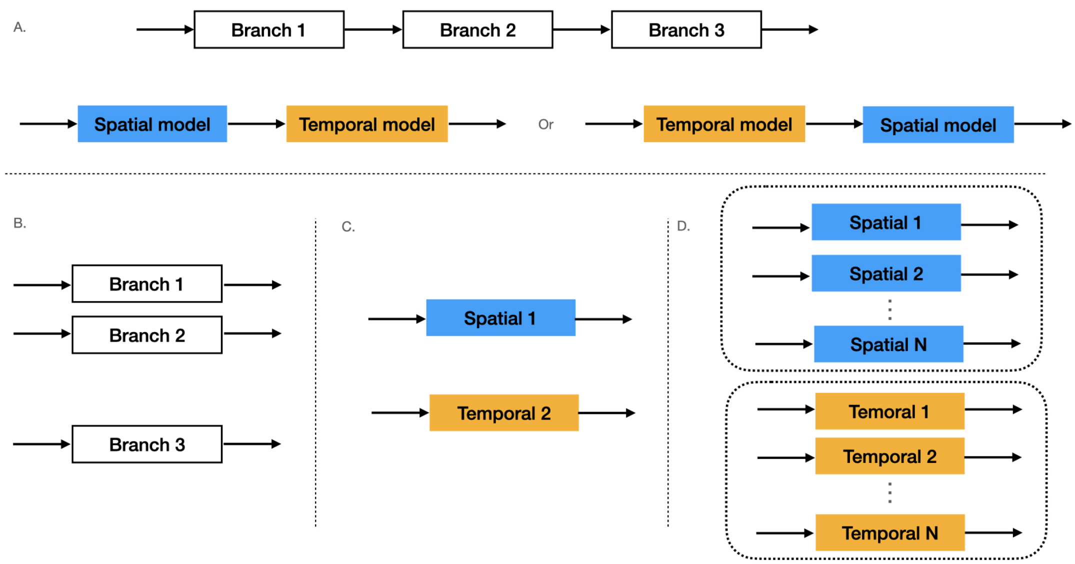


Figure 3.4: The differences of sequential and parallel modelling in machine learning and spatio-temporal modelling. A. Sequential modelling of ensemble learning and spatio-temporal modelling B. Parallel modelling of ensemble learning C. Parallel modelling of spatio-temporal modelling D. Parallel modelling of spatio-temporal modelling in subcategory

in sequential modelling, applying the spatial-related models to get the spatial information from the raw data and then getting the processed features into the temporal models to obtain the temporal extraction for the final results, which can be seen in Fig. 3.4 A. Zhao et al. proposed the typical sequential architecture with the separate spatial and temporal extraction applied to spatio-temporal modelling in the early research [134] — the spatial dependence referred to the topology of the traffic road network that is captured by GCN; and the temporal dependence was represented by temporal dynamics, with the extraction of the GRU. Zhu et al. were inspired and developed the sequential modelling for each moment based on Zhao et al. by introducing an attention mechanism to capture the importance of information at each time slot. For this idea of construing the spatio-temporal modelling, we can see so many similar consideration in the following research [25, 32, 115, 121, 122].

Yu et al. and Diao et al. showed the basic structure of the ST (spatio-temporal) blocks that involved the stack of multiple spatial and temporal submodels in capturing the dependencies in sequence in the blocks [25, 121]. A stack of the ST blocks is also under sequential modeling to capture the spatio-temporal dependencies for more extended time information.

3.3.0.2 Parallel modelling in spatio-temporal architecture

In paralleling modelling, the base submodels are constructed in parallel structure, the principle of which is to exploit the independence between the submodels and make up the ignored information of the other learners. Depending on the dependencies ' characteristics, the base submodels can be the same methods. Guo et al. proposed the typical parallel architecture for processing spatio-temporal dependency in sub-categories [42]. The input data is divided into sub-categories (hourly, daily, weekly) first, fed into three branches with the same blocks extracted the spatiotemporal pattern from hourly, daily, and weekly levels. The following work, such as [46,50,71,88,126,130] considered parallel structure under the similar idea and have different branches for extracting the sub-category dependency. With Han et al. [46] and Hong et al. [50] further developed the spatio-temporal feature extraction of daily, weekly, and recent information on sub-models in parallel modelling, The other work employed more branches on extracting multiple fine-grained dependencies in parallel structure. Sun et al. proposed the models with seven branches on extracting spatial and temporal features in sub-categories separately and then fused for representing spatio-temporal dependency [88], which is as shown in Fig 3.4 D. Luo et al. mined the heterogeneous information in three branches that were constructed by different sub-models [71]. Within the branches, the parallel modelling can be achieved with sub-models which are constructed by parallel modelling as well. Zhang et al. came up the models with spatial branch and temporal branch, which are with sub-branches on extracting fine-grained spatial and temporal dependency separately [126].

3.3.0.3 Spatial-temporal dependency

As that researchers prefer to model the spatial and temporal dependency separately, it is also necessary to consider how to fuse the components to can make best utilize of the hidden spatio-temoral relationship from the combination. With the discovery of the effective and informative spatial-temporal expression in traffic flow prediction, multiple fusion methods have been employed in feature fusion for the different features, thus further contributing to the improvement of the model performance. Even the external factors, such as weather conditions, car accidents, and events hosted near the target, seems as no directly relationship for the prediction, some work still shows that the availability of the external data enhance the model performance [124, 141]. Hence, the motheds on fusing the features in the current traffic flow forecasting field is worth discussing for integrating the spatial and temporal features extracted from the raw data. In fact, it is through the information superposition of the input to enrich the obtained features, thereby improving the performance of the model.

Concatenation Feature Concatenation is a common method of feature fusion. It is to contact the different feature vectors directly in the same order. Assume that we have $v_1 \in R^m$ and $v_2 \in R^n$, the fused vector could be $v = [v_1, v_2] \in R^{m+n}$. The concatenation of the features has been applied in many types of research from different perspectives. [137] concatenated the information of the day-of-week $v_1 \in R^7$ and time-of-day $v_2 \in R^T$of each time step into the temporal vector $v \in R^{7+T}$. Yao et al. [114] reconstructed the temporal representation that reserved both short-term and long-term dependencies for traffic flow predicting by directly concatenating the short-term representation $h_{i,t}$ and long-term representation $h^P_{i,t}$ as $h^c_{i,t}$. Yao et al. took the temporal representation by the concatenation of the spatial dependency and the external factors before feeding it to the LSTM. The extracted information is further concatenated with the feature from the semantic view for the final demand prediction [115].

Wu et al. and Li et al. introduced the graph attention network (GAT) with a multi-head mechanism to capture the dynamic expression of the node feature [107] [60]. The features under the K independent attention mechanisms are concatenated after the extraction. Cui et al. enriched the spatial information by concatenating the features extracted from the k hops of neighborhood on the traffic road graph [22]. Guo et al. [41] concatenated the region feature j with the road feature i if the segment i belongs to the region j.

Element-wise operation As we can see that $v = [v_1, v_2] \in R^{m+n}$, the dimension (number of channels) of the features v has increased after the operation of concatenation, but the processed feature keeps its original information as v_1, v_2. The element-wise operation is to richer the information but keeps the same dimension of the feature after processing. For example, as the element-wise product, assume that we have $v_1 \in R^m$ and $v_2 \in R^m$, the fused vector could be $v = v_1 \circ v_2 \in R^m$.

When the different components are fused, the different weights are learned from the data, and the final results can be regarded as obtained by the linear weighting method. Wang et al. applied the element-wise operation to fuse the closeness TD_{t_c}, period TD_{t_p} and trend TD_{t_t} representation from three predicted branches to the final result [97] . $TD_t = W_c \odot TD_{t_c} + W_p \odot TD_{t_p} + W_t \odot TD_{t_t}$, where TD_t is the final result, and the W_c, W_p, W_t are the weight matrix of the three temporal predictors respectively. [42] considered the hourly, daily and weekly components in the same way. Other examples (for example, [124] [19] [60] [22] [82]) with the application of the element-wise addition and product will not be listed in details.

Zheng et al. [137] employed a gated fusion mechanism which can adaptively control the spatial and temporal information $H^l = z \odot H_S^l + (1 - z) \odot H_T^l$, where H_S^l and H_T^l represents the spatial and temporal dependency obtained from the last layer, and z is the gate. [123] mentioned the fusion methods, such as pairwise sum and pooling approaches, applied to the node feature. It studied the gated attention-based fusion method with the consideration of a multi-head mechanism and outperformed the pairwise sum and pooling on the graph nodes with fewer number parameters.

Likelihood Since the random walk-based network embedding approaches constructed a co-occurrence matrix to help describe the spatial dependency and temporal dependency [2] and received a relatively satisfactory result, Liu et al. [69] proposed to compare the co-occurrence matrix generated by the random walk approach, to maximize the likelihood of the time-evolving traffic graphs. It jointly captured spatial-temporal dependency instead of extracting the spatial and temporal correlations from two independent sources.

3.4 Summary

In this chapter, we go through the spatial and temporal dependencies and detail the subcategories contributing to solving traffic flow forecasting problems. This section also highlights factors and motivates to work towards more performant solutions.

Chapter 4

Discussion on perspectives of the current solutions

When categorizing solutions to traffic flow prediction, we may divide by two main aspects: one is the introduction of new features, which can be referred to as the introduction of the graph-based neural network for extracting the spatial information from traffic data, and the other is the detailed representation of previous features, for example, that of the rush hours in each weekday being different. The difficulty of each is applying the current techniques to represent the features with the appropriate informative expression.

4.1 Feature engineering on temporal perspective

The topic for further research in recent years has trended towards discovering informative feature engineering for inputs into traffic flow forecasting models. For example, research tends to focus on digging for much more appropriate approaches to build the general dependency under the detailed categories, such as temporal position embedded in the transformer, which enhances the ability of temporal order expression instead of the time-series models to describe the general temporal dependency. For example, the Dynamic Time Warping (DTW) algorithm is applied as a clustering method to describe the locality of the spatial dependency, etc., instead of the spatial models to represent general spatial features. This direction requires researchers to understand the factors that may affect traffic flow forecasting comprehensively. It leads to feature reconstruction considering the multiple facts from different perspectives of the raw data. In other words, previous work extracts the features from the raw data with the spatial and temporal models, while current work aims to reconstruct the feature vectors before feeding the raw data. Feature engineering can better reflect and enhance the spatial and temporal correlations in real-world traffic conditions.

The multiple temporal correlations must be combined in the long-term prediction [111]. On the one hand, temporal features are represented as the short-term dependency affected by the most recent historical time step. On the other hand, temporal features are also under the influence of periodic solid characteristics in daily and weekly cycles. Most recent work employs the idea of trading the utilization of the short-term and long-term in the temporal feature pre-construction, the variables of which are always in hourly, daily, and weekly periodicity to reflect the variation in the time domain [17, 30, 53, 78, 90, 98, 99, 126]. Multi-dimensional time dependency shows that the periodic variation of the hour, day, and week can be fused as the overall temporal input to the model based on the task required.

4.2 Feature engineering on spatial perspective

Besides multiple temporal dependencies, spatial features can be enriched by multi-dimensional representation [78, 90, 126, 128, 133]. Zhang et al. [133] focused on obtaining richer spatial information by introducing the content similarity adjacency matrix, the transportation neighborhood adjacency matrix, and the graph betweenness adjacency matrix to represent spatial correlations in traffic flow graphs as geographical correlations, region similarity, and road connectivity. Li et al. [108] proposed a multi-scale graph convolutional layer to assign weights to three spatial dependencies. The three spatial features referred to the normalized adjacency matrix with the self-connected unit represented by the geographic adjacency matrix, the extraction of hidden spatial dependencies represented by the self-adaptive matrix, and the similarity of the traffic patterns represented by the similarity matrix. Wang et al. [98] pointed out that existing GCN methods adopt the spatial feature of the latest structure (e.g., k-hops), ignoring the information of the previously obtained stages (e.g., before $k-1$-hops). They utilized the completed spatial information obtained from the previous neighboring range and proved the efficiency of the spatial enhancement.

Guo et al. [43] considered feature vectors separately embedded in spatial and temporal respective. In the temporal dimension, they manually defined the global and local periodicity. They concatenated them with the time steps X of the historical traffic records, finally leading to a new temporal input. In the spatial dimension, the GCN with Laplacian smoothing form is employed subsequently to obtain the spatial embedding vectors. This paper also discussed spatial heterogeneity, which referred to the observation that the traffic records generated in different locations obtained different traffic patterns and further discussion on regional solutions proposed to model it. Fang et al. [31] proposed a convolution model named Dilated Attention Graph Convolution (DAGC) that can generate both spatial and temporal dependency. DAGC captures the non-local spatial correlations in multi hops as the spatial input and the adjusting parameters are also applied to fuse the temporal correlations as the temporal input.

4.3 Feature engineering on Spatio-temporal perspective

Besides embedding the spatial and temporal features separately, Bai et al. [87] adopted the MLP to learn the new Spatio-temporal features from the raw input, which can directly be used as the predictors in the proposed model. It has been proved that direct extraction of the new features with MLP works well in image recognition and voice translation. Further, it is more effective than the manual design of new features from raw data. Jiang et al. [54] developed a similar architecture to capture the dependencies as previous research. Nevertheless, they introduced a Spatio-temporal relation matrix to represent the traffic road topology as the spatial input. It is because the Spatio-temporal relation matrix refers to the space-time constraint on the traffic road conditions. As the spatial proximity matrix is usually applied to represent the relationship between the target road segment and its corresponding neighbours, it is utilized to depict the topology of the road network in this work. Furthermore, they supposed the historical time interval to represent the traffic speed time series in the temporal dimension, which is the standard approach as before. However, because they adopted a cross-correlation function to fuse the time information in each time interval, the after-fused Spatio-temporal relation matrix is defined as the one that combined the Spatio-temporal information of the road traffic in a matrix before feeding into the model.

4.4 Feature engineering on extra factors

Except for reconsidering the representation of the spatial and temporal feature vectors for traffic flow data, Zhu et al. [141] introduced the external factors to reconstruct the input to the model. They employed the other factors which could affect the traffic conditions as the additional attributes, such as static geographic information, which is a static factor, and weather condition, which is the dynamic factor because of its time-varying determinant. The static factors, dynamic factors, and flow values are combined as traffic characteristic information at time t in the proposed model.

4.5 Region-level solutions

Another direction of work in traffic flow forecasting is the proposition of region-level solutions for spatial feature extraction in heterogeneous regions. As previous work paid more attention to capturing spatial dependencies at the graph level to depict information of all the nodes on the graph, research at the region level tries to mine spatial relationships among the irregularity of the regions [14, 90, 98, 113, 127, 133, 140]. For example, every day, the commuters that leave residential areas and drive to their workplaces could lead to a morning peak and the opposite to an evening peak. It displays an instinctive transition flow due to regional heterogeneity at different periods in a day. Hence, it can be concluded that all the nodes on the traffic road network are not isolated but have functional similarities in certain regions. It also leads to introducing the community algorithm [90, 113] for capturing the regional spatial dependency to enrich the extracted spatial features for prediction. The community structure implies that the nodes with high similarities could belong to the same community, referring to the same functionality in a traffic road network.

In [137], a group spatial attention mechanism was proposed based on the sub-graph partition of the intra-group and inter-group spatial attention parts. However, the graph-wide computation takes time and memory consumption. The Louvain algorithm, proposed by Blondel et al. [9], was adopted in [90] to discover different functional communities in road networks. It contributed to constructing the functional similarity graph (FSG) for the representation ability of the spatial dependency. To avoid the bias of the static adjacency matrix, which only represents the connectivity of each node, Zhang et al. [127] employed the fuzzy-graph generation network to enhance the expression of the intrinsic correlation for regional spatial features. The method utilized the fuzzy logic relationship among nodes on the traffic road network to generate similar clusters.

4.6 Summary

In this chapter, we reviewed current solutions' techniques to model spatial and temporal features. We categorized the perspectives from four aspects: temporal, spatial, spatial-temporal, and other factors.

Chapter 5

Methodology

5.1 EDA

5.1.1 Datasets

For our experiments, we adopt datasets with different sizes. METR-LA(S) and PeMS-04(S) are the small datasets tracing from 01/03/2012 to 08/03/2012 and 01/01/2018 to 08/01/2018, respectively. PeMS-04(L) is a large dataset with traffic flow data from 1/1/2018 until 2/28/2018. We decided to adopt traffic speed as the predictor in the experiment section because traffic speed can better describe the fluctuation of traffic conditions. Hence, all the datasets have only one feature channel: $C = 1$.

As to explore whether the different historical periods of inputs could reflect different amounts of information to the prediction, we use a time slot of 5 mins with historical lengths of 3, 6, and 12; thus, 15-, 30-, and 60 mins are used to forecast the future traffic conditions in the next step.

