# DYNAMIC LEARNING BY GRAPH CONVOLUTIONAL NETWORK FOR TRAFFIC PREDICTION

**ATKIA AKILA KARIM**
**Exam Roll: 2323201**
**Session: 2022-23**
**Registration Number: 2017-515-180**

A Thesis
Submitted to the Master of Science in Software Engineering Program Office
of the Institute of Information Technology, University of Dhaka
in Partial Fulfillment of the
Requirements for the Degree

**MASTER OF SCIENCE IN SOFTWARE ENGINEERING**

Institute of Information Technology
University of Dhaka
DHAKA, BANGLADESH

DYNAMIC LEARNING BY GRAPH CONVOLUTIONAL NETWORK FOR TRAFFIC PREDICTION

ATKIA AKILA KARIM

Approved:

| *Signature* | *Date* |
| --- | --- |
| | 01.08.2024 |
| Supervisor: Dr. Naushin Nower | |

To *my parents*,
for always believing in me

# Abstract

Traffic prediction is crucial for Intelligent Traffic Systems (ITS), traffic control, and traffic management systems. Complex spatial and temporal interactions of traffic networks make traffic prediction tasks challenging. Many cities around the world experience severe traffic congestion. In recent decades, there has been a shift in prediction techniques owing to advancements in big data and computational tools. Deep learning-based models have been popular in traffic prediction research due to their capability to handle large traffic data. Among these methods, Graph Convolutional Network (GCN) has attracted researchers' attention for its graph structure that can represent traffic networks better. However, traditional GCN have limitations due to their use of static adjacency matrix and parameter sharing, which hinder their ability to capture dynamic spatial patterns of traffic networks. Moreover, deep learning-based models are highly sensitive to noise and missing data due to sensor malfunctions, communication errors, and many more. While various missing data imputation techniques exist, they often apply preprocessing before prediction which creates an extra processing step. It causes bias in prediction results and requires extra computational resources. In addition, the spatio-temporal nature of traffic makes missing data handling more challenging.

To overcome this, a robust traffic prediction model is proposed which represents the traffic road network as a dynamic graph and uses a probabilistic adjacency matrix to identify the dynamic impacts of adjacent roads on the target road in GCN. In addition, to find the similarity among the nodes, node-specific learning

is employed in GCN rather than sharing parameters like traditional GCN. This node-specific learning helps the proposed model to learn detailed characteristics of road networks. For temporal feature extractions, a Gated Recurrent Unit (GRU) is used to capture the local trend of traffic flow and an attention mechanism is to capture the global trend of traffic flow. The performance of the proposed model is compared with baseline models using two real-world datasets. Experimental results show that the proposed model is effective in predicting both short and long-term traffic flow. Moreover, the proposed model performs well under various noise and missing data. The experiment proves that the proposed robust traffic prediction model can inherently handle missing data and noise in traffic networks.

# Acknowledgments

"*All praises are due to Allah*"

Firstly, I express my deep gratitude to Allah for providing me with the opportunity and the strength to continue my research effectively.

I wish to extend my heartfelt thanks to my supervisor, Dr. Naushin Nower, Institute of Information Technology, University of Dhaka, for her unwavering guidance, support, and invaluable feedback throughout my research. Her expertise, patience, and encouragement have been crucial in shaping my work and helping me navigate challenges.

I am also thankful to the faculty and thesis committee members of the Institute of Information Technology, University of Dhaka, for their insightful feedback, which significantly improved my thesis.

I am appreciative of my classmates for their support. Finally, I am deeply indebted to my family, who have always been there for me whenever I needed them.

# List of Publications

1. [*Published*] Atkia Akila Karim and Naushin Nower Long-Term Traffic Prediction Based on Stacked GCN Model" in *Knowledge Engineering and Data Science, 2023.*

2. [*Published* ] Atkia Akila Karim and Naushin Nower "Robust Traffic Prediction Using Probabilistic Spatio-Temporal Graph Convolutional Network"in *International Conference on Engineering Applications of Neural Networks (pp. 259-273). Cham: Springer Nature Switzerland, 2024.*

3. [*Published*] Atkia Akila Karim and Naushin Nower "Probabilistic Spatio-temporal Graph Convolutional Network for Traffic Forecasting" in *Applied Intelligence Journal, Springer, 2024.*

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Traffic prediction focuses on forecasting future traffic patterns in specific road networks using historical and current traffic data [1]. In megacities, severe traffic congestion poses significant challenges, adversely affecting socioeconomic development [2]. Accurate traffic flow prediction plays a crucial role in urban traffic planning, traffic management, and traffic control by aiming to reduce congestion. It is crucial for the effective functioning of intelligent transportation systems (ITS). ITS relies on accurate predictions to facilitate real-time traffic management decisions, thereby enhancing the overall traffic flow and reducing delays.

Moreover, accurate traffic flow predictions are indispensable for recommending time-saving routes to drivers, contributing to more efficient and less stressful commutes. The primary causes of traffic congestion include increasing population, urbanization, traffic mismanagement, and inadequate infrastructure [3]. The economic costs of traffic congestion are rising globally. For example, INRIX, (a major transportation analytics provider), estimates that the typical commuter in the top 1000 cities in the world will spend 99 hours stuck in traffic congestion in the year 2020 [4]. As a consequence of this, an estimated economic cost of 1,036 dollars per commuter was incurred as a result of wasted time as well as fuel [4]. In Dhaka city, traffic congestion costs five million working hours daily and results in an annual loss of 200-550 billion takas [5]. According to a World Bank report, the average driving speed in Dhaka has decreased from 21 km/hr to 5 km/hr over the last decade, and this decline is expected to continue. One promising way to mitigate traffic congestion is to accurately predict traffic flow since it helps users to make better route planning and city planners to effectively manage traffic throughout the city. Effective traffic prediction can lead to better utilization of existing infrastructure, delaying the need for costly expansions.