- METR-LA: depicts the traffic flow records of highway networks in Los Angeles [52] in real-time by loop detectors, with 207 sensors and its traffic records from Mar.1 to Mar.8, 2012. The data set is divided into a training set with the first three days,

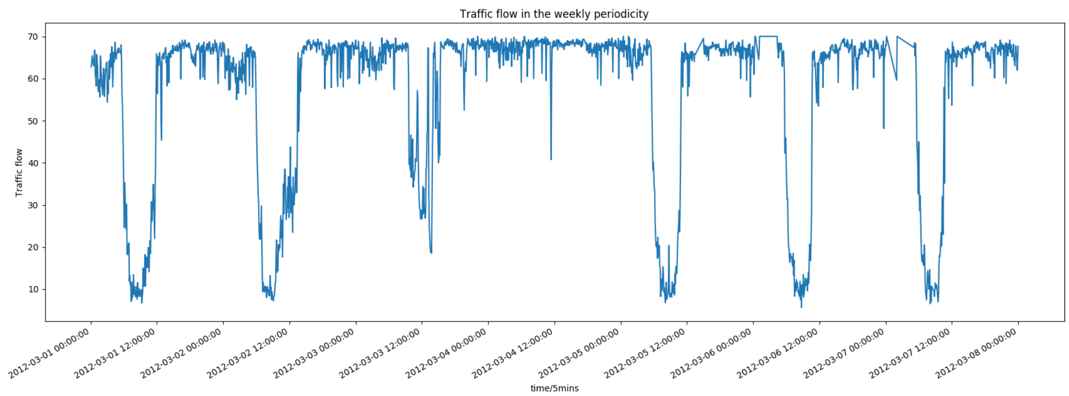


Figure 5.1: Traffic flow data in a week of METR_LA in the Los Angeles County

a validation set from 04/03/2012 to 06/03/2012, and the rest of the days as a test set. We select this dataset because it is relatively small and because the records have much more complexity and fluctuation every minute on the road.

- PeMS04(S)(L): describes traffic flow changes in the San Francisco Bay Area. The records are fron with 307 sensors on 29 roads. It was collected from California Transportation Agencies Performance Measurement System (PeMS) [20] in District04 over two months. The system collects data in real-time every 30s, and the data was aggregated as expected in each experiment. The dataset was created over 59 days, from 1/1/2018 until 2/28/2018, and the total number of time slots is 16,992. For data split, the training, validation, and test datasets of PeMS04(S) and PeMS04(L) were with proportions of around 0.8, 0.1, and 0.1, respectively. This dataset has many records to reflect real San Francisco Bay Area traffic conditions.

  In both datasets, the time step has been aggregated into 5 minutes. METR-LA(L) is only used to compare the performance with STGCN and ASTGCN.

Dataset	Node	Time Step
METR-LA(S)	207	2016
METR-LA(L)	207	34272
PEMS04(S)	307	2016
PEMS04(L)	307	16992

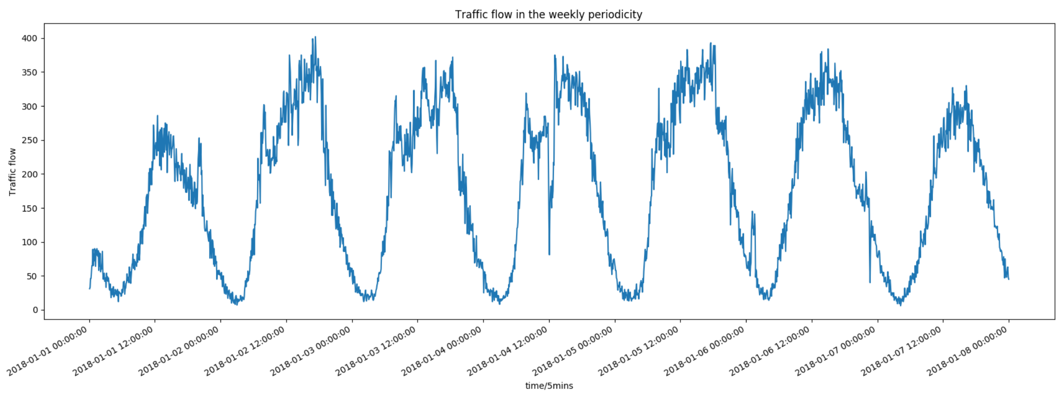


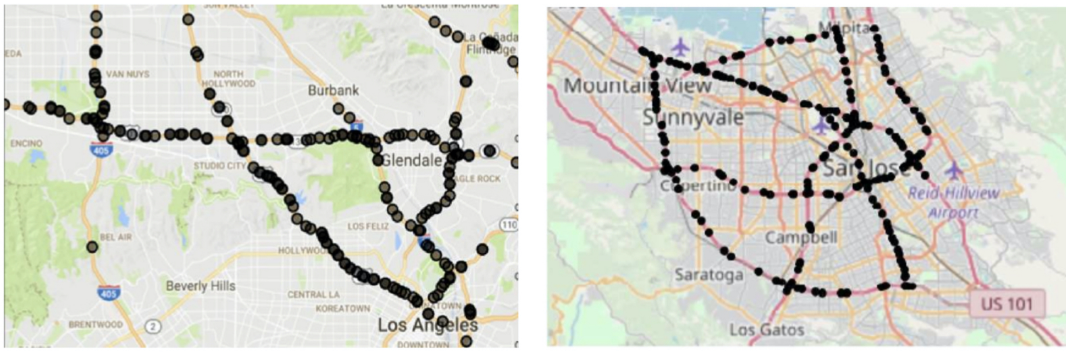
Figure 5.2: Traffic flow data in a week of PeMSD4 in the San Francisco Bay Area

5.1.2 Data patterns

- Temporal dependency

  From Fig. 5.1 and Fig. 5.2, we can see the short-term temporal dependency on both of the figures that the tendency changes every time slot, and the tendency of each day is slightly different; the long-term temporal dependency as a daily pattern can be seen from the figures as well. Besides, as seen from Figure 5.2 the daily periodicity has been shown in the figures. Ever, the weekly pattern of the data shows the similarities of the tendency within the group of weekends and weekdays.

- Spatial dependency
  Seen from Fig. 5.3, the topological information has been shown on the map with the traffic records of the sensors on the road.



a. The map of the traffic flow records of METR_LA in Los Angeles

b. The map of the traffic flow records of PeMSD4 in San Francisco Bay Area

Figure 5.3: The traffic flow data of METR_LA [64] and PeMSD4 [37]

5.2 Model construction

5.2.1 Introduction

Two experiments (Chapter 6 and 7) have been proposed to explore: 1. the effect of the different feature extraction architectures for the spatial and temporal dependency on the traffic flow forecasting task; 2. the effect of a different order of spatial and temporal components on the sequential modeling; 3. the impact of the structure constructed to extract semantic information of the Spatio-temporal dependency on traffic flow forecasting. In order to achieve the purposes of the experiment, this thesis mainly compares different spatial and temporal models as temporal components and spatial components to explore the impact of different feature extraction structures on traffic flow prediction. The construction order for the spatial and temporal components has been investigated via sequential modeling. Further, based on feature extraction pyramids, the framework extracting the features with different semantic information, our work considers models with different levels and scales to extract features with sufficient informative semantics. The central mythology of this thesis is 1. Temporal modeling on extracting temporal features (GRU for the first experiment and CNN for the second); 2. Spatial modeling on extracting spatial features(GCN w/o Chebyshev polynomials); 3. Sequential modeling; 4. Multi-scale feature fusion.

5.2.2 Temporal dependency modeling

Traffic flow forecasting uses historical traffic flow data and real-time information to predict road conditions in the following steps. At the beginning of the related research, the models mainly focus on the time dimension. On the one hand, it is because the fluctuation of traffic data is affected by each time change, which is concluded as the short-term time feature. On the other hand, the traffic flow shares the same patterns in daily, weekly, or even more extended periods; it is also proven to own periodicity features in long time distances.

Recurrent neural network (RNN) has gained much attention because of its ability to process sequential data, which cannot be modeled by classical neural networks for temporal modeling. The RNN mechanism is to retain knowledge about the past to achieve future forecasting. The connection between the units of the recurrent neural network (RNN), such as long-term, short-term memory (LSTM), and gated recurrent unit (GRU), can form a directed loop, which helps to handle the contextual correlation between the time inputs.

On the other side, based on the advantage of processing in large-scale parallel structure and increasing the speed of network training simultaneously, [135] uses the CNN structure to model the temporal dependency named the temporal convolutional network. The results showed that it is better than the latest LSTM and typical GRU models. Moreover, because TCN is more flexible in processing the length of past information, it avoids the problems of gradient dispersion and gradient explosion in RNN. It has generally been regarded as a formidable alternative to RNNs and is worth further exploration [58] [6].

Hence, in this paper, we adopted variant GRU and temporal CNN as two typical models of temporal dependency modeling. The details on how to employ the temporal models in the experiments will be demonstrated in chapters 7 and 8.

5.2.3 Spatial dependency modeling

The spatial structure of the traffic network is non-Euclidean and directional since the transportation networks, in reality, are essentially in a topological graph structure. To accurately describe and use the spatial dependence, the graph neural network and its variants are appropriate to express the traffic network mathematically as required. In this thesis, we adopt two different GCN (w/o Chebyshev polynomials filter) as two typical models of spatial dependency modeling in a static structure. The details on how to employ the spatial models in the experiments will be demonstrated in chapters 7 and 8.

5.2.4 Spatio-temporal dependency modeling

As mentioned in Chapter 3, the relationship between spatial and temporal dependency is also due to the construction of the spatial and temporal components in the models; here, we consider the sequential modeling to extract the Spatio-temporal dependency because non-sequential modeling usually introduces a different fusion method to obtain the Spatio-temporal feature which might lose information on expression if an inappropriate method is adopted. The details of sequential modeling will be discussed in the following section.

5.2.5 Sequential modeling

Recent studies on mining the Spatio-temporal information behind the traffic flow data are mainly in two approaches: sequential modeling and non-sequential modeling.

Sequential modeling extracts the spatial and temporal features by modeling the related information in a certain order [121, 134, 141]. For instance, the Spatio-temporal model's construction could be the spatial-temporal or temporal-spatial sequence. The sequential modeling extracts the Spatio-temporal features in the specific order designed by the researchers. Non-sequential modeling fuses the extracted spatial and temporal features by employing parallel modeling when extracting the spatial and temporal features separately from the sub-models and then adopting the specific fusion method to obtain the Spatio-temporal feature. The architecture of Spatio-temporal modeling can be seen in detail in Fig. 3.4. Guo et al. [42] proposed a framework with Spatio-temporal blocks that combine the spatial correlations from the graph convolutions component and time-series feature from the temporal module constructed as a stack of convolutions layers. Even though Li et al. [64] defined the solution to traffic flow forecasting as the joint combination of diffusion convolutions and sequential models for the Spatio-temporal modeling.

In this thesis, the architecture of the Spatio-temporal model is based on sequential modeling because parallel modeling requires fusing the spatial and temporal, which might cause loss of information that affects the feature expression if the fusion method is not appropriately selected. The exploration of the order of spatial and temporal components in sequential modeling has been shown in Chapter 6.

Furthermore, for multi-scale Spatio-temporal dependency extraction, given the inspiration from object detection in computer vision, we consider the feature extraction pyramid as a basic framework for extracting the semantics of Spatio-temporal features in this thesis, since it can extract the semantic information of features from different levels in sequence.

Overall, combined with the sequential modeling and the structure of the feature extraction pyramid, we propose a basic construction for Spatio-temporal feature extraction in different levels to enhance the expression of the Spatio-temporal dependency.

5.2.6 Multi-scale feature fusion

In this thesis, we consider the representation of multi-scale features in two aspects: the sequential modeling of the Spatio-temporal feature that constructed the spatial and temporal components in sequence and with different scales (kernels) in each layer; a residual connection is introduced to enhance the feature expression at different levels. The informative expression of multi-scale feature fusion outweighs that of the features from single domains, the details of which we discuss in Chapter 7.

5.3 Summary

In the chapter, we explored and analyzed various datasets to better understand and select a subset of specific datasets for our upcoming experimentation.

Chapter 6

Spatio-Temporal dependency with sequential modeling

6.1 Introduction

A large subset of work discusses feature extraction from the temporal and spatial perspectives. However, only a few discuss the effects of model design on the extraction of dependencies and the final prediction. Moreover, most models regard all features equally and omit discussion on the semantic representation of the dependencies from different perspectives, which may contribute to the final results. In this chapter, we identify the following key points:

- As we know, feature engineering plays an essential role in deep learning modeling. Hence, we explore the effect of different feature extraction architectures for Spatio-temporal dependency in traffic flow forecasting tasks. In detail, whether the sequence of the feature extraction could be the critical factor in predicting or which is the better sequence when modeling the traffic flow prediction problem.
- Most research adopts fixed time slots as a historical time window. However, a different historical period for a prediction may not contain the same amount of information,

thus leading to the exploration of the effect of different periods as input for predictions. We further explore how different time window periods may be a factor for prediction.

- We evaluate the regression task in traffic flow prediction using the Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE), or Root Mean Squared Errors (RMSE).

  However, the models are also expected to have an explanation of how the features could explain the results or to what extent they can explain them.

  The results should represent information relating to the datasets or precisely the information that MAE or RMSE cannot measure. Based on this rule, we explore the measurement of various models (GCN, GRU, two-layer GCN, and the sequential spatial and temporal models) mentioned in this thesis. It is done to explore how spatial, temporal, and Spatio-temporal dependency can reflect the information in traffic flow forecasting tasks and to what extent.

6.2 Methodology

Our solution focuses on solving problems in traffic flow prediction, which rely on the spatial and temporal correlations of the obtained traffic flow data and the discussion on the relationship between each other.

6.2.1 Spatial dependency modeling

Much research pays attention to extracting spatial dependencies via the GCN since the GCN considers traffic flow data in a graph structure. At the same time, the classic convolutional neural network focuses on grid data extraction that violates the traffic graph structure.

GCN aggregates the information of all the nodes from the current layer by a nonlinear

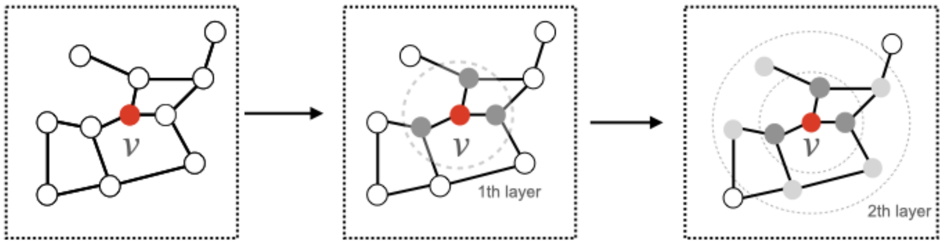


Figure 6.1: lth layer of GCN

function [139]:

$$X^{l+1} = f(X^l, A). \tag{6.1}$$

A is the adjacent matrix of the traffic graph. X^l represents the information of all nodes in layer l that should be updated, and X^{l+1} is the updated layer. We can also stack two layers of GCN, which represent that each node in the current layer can aggregate the features of its 2-hops neighbors to update its features(See Fig. 6.1)

f is the update function that aggregates information from the neighboring nodes. It could be that the GRU function is a Gated Graph Neural Network (GGNN) [66], message function in Message Passing Neural Network (MPNN) [38] or activation function in GraphSage [45] as the model requires.

6.2.2 Temporal dependency modeling

In this thesis, we construct an unweighted graph to depict the spatial relations S in the traffic network. The node v_i is the node i in the road network. The node v_i contains flow information x_t^i at t time slots. In order to model the temporal correlations, we employ the gated recurrent unit (GRU) model.

The given input information of a node h_t^i includes the historical information at t time slot for node v_i; the GRU model will then filter the related information of the obtained updated state and handle the ones that should be maintained in the long-term.

$$\begin{aligned} u_t &= \sigma_{GRU}(W_u * [x_t^i, h_{t-1}^i] + b_u) \\ r_t &= \sigma_{GRU}(W_r * [x_t^i, h_{t-1}^i] + b_r) \\ c_t &= tanh(W_c[x_t^i, r_t * h_{t-1}^i] + b_c) \\ h_t^i &= u_t * h_{t-1}^i + (1 - u_t) * c_t \end{aligned} \tag{6.2}$$

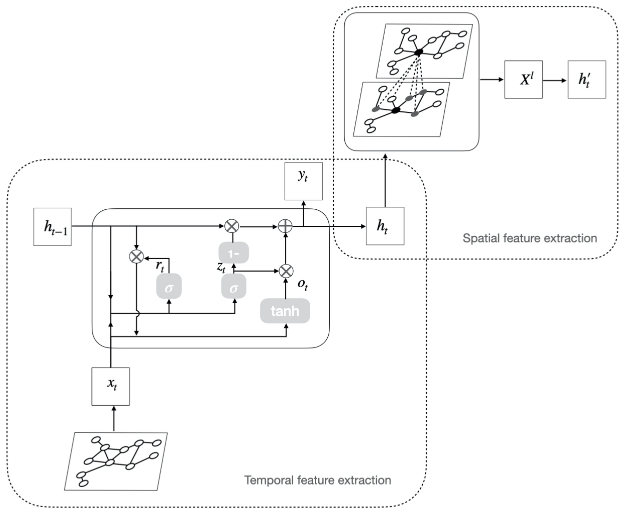
where σ_{GRU} is RELU $(\cdot)$ which is a nonlinear activate function. $W_u, W_r, W_c, b_u, b_r, b_c$ are the learnable parameters. r_t is the reset gate, the output of 0 or 1, to control the previous state information so that the irrelevant information with forecasting can be abandoned. u_t is used to determine to update the quantity of the state information quantity that should bring to the new state. Meanwhile, c_t aims to store the previous related information, and h_t^i is the output state at the current time step.

6.2.3 Spatio-temporal feature engineering

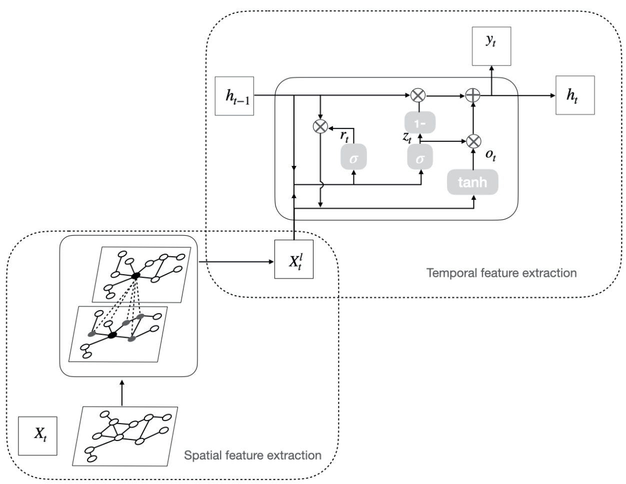
As Feature Engineering is the primary process in deep learning applications, it requires the transformation of the raw data into more expressive features to represent the essence of the problem so that the application of extracted spatial and temporal features can improve the model performance.

Since we are here to discuss the relationship between spatial dependency and temporal dependency modeling in traffic flow prediction, we consider the model structure's design-whether the feature extraction sequence or the feature fusion would be the primary explanation for the model performance, as from the **Fig**. 3.4, the structure can be designed as a spatial-temporal sequence, temporal-spatial sequence, and the Spatio-temporal relation.

Temporal-spatial sequence Based on the discussion above, the temporal features of each node can be extracted from GRU and updated as the expressive information h_t^i at time slot t. After the temporal features are obtained, the spatial dependency of the graph can be described as $f(H_t, A)$ at time slot t with all the temporal features h_t gathered. As the spatial-temporal sequential modeling (**Fig**. 6.2(a)) prefers temporal feature obtained



(a) The structure of Temporal-Spatial sequence modeling



(b) The structure of Spatial-Temporal sequence modeling

Figure 6.2: The design of the model architecture considering the temporal and spatial sequence

and then the spatial feature extraction, the formulation can be based on Equation. 6, thus:

$$\begin{aligned}
u_t &= \sigma_{GRU}(W_u * [x_t^i, h_{t-1}^i] + b_u) \\
r_t &= \sigma_{GRU}(W_r * [x_t^i, h_{t-1}^i] + b_r) \\
c_t &= tanh(W_c * [x_t^i, r_t * h_{t-1}^i] + b_c) \\
h_t^i &= u_t * h_{t-1}^i + (1 - u_t) * c_t \\
H_t &= aggregate(h_t^i) \\
H_t' &= GCN(H_t, A)
\end{aligned} \tag{6.3}$$

.

Spatial-temporal sequence Based on the discussion above, the consideration of l layer spatial dependency of X_t^l, the whole traffic network at a specific time slot t can be described as the results of $f(X_t^{l-1}, A)$. It represents the spatial information of the graph obtained at time slot t. As the spatial-temporal sequential modeling(**Fig**. 6.2(b)) prefers the spatial feature extraction, and then the temporal feature obtained, the formulation can be as:

$$\begin{aligned}
X_t^l &= GCN(X_t^{l-1}, A) \\
u_t &= \sigma_{GRU}(W_u * [X_t^l, h_{t-1}] + b_u) \\
r_t &= \sigma_{GRU}(W_r * [X_t^l, h_{t-1}] + b_r) \\
c_t &= tanh(W_c * [X_t^l, r_t * h_{t-1}] + b_c) \\
h_t &= u_t * h_{t-1} + (1 - u_t) * c_t
\end{aligned} \tag{6.4}$$

6.3 Experiments

6.3.1 Model construction

All the neural networks are built under the Pytorch. Partial model parameters were adopted with the consideration from work [117,134]. The Adam is selected as an optimizer to optimize the model parameters, with the learning rate set at 10^{-3}. The exponential decay method is adopted for the learning rate. It uses a more extensive learning rate at the beginning to quickly obtain a better solution and gradually reduces the learning rate

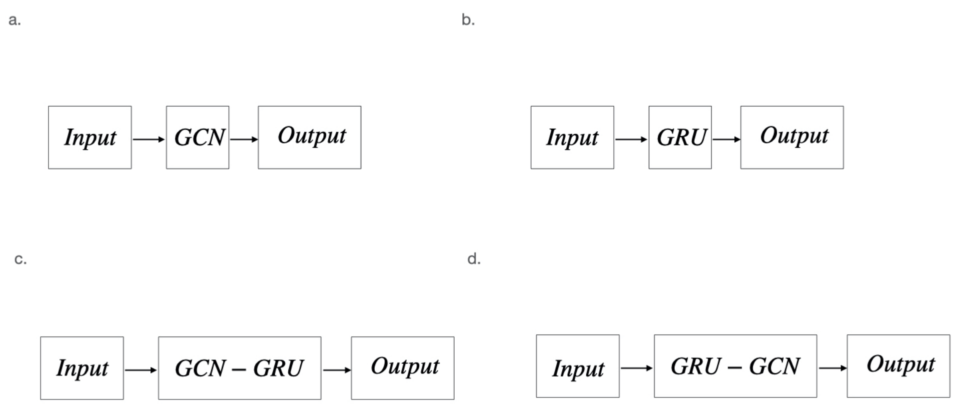


Figure 6.3: The different structures of the models in the experiment section. a.GCN, b.GRU, c.GRGCN, d.GCGRU

as the iteration continues. The batch size is 128. The historical time window size has been set to [3, 6, 12] for the 15-, 30-, 60 mins periods to forecast traffic conditions in the next time step.

An appropriate number of hidden layer units is essential to the model's performance. The number of neurons in a hidden layer decides how much information can be conveyed from the input layer to the next layer. On the one hand, fewer neurons will lead to underfitting; on the other hand, too many neurons can cause overfitting. As we set the hidden units based on the information from the input, we choose the hidden units from [8, 16, 32, 64, 128, 256, 512] to get the best one for the feature expression. As a result, the METR-LA and PeMS04 adopt 128 as the hidden units based on the parameters selection experiment.

6.3.2 Baseline

From the feature engineering perspective, we try to find whether incorporating spatial topology will improve model performance. We want to discover the impacts of the differences when considering the sequence or combination for modeling with spatial and temporal feature extraction. Hence, we compare the proposed work for sequential spatial-temporal

(GRU-GCN) modeling (named GCGRU) and sequential temporal-spatial (GCN-GRU) modeling (named GRGCN) with **GRU** (one layer) and **GCN** (one layer).

GCGRU is constructed with the GCN layer for the spatial dependency extraction and the GRU layer for the temporal feature extraction.

GRGCN has one GRU layer for the temporal dependency extraction and a GCN layer for the spatial feature extraction.

Figure 6.3 shows the ability of each model when extracting the time-series feature and spatial feature in sequence from traffic flow data.

Finally, we also consider how different layers of GCN could affect the spatial information in the extracted spatial features. Hence, two-layered GCN is set to the baseline as well. We train the models by minimizing the mean square error using the MSEloss loss function.

6.3.3 Evaluation metrics

The evaluation metrics are for evaluating the performance of the model comparison.

1. Mean Absolute Error(MAE): MAE is to measure the prediction errors between the observations and values; the smaller, the better.

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - x_i| \tag{6.5}$$

2. Mean Absolute Percentage Error (MAPE): MAPE is the average of the percentage errors that divides each error individually by the demand.

$$MAPE = (\frac{1}{n})\sum_{i=1}^{n}\frac{|y_i - x_i|}{x_i} \tag{6.6}$$

3. Root Mean Squared Error(RMSE): RMSE measures the differences between estimators by models and the values obtained from the datasets. The smaller the values

observed, the better the performance of the model.

$$RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - x_i)^2} \tag{6.7}$$

4. Accuracy(ACC): Accuracy is a measure of the prediction precision.

$$Accuracy = 1 - \frac{||Y - \hat{Y}||_F}{||Y||_F} \tag{6.8}$$

5. Coefficient of Determination(R^2): The coefficient of determination describes the model's ability to how well the prediction results can represent the original values.

$$R^2 = 1 - \frac{\sum_{i=1}(Y_t - \hat{Y}_t)^2}{\sum_{i=1}(Y_t - \bar{Y})^2} \tag{6.9}$$

6. Explained Variance Score(Var): Explained Variance Score also expresses the proportion of the variation in the dependent variable that can predict the independent variable. However, the difference between R^2 and Var is that the explained variance adopts the slight variance to determine how well the variance is explained, while R^2 applies the raw sums of squares. Hence, the two scores should be the same if the results are unbiased.

$$Var = 1 - \frac{Variance\{Y - \hat{Y}\}}{VarianceY} \tag{6.10}$$

6.4 Results and discussion

Table 6.1 and 6.2 demonstrate the results of the proposed GCGRU and GRGCN baselines on the datasets PeMS04. Our proposed model achieves better performance with statistical significance in all three evaluation metrics compared with one-layer GRU and GCN. We can easily observe that:

- When adopting the same number of layers on feature extraction, the sequential feature extraction for the Spatio-temporal dependency modeling could better contribute to the performance than the individual feature extraction with the baselines. The proposed GCGRU and GRGCN show that Spatio-temporal dependency can better explain the dataset since the higher proportion (R^2 and Explained_VAR) accounts for the variation of a given data set.

- When considering the sequential feature extraction, adopting the spatial feature model first and then temporal feature extraction could perform well.

- Compared with one-layer GCN adoption, two-layer GCN improves performance. It can perform better than the scenarios considering the Spatio-temporal feature combination in some cases.

Model	PeMS04(15/30/60 mins)		
	MAE	MAPE	RMSE
GRU	66.13/67.83/69.84	1.09/1.07/1.13	81.16/83.37/84.92
GCN	81.26/80.45/106.90	0.77/0.43/0.52	103.51/107.18/141.76
GCN(2)	33.06/34.43/44.17	0.34/0.36/ 0.45	48.46/50.19/60.24
GCGRU	34.61/34.98/39.68	0.38/0.35/0.44	50.05/50.94/56.45
GRGCN	37.04/ 37.86/39.54	0.36/ 0.37/0.38	53.65/54.63/57.12

Table 6.1: Performance comparison (MAE/MAPE/RMSE) of different approaches on the dataset PeMSD4

Model	PeMS04(15/30/60 mins)		
	ACC	R^2	Explained_VAR
GRU	0.70/0.66/0.55	0.70/0.69/0.59	0.76/0.74/0.72
GCN	0.60/0.59/0.46	0.60/0.58/0.38	0.78/0.73/0.56
GCN(2)	0.81/0.80/0.74	0.88/0.88/0.77	0.88/0.88/0.81
GCGRU	0.81/0.80/0.79	0.88/0.87/0.87	0.88/0.87/0.87
GRGCN	0.79/0.79/0.87	0.86/0.85/0.87	0.86/0.86/0.87

Table 6.2: Performance comparison (ACC/R^2/Explained_VAR) of different approaches on the dataset PeMSD4

Benefits of Spatio-temporal dependency

Given the results of applying the individual spatial and temporal models for feature extraction, Spatio-temporal dependency modeling could outperform. Hence, feature engineering, or precisely feature fusion, can improve the model's performance. The results demonstrate that the Spatio-temporal feature extraction can better use features with different characteristics and jointly model these different features.

Benefits of informative spatial information

Compared with the one-layer GRU and one-layer GCN, The proposed GCGRU and GRGCN incorporate spatial topology and time series information under the sequential feature extraction. The results show that the feature fusion of both spatial and temporal dependency could outperform the individual model.

Besides, when applying two-layer GCN, it has achieved a significant improvement in results, showing that much more spatial information could be captured by more layers adopted. The results of it show that it could perform the same as employing the one-layer Spatio-temporal layer (GCGRU and GRGCN).

Benefits of long-term temporal dependency The MAE, MAPE, and RMSE of one-layer GCN show that the different historical time windows could affect the expression of features in traffic flow data, which performs well for short-term forecasting but is unsuitable for long-term prediction. Compared with one-layer GRU, it leads to the idea that the long-term historical period may contain long-term temporal dependency because the better results when applying GRU in the longer historical period.

The results for the dataset METR-LA:

Table 6.3 and 6.4 demonstrate the results of the proposed GCGRU and GRGCN baselines on the datasets METR-LA. Compared with the performance of dataset PeMS04, the results of datasets METR-LA do not show significant differences in the evaluation metrics. GRU outperforms other models on the dataset because of the highest R^2 and Explained_VAR, representing the best expression of the temporal features extracted by GRU. At the same time, GCN shows the worst performance with the highest evaluation metrics in MAE, MAPE, and RMSE and the lowest feature expression in R^2 and Ex-

Model	METR-LA(15/30/60 mins)		
	MAE	MAPE	RMSE
GRU	5.30/5.19/5.00	0.12/0.11/0.12	7.07/6.63/6.81
GCN	6.16/6.13/6.29	0.13/0.14/0.15	8.11/8.82/8.96
GCN(2)	5.52/5.46/5.50	0.13/0.12/0.12	8.05/8.14/8.09
GCGRU	5.45/5.47/5.64	0.13/0.13/0.13	7.94/8.06/8.32
GRGCN	5.82/5.32/5.30	0.13/0.12/0.11	8.03/7.66/7.73

Table 6.3: Performance comparison (MAE/MAPE/RMSE) of different approaches on the dataset METR-LA

Model	METR-LA(15/30/60 mins)		
	ACC	R^2	Explained_VAR
GRU	0.88/0.89/0.89	0.69/0.71/0.70	0.70/0.72/0.71
GCN	0.86/0.86/0.86	0.57/0.49/0.48	0.58/0.50/0.49
GCN(2)	0.86/0.86/0.87	0.58/0.57/0.57	0.58/0.57/0.57
GCGRU	0.86/0.86/0.86	0.59/0.58/0.55	0.59/0.58/0.55
GRGCN	0.86/0.87/0.87	0.58/0.61/0.61	0.58/0.62/0.61

Table 6.4: Performance comparison (ACC/R^2/Explained_VAR) of different approaches on the dataset METR-LA

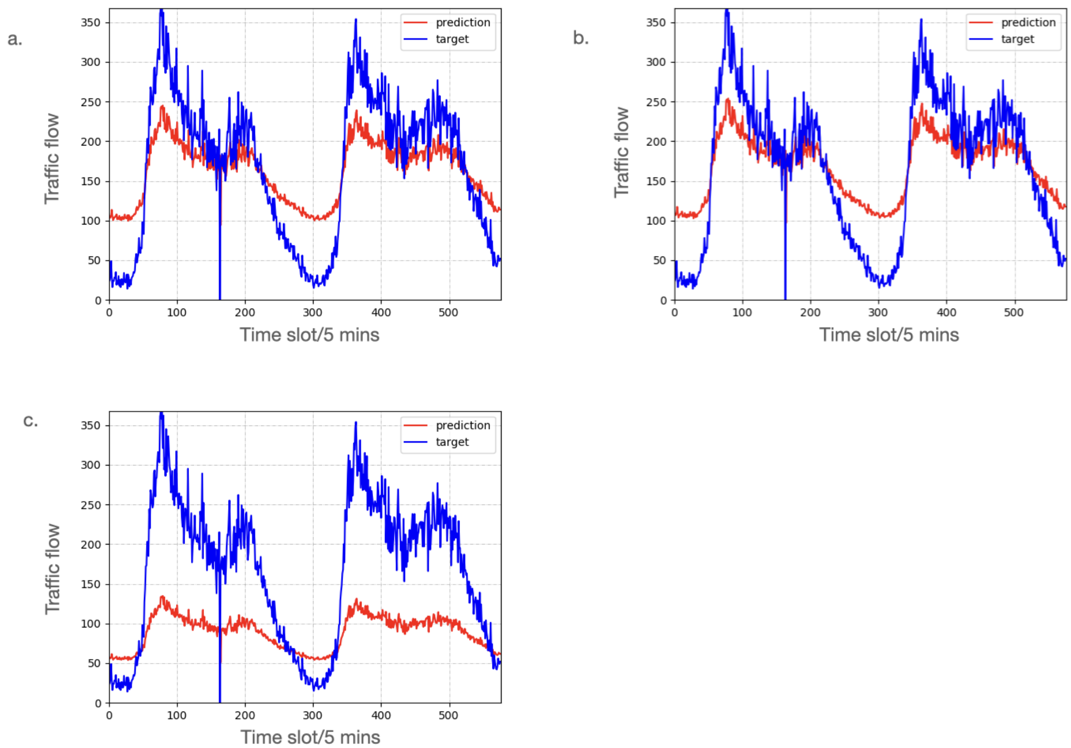


Figure 6.4: GRU captured the overall trend in PeMS04 but can not reach extreme values (historical period in 15-, 30-, 60 mins)

plained_VAR. The proposed GCGRU and GRGCN perform better than GCN but worse than GRU.