However, traffic prediction is a challenging task because traffic data has complex spatiotemporal dependence, compassing both spatial and temporal information regarding traffic patterns and behavior. Spatial data provides essential context for comprehending traffic flow within a specific area, while temporal data aids in identifying recurring traffic patterns, such as rush hours or variations in traffic volume between weekdays and weekends. Several deep learning-based methods have been proposed in recent years to predict traffic since these methods have the ability to process vast data sets and extract traffic features through multiple layers [6].

Among these methods, the Graph Convolutional Network (GCN) has gained attention from researchers for its ability to capture the spatial properties of traffic networks through its graph structure [7]. Additionally, recurrent neural network (RNN) [8] and their variants such as Gated Recurrent Unit (GRU) [9] and Long Short-Term Memory (LSTM) [10] are employed to capture the temporal features of traffic networks. Recent studies have explored combinations of GCN with GRU [11, 12, 13] or GCN with LSTM [14, 15, 16] to simultaneously capture spatial-temporal features.

However, traditional GCN-based models consider road networks as static graphs and employ a static adjacency matrix, which makes GCN unable to capture complete information about spatial dependency [17, 13]. The static adjacency matrix implies equal influence from all upstream nodes on target downstream nodes, which does not reflect real-world scenarios [15]. The impact of adjacent nodes changes over time. Moreover, GCN-based models share parameters among nodes which is only capable of identifying common patterns among adjacent nodes of traffic series [18]. It produces suboptimal solutions. Traffic series exhibit diverse patterns—ranging from similar to dissimilar patterns due to distinct attributes across all nodes in the traffic network. Nonadjacent nodes can exhibit similar patterns as the target node, this can happen because of the presence of POIs such as markets, hospitals, etc.

Moreover, deep learning models are heavily reliant on traffic data, which often suffers from issues like overfitting, noise, and data shortages [19]. Consequently, to address real-world scenarios, traffic prediction models must not only be accurate but also robust enough to handle data imperfections. In existing literature, missing values and noise are either ignored or addressed before the training process using imputation techniques. However, these methods require a two-step processing approach and substantial datasets and computational resources for imputation and model training [20]. Hence there is a need for a traffic prediction model that can accurately capture the spatiotemporal dependencies and is robust enough to handle noise and missing data. In this thesis, a robust traffic prediction model is proposed considering the limitations of the existing approaches.

In this chapter, the motivation behind this work and challenges have been discussed. This chapter also carries the research questions and contributions regarding the proposed traffic prediction model. A section on how this thesis has been organized is mentioned at the end of this chapter.

## 1.1 Motivation

Traffic congestion leads to substantial economic losses, primarily due to wasted time and fuel. Accurate traffic prediction models are essential for optimizing route planning and traffic management, thereby mitigating these losses and enhancing overall economic efficiency. Studies suggest that 50 to 70 percent of these losses can be reduced through proper traffic management and actions. Specifically, in Dhaka, the capital of Bangladesh, reducing traffic congestion could contribute to massive economic growth, potentially increasing the country's GDP by 35% [21]. Beyond financial implications, traffic congestion adversely affects health and the environment. Effective traffic prediction models can reduce fuel consumption and emissions, promoting environmental sustainability. Accurate predictions provide drivers with real-time information on optimal routes, thus reducing travel time and stress, and improving the daily commuting experience and overall quality of life. Significant advancements have been made in traffic prediction research. Recently, GCN has been employed to capture the spatial characteristics of traffic data, while variants of RNN have been utilized to capture temporal characteristics. However, traditional GCN, which rely on a predefined adjacency matrix, are limited in their ability to capture dynamic spatial dependencies in traffic data. The influence of traffic from adjacent roads on the target road can change over time so a static adjacency matrix cannot effectively respond to spatio-temporal changes [17]. Additionally, GCN employs shared parameters among nodes, limiting the model's ability to accurately capture the unique characteristics of individual roads, as it only considers similar patterns of adjacent roads [18]. However, non-adjacent roads can also exhibit similar patterns. Existing models also tend to be sensitive to noise and abnormalities in data. This research aims to develop a traffic flow prediction model that addresses the limitations of traditional GCN and demonstrates robustness in handling missing and noisy data.

## 1.2 Research Questions

Despite the development of numerous GCN-based models for traffic prediction, significant limitations remain in their ability to capture the dynamic spatial dependencies inherent in traffic networks. Additionally, existing models struggle to handle missing data, reducing their effectiveness in real-world scenarios where noise and missing data are prevalent. These challenges lead to the following research question.

1. How to learn dynamic spatial patterns by Graph Convolutional Network (GCN) for better traffic prediction?

   More specifically, this research question will be answered by the following sub-questions.

   (a) How can dynamic spatial patterns be captured by GCN?

   To answer this question, a probabilistic adjacency matrix is proposed which uses Bayesian inference to calculate the congestion propagation probability in traffic networks. Unlike traditional GCN that use a predefined adjacency matrix, the proposed probabilistic adjacency matrix can capture the dynamic spatial pattern of the traffic network. While parameter sharing reduces the number of parameters, it can result in biased outcomes as traffic networks show diverse patterns. So, rather than shared parameters, node-specific learning is incorporated into the GCN to enable the model to learn the unique characteristics of each road in the traffic network.