Compared with the results obtained from PeMS04, we investigate the results of METR-LA from the relationship between the model and data:

- **The amount of training data** The length of the training set for PeMS04 and METR-LA is 45 days and seven days. The amount of training data leads to the differences in the effects of mining the hidden patterns and correlations within extensive data sets to predict outcomes. Hence, the performance of all the models with METR-LA tends to show fewer differences in the results.
- **The temporal trend of the dataset.** See Fig. 6.4 and Fig. 7.1; the figures show

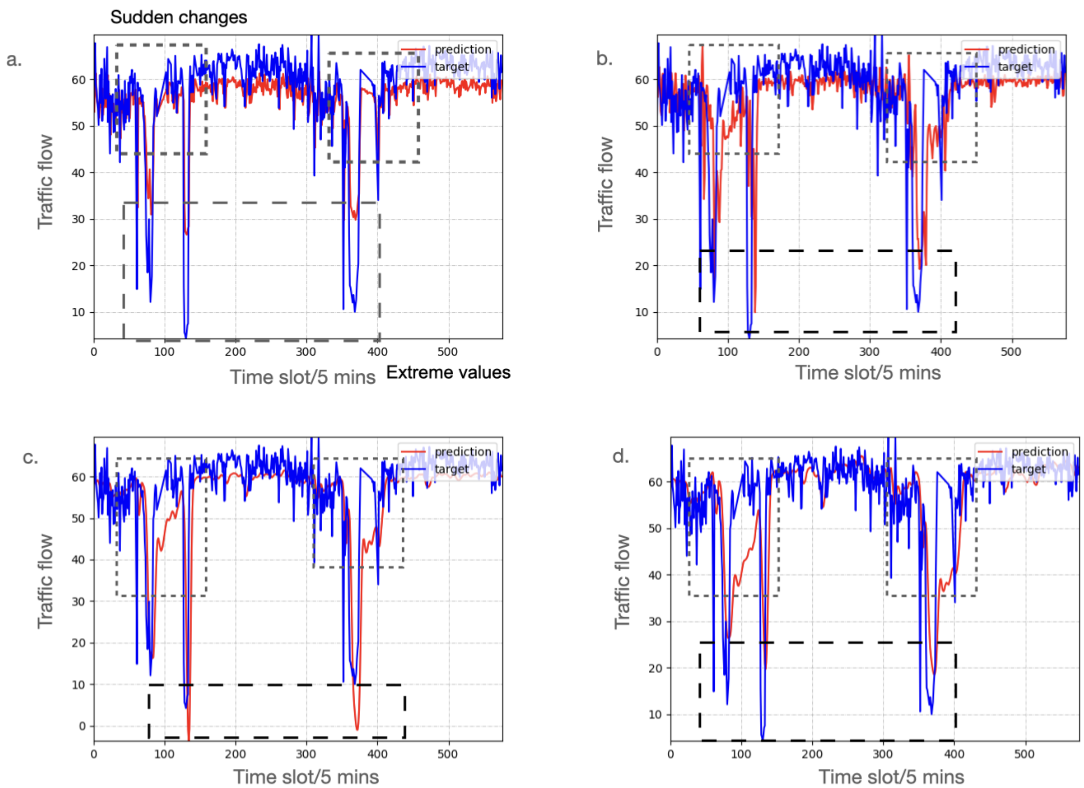


Figure 6.5: The differences in the feature expression in METR-LA, when the historical period is 30 mins with GRU, GCN, GRGCN, and GCGRU, applied. a.GRU (captures the major tendency but does not reach the minimum and maximum value), b.GCN (reaches the extreme values of the trend), c.GRGCN (reaches the extreme values of the trend but delays in describing the sudden fluctuations), d.GCGRU (less ability to reach extreme values and delay in capturing the sudden fluctuations)

the fluctuation of the traffic flow in two days for the two datasets. The changes of the selected sensor of PeMS04 display a similar periodical change in two-day periods. At the same time, METR-LA shows an unstable temporal tendency that frequently changes irregularly in the same given period. The models cannot perform well with datasets that show remarkably different distributions. Hence, the proposed model should improve robustness for different growing trends.

- **The temporal expression of the base models.** From the temporal perspective, the temporal pattern of both datasets can be captured by the temporal models (GRU). Nevertheless, the amount of the time-series information extracted by GRU shows the limitation of the application. In PeMS04, GRU can describe the overall trend (6.4), but hard to capture the sudden changes to the extreme values. The same problem exists in METR-LA: GRU can depict a significant trend but can not reach the related minimum and maximum limits. The explanation is that the forecast by GRU is affected by the expression of the historical data because the memory of the previous trend in the last time steps influences the tendency in future steps, thus limiting the tendency to express extreme values. The expected extraction of the temporal feature is supposed to depict the overall trend and the ability to capture the sudden changes.

- **The spatial expression of the base model.** See the Table. 6.1, R^2 of GCN in 30-mins prediction is 0.58, showing that the spatial feature extracted by the GCN does not contribute to the prediction so much. However, from Fig. 7.1.b, we can see that model may not be able to capture the fluctuating trend in METR-LA, but it can reach the values for both maximum and minimum.

- **The ability of feature fusion.** As the proposed model performs better than GRU and GCN in PeMS04, it is essential to understand how the spatial and temporal correlations create trends. GCGRU and GRGCN improve the model performance in expressing Spatio-temporal dependency but enhance the ability in different perspectives. From Fig. 7.1 a., b., and c., we can see that GCN and GRGCN show the same ability to capture the extremes as that in raw data within the tendency. So GRGCN enhances the ability to reach extreme values. Alternatively, exactly, GRGCN enhances spatial modeling in prediction when considering the temporal correlations. From Fig. 7.1 a., b., and d., we can see that GRU and GCGRU both cannot reach the maximum and minimum in trend. Thus GCGRU enriches the ability of temporal modeling.

The ability of the feature expression for both GCGRU and GRGCN relies on the combination of the feature extracted from the based models (GRU and GCN). Hence, the proposed models are expected to adopt a much more robust feature fusion method for better expression by capturing the primary trend and extending the tendency range to extreme values.

6.5 Summary

In this chapter, we mainly explored whether consideration of spatial features in traffic flow prediction problems will affect the results; and whether the different sequential modeling of feature extraction for the spatial dependency and temporal dependency on the Spatio-temporal feature extraction could influence the model's performance. Experiments show that considering the Spatiao-temporal model outperforms the sole spatial and temporal models. Moreover, the sequential modeling for the Spatio-temporal dependency extraction will be affected based on the architecture designed. On two real-world datasets, the results indicate the great potential for exploring Spatio-temporal feature engineering from the input. Further, comparing two-layered GCN and our proposed models reflects that the richer spatial information could improve model performance. Because the two-layered GCN aggregates not only the information from nearby but also the neighboring information at two hops, it indicates that dynamic spatial dependency could achieve better model performance and that the multi-scale spatial information may also contribute to the performance.

Chapter 7

Spatio-temporal dependency with multi-scale semantics

7.1 Introduction

Although encouraging solutions have been proposed to capture the Spatio-temporal correlations and present non-linear approaches in the past several decades, they still face some challenges. Researchers still need help to fully exploit the complex spatial-temporal dependency from traffic flow data because of the relationship between the dynamic periodic fluctuation and the complicated topology of the road network, leading to high randomness and non-linearity characteristics in traffic flow data.

As a fundamental time series forecasting problem, it is hard to balance the exploration of long-term and short-term dependency modeling. The difficulty has prevented researchers from finding an answer for trading off the long-term and short-term temporal dependency when modeling: Long-term time dependency needs to consider periodic information. In contrast, short-term correlations focus on capturing fine-grained temporal relations because the traffic flow changes dynamically and irregularly with local timing. Hence, the relationship of different temporal scales should be considered in the model.

Each node on the graph contains different spatial information, and it is because each

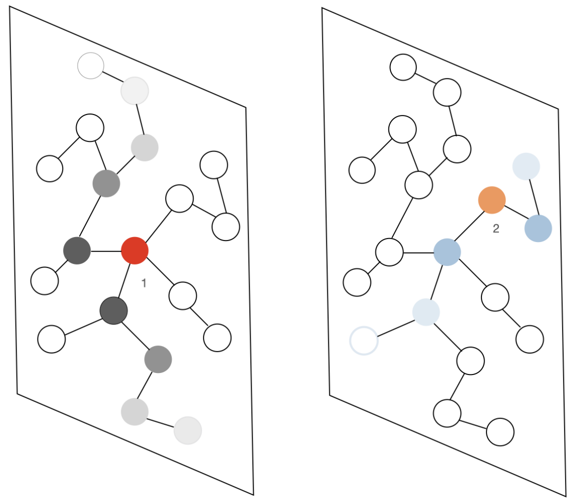


Figure 7.1: Node 1 requires the spatial information captured from the neighbors within 5-hops, while that of Node 2 should be within two hops

node has a different influence on its neighbours. Previous studies only assumed that the spatial correlations for neighbouring node targets are within the same hop. For example, they all assumed spatial information of the central node was extracted from the spatial dependence within k surrounding hops. However, the spatial information of each node on the map is from different scales. For instance, given a city-wide traffic network, this great event could actively affect the linked adjoining roads to differing extents when traffic congestion happens in a specific area. Thus, traffic jams last in a radiation zone, and the different nodes in the spot could be affected on different scales. If the size of spatial scales for the target is different, capturing or fetching the correct information on the spatial dependency from the traffic map is easier. As seen from Fig.7.1, the sufficient spatial information of Node 1 contains neighbors within five hops, while that of Node 2 should be within two hops. Therefore, the traffic flow changes due to the effect of the different spatial scales on the central node within the surroundings instead of the equal scale on the traffic map.

Less research discusses how each component in the model contributes to the final prediction. The solutions from the previous work give overall ideas on capturing the Spatio-temporal features from the traffic flow data in two ways: one, to capture the distinct spatial and temporal features from the corresponding models in sequence or parallel [68]; two, construct a Spatio-temporal block to extract the spatial and temporal features. However, they do not explain how each component can contribute to the final prediction. In other words, previous solutions inspire the current research on Spatio-temporal modeling but ignore the analysis of the methods on how to design the architecture of each component to traffic flow forecasting. Since the amount of information can be conducive to finding a more graphic representation of the raw data based on the architecture of the network, the discussion on how to design the Spatio-temporal model architecture for capturing features is also essential to the problem. Guo et al. [1] implied that simultaneous architecture could be used and found the importance of describing spatial-temporal correlations in long-term time-series prediction, which is compared with separate utilization of spatial dependency and temporal dependency.

Even though some studies tried to explain the purpose of the proposed spatial and temporal components, the input of the solution is expected to be manually pre-processed and thus is not considered an end-to-end architecture. Time costs are introduced for pre-processing, which is inappropriate in the objective case.

Further, the quality of the features plays an essential role in deciding model performance because it integrates the correlated features of the target to enhance the representation of the raw data. The primary benefit of feature-level fusion is to utilize the representativeness of the different algorithms to extract correlated features, thereby generating a set of salient features which can improve final accuracy. In the application of deep learning in the work of computer vision and image processing, researchers have realized that to describe how the content of data at the semantic level is the premise of making full use of information from the dataset. Lin et al. [67] bring the idea that a sufficient multi-scale feature representation with expressive semantic information would improve the expression of datasets. The definition of multi-scale is the method of sampling different granularity of the signal or input.

Moreover, most of the solutions in traffic flow forecasting consider the fixed input sequence as the historical period horizon to predict the traffic flow in the next time steps. However, more work should have discussed finding the appropriate input length when considering the different Spatio-temporal information in the prediction horizon. The relationship between the input sequence length and the horizon period is rarely explored by the literature [91], leading to the exploration of the effect of the information contained when deciding the historical horizon to address traffic flow forecasting.

Moreover, for the regression task of traffic flow prediction, the evaluation of most work focuses on the discussion of Mean Absolute Errors (MAE) and Root Mean Squared Errors (RMSE). However, it is expected to have an explanation of how the features could explain the results or to what extent it can explain it when we want to discuss the contribution of spatial and temporal components to the final prediction. It requires the evaluation of the coefficient of determination (R^2) and explained variance score (Var) to express the proportion of each component that can predict the results. So, we have to explore the evaluation metrics for the models mentioned in the current literature (spatial models, temporal models, and sequential Spatio-temporal models), thus providing a detailed exploration of how the models can reflect the corresponding information in traffic flow forecasting tasks.

With the above concerns, in this work, we propose the Hierarchical Multi-scale Spatio-Temporal Semantics Graph Convolutional Network (HMSTGCN) to explore the semantics expression at different scales of the Spatio-temporal correlations in the traffic flow network. HMSTGCN can simultaneously capture the spatial and temporal dependency from traffic flow data and mine the hidden Spatio-temporal relationships within the multi-scales of each target on the map. We rely on the architecture that adopts the different sizes of kernels on both spatial and temporal modeling via a top-down pathway and combines low-level but semantically strong features with high-level but semantically weak features via skip connection. This structure trades off time series of short-term and long-term dependency in a temporal perspective and unites the spatial information of each node from different scales. We identify the contribution of this work as follows:

- We propose a new end-to-end spatial-temporal model that focuses on information utilization and feature fusion from multiple temporal and spatial scales to improve the semantic correlations of the Spatial-temporal feature. Spatial-temporal dependency learned within multi-scales is extracted from the different layers in a hierarchical architecture, thus enabling effective time-series prediction for traffic flow forecasting, especially in the long-term prediction.

- We discuss how each spatial and temporal component and the detailed residual parts can contribute to the final prediction with an ablation study.

- We provide an analysis of the structure of the construction of our spatial components and temporal components to discover the impact of the scale size on spatial information on the traffic map and the number of temporal layers to exploit the hidden temporal dynamics in different scales.

- We conduct several experiments on mining different amounts of Spatiotemporal information contained within different input lengths of historical periods for enabling model design. We consider the relationship between the input sequence and the prediction horizon and whether the different lengths of the input sequence could be a factor for time-series prediction in traffic flow.

- We show the coefficient of determination (R^2) and explain the variance score (Var) to express the proportion of each component that can predict the results.

7.2 Related work

Temporal modeling During the beginning of traffic flow forecasting research, researchers defined the problem as a time series prediction task because the previous states of traffic conditions can predict the future traffic conditions of a road. According to different historical periods, temporal dependency could be divided into short-term and long-term temporal dependency in traffic flow forecasting.

Compared with the statistical methods, Neural-network-based models outperformed others in capturing non-linear temporal dependency. Because the output of RNN is from the previous step and is fed as the input to the current step, the mechanism models how previous information is updated to the next time step in real-world scenarios. The Recurrent neural network (RNN) and its variants (Long short-term memory, Gated Recurrent Unit) have gained much attention. Although RNN-based approaches adopted gate mechanisms to regulate the flow of information to keep the valuable data in a sequence and throw away the less meaningful data, the dropped information might be utilized in the future. Also, because of the time-consumption problem in RNN, some work considers Gated CNN as the temporal component for time series modeling, which processes the input in large-scale parallel structure and increases the speed of network training.

With further study in temporal modeling, the temporal convolutional network (TCN) was proposed with an architecture that combined the dilated convolutions and residual connections to adapt for sequential data modeling. Duranton et al. [135] demonstrate that TCN outperforms the latest LSTM of RNN in temporal modeling. Furthermore, it avoids the gradient dispersion and explosion problems in RNN. However, even though the TCN seems to be an impressive architecture, Zhang et al. indicated that the number of TCN layers could be an additional concerned topic on short-term flow prediction [125]. It is because the shallow structure of the TCN architecture could be challenging for complex temporal relationships, while the deeper layers could result in over-fitting.

Spatial modeling The consideration of spatial models employed in traffic flow forecasting is based on the topology of the traffic network. In the beginning, the road segments were regarded as 2D structured data, which allowed transportation research [28, 114, 115, 138] to apply CNN to exploit the grid-based structure in traffic flow data. In contrast, recent research [84, 111] pointed out that the graph-based network outperforms traffic network modeling because the transportation map under real conditions has more complicated non-euclidean spatial correlations in graph-wide topological structures.

Sequential modeling Recent studies on mining the Spatio-temporal information behind traffic flow data are mainly done via sequential modeling and non-sequential modeling. Sequential modeling extracts the spatial and temporal features by modeling the related information in a certain order [121,134,141]. Non-sequential modeling fuses the extracted spatial and temporal features in the required way by employing parallel extraction for spatial and temporal features separately and then adopting a fusion method for the Spatio-temporal feature. For example, Guo et al. proposed the framework with the Spatio-temporal blocks that combine the spatial correlations from the graph convolutions component and time-series feature from the temporal module constructed by a stack of convolution layers [42]. Even though Li et al. defined the solution to traffic flow forecasting as the joint combination of diffusion convolutions and sequential models for Spatio-temporal modeling [64], it is under the same perspective as we have for the solution as the model for spatial and temporal components.

Multi-scale Spatio-temporal feature

Yang et al. designed the Spatial-temporal block considering multi-scales of Spatio-temporal correlations, the construction of which are the combination of the Gated DC layers and the Graph Attention layers [111]. They pointed out the effectiveness of Spatio-temporal modeling when considering the feature extraction in different scales that could contribute to the precision of the long-term trend by utilizing the spatial context and the granularity of temporal correlations in different scales.

Observing fluctuation in the temporal domain shows that high-level temporal features change slowly. In contrast, the low-level features quickly change, thus implying that temporal correlations are sensitive to the precise local timing with temporal coherency [48]. The employed temporal models must dynamically adapt the timescales for the various time-series inputs. The most popular approach is to manually pre-process the dataset into various timescales and adopt parallel extraction for the inputs under different scales [42,126] instead of learning the multiple temporal scales directly from the traffic flow data.

The solution focuses on multi-scale spatial feature extraction for traffic flow forecasting problems that can be inspired by object detection. As objects in the image are of different sizes and locations, multi-scale architectures, such as the feature pyramid network, can combine low_level features to distinguish simple objects and high_level features to detect complex objects for an informative feature expression. For small targets, large-scale feature maps cannot deliver the crucial resolution information to distinguish small targets, so it is necessary to provide a feature map that can capture objects on a small scale.

7.3 Problem statement

The traffic flow forecasting problem can generally be defined as a time series problem since it aims to predict future traffic indicators over time steps.

Definition 1: Feature matrix X^{N*D}. For each node i, it has characteristics x_i, which can be represented in a feature matrix X^{N*D}. In the feature matrix X^{N*D}, N represents the number of nodes, and D is denoted as the number of features of each node and can also be regarded as the dimension of the feature vector. A node has given historical traffic indicators that can be denoted as $X_t^i = [x_{t-T+1}^i, \ldots, x_t^i]$ over the past T time slots.

Definition 2: Traffic graph $G = (V, E)$. V is a set of nodes on the traffic graph, and E is a set of edges. $V = \{v_1, v_2, \ldots, v_N\}$ refers N nodes on the graph, and each node represents the sensor's location. The edge E refers to the relation between two nodes, represented by the adjacency matrix $A \in R^{N \times N}$, containing the topological information of the road network. The adjacency matrix plays a vital role in accurate spatial modeling in GNN. It is because the expression of the node feature highly relies on the neighbours' information. The graph adjacency matrix aims to incorporate the information of one from its neighbor nodes and distant ones. In other words, the adjacency matrix is the key to describing the spatial relations of the nodes in the graph [109]. If the road segment is related to the adjacent road, the element is 1 in the matrix. Otherwise, it is 0.

Definition 3: Traffic flow forecasting: The future traffic condition over the next T' time slices of all nodes on the graph is represented as $Y = [y_{t+1}, \ldots, y_{t+T'}]$. Hence, the solution for the traffic flow prediction with spatial correlations is: $Y = f(X, G) = f([x_{t-T+1:t}], G)$. $[y_{t+1}, \ldots, y_{t+T'}] = f([x_{t-T+1}, \ldots, x_t])$, where

$$T' = \begin{cases} 1, & one - step\ prediction \\ n, & multi - step\ prediction \end{cases}$$

. In this thesis, T' is set to 1, which means we predict the future traffic conditions of the next time steps the same as the historical period.

7.4 Methodology

Yao et al. [115] mentioned that weak correlations within the specialty can hurt performance for spatial modeling. Hence, the HMSTGCN tries to mine the Spatio-temporal dependency in different scales to enhance the correlations behind the traffic flow data.

7.4.1 Graph convolutional network for the spatial component

GCN methods [35] apply to the graph-structured traffic network because of the ability to model non-euclidean data. It is to capture information on the neighboring nodes and edges. The two methods are from the spectral domain and spatial domain, respectively. In the spectral domain, the basic idea of graph convolution operations is to first convert the topological graph signal from the spatial domain to the spectral domain by the Fourier Transform as $f(t)->\hat{f(t)} = U^T f(t)$, and multiply it with the convolution kernel $g_\theta(\lambda)$ in the spectral domain. And then transform the outcome back to the spatial domain through the Inverse Fourier Transform. It is because the convolution operation in the spatial domain is the same as the multiply operation in the spectral domain. The extracted feature in the spectral domain is $\hat{y} = \hat{f(t)} \cdot g_\theta$. Since we are working on the dependency in the spatial domain, the extracted feature should be converted by the Inverse Fourier Transform as $y = U\hat{y}$, where U is the eigenvalue that is related to the Laplacian matrix of a graph. Laplace matrix is a vital matrix used in graph theory. In a graph $G = (V, E)$, the normalized Laplacian matrix of a graph is defined as $L = D - A$, where D is the degree matrix of the graph, and A is the adjacency matrix of the graph. Hence, we can have

$$H^l = \sigma(LH^{l-1}W^l) \tag{7.1}$$

to describe the update operation in the spectral domain, where W^l is the weighted parameter matrix of the lth layer, and σ is a non-linear activation function, such as ReLU. The Laplacian matrix L is multiplied by the feature matrix H. However, there are two problems with this operation: it ignores the impact of the node itself. We can have $L^{sym} = D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}} = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}} = I_a + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, which introduces the Symmetric normalized Laplacian to solve the self-transmission problem.

$$\begin{aligned} H^{l+1} &= \sigma(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}H^lW^l) \\ &= \sigma(D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}}H^{l-1}W^l) \\ &= \sigma((I_n - D^{-\frac{1}{2}}AD^{-\frac{1}{2}})H^{l-1}W^l) \end{aligned} \tag{7.2}$$

is proposed to solve the mentioned problem.

Nevertheless, we take the original definition as an example. Since $L = U\Lambda U^T$, U is the eigenvalue and Λ is the eigenvector, we can expend the formulation:

$$S = U\hat{y} = U\hat{f(t)} = Ug_\theta U^T f(t) \tag{7.3}$$

The hidden state of a node refers to the information of a node we mentioned above, for example, defined as $h_v^{t+1} = f(x_v, x_c, h_n^t, x_n)$, which represents the hidden state of node v in the layer $t+1$ is updated by the aggregation of the embedded node feature, the edge feature, the hidden state of the neighbors, and the edge feature of the neighbors. Because the node information and the Laplacian matrix can be obtained directly from a graph, the difference of the spectral GCN lies in the filter g_θ.

The filter can be restricted to the Chebyshev polynomials [57] for parameter reduction. The Chebynet method defines the value of the convolution kernel in the spectral domain as a function related to the eigenvalue of a graph, which can be used to approximate the Chebyshev polynomials. Since $g_\theta = \sum_{k=0}^{K} \theta_k T_k(\Lambda)$, the function can be as :

$$\begin{aligned} S &= Ug_\theta U^T f(t) \\ &= U \sum_{k=0}^{K} \theta_k T_k(\Lambda) U^T f(t) \\ &= \sum_{k=0}^{K} \theta_k T_k(U(\Lambda)U^T) f(t) \\ &= \sum_{k=0}^{K} \theta_k T_k(L) f(t) \end{aligned} \tag{7.4}$$

Because the T_k is the Chebyshev polynomials, θ_k is the coefficient to be learned. Kipf and Welling [57] proposed work that simplified the Chebynet based on the Chebyshev polynomials. It only considered the 1st-order Chebyshev polynomial, and each convolution kernel has only one parameter. The stack of the Chebyshev layers represents the scales of the spatial information extracted. We can see from the Figure. 7.2 c. that the spatial feature extraction has different scales from the spatial component.

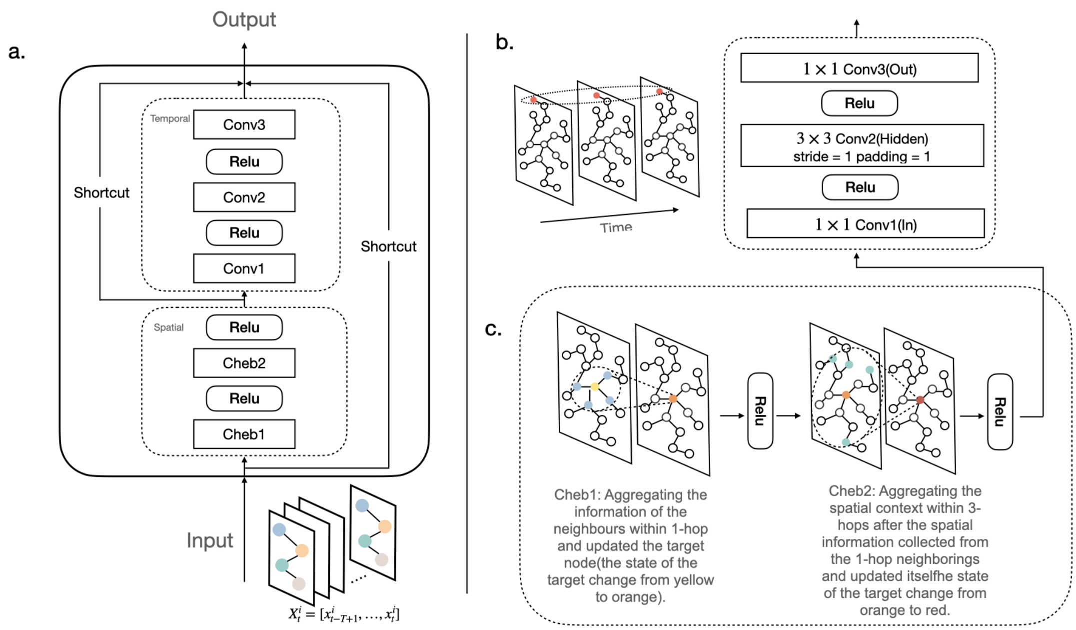


Figure 7.2: Fig a. illustrates the structure of the HMSTGCN. b. shows the temporal feature extraction. c. shows the spatial feature extraction in different spatial scales

7.4.2 Convolutional neural network for the temporal component

The RNN-based models adopted gate mechanisms to regulate the flow of information, which enhances the temporal dependency since the gates keep the valuable data in a sequence and throw away the less meaningful ones. However, the dropped information might be utilized and contribute to long-term temporal correlations. Also, because of the time cost of the model training in RNN-based variants, some work considers CNN to handle long temporal sequences through multi-layer structures that formed as hierarchical representations to capture the temporal correlations. The design allows parallel computation and controllable training procedures to alleviate the gradient explosion problem.

The basic temporal convolutional layer contains a 2-D causal convolution with K size kernel to capture the temporal correlations in K-hops. We adopted the temporal convolutional layers in multiple layers to extract the time-series information in a certain period (as seen from Fig. 7.2 b.) within different scales by defining the different kernel sizes, thus, the output of a temporal convolution layer can be regarded as a transformation of the temporal information on the time axis within a certain period. The convolution kernel $g(n)_i$ is designed to map the input in a specific ith temporal convolutional layer, the size of which varies based on the layers. It is in a definition as:

$$T = g(n)_i \odot F(t) \tag{7.5}$$

, where $F(t)$ is the extracted information from the previous layer. And

$$F(t) = \begin{cases} S, \ if \ i = 1 \\ F(t)_{t-1}, \ if \ i \ ! = 1 \end{cases}$$

. Furthermore, residual connections are implemented after the stacked temporal convolutional layers to fuse the temporal information with different semantics from high to low levels.

7.4.3 Residual mechanism

We designed residual connections to reconstruct the traffic Spatio-temporal features by fusing the semantic context learned from lower levels. We introduce the Spatio-temporal semantics from the two parts through residual links to help the Spatio-temporal information reconstruction. First is Fig. 7.2 a. demonstrates that the spatial and temporal components provide the features extracted within different spatial and temporal scales to achieve feature expression, and the residual connections enhance the information semantics with features from the lower level.

7.5 Experiment

7.5.1 Model construction

All experiments are compiled and tested on a Linux environment (GPU: NVIDIA GeForce GTX 2060 Ti). The observed periods are in 3/6/12 slots to forecast traffic conditions in the next 15-, 30-, and 60- minutes. The learning rate is 0.01 for all experiments. The batch size of the experiments is set to 128, and the training epoch is 50, where we set the early stop mechanism to avoid over-fitting so that our model converges to an end at around epoch 8.

7.5.2 Baseline

CNN:Convolutional Neural Networks can be used for sequence modeling, which captures the sequential data's temporality by stacking the layers. It can detect the potential temporal dependencies of the targets between the different scales and are not next to each other based on the kernel size of the CNN.

Chebynet [83]: Graph convolutional network (GCN) is a class of graph neural networks for processing data that is on graph-based structure with the spectral framework to apply convolutions in spectral domains. It introduces the Chebyshev Polynomials Approximation as the kernel. It can depict the topological relationship between the central road and the neighbours in the surrounding regions.

STGCN [121]: STGCN proposed a framework that combines the graph convolutional layers and convolutional structures on the time axis to extract the spatial and temporal dynamics.

ASTGCN [42]: ASTGCN combined the spatial-temporal attention mechanism to capture the dynamic spatial-temporal characteristics of traffic data simultaneously

ScaCNN: We propose a Multi-scale Convolutional Neural Network named ScaCNN in our work. This model is a multi-scale CNN that is constructed with the multi-layer convolutional network structures with different sizes of kernels that formed as hierarchical representations for capturing the temporal features in different scales, which can extract the temporal dynamics on the time axis, the kernels of each layer are 1, 3, 1.

ScaCheb: We propose a Multi-scale Chebynet network named ScaCheb in our work. This model is a multi-scale Chebynet network constructed with the multi-layer Chebynet structures formed as hierarchical representations for capturing the spatial features which can represent the local spatial context with different scales. The kernels of each layer are 1, 3.

ResCNN: We propose a Resnet-like Convolutional Neural Network named ResCNN in our work. This model is constructed based on ScaCNN by introducing the residual connection to fuse the temporal features from a low level. It is under the hierarchical structure of the sequence for capturing the temporal features containing different semantic information from different levels.

ResCheb: We propose a Resnet-like Chebynet named ResCNN in our work. This model is constructed based on the ScaCheb with the introduction of the residual connection to fuse the spatial features from the low-level feature. It is under the hierarchical structure of the sequence for capturing the spatial features containing different semantic contexts from different levels.

We use the default settings described in the other papers' baseline models.

7.5.3 Evaluation metrics

We evaluate the performances of all baselines by three commonly used metrics in traffic forecasting.

1. Mean Absolute Error (MAE): MAE is to measure the prediction errors between the observations and the values. The smaller, the better.

$$MAE = (\frac{1}{n}) \sum_{i=1}^{n} |y_i - x_i| \tag{7.6}$$

2. Mean Absolute Percentage Error (MAPE): MAPE is the average of the percentage errors that divides each error individually by the demand.

$$MAPE = (\frac{1}{n}) \sum_{i=1}^{n} \frac{|y_i - x_i|}{x_i} \tag{7.7}$$

3. Root Mean Squared Error (RMSE): RMSE measures the differences between estimators by models and the values obtained from the datasets. The smaller the values observed, the better the performance of the model.

$$RMSE = \sqrt{(\frac{1}{n}) \sum_{i=1}^{n} (y_i - x_i)^2} \tag{7.8}$$

4. Accuracy (ACC): Accuracy is a measure of the prediction precision.

$$Accuracy = 1 - \frac{||Y - \hat{Y}||_F}{||Y||_F} \tag{7.9}$$

5. Coefficient of Determination (R^2): The coefficient of determination describes the model's ability to show well that the prediction results can represent the original values.

$$R^2 = 1 - \frac{\sum_{i=1}(Y_t - \hat{Y}_t)^2}{\sum_{i=1}(Y_t - \bar{Y})^2} \tag{7.10}$$

6. Explained Variance Score (Var): Explained Variance Score also expresses the proportion of the variation in the dependent variable that can predict the independent variable. Nevertheless, the difference between R2 and Var is that the explained variance adopts the slight variance to determine how well the variance is explained, while R2 applies the raw sums of squares. Hence, the two scores should be the same if the results are unbiased.

$$Var = 1 - \frac{Variance\{Y - \hat{Y}\}}{Variance Y} \tag{7.11}$$

Model	METR_LA(15/30/60 mins)		
	MAE	MAPE	RMSE
CNN	3.13/3.37/3.91	6.32/7.00/8.27	5.55/6.16/6.93
Chebynet	2.84/3.05/3.42	5.90/6.32/ 7.24	4.50/4.89/5.62
ScaCNN	3.54/3.48/4.18	7.44/7.52/9.34	5.25/5.31/6.14
ScaCheb	2.92/2.97/3.29	6.24/6.31/6.99	4.55/4.82/5.40
HMSTGCN	2.59/2.71/3.08	5.10/5.37/6.13	4.34/4.50/5.18

Table 7.1: Performance comparison (MAE/MAPE/RMSE) of the consideration on feature from single domain and our Spatio-temporal feature extraction method on the dataset METR_LA