   (b) How can the traffic prediction model be robust enough to handle noise and missing data inherently?

   To answer this question, a traffic prediction model is proposed that can inherently handle abnormalities in data. Using Bayesian inference in the adjacency matrix makes the model handle missing data naturally and flexibly. In Bayesian inference, missing data is interpreted as unknown parameters with a probability distribution that is inferred using the available information and the prior beliefs. So the traffic prediction model does not need a preprocess or imputation technique.

## 1.3 Contribution and Achievement

To solve the limitations of existing work, a robust probabilistic spatiotemporal Graph Convolutional Network (R-PST-GCN) model is proposed that can predict in a noisy environment. In the R-PST-GCN, a modified GCN is used that can capture the dynamic spatial characteristics in traffic data. Rather than using a static adjacency matrix like GCN, a probabilistic adjacency matrix is used that helps the model learn the dynamic spatial characteristics of traffic networks. Bayesian inference is used to learn the dynamic traffic propagation patterns in the road network from historical data. It helps the model handle the missing data. Node-specific learning is also used to learn the unique characteristics of the roads. Those modifications in GCN make the proposed model capable of handling noisy data without any imputation or preprocessing of data. To evaluate the performance of our model we experimented with two real-world datasets. We used one city road dataset and one highway dataset as both have different patterns. Experimental results show that our model outperforms baseline methods while handling noisy or missing data. The performance of our model is steady throughout different noisy data. An ablation study is also carried out to see which part of the model contributes to handling missing data.

## 1.4 Organization of the Thesis

This section provides an overview of the subsequent chapters. The chapters are organized in the following way:

- **Chapter 2 Literature Review:** The chapter starts with presenting complex dependencies of the traffic network that need to be considered for accurate traffic prediction. Next, existing traffic prediction models are classified. Based on this classification, the methodology, strengths and weaknesses of existing traffic prediction approaches are discussed.

- **Chapter 3 Robust Traffic Prediction using Probabilistic spatiotemporal Graph Convolutional Network:** This chapter presents the methodology for a robust traffic prediction model called Robust Probabilistic Spatiotemporal Graph Convolutional Network (R-PST-GCN).

- **Chapter 4 Simulation and Result Analysis:** In this chapter, details on the experimental setup, implementation, and result analysis are provided. The proposed approach is compared with state-of-the-art methods to demonstrate its effectiveness. To check the robustness of the model perturbation test has been done.

- **Chapter 5 Conclusion:** In this chapter, the whole thesis is summarized as well as the future work is presented.

# Chapter 2

# Literature Review

Traffic flow prediction is a critical component of intelligent transportation systems, aiming to enhance traffic management, reduce congestion, and improve overall urban mobility. Understanding and predicting traffic flow is inherently complex due to the various factors influencing it. This chapter delves into the essential characteristics of traffic networks that impact prediction accuracy. It begins by exploring spatial dependencies, where traffic conditions at one location are influenced by those in adjacent areas. Following this, the chapter examines temporal dependencies, highlighting how traffic patterns evolve over different times of the day, days of the week, and seasons. The influence of sudden incidents on traffic flow is also discussed, emphasizing the need for real-time data and rapid response mechanisms for effective traffic management.

Subsequent sections review the extensive body of research dedicated to traffic prediction. Various methodologies, including parametric, non-parametric, and deep learning techniques, are studied for their efficacy in capturing the complex spatial and temporal dependencies of traffic networks. The chapter also discusses the robustness of traffic prediction models, particularly their ability to handle missing data and noise, which are common challenges in real-world traffic data.

By integrating insights from these diverse studies, the chapter aims to provide a comprehensive understanding of the current state of traffic flow prediction research, identifying key advancements and persistent challenges in the field.

## 2.1 Background Study

Traffic flow exhibits a complex pattern influenced by various factors. It shows both spatial and temporal dependencies, as traffic conditions vary by location and change over time. Additionally, sudden events such as rain or accidents significantly impact traffic flow in subsequent periods. This section aims to detail these dependencies of the traffic network that are crucial to capture for improving traffic prediction accuracy.

### 2.1.1 Spatial Dependencies

Spatial dependencies in a traffic network refer to how traffic conditions at one location are influenced by conditions at surrounding locations. Traffic flow is not isolated to a single point but is interconnected with adjacent roads, intersections, and regions. In Figure 2.1 from [22], we can see that the traffic speed patterns of Road 1 and Road 2 are similar. However, the trend of traffic speed on Road 3 is quite different. The reason is Road 1 and Road 2 are adjacent roads and have the same direction, whereas Road 3 is in the opposite direction. Understanding these spatial relationships between different road segments is crucial for anticipating