7.6 Results

7.6.1 Performance and discussion

As shown in Table 7.17.2 and Table 7.3,7.4, our models achieve outstanding performance in the listed datasets in different sizes, which indicates the effectiveness of our model. We can easily observe that:

Model	METRA_LA(15/30/60 mins)		
	ACC	R^2	Explained_VAR
CNN	90.58/89.54/88.26	79.47/74.79/68.10	79.61/75.06/68.70
Chebynet	92.36/91.70/90.46	86.40/84.03/78.90	86.52/84.29/79.40
ScaCNN	91.09/90.99/89.57	81.75/81.33/75.20	81.82/81.45/75.31
ScaCheb	92.27/91.83/90.84	86.21/84.56/80.57	86.40/84.81/81.09
HMSTGCN	92.63/92.37/91.22	87.20/86.36/82.16	87.25/86.45/82.17

Table 7.2: Performance comparison (ACC/R^2/Explained_VAR) of the consideration on feature from single domain and our Spatio-temporal feature extraction method on the dataset METR_LA

Model	PeMSD4(S)(15/30/60 mins)			PeMSD4(L)(15/30/60 mins)		
	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE
CNN	18.35/19.59/23.39	14.85/16.61/21.10	28.14/29.19/33.40	22.09/24.33/29.71	14.74/16.68/21.17	34.60/37.13/43.70
Chebynet	40.90/30.74/27.37	53.6/43.01/34.08	54.11/41.68/37.54	47.39/35.67/33.29	53.37/41.46/34.32	63.71/49.23/46.67
ScaCNN	16.82/16.83/18.54	16.43/15.82/19.65	25.23/25.14/27.02	19.54/20.26/23.15	15.21/16.22/19.32	30.32/31.02/34.57
ScaCheb	18.01/18.51/20.96	18.92/20.35/22.44	26.41/26.90/29.76	21.06/21.43/25.29	19.04/18.75/21.42	31.66/31.86/36.96
HMSTGCN	16.39/16.58/17.51	12.59/12.85/13.88	25.30/25.42/26.90	19.09/19.47/20.95	12.51/12.80/14.01	30.20/30.56/32.82

Table 7.3: Performance comparison (MAE/MAPE/RMSE) of different approaches on the dataset PeMSD4(S/L)

Model	PeMSD4(S)(15/30/60 mins)			PeMSD4(L)(15/30/60 mins)		
	ACC	R^2	Explained_VAR	ACC	R^2	Explained_VAR
CNN	86.44/85.73/83.36	93.90/93.36/91.19	94.02/93.74/92.21	86.70/85.52/82.66	94.04/93.23/90.66	94.12/93.46/91.29
Chebynet	73.2/79.30/81.56	77.89/86.42/89.10	81.72/88.22/90.40	75.20/80.80/81.63	80.62/88.12/89.36	82.56/88.65/89.98
ScaCNN	87.90/87.84/86.82	95.11/95.12/94.34	95.18/95.28/94.62	88.36/88.00/86.49	95.42/95.24/94.13	95.46/95.31/94.33
ScaCheb	87.28/86.91/85.29	94.62/94.38/93.06	94.75/94.70/93.73	87.83/87.23/84.98	95.00/94.61/92.77	95.07/94.75/93.12
HMSTGCN	87.92/87.85/87.13	95.09/95.05/94.44	95.13/95.12/94.52	88.45/88.30/87.45	95.42/95.32/94.60	95.45/95.37/94.69

Table 7.4: Performance comparison (ACC/R^2 /Explained_Var) of different approaches on the dataset PeMSD4(S/L)

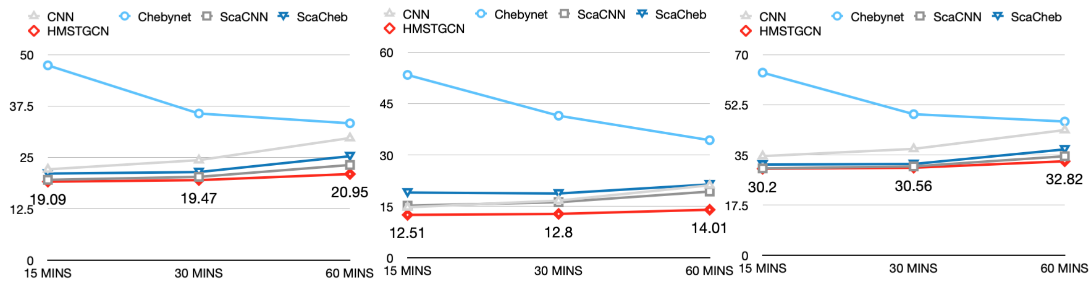


Figure 7.3: The comparison among the impact of multi-scale Spatio-temporal correlations (HMSTGCN) with the single-scale correlation (CNN, Chebynet) and multi-scale spatial (ScaCheb) and temporal (ScaCNN) correlation in PeMSD4 dataset.

- Spatio-temporal correlation can enhance the performance of traffic flow forecasting, surpassing that of the individual feature extraction in the single domain (GCN, CNN).
- The introduction of semantic information on spatial and temporal features can improve the performance of traffic flow forecasting in the corresponding domain. Moreover, the multi-scale Spatio-temporal features outperform all the single domain or single-scale features. It reveals the importance of capturing the Spatio-temporal semantics in traffic flow forecasting.
- The combination of the spatial and temporal features which fuse the information in the different scales will contribute to the performance, with significant improvements in long-term forecasting.

Benefits of Spatio-temporal dependency

The 1-D CNN is designed for considering temporal features in sequential data by stacking the multiple layers, while 2-D CNN can extract the temporal information over the time axis. Here, we adopted the one-layer 2-D CNN to depict the information transferred in the time domain. The GCN (Chebyshev polynomials) is to extract the spatial information from the k-hops neighbors, and it is constructed in a 2-layer structure.

Compared to the performance of the single factor (GCN, Chebyshev polynomials, and CNN), which captures the spatial information or temporal correlations of the traffic flow for 15-, 30-minute, and 60 minutes ahead predictions on the tested datasets, HMSTGCN obtains superior results on all the datasets implying the effect of Spatio-temporal dependency to the traffic flow forecasting. Our model considers the Spatio-temporal correlations to receive better precision, indicating that our model can capture the Spatio-temporal feature from traffic flow data, which contains the complex and non-linear Spatio-temporal co-relations instead of only temporal or spatial features.

We can see from Table 7.1 for the traffic forecasting period in 15-, 30- 60-mins of dataset METR_LA, and our model performs better compared with the Chebynet model in spatial feature extraction, and the apparent improvement compared to the temporal model (CNN).

Benefits of Multi-scales dependency

We visualize the results (MAE/MAPE/RMSE) of the PeMSD4 dataset to better view the impact of the multi-scale dependency compared with the single-scale dependency from a spatial and temporal perspective. From Fig. 7.3, the benefits of extracting the multi-scale correlations from spatial (Blue), temporal Gray), and Spatio-temporal (Red) perspectives are shown clearly. Comparing CNN, a one-layer temporal model for single-scale temporal feature extraction with the light grey colour in Fig. 7.3, with ScaCNN, a multi-scale temporal model with the dark grey colour in Fig. 7.3, reveals the difference in the results when considering multi-scales temporal dependency in traffic flow forecasting. By comparing Chebynet (light blue) with the ScaCheb (dark blue), the models of single-scale and multi-scale spatial models in two layers, the benefits of employing sufficient semantic context on spatial feature fusion can be explicitly demonstrated. The features from the single scales contain less semantic information than those extracted from the multiple scales of spatial and temporal dependency. The expressive spatial and temporal semantics contribute to traffic flow forecasting, seen from Figure 7.4 as the R^2 and explained_var, are much higher than that from the single scale.

An exception is shown in Table 7.1, where multi-scale CNN receives the worse results in MAE and MAPE when compared with that of single-scale CNN. However, the multi-scale CNN performs well overall since it performs well in other aspects. Besides, the results of R^2 and explained_var also imply that sufficient temporal semantics will contribute to a better explanation in modeling.

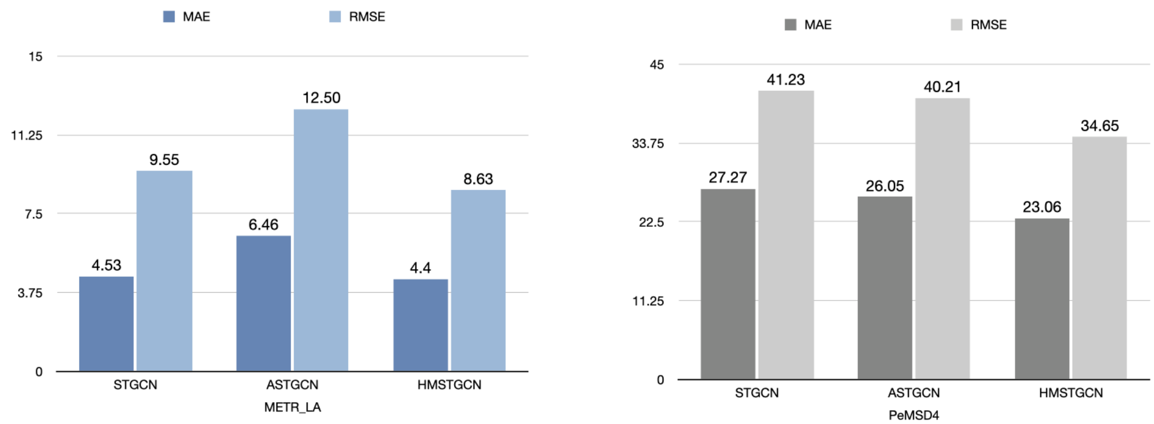


Figure 7.4: Forecasting performance comparison of STGCN, ASTGCN, and our model on METR_LA(L) and PeMSD4(L).

		MAE	MAPE(%)	RMSE	ACC	R^2	Explained_VAR
METR_LA	w/o Temporal	2.67/2.81/3.27	5.51/5.62/6.86	4.28/4.42/5.21	92.73/92.51/91.16	87.64/86.70/81.80	87.72/86.79/82.02
	w/o Spatial	2.95/2.69/3.65	6.14/5.43/7.62	4.52/4.65/5.45	92.33/92.14/90.75	86.19/85.59/80.32	86.24/85.67/80.55
	w/o ResTemporal	3.54/3.48/4.18	7.44/7.52/9.34	5.25/5.31/6.14	91.09/90.99/89.57	81.75/81.33/75.20	81.82/81.45/75.31
	w/o ResSpatial	2.92/2.97/3.29	6.24/6.31/6.99	4.55/4.82/5.40	92.27/91.83/90.84	86.21/84.56/80.57	86.40/84.81/81.09
	HMSTGCN	2.59/2.71/3.08	5.10/5.37/6.13	4.34/4.50/5.18	92.63/92.37/91.22	87.20/86.36/82.16	87.25/86.45/82.17
PeMSD4_S	w/o Temporal	19.49/21.88/22.50	14.92/29.67/20.12	29.54/30.69/33.02	85.96/84.72/83.95	93.31/92.56/91.51	93.61/93.65/91.89
	w/o Spatial	16.28/16.90/19.49	12.55/15.89/21.50	24.98/25.40 /28.40	88.04/87.80/86.32	93.79/95.04/95.21	93.90/95.14/95.27
	w/o ResTemporal	16.82/16.83/18.54	16.43/15.82/19.65	25.23/25.14/27.02	87.90/87.84/86.82/	95.11//95.12/94.34	95.18/95.28/94.62
	w/o ResSpatial	18.01/18.51/20.96	18.92/20.35/22.44	26.41/26.90/29.76	87.28/86.91/85.29	94.62/94.38/93.06	94.75/94.70/93.73
	HMSTGCN	16.39/16.58/17.51	12.59/12.85/13.88	25.30/25.42/26.90	87.92/87.85/87.13	95.09/95.05/94.44	95.13/95.12/94.52
PeMSD4_L	w/o Temporal	22.70/25.22/26.56	15.10/28.14/19.92	35.04/36.14/39.53	86.62/86.00/84.78	93.84/93.53/92.21	94.11/93.87/92.34
	w/o Spatial	19.16/19.42/22.88	12.63/16.58/20.87	30.09/29.90/34.06	88.47/88.51/86.84	95.47/95.54/94.26	95.51/95.60/94.33
	w/o ResTemporal	19.54/20.26/23.15	15.21/16.22/19.32	30.32/31.02/34.57	88.36/88.00/86.49	95.42/95.24/94.13	95.46/95.31/94.33
	w/o ResSpatial	21.06/21.43/25.29	19.04/18.75/21.42	31.66/31.86/36.96	87.83/87.23/84.98	95.00/94.61/92.77	95.07/94.75/93.12
	HMSTGCN	19.09/19.47/20.95	12.51/12.80/14.01	30.20/30.56/32.82	88.45/88.30/87.45	95.42/95.32/94.60	95.45/95.37/94.69

Table 7.5: Ablation study on METR-LA and PeMSD4(S/L)

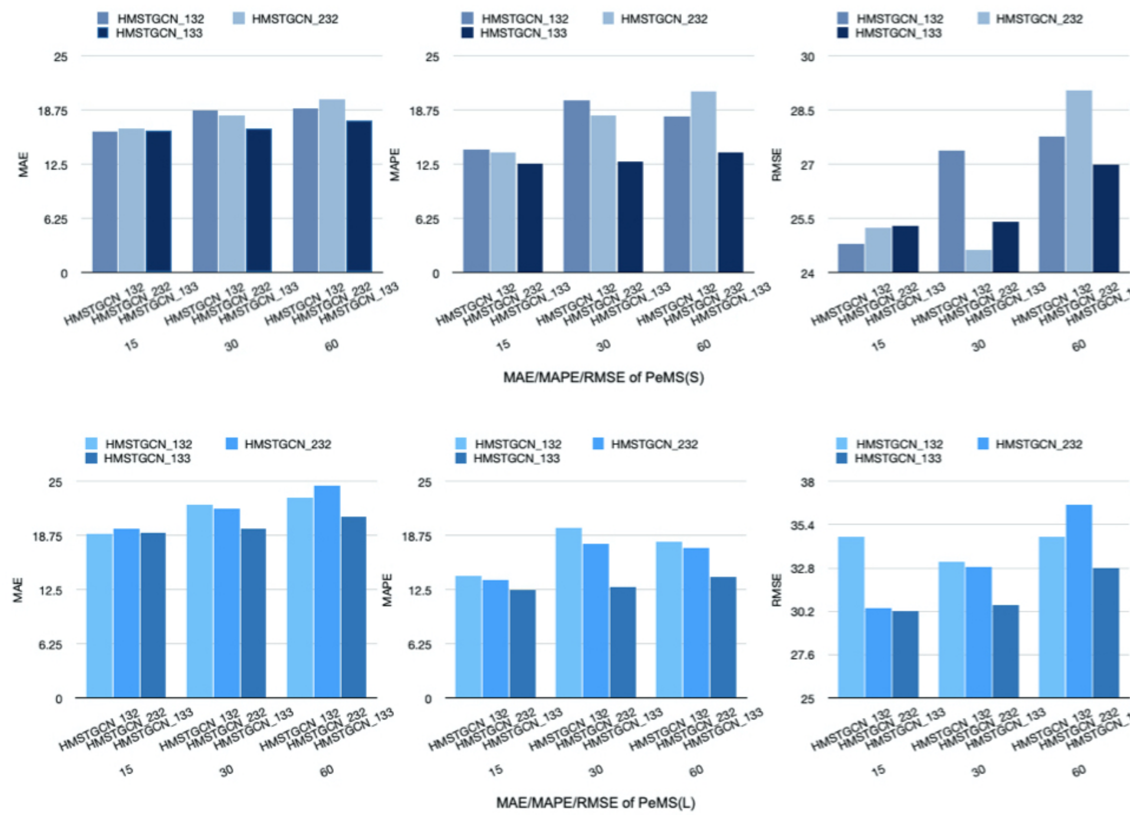


Figure 7.5: Performance comparison of the results (MAE, MAPE, RMSE) on PeMSD4(S) and PeMSD4(L).

Benefits of multi-scale Spatio-temporal dependency in long-term prediction As seen in Fig 7.3, there is significant improvement when utilizing multi-scale Spatio-temporal correlation in the traffic forecasting period within 15-, 30- 60-mins of dataset METR_LA. The conclusion goes the same way when comparing the results in Table7.3 and 7.4. In particular, the performance of the HMSTGCN validates the necessity to consider the multi-scale Spatio-temporal dependency in the long-term trend for traffic flow forecasting.

Fig. 7.4 compares the forecasting performance of the STGCN, ASTGCN, and our model in the 60-mins historical period. The reason why we adopt 60-mins as the predicting period relies on the fact that [68] pointed out that the ability of an effective model depends on depicting the long-term dependence of the periodicity and trend accurately because the complex periodic correlations make long-range forecasting of traffic data more difficult than other predicting tasks. For example, forecasting the future 12-time steps is generally more complex than the next 1st-3rd time steps. Moreover, traffic data contains spatial dynamics that are also diverse due to the intricate traffic flow on the road network. STGCN also considers the structure of the Spatial-temporal GCN model as us, and ASTGCN adopted the attention mechanisms to enhance the ability to capture the temporal correlation of long sequences. We can see from Fig. 7.4 that compared with STGCN and ASTGCN, our model performs best, demonstrating the impact of the multi-scale Spatio-temporal features that can contribute to traffic flow forecasting, especially in long-distance prediction.