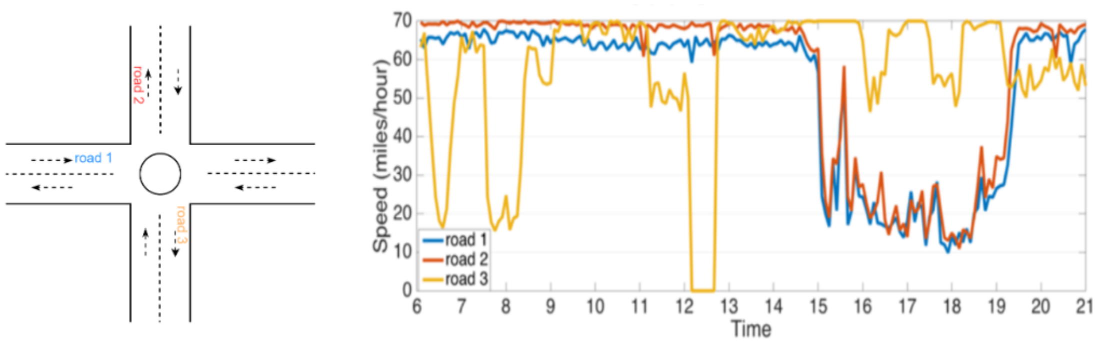
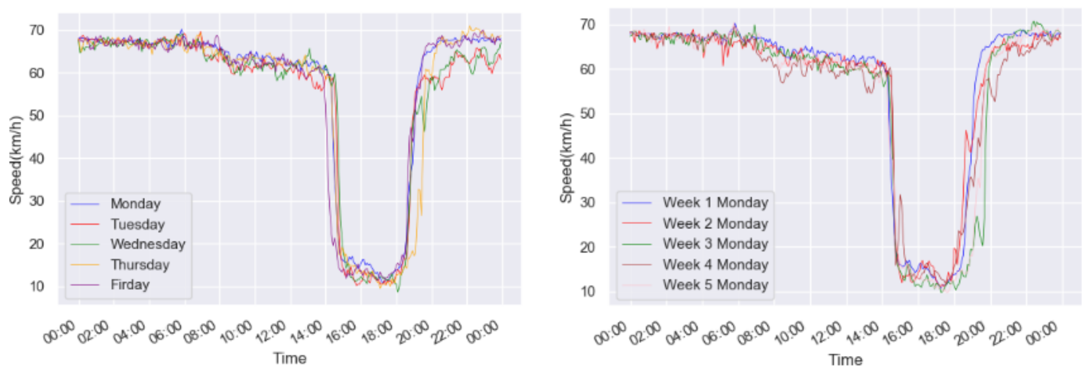



Figure 2.1: Spatial Characteristics of Traffic Network

how traffic conditions in one area may be affected by its adjacent areas. Capturing accurate spatial information by predictive models improves the accuracy and utility of traffic predictions, thereby enhancing traffic management and reducing congestion.

### 2.1.2 Temporal Dependencies

Temporal dependencies in a traffic network refer to how traffic conditions at a given time are influenced by previous traffic states. Traffic flow is inherently dynamic, exhibiting patterns that change over different times of the day, days of the week, and seasons of the year. For instance, rush hour traffic in the morning and evening shows regular peaks due to commuting patterns, while weekends may have different flow characteristics due to recreational activities. In Figure 2.2 from [23], we can see the daily and weekly periodicity of traffic speed for the PeMSD7 dataset.




(**a**) daily periodicity of traffic speed (**b**) weekly periodicity of traffic speed

Figure 2.2: Temporal Characteristics of Traffic Network

Understanding temporal dependencies is crucial for accurate traffic prediction. These dependencies account for the temporal correlations in traffic data, such as how a traffic jam at a time can cause ripple effects in subsequent periods. Additionally, temporal factors include routine variations like day-night cycles and weekly schedules. By capturing temporal patterns and variations by predictive

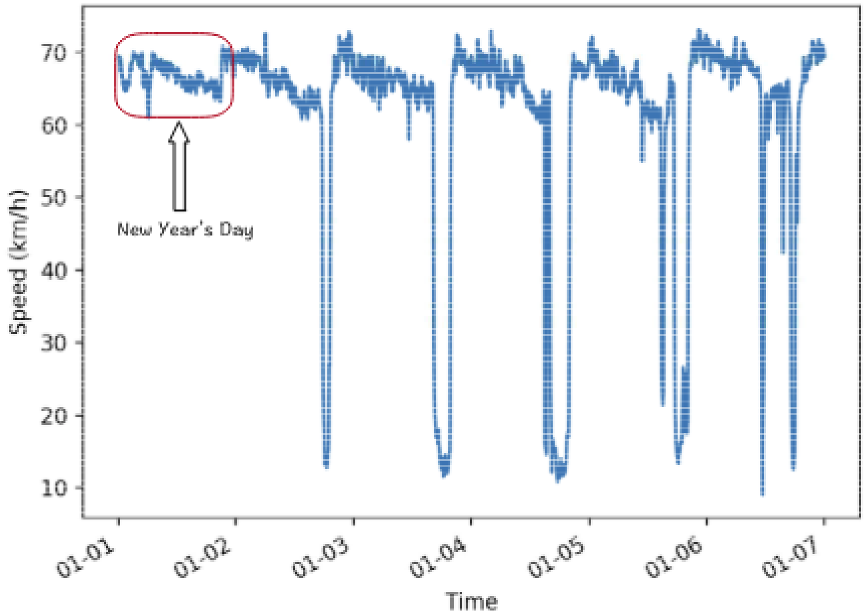



Figure 2.3: Effects of New Year's Day versus Normal Day in San Francisco Bay Area

models, it is possible to have more reliable traffic predictions.

### 2.1.3 Sudden Incident and External Factors

Sudden incidents in a traffic network, such as accidents, adverse weather conditions, or unexpected road closures, can significantly disrupt normal traffic flow. These events introduce abrupt changes that are difficult to predict but have substantial impacts on traffic conditions. For instance, an accident can create immediate congestion, affecting not only the incident location but also causing delays and rerouting in adjacent areas. Traffic speed can also be influenced by external factors like weather conditions, holidays, and other special events. As depicted in Figure 2.3 from [23], there is a distinct disparity in traffic speed between holidays and normal days. We can see that the traffic pattern of holidays doesn't match with normal days. Furthermore, Figure 2.4 from [23], illustrates that traffic speed on heavily rainy days is markedly lower compared to sunny days.

The impact of sudden incidents highlights the importance of real-time data

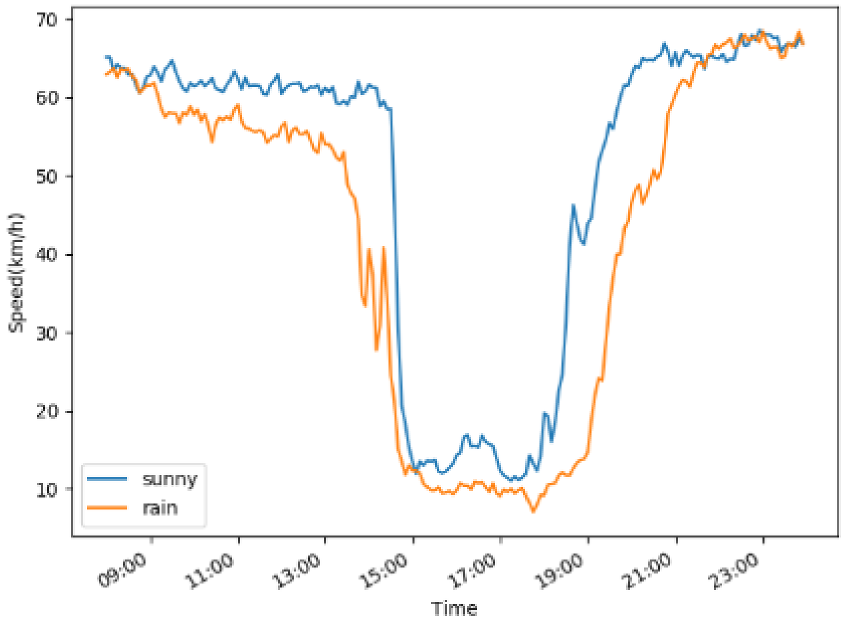



Figure 2.4: Effects of Sunny versus Rainy in San Francisco Bay Area

and rapid response mechanisms in traffic management. Predictive models must account for these anomalies to enhance their accuracy and robustness. This includes integrating data from various sources, such as traffic sensors, cameras, and weather forecasts, to quickly identify and respond to incidents.

## 2.2 Related Work

To capture the dependencies of the traffic network discussed in the previous section, extensive research has been conducted on traffic prediction. Numerous authors have explored various techniques to accurately forecast traffic flow, taking into account different aspects and related characteristics. Broadly, three major categories of traffic flow prediction models emerge (i) parametric techniques, (ii) non-parametric techniques, and (iii) deep learning techniques.

Parametric models derive parameters from original data analysis, employing predetermined regression functions for subsequent traffic predictions. Autoregressive Integrated Moving Average (ARIMA) and Kalman filter (KF) model are the most studied statistical techniques for predicting time series data. The Autoregressive Integrated Moving Average (ARIMA) model [24] and its variants [25] are widely adopted in traffic prediction. Another approach involves the Kalman filter (KF) model [26], predicting future traffic information based on the analysis of current and previous traffic conditions. However, these models exhibit limitations, relying on static system model assumptions, and neglecting the dynamic nature, non-linearities, and spatial features inherent in traffic data.

Addressing the limitations of parametric models, non-parametric approaches offer solutions by analyzing historical data without imposing assumptions. Common non-parametric models are Random Forest [27], Support Vector Machine [28], Fuzzy Logic [29], Bayesian Network [30], K-Nearest Neighbors [31], and neural networks. These models individually or in combination, perform well in handling spatiotemporal data. However, these models face challenges with large-scale traffic data.

### 2.2.1 Deep Learning Model

To overcome the limitation of non-parametric models, deep learning networks [6] have emerged. It can process vast data sets and extract traffic features through multiple layers. Initial deep learning models solely employ RNN [8], its variants, gated recurrent unit (GRU) [9], and long short-term memory (LSTM) [10] for traffic prediction, which can capture the temporal features only. However, the change in traffic data is influenced by its surrounding urban road network, making it impossible for these models to accurately predict the traffic condition on the road since they only capture the temporal aspects of traffic flow while ignoring its spatial features.

To solve this problem, it is essential to fully utilize spatial and temporal dependencies. Based on this, numerous studies have included CNN or GCN to capture spatial features with variants of RNN to capture temporal features.

#### 2.2.1.1 Convolutional Neural Network (CNN)-based Model

Many researchers used hybrid models to capture both spatial and temporal features. By structuring the road network as a grid, CNN processes inputs as a series of cells, with each cell denoting the vehicle count, speed, etc within a defined area at a specific time interval. CNN is used for extracting spatial features from traffic data.

For example, the CNN-LSTM Traffic Flow Prediction (CLTFP) model [32], leverages CNN and LSTM to address the spatial-temporal characteristics of traffic flow data. CNN is employed to extract spatial features while LSTM captures short-term temporal variations in the data. The model then utilizes a linear regression layer, with an added L1 norm constraint. Evaluated on freeway corridor data, CLTFP demonstrates its success in the comprehensive utilization of spatial distribution and short-term temporal variability.

Another model [33] is proposed to improve the CNN-LSTM model for capturing both spatial and temporal features of GPS trajectory data. This model comprises two heterogeneous sub-DNNs: one focused on learning spatial features using a CNN without pooling layers, while the other incorporates a combination of CNN and LSTM to capture local dependencies and preserve temporal information over periods. By merging GPS trajectory data processed through separate channels in the output layer, the hybrid DNN produces inflow and outflow grid data. To address the complexity of the CNN-LSTM architecture and mitigate training challenges, the model integrates a greedy policy approach. Experimental results demonstrate that the proposed hybrid CNN-LSTM model significantly outperforms existing methods in terms of prediction accuracy. Furthermore, the incorporation of the greedy policy enhances training efficiency and reduces computation time. However, this model can not handle the sudden incident in the traffic network.