7.6.2 Ablation study

In this part, we will conduct ablation studies to investigate the effect of the spatial, temporal, and residual components in our model; we evaluate four variants by removing the whole Temporal part, the Temporal Residual part, the whole Spatial part, and the Spatial Residual part from HMSTGCN separately. The model without the whole Temporal or Spatial part is the structure of ResCheb and ResCNN.

We design four variants of our spatial-temporal framework in the following way: W/o Temporal removes the whole Temporal part, remaining the two-layer Chebynet with the scale within the size of 1 and 3, and the residual connection with the scale size of 2. It is to verify whether they are interchangeable. W/o Spatial removes all the components that extract the spatial feature in traffic flow data, with only the ResCNN structure left as a three-layer CNN with kernel size as that of ScaCNN and an extra residual link to receive the temporal feature from the low level. W/o ResSaptial is to remove the Residual connection that conveys the spatial feature from the low level to a high level, while w/o ResTemporal removes the residual links that transfer the low-level spatial feature, leaving the structure of ScaCNN that the kernel size of it are 1, 3, 1.

The result is shown in Fig. 7.5. The results of w/o Temporal and w/o Spatial suggest that the spatial and temporal correlations hidden in the traffic flow data can contribute to the prediction separately. From an architectural aspect, compared with individual modeling, jointly considering the temporal and spatial components in our model does make a significant difference. However, the distinct impacts of each part are not the same in the specific dataset. It is noticed that the sole spatial component makes a significant contribution compared to the sole temporal component for forecasting the traffic flow in different regions. With the METR_LA dataset, which reveals the information in the LA area, the results of R^2 worth an average of 85.38% and 84.03% for the spatial part and temporal part, respectively, and do not make an impact as that in the PeMSD4 area, no matter the size of the dataset. The R^2 of the corresponding components is worth an average of 92.48%, 94.68% in the PeMSD4(S) dataset, and 93.9%, 95.09% in PeMSD4(L) dataset, which present a different impact of the components to different datasets. In addition, the data contains spatial and temporal information about the distinct area that is unique from other regions. The model is expected to mine the characteristic hidden relationships from the dataset as the Spatio-temporal relationship is not the same in each area.

Compared with the component with residual connections, the results of w/o ResSpatil and w/o ResTemporal reveal that further detailed construction on the residual connections of spatial and temporal information from the low level is necessary for the model on utilizing features fused at a different level. In the experiments to METR_LA dataset, compared with the spatial component with the residual connections, removing residual low-level links of spatial and temporal information could decrease the performance. However, the results in the PeMSD4 dataset demonstrate that the sole spatial or temporal feature fusion from the low level might not contribute to the results. It shows that the accuracy could increase

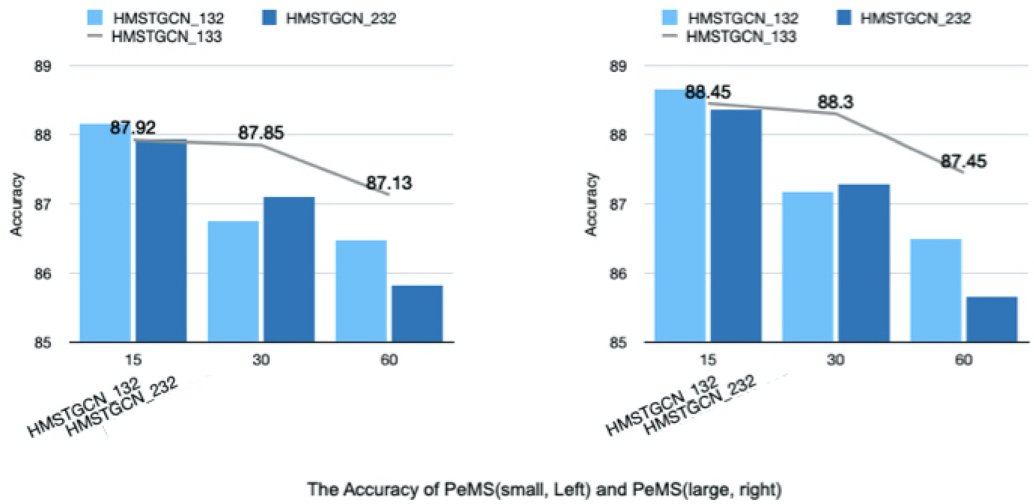


Figure 7.6: Performance comparison of the accuracy on PeMSD4(S) and PeMSD4(L).

when removing the spatial residual links from the spatial component of the models applied to the PeMSD4 dataset, no matter the size of the datasets. However, compared with our model (HMSTGCN), it indicates that the performance of the Spatio-temporal feature from different levels is much better than the individual models on capturing the features from different granularity in the single domain because our model outperforms all the ablation experiments.

7.6.3 Architecture study

To further discuss the possibility of our model and its powerful ability to capture Spatio-temporal features, we design experiments to explore the model's architecture based on the current structure. Recall that our HMSTGCN comprises spatial and temporal blocks composed of multi-scale layers to capture the corresponding correlations and add an extra residual connection to fuse the feature from different levels. The designed Chebynet is constructed in a two-layer structure for the spatial block extracting the spatial correlations from specific neighbours with different hops. We compose the kernel sizes of the spatial block in (1, 3) and (2, 3). The temporal scales of the HMSTGCN do not only rely on the kernel sizes of each layer but also the number of layers for temporal feature extraction. The temporal block is constructed in (1, 3, 1) and (1, 3). Hence, the Spatio-temporal scales of the control group are designed as ((1, 3), (1, 3)), ((1, 3), (1, 3, 1)), ((2, 3), (1, 3)), named as HMSTGCN(1, 3, 2), HMSTGCN(1, 3, 3) and HMSTGCN(2, 3, 2).

Figure 7.5 illustrates the impact of the model architecture on small and large PeMSD4 datasets. It shows how the different number of temporal layers and spatial sizes contribute to the three variants. We can see from the Figure that our model(HMSTGCN(1, 3, 3)) surpasses in 60-mins period prediction in both datasets, implying that our model has a powerful ability to capture Spatio-temporal correlations in the long term. Further, it should be noticed that the HMSTGCN(1, 3, 2) outperforms others in the 15-minute forecasting. Figure 7.6 also demonstrates that the HMSTGCN(1, 3, 3) receives the best performance in the long-term prediction while HMSTGCN(1, 3, 2) outperforms in the related short-term forecasting within 15 mins.

Thus, after exploring the architecture of the proposed experiments, two points should be noticed: first, the amount of information leads to the design of the size of the structure. From Figure 1, we can see that the 15-minute period contains less information than that in 30-and 60-mins, thus guiding to the fact that fewer temporal layers can help to avoid the outfitting. It is verified by the results that HMSTGCN(1, 3, 2) outperformed HMSTGCN(1, 3, 3) in the 15-minute forecasting; second, the larger differences within the sizes of the spatial scales in different layers could enhance the model's capacity for capturing correlations. The idea was also demonstrated by HMSTGCN(1, 3, 2) outperformed HMSTGCN(2, 3, 2).

7.7 Summary

This chapter presents a hierarchical semantic model for depicting spatial-temporal correlations with informative expressions. Our model captures spatial-temporal features by combining the Chebynet graph and convolutional neural networks. We employed the different scales for each component to balance the granularity of the corresponding feature expression, and our method can learn the features automatically without manually re-processing the dataset. We also designed experiments to discuss the contribution of each spatial and temporal component to forecasting. We analyze how the construction of the different components will also contribute to forecasting, and the results show that different amounts of information contained in the historical period lead to different depths of each component. We came up with a solution by introducing the Spatio-temporal semantics into traffic flow forecasting, but we need to discuss the effect of different weights on the scales. In future work, we will study more scalable methods for extracting Spatio-temporal semantics and look for approaches to dynamically depict the spatial dependencies.

Chapter 8

Future directions and Conclusion

We have shown various perspectives for finding solutions to traffic flow forecasting via relevant spatial-temporal models in recent years, which were: first, introducing graph-based models for capturing the topological structure of the traffic road network instead of considering the extraction of grid-based spatial information as in the previous work; and then, proposing the detailed sub-categories of the spatial and temporal dependencies, such as the periodicity in hourly, daily, and weekly under the category of the long-term temporal dependency; and, finally, expressive feature engineering for solutions to the traffic flow prediction problem. Besides the work we mentioned in previous sections, some other techniques can potentially handle problems in this area.

8.1 Future work

8.1.1 The reconstruction of the graph structure

With the success of the region-level solution proposed, we can catch sight of the idea behind it: the high similarities among nodes will contribute to the occurrence of node groups with similar properties. Based on its inspiration, the future solution should consider an appropriate graph partition to reduce the redundant graph information for the targets, thus improving the efficiency and performance of the model.

In a realistic scenario, the graph-structured traffic road network is on a large scale brings unnecessary information for prediction, which directly reflected by the time-consuming problem, which is always a heated discussion.

Two reasons can explain it in a large graph: First, compared with all the nodes on the traffic graph, the links to the target nodes are limited when the adjacency matrix is constructed graph-wide. Hence, the significant traffic graph shows the highly sparse characteristic when all the noted are connected only with their neighbors. Also, the traffic network generally enlarges in city size, which brings enormous topological edge information to the graph matrix. The other relevant factor is an increasing number of features proposed to model the scenario in recent work. The feature aggregation for each node could lead to extensive efforts because more features should be considered in the computation. Researchers have tried to introduce graph reduction to tackle this issue in the large traffic graph in recent years. We can also see that Guo et al. [41] introduced the traffic graph's spectral clustering(pooling) of the traffic graph on the Laplacian matrix of the adjacent matrix to reconstruct the graph structure. However, it showed that the reduced graph contributes to the model performance when applied in the short-term prediction task, while long-term prediction did not receive the improvement. However, Wu et al. [109] pointed out that the partial information of the graph will be lost if the graph reduction is introduced. Therefore, whether and when to investigate the performance on the reduced graph could be the future research.

8.1.2 Dependency trade-off

Even though the models for traffic flow forecasting emphasize the importance of the spatial correlations and temporal correlations, it is still an open question on how to balance the contribution of the temporal correlations and spatial correlations in the final prediction [69]. Partially because the existing methods applied to the traffic flow prediction have received relatively satisfying results without considering the spatial or temporal contribution weights; in other words, does it mean that the importance of temporal dependency should outweigh that of spatial dependency, or in reverse? The typical architecture of the traffic flow prediction is to separately consider the spatial and temporal features, whether sequential or parallel. However, balancing the importance of the extracted temporal and spatial information and utilizing it to the fullest must be considered.

8.1.3 Multi-source Spatial and Temporal correlations

Most previous works focused on extracting the spatio-temporal dependency over a single source. For example, researchers investigated the secret relationship between spatial and temporal perspectives from traffic flow data. However, Fang et al. [33] discovered that different data sources from the same/similar traffic areas existed the similar distributions in both temporal and spatial dimensions, which guide future research in exploring the hidden traffic relationship from the target area. They verified that with the traffic flow graph of taxi and bike data in NYC and captured the spatio-temporal dependency in a specific NYC area. It implied that the trajectory data from different sources could be combined to extract the hidden complex spatial-temporal dependencies for traffic flow prediction. Yao et al. showed another way to the different sources of the data [115]. They provided the graph's spatial and semantic views and combined the extracted features for the forecast. Therefore, future research should emphasize exploring the shared relationship of multi-sources data from the same target area since it could potentially contribute to forecasting.

8.1.4 Representation Learning

Inspired by Transformer, we can see the ability to express context information of the model by position embedding. Guo et al. [43] adapted the spatial position embedded method and temporal position embedded method to induce the order information in spatial dependency and temporal dependency, which resulted in a more accurate prediction. Hence, applying several appropriate learning representations on the raw data, which better represent different dependencies of the datasets, can potentially contribute to the model performance. Except for the models for the spatial and temporal dependency representation we mentioned in this paper, Zhang et al., [126] revealed that the Graph Embedding Technology such as DeepWalk [80], Node2Vec [40] can extract and mine the hidden spatial patterns because it can contain the details of spatial dependency behind the graph-structured data.

Zheng et al. consider spatial embedding and temporal embedding to provide static and dynamic representations among the traffic sensors and fuse them as spatio-temporal embedding to obtain the time-variant vertex representations for the combined static-dynamic spatio-temporal feature expression [137]. Wang et al. [101] proposed an adaptive GCN with multi-channel to inspire future work in extracting informative spatial features on traffic graphs. It extracted the spatio-temporal information by adopting two convolution operators from the given feature and topology information. Therefore, informative representation learning is still required to contribute to mining the hidden Spatial-temporal patterns and provide a comprehensive understanding of the overall behavior of the data.

8.2 Conclusion

In this paper, we believe the rigorous understanding of the task can contribute to the appropriate design of an efficient deep-learning-based solution to the traffic problem. Hence, we analyzed the traffic flow forecasting problem from spatial and temporal perspectives at a fine-grained level, which could inspire future researchers to consider designing the model to simulate specific scenarios. At first, we showed the problem definition of traffic flow forecasting from a sole temporal perspective to a spatio-temporal perspective, with which the corresponding techniques for spatial and temporal feature extraction have also been provided, plus the architecture, such as seqwseq and transformer. Then, we categorized and summarized a new taxonomy of the fine-grained dependency in a sub-category of spatial and temporal features of traffic flow forecasting problems, and the methods for addressing the dependency had also been provided. Because of the spatial and temporal feature extraction for constructing spatio-temopral dependency, we provided the guidance on how to extract spatio-temporal when considering building the ST architecture - sequential modelling, parallel modelling, and fusion methods. Also, we gave a discussion on the focus on how current work pays attention to when addressing the spatio-temporal feature extraction-multi scale temporal information, such as weekly, daily, and recent temporal data; and spatial information, such as k-hops neighboring information, and solution focusing on fusing region-level graph information. We point out future directions and discuss the possibilities of that in traffic flow forecastings, such as the reconstruction of the given graph information, balancing employment of temporal and spatial information, utilization of multi-source data to capture the hidden shared spatiotemporal information in the target area, and informative representation learning for the traffic context information.

References

[1] Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. 33.

[2] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. Watch your step: Learning node embeddings via graph attention. *arXiv preprint arXiv:1710.09599*, 2017.

[3] Mohammad Asghari, Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, and Yaguang Li. Price-aware real-time ride-sharing at scale: an auction-based approach. In *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 1–10, 2016.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. A3t-gcn: attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information*, 10(7):485, 2021.

[6] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[7] Zhengwei Bai, Wei Shangguan, Baigen Cai, and Linguo Chai. Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic. In *2019 Chinese Control Conference (CCC)*, pages 8600–8605. IEEE, 2019.

[8] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[9] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[10] Azzedine Boukerche, Xin Fei, and Regina B Araujo. An optimal coverage-preserving scheme for wireless sensor networks based on local information exchange. *Computer Communications*, 30(14-15):2708–2720, 2007.

[11] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Localization systems for wireless sensor networks. *IEEE wireless Communications*, 14(6):6–12, 2007.

[12] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications*, 31(12):2838–2849, 2008.

[13] Azzedine Boukerche, Yanjie Tao, and Peng Sun. Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Computer networks*, 182:107484, 2020.

[14] Azzedine Boukerche and Carl Tropper. A static partitioning and mapping algorithm for conservative parallel simulations. In *Proceedings of the eighth workshop on Parallel and distributed simulation*, pages 164–172, 1994.

[15] Azzedine Boukerche and Jiahao Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, 181:107530, 2020.

[16] Alberto Bull, NU CEPAL, et al. *Traffic Congestion: The Problem and how to Deal with it.* ECLAC, 2003.

[17] Ling Cai, Krzysztof Janowicz, Gengchen Mai, Bo Yan, and Rui Zhu. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755, 2020.

[18] Clayson Celes, Fabricio A Silva, Azzedine Boukerche, Rossana Maria de Castro Andrade, and Antonio AF Loureiro. Improving vanet simulation with calibrated vehicular mobility traces. *IEEE Transactions on Mobile Computing*, 16(12):3376–3389, 2017.

[19] Cen Chen, Kenli Li, Sin G Teo, Xiaofeng Zou, Kang Wang, Jie Wang, and Zeng Zeng. Gated residual recurrent graph neural networks for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 485–492, 2019.

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

[21] Chen Chen, Bin Liu, Shaohua Wan, Peng Qiao, and Qingqi Pei. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system.

[22] Zhiyong Cui, Kristian Henrickson, Ruimin Ke, and Yinhai Wang. Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11):4883–4894, 2019.

[23] Xingyuan Dai, Rui Fu, Yilun Lin, Li Li, and Fei-Yue Wang. Deeptrend: A deep hierarchical neural network for traffic flow prediction. *arXiv preprint arXiv:1707.03213*, 2017.

[24] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.