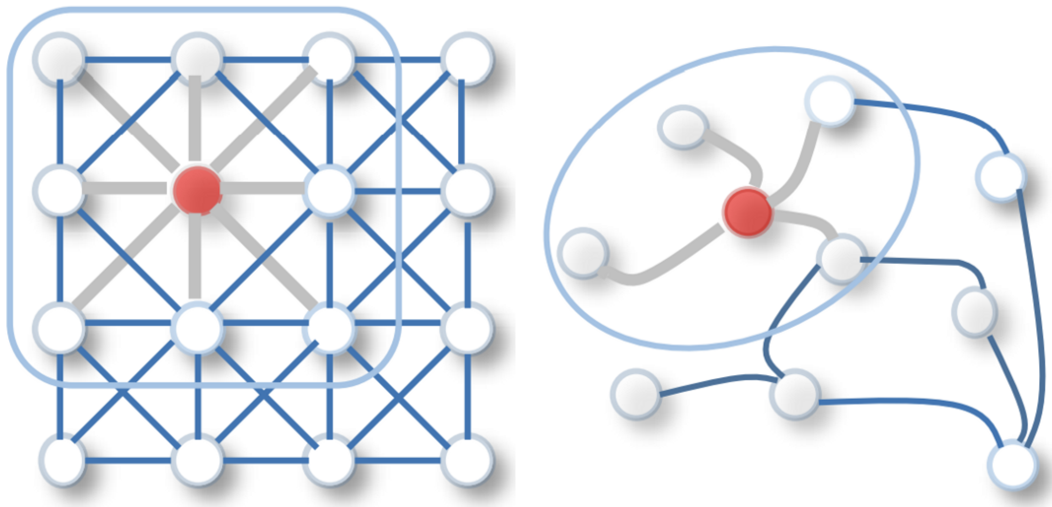
Attentive Traffic Flow Machine (AFTM) [34] employs two ConvLSTM networks to forecast short-term and long-term traffic flow patterns. Utilizing trajectory data from bikes and taxis in Beijing and New York City, AFTM constructs grid maps based on latitude and longitude coordinates. For short-term prediction, the model first extracts spatial and temporal features using the ResNet architecture, followed by individual AFTM modules to capture sequential and periodic traffic flow properties. These features are then integrated using a fusion model to generate subsequent time sequence predictions. AFTM incorporates an attention mechanism to detect sudden incidents, enhancing the model's ability to focus on critical features. While AFTM demonstrates remarkable performance, it requires substantial computational resources and memory. In the long-term prediction scenario, AFTM employs a similar short-term prediction setup but augments it with four additional ConvLSTM layers to predict traffic flow for subsequent time intervals.

However, CNN-based models encounter limitations in handling complex topological structures inherent in road networks. CNN cannot adequately characterize the spatial features of the road network because it was designed for Euclidean space, which includes images, regular grids, and so on [35]. In Figure 2.5a from [36], we can see that CNN applied convolution operation in Euclidean space treating traffic networks as grid data. However, the traffic network cannot be presented as regular grid data as connections between roads are not fixed.

#### 2.2.1.2 Graph Convolutional Network (GCN)-based Model

GCN represents the traffic network as a graph and each road segment/sensor as a node. They have gained significant attention in recent years due to their effectiveness in irregular data domains. CNN is well-suited for regular grid-like data such as images where convolutional filters slide over the input space. In contrast, graphs lack a fixed grid structure, and nodes can have varying degrees of connectivity. GCN extends the concept of CNN to work with graph-structured data. In Figure

2.5 from [36], we can see that GCN adapts the convolution operation to work with these irregular structures by aggregating information from a node's neighborhood, similar to how convolutional filters aggregate information from neighboring pixels in images.



(a) Convolutional Neural Network (CNN) (b) Graph Convolutional Network (GCN)

Figure 2.5: Comparison between CNN and GCN

At each layer of the GCN, nodes aggregate information from their neighboring nodes, through a weighted sum operation. GCN utilizes an static adjacency matrix and feature matrix to extract the information from the road network. As a result, GCN has recently been widely used in traffic prediction over CNN.

For example, T-gcn model [11] integrates GCN with GRU to capture the topological structure of urban road networks with the dynamic variation of traffic information over time. This hybrid architecture allows the T-GCN model to capture the complex interactions within traffic data. They used traffic speed for traffic prediction. LSTM and GRU are popular for extracting temporal dependencies. They used GRU as it has a relatively simple structure, faster training ability and fewer parameters than LSTM. In the loss function, they used L2 regularization to avoid overfitting. Linear interpolation is used to fill the missing data. When evaluated on two real-world traffic datasets and compared with the HA model, the ARIMA model, the SVR model, the GCN model, and the GRU model, the T-GCN model achieves the best prediction results under different prediction horizons. The drawbacks of this model are, that its graph modeling only considers the geographical distance information of the road segments, and does not consider the dynamic changes of the transportation network in the long-term prediction. The graph based on distance modeling is always static, while traffic propagation is not static.

Following T-GCN, the A3TGCN model is proposed which uses an attention mechanism on top of the TGCN model. This method combines the power of GRU to capture short-term time series trends and GCN to model spatial dependencies based on road network topology. Furthermore, it incorporates an attention mechanism to dynamically adjust the significance of different time points, thereby aggregating global temporal information. The model is tested in the urban road network-based traffic prediction task using two real datasets, namely, SZ-taxi and Los-loop. However, the method can only capture shared patterns among all traffic series and rely on a pre-defined adjacency matrix.

Attribute-augmented spatiotemporal graph convolutional network model (AST-GCN) [37] incorporated external factors such as weather, and points of interest with the spatial and temporal features. It acknowledges that accurate traffic predictions require considering not only historical traffic data but also external factors like weather conditions and the distribution of points of interest (POI). While spatiotemporal models integrating graph convolutional networks and recurrent neural networks have gained attention, they often neglect these external factors. To bridge this gap, AST-GCN treats external factors as dynamic and static attributes, devising an attribute-augmented unit to encode and integrate them seamlessly into the spatiotemporal graph convolution model. They used GCN to capture the spatial dependencies and GRU to capture the temporal dependencies. They used L2 regularization as a loss function to avoid overfitting.

Linear interpolation is used to fill the missing data.

The limitation of the model is that it uses a static adjacency matrix that cannot capture the dynamic spatial dependencies. Moreover, they used traditional GCN that shares parameters between the nodes, which hampers node-specific pattern learning.

To expand the receptive field of graph convolution AST-GCN-LSTM model [14] is proposed. It employs Local Spectral Graph Convolution (LSGC) to extract spatial correlation features from K-order local neighborhoods of road segment nodes, thereby considering high-order neighborhood information. This approach goes beyond conventional methods that only focus on first-order neighbor nodes. Additionally, the model incorporates an external attribute enhancement unit to extract and integrate external factors such as weather conditions, point of interest data, and time, recognizing their impact on traffic flow. Used LSTM to capture the temporal feature. However, LSTM is time consuming than GRU. The limitations of the model are that it uses a static adjacency matrix that cannot capture the dynamic spatial dependencies. Moreover, they used traditional GCN that shares parameters between the nodes, which hampers node-specific pattern learning.

Though commonly an undirected graph is used to model spatial dependency of traffic network, DCRNN model [22] proposed bidirectional random walks in directed graphs for spatial dependencies. It integrates diffusion convolution into the architecture of the GRU, resulting in the Diffusion Convolutional Gated Recurrent Unit (DCGRU). For temporal modeling, DCRNN employs a Sequence-to-sequence (Seq2Seq) architecture, with both the encoder and decoder composed of DCGRU. During training, the historical time series data is fed into the encoder, and its final states initialize the decoder, which then generates predictions based on previous ground truth observations. However, capturing sudden incidents is hard for this model. Moreover, the encoder and decoder structure for long-term prediction makes the model's computational cost higher.

To handle sudden incidents Attention-based Spatial-Temporal Graph Convolutional Network (ASTGCN) model [38] is proposed. It comprises three distinct components, each dedicated to modeling different temporal properties of traffic data, including recent, daily-periodic, and weekly-periodic dependencies. These components incorporate a spatial-temporal attention mechanism to effectively capture dynamic correlations, along with spatial-temporal convolutions that utilize graph convolutions for spatial patterns and standard convolutions for temporal features. The outputs from these components are weighted and fused to generate final prediction results. However, using the spatial-temporal attention mechanism separately for the three components makes the model computationally costly.

All of the aforementioned models utilize a static adjacency matrix. This static adjacency matrix implies that all adjacent nodes have the same influence on the target node, which does not accurately reflect real-world scenarios. GCN-based models employing static adjacency matrices often fail to capture the dynamic spatial features of traffic. In reality, the impact of adjacent roads varies over time and context. Therefore, models relying on static adjacency matrices cannot adequately represent the fluctuating influences within a traffic network. GCN often yields suboptimal predictions due to their inherent smoothing effects, which can obscure critical traffic dynamics and peaks [11].

To overcome this limitation, the paper [15] introduces a novel approach called Location Graph Convolutional Network (Location-GCN). Location-GCN resolves this issue by introducing a learnable matrix into the GCN mechanism, where the absolute values of this matrix represent the distinct influence levels among different nodes, allowing the model to dynamically learn spatial patterns during training. Additionally, the paper incorporates Long Short-Term Memory (LSTM) and Trigonometric function encoding to capture both short-term and long-term periodic information in the traffic data. The proposed model is then compared with baseline models and evaluated using real-world traffic flow datasets.

However, it is extremely difficult for a random trainable matrix to converge to an adjacency matrix that shows the graph structure correctly. It can not learn how the influence weights among different nodes change over time.

USTGCN model [39] utilizes a spatio-temporal adjacency matrix to facilitate the aggregation of spatio-temporal traffic features from nodes across previous timestamps and spatial traffic features from current timestamp neighbor nodes. It aggregates the last one week's timestamps in the adjacency matrix. By leveraging spectral graph convolution, USTGCN performs spatiotemporal aggregation in a unified manner, incorporating a temporal weight parameter to learn the importance of traffic features from different timestamps. Stacking multiple layers of spatiotemporal convolutions enables USTGCN to capture information from the K-hop neighborhood for each target node at different timestamps, enhancing the model's receptive field. However, the model is computationally expensive because of the aggregation of one-week traffic data.

Following that, DCGCN model [13] particularly focuses on fine-grained lane-level traffic prediction. It leverages GCN with a data-driven adjacency matrix to capture spatial features, treating different lanes as individual nodes. The proposed data-driven adjacency matrix combines distance-based relationships with dynamic lane correlation, allowing it to adapt to spatio-temporal changes. Temporal features are extracted using a GRU. A gating mechanism is employed to fuse spatial and temporal features, resulting in the final spatiotemporal features for lane-level traffic prediction.