[25] Zulong Diao, Xin Wang, Dafang Zhang, Yingru Liu, Kun Xie, and Shaoyao He. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 890–897, 2019.

[26] Loan NN Do, Neda Taherifar, and Hai L Vu. Survey of neural network-based models for short-term traffic state prediction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(1):e1285, 2019.

[27] Chun Jiao Dong, Chun Fu Shao, Cheng-Xiang Zhuge, and Meng Meng. Spatial and temporal characteristics for congested traffic on urban expressway. *Journal of Beijing Polytechnic University*, 38(8):128–132, 2012.

[28] Shengdong Du, Tianrui Li, Xun Gong, Yan Yang, and Shi Jinn Horng. Traffic flow forecasting based on hybrid deep learning framework. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 1–6. IEEE, 2017.

[29] Gilles Duranton and Matthew A Turner. The fundamental law of road congestion: Evidence from us cities. *American Economic Review*, 101(6):2616–52, 2011.

[30] Mengyuan Fang, Luliang Tang, Xue Yang, Yang Chen, Chaokui Li, and Qingquan Li. Ftpg: A fine-grained traffic prediction method with graph attention network using big trace data. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[31] Shen Fang, Véronique Prinet, Jianlong Chang, Michael Werman, Chunxia Zhang, Shiming Xiang, and Chunhong Pan. Ms-net: Multi-source spatio-temporal network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[32] Shen Fang, Qi Zhang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Gstnet: Global spatial-temporal network for traffic flow prediction. In *IJCAI*, pages 2286–2293, 2019.

[33] Ziquan Fang, Lu Pan, Lu Chen, Yuntao Du, and Yunjun Gao. Mdtp: A multi-source deep traffic prediction framework over spatio-temporal trajectory data. *Proc. VLDB Endow.*, 14(8):1289–1297, 2021.

[34] Dong Feng, Zhongcheng Wu, Jun Zhang, and Ziheng Wu. Dynamic global-local spatial-temporal network for traffic speed prediction. *IEEE Access*, 8:209296–209307, 2020.

[35] Xinxin Feng, Xianyao Ling, Haifeng Zheng, Zhonghui Chen, and Yiwen Xu. Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2001–2013, 2018.

[36] Chang Gang, Wang Shouhui, and Xiao Xiaobo. Review of spatio-temporal models for short-term traffic forecasting. In *2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, pages 8–12. IEEE, 2016.

[37] Liang Ge, Siyu Li, Yaqian Wang, Feng Chang, and Kunyan Wu. Global spatial-temporal graph convolutional network for urban traffic speed prediction. *Applied Sciences*, 10(4):1509, 2020.

[38] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Message passing neural networks. In *Machine Learning Meets Quantum Physics*, pages 199–214. Springer, 2020.

[39] John F Gilmore and Naohiko Abe. Neural network models for traffic control and congestion prediction. *Journal of Intelligent Transportation Systems*, 2(3):231–252, 1995.

[40] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[41] Kan Guo, Yongli Hu, Yanfeng Sun, Sean Qian, Junbin Gao, and Baocai Yin. Hierarchical graph convolution networks for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 151–159, 2021.

[42] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.

[43] Shengnan Guo, Youfang Lin, Huaiyu Wan, Xiucheng Li, and Gao Cong. Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[44] Hadi Habibzadeh, Tolga Soyata, Burak Kantarci, Azzedine Boukerche, and Cem Kaptan. Sensing, communication and security planes: A new challenge for a smart city system design. *Computer Networks*, 144:163–200, 2018.

[45] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[46] Yong Han, Shukang Wang, Yibin Ren, Cheng Wang, Peng Gao, and Ge Chen. 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, 2019.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[48] Salah Hihi and Yoshua Bengio. Hierarchical recurrent neural networks for long-term dependencies. *Advances in neural information processing systems*, 8, 1995.

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

[50] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kung Fu, Zheng Wang, Xiaohu Qie, and Jieping Ye. Heteta: Heterogeneous information network embedding for estimating time of arrival. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2444–2454, 2020.

[51] Wei-Chiang Hong, Yucheng Dong, Feifeng Zheng, and Shih Yung Wei. Hybrid evolutionary algorithms in a svr traffic flow forecasting model. *Applied Mathematics and Computation*, 217(15):6733–6747, 2011.

[52] Hosagrahar V Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Communications of the ACM*, 57(7):86–94, 2014.

[53] JQ James, Christos Markos, and Shiyao Zhang. Long-term urban traffic speed prediction with deep learning on graphs. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[54] Manrui Jiang, Wei Chen, and Xiang Li. S-gcn-gru-nn: A novel hybrid model by combining a spatiotemporal graph convolutional network and a gated recurrent units neural network for short-term traffic speed forecasting. *Journal of Data, Information and Management*, 3(1):1–20, 2021.

[55] Shan Jiang, Wentian Chen, Zhiheng Li, and Haiyang Yu. Short-term demand prediction method for online car-hailing services based on a least squares support vector machine. *IEEE Access*, 7:11882–11891, 2019.

[56] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey, 2021.

[57] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[58] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[59] Kyungeun Lee, Moonjung Eo, Euna Jung, Yoonjin Yoon, and Wonjong Rhee. Short-term traffic prediction with deep neural networks: A survey. *arXiv preprint arXiv:2009.00712*, 2020.

[60] Duo Li and Joan Lasenby. Spatiotemporal attention-based graph convolution network for segment-level traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[61] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, 32:5243–5253, 2019.

[62] Yaguang Li, Dingxiong Deng, Ugur Demiryurek, Cyrus Shahabi, and Siva Ravada. Towards fast and accurate solutions to vehicle routing in a large-scale and dynamic environment. In *International Symposium on Spatial and Temporal Databases*, pages 119–136. Springer, 2015.

[63] Yaguang Li and Cyrus Shahabi. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *Sigspatial Special*, 10(1):3–9, 2018.

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

[65] Yang Li and José MF Moura. Forecaster: A graph transformer for forecasting spatial and time-dependent data. *arXiv preprint arXiv:1909.04019*, 2019.

[66] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[67] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[68] Aoyu Liu and Yaying Zhang. Spatial-temporal interactive dynamic graph convolution network for traffic forecasting. *arXiv preprint arXiv:2205.08689*, 2022.

[69] Zhining Liu, Dawei Zhou, and Jingrui He. Towards explainable representation of time-evolving graphs via spatial-temporal graph attention networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2137–2140, 2019.

[70] Bin Lu, Xiaoying Gan, Haiming Jin, Luoyi Fu, and Haisong Zhang. Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1025–1034, 2020.

[71] Wenjuan Luo, Han Zhang, Xiaodi Yang, Lin Bo, Xiaoqing Yang, Zang Li, Xiaohu Qie, and Jieping Ye. Dynamic heterogeneous graph neural network for real-time event prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3213–3223, 2020.

[72] Xiaolei Ma, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044, 2015.

[73] Abdelhamid Mammeri, Azzedine Boukerche, and Zongzhi Tang. A real-time lane marking localization, tracking and communication system. *Computer Communications*, 73:132–143, 2016.

[74] K-R Müller, Alexander J Smola, Gunnar Rätsch, Bernhard Schölkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector machines. In *International Conference on Artificial Neural Networks*, pages 999–1004. Springer, 1997.

[75] Horacio ABF Oliveira, Azzedine Boukerche, Eduardo F Nakamura, and Antonio AF Loureiro. Localization in time and space for wireless sensor networks: An efficient and lightweight algorithm. *Performance Evaluation*, 66(3-5):209–222, 2009.

[76] R Keith Oswald, William T Scherer, and Brian L Smith. Traffic flow forecasting using approximate nearest neighbor nonparametric regression. Technical report, 2000.

[77] Jun Ouyang, F Lu, and X Liu. Short-term urban traffic forecasting based on multi-kernel svm model. *Image Graph*, 15:1688–1695, 2010.

[78] Chengsheng Pan, Jiang Zhu, Zhixiang Kong, Huaifeng Shi, and Wensheng Yang. Dc-stgcn: Dual-channel based graph convolutional networks for network traffic forecasting. *Electronics*, 10(9):1014, 2021.

[79] Zhao Peng, Wang Jian-Wei, Sun Mao-Peng, and Zhou Ya-Xin. Vehicle scheduling for mountainous expressway traffic emergency. *China Journal of Highway and Transport*, 2018.

[80] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[81] Nicholas G Polson and Vadim O Sokolov. Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, 79:1–17, 2017.

[82] Amit Roy, Kashob Kumar Roy, Amin Ahsan Ali, M Ashraful Amin, and AKM Rahman. Unified spatio-temporal modeling for traffic forecasting using graph neural network. *arXiv preprint arXiv:2104.12518*, 2021.

[83] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[84] Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*, 2021.

[85] Qi Shi and Mohamed Abdel-Aty. Big data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. *Transportation Research Part C: Emerging Technologies*, 58:380–394, 2015.

[86] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal net-

work data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 914–921, 2020.

[87] Bin Sun, Duan Zhao, Xinguo Shi, and Yongxin He. Modeling global spatial–temporal graph attention network for traffic prediction. *IEEE Access*, 9:8581–8594, 2021.

[88] Junkai Sun, Junbo Zhang, Qiaofei Li, Xiuwen Yi, Yuxuan Liang, and Yu Zheng. Predicting citywide crowd flows in irregular regions using multi-view graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2348–2359, 2020.

[89] Cong Tang, Jingru Sun, Yichuang Sun, Mu Peng, and Nianfei Gan. A general traffic flow prediction approach based on spatial-temporal graph attention. *IEEE Access*, 8:153731–153741, 2020.

[90] Jinjun Tang, Jian Liang, Fang Liu, Jingjing Hao, and Yinhai Wang. Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network. *Transportation Research Part C: Emerging Technologies*, 124:102951, 2021.

[91] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Choudhury, and A. K. Qin. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020.

[92] Chenyu Tian and Wai Kin Chan. Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intelligent Transport Systems*, 15(4):549–561, 2021.

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

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

[95] Leandro Aparecido Villas, Azzedine Boukerche, Guilherme Maia, Richard Werner Pazzi, and Antonio AF Loureiro. Drive: An efficient and robust data dissemination protocol for highway and urban vehicular ad hoc networks. *Computer Networks*, 75:381–394, 2014.

[96] Eleni I Vlahogianni, Matthew G Karlaftis, and John C Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

[97] Dongjie Wang, Yan Yang, and Shangming Ning. Deepstcl: A deep spatio-temporal convlstm for travel demand prediction. In *2018 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2018.

[98] Fucheng Wang, Jiajie Xu, Chengfei Liu, Rui Zhou, and Pengpeng Zhao. On prediction of traffic flows in smart cities: a multitask deep learning based approach. *World Wide Web*, 24(3):805–823, 2021.

[99] Jingcheng Wang, Yong Zhang, Yun Wei, Yongli Hu, Xinglin Piao, and Baocai Yin. Metro passenger flow prediction via dynamic hypergraph convolution networks. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[100] Senzhang Wang, Jiannong Cao, and Philip Yu. Deep learning for spatio-temporal data mining: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[101] Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery & data mining*, pages 1243–1253, 2020.

[102] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, pages 1082–1092, 2020.

[103] Xinglei Wang, Xuefeng Guan, Jun Cao, Na Zhang, and Huayi Wu. 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*, 119:102763, 2020.

[104] Yue Wang, Xiang Lian, and Lei Chen. Efficient simrank tracking in dynamic graphs. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 545–556. IEEE, 2018.

[105] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2496–2505, 2018.

[106] Billy M Williams. Multivariate vehicular traffic flow prediction: evaluation of arimax modeling. *Transportation Research Record*, 1776(1):194–200, 2001.

[107] Tianlong Wu, Feng Chen, and Yun Wan. Graph attention lstm network: A new model for traffic flow forecasting. In *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, pages 241–245. IEEE, 2018.

[108] Xuesong Wu, Jie Fang, Zhijia Liu, and Xiongwei Wu. Multistep traffic speed prediction from spatial–temporal dependencies using graph neural networks. *Journal of Transportation Engineering, Part A: Systems*, 147(12):04021082, 2021.

[109] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[110] Zhipu Xie, Weifeng Lv, Shangfo Huang, Zhilong Lu, Bowen Du, and Runhe Huang. Sequential graph neural network for urban road traffic speed prediction. *IEEE Access*, 8:63349–63358, 2019.

[111] Chenming Yang, Zhiheng Zhou, Hui Wen, and Liang Zhou. Mstnn: A graph learning based method for the origin-destination traffic prediction. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[112] Lijin Yang, Qing Yang, Yonghua Li, and Yuqing Feng. K-nearest neighbor model based short-term traffic flow prediction method. In *2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)*, pages 27–30. IEEE, 2019.

[113] Xu Yang, Qiang Zhu, Peihao Li, Pengpeng Chen, and Qiang Niu. Fine-grained predicting urban crowd flows with adaptive spatio-temporal graph convolutional network. *Neurocomputing*, 446:95–105, 2021.

[114] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5668–5675, 2019.

[115] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[116] Jiexia Ye, Juanjuan Zhao, Kejiang Ye, and Chengzhong Xu. How to build a graph-based deep learning architecture in traffic domain: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[117] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. Multi-stage attention spatial-temporal graph networks for traffic prediction. *Neurocomputing*, 428:42–53, 2021.

[118] Maram Bani Younes and Azzedine Boukerche. Intelligent traffic light controlling algorithms using vehicular networks. *IEEE transactions on vehicular technology*, 65(8):5887–5899, 2015.

[119] Maram Bani Younes and Azzedine Boukerche. A performance evaluation of an efficient traffic congestion detection protocol (ecode) for intelligent transportation systems. *Ad Hoc Networks*, 24:317–336, 2015.

[120] Maram Bani Younes and Azzedine Boukerche. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wireless Networks*, 24:2451–2463, 2018.

[121] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[122] Chenhan Zhang, JQ James, and Yi Liu. Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Access*, 7:166246–166256, 2019.

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

[124] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[125] Ke Zhang, Fang He, Zhengchao Zhang, Xi Lin, and Meng Li. Graph attention temporal convolutional network for traffic speed forecasting on road networks. *Transportmetrica B: transport dynamics*, 9(1):153–171, 2021.

[126] Shaokun Zhang, Yao Guo, Peize Zhao, Chuanpan Zheng, and Xiangqun Chen. A graph-based temporal attention framework for multi-sensor traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[127] Shuai Zhang, Yong Chen, and Wenyu Zhang. Spatiotemporal fuzzy-graph convolutional network model with dynamic feature encoding for traffic forecasting. *Knowledge-Based Systems*, 231:107403, 2021.

[128] Tianpu Zhang, Weilong Ding, Tao Chen, Zhe Wang, and Jun Chen. A graph convolutional method for traffic flow prediction in highway network. *Wireless Communications and Mobile Computing*, 2021, 2021.

[129] Weibin Zhang, Yinghao Yu, Yong Qi, Feng Shu, and Yinhai Wang. Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning. *Transportmetrica A: Transport Science*, 15(2):1688–1711, 2019.

[130] Xiyue Zhang, Chao Huang, Yong Xu, and Lianghao Xia. Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1853–1862, 2020.

[131] Zhengchao Zhang, Meng Li, Xi Lin, Yinhai Wang, and Fang He. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation research part C: emerging technologies*, 105:297–322, 2019.

[132] Zhenxia Zhang, Azzedine Boukerche, and Richard Pazzi. A novel multi-hop clustering scheme for vehicular ad-hoc networks. In *Proceedings of the 9th ACM international symposium on Mobility management and wireless access*, pages 19–26, 2011.

[133] Zikai Zhang, Yidong Li, Haifeng Song, and Hairong Dong. Multiple dynamic graph based traffic speed prediction method. *Neurocomputing*, 461:109–117, 2021.

[134] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3848–3858, 2019.

[135] Wentian Zhao, Yanyun Gao, Tingxiang Ji, Xili Wan, Feng Ye, and Guangwei Bai. Deep temporal convolutional networks for short-term traffic flow forecasting. *IEEE Access*, 7:114496–114507, 2019.

[136] Zheng Zhao, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.

[137] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1234–1241, 2020.

[138] Zibin Zheng, Yatao Yang, Jiahao Liu, Hong-Ning Dai, and Yan Zhang. Deep and embedded learning approach for traffic flow prediction in urban informatics. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3927–3939, 2019.

[139] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[140] Qiang Zhou, Jing-Jing Gu, Chao Ling, Wen-Bo Li, Yi Zhuang, and Jian Wang. Exploiting multiple correlations among urban regions for crowd flow prediction. *Journal of Computer Science and Technology*, 35:338–352, 2020.

[141] Jiawei Zhu, Qiongjie Wang, Chao Tao, Hanhan Deng, Ling Zhao, and Haifeng Li. Ast-gcn: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access*, 9:35973–35983, 2021.

[142] Li Zhu, Fei Richard Yu, Yige Wang, Bin Ning, and Tao Tang. Big data analytics in intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):383–398, 2019.

[143] Hans-Georg Zimmermann, Christoph Tietz, and Ralph Grothmann. Forecasting with recurrent neural networks: 12 tricks. In *Neural Networks: Tricks of the Trade*, pages 687–707. Springer, 2012.