As this model uses shared parameters, it is unable to learn how the influence weights among different nodes change over time. Moreover, the model used GCN and GRU separately, which limits the model's ability to capture interconnected spatial and temporal dependencies.

Another dynamic adjacency matrix-based model GCN is AGCRN model [18] which has two modules: the Node Adaptive Parameter Learning (NAPL) module, designed to capture node-specific patterns, and the Data Adaptive Graph Generation (DAGG) module, which automatically infers inter-dependencies among different traffic series. The pre-defined graph cannot contain complete information about spatial dependency and is not directly related to prediction tasks, which may result in considerable biases. To mitigate this problem, they used an adaptive adjacent matrix. These modules are integrated into an Adaptive Graph Convolutional Recurrent Network (AGCRN), which leverages recurrent networks and the two modules to automatically capture spatial and temporal correlations within traffic series data.

However in the hash learning environment, such as when there is a large range of data and insignificant node features, the above methods that generate the adjacency matrix by embedding matrix may face failure [16]. In addition, the above-mentioned methods did not consider missing and noisy data which is very frequent in real-life environments.

### 2.2.2 Robustness of Traffic Prediction Model

The performance of traffic forecasts can be significantly impeded by the missing data. Although deep learning models demonstrate impressive performance, they heavily depend on traffic data, which frequently encounters challenges such as overfitting, noise, and insufficient data [19]. Therefore, to effectively tackle real-world situations, traffic prediction models need to not only achieve accuracy but also exhibit robustness in handling data imperfections. In current research, missing values and noise are typically either disregarded or dealt with before training through imputation techniques. Many techniques for data imputation have been developed to address the problem.

A Probabilistic principal component analysis-based model [40], has suggested minimum data imputation. It uses a data optimization algorithm, P-MDIO, which leverages spatio-temporal information to minimize the number of missing points requiring imputation across the transportation network. Additionally, the classic STARIMA model is adapted to handle incomplete traffic data, enabling real-time prediction alongside data imputation. Experimental results using real-time traffic data from Taipei demonstrate the effectiveness of combining the P-MDIO algorithm with the STARIMA model, under missing data conditions.

Bayesian Gaussian CANDECOMP/PARAFAC (BGCP) tensor decomposition model [41] extends Bayesian matrix factorization to higher-order cases, allowing it to impute missing entries in spatiotemporal traffic data. By placing conjugate priors on hyper-parameters, they derive posterior distributions and develop an efficient Markov chain Monte Carlo (MCMC) algorithm for model estimation. Their approach achieves stable performance across varying missing rates and non-random correlated missing conditions, providing uncertainty measures for missing values through multiple imputation. They organize spatiotemporal traffic data into matrix, third-order tensor, and fourth-order tensor representations, finding that the BGCP model performs consistently well with a third-order tensor structure. Experimental results on five synthetic datasets validate their approach's effectiveness in imputing missing data, particularly with the third-order tensor representation.

GMN model [19] infers traffic state step by step using the graph Markov process. By integrating spectral graph convolution operations, the SGMN enhances prediction performance and efficiency. Experimental evaluations demonstrate the effectiveness of both GMN and SGMN in predicting traffic states across various datasets, with SGMN exhibiting better performance, particularly in datasets with higher missing rates.

LSTM-M model [42] revises the traditional LSTM architecture to capture complex missing patterns in the data. In LSTM-M, influence factors are incorporated to model decay influence in memory, and a masking vector is introduced to simulate the residual between predicted and ground-truth values. This allows the model to effectively predict missing observations and learn prediction residuals via nonlinear functions within the LSTM unit.

One drawback of those methods is they need pre-processing procedures like imputation techniques as they are unable to handle missing data inherently. These actions may add biases and need computational resources [20].

## 2.3 Summary

Although deep learning-based traffic prediction models have gained attention in the traffic prediction field, they struggle to capture complex dynamic spatiotemporal dependencies. While CNN is commonly used for spatial dependencies, CNN can not fully capture the dynamic spatial dependencies of traffic networks as it treats traffic networks as grid data. However, in real life, traffic networks have quite complex structures. GCN is more fitted for capturing spatial dependencies as its graph structure can represent the road connection more accurately. Unfortunately, GCN faces limitations too. It requires a predefined adjacency matrix, that makes the graph fixed. It hinders the capability of the model to capture the dynamic spatial dependencies. So there is a need for a data-driven adjacency matrix that is not rigid and can capture the dynamic spatial dependencies in traffic networks. Another limitation of GCN is that it shares weight among its nodes. As sharing parameters can reduce parameter numbers, most models tend to use shared parameters. However, it produces sub-optimal results. Every road can have its distinct characteristics, so it is important to learn those characteristics to produce better prediction performance. Another limitation of deep learning models is that they are quite sensitive to noise and missing data. Traditional approaches for handling missing data involve data preprocessing before model training, which can introduce biases. So there is a need for a robust prediction model that can seamlessly handle missing data and capture dynamic spatiotemporal characteristics. Integration of such mechanisms into prediction processes can enhance accuracy and reliability while reducing reliance on pre-processing steps and additional computational resources. Further, dynamic adjacency matrices and node-specific learning in GCN can improve prediction capabilities.

# Chapter 3

# Robust Traffic Prediction using Probabilistic spatiotemporal Graph Convolutional Network

Traffic forecasting is inherently challenging because it involves intricate intra-dependencies at different times (temporal) and inter-dependencies across different locations (spatial), leading to complex spatiotemporal interactions. Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been commonly used to address spatial and temporal dependencies respectively. However, CNN was initially introduced for grid-shaped networks and it works on Euclidean distance thus it can not properly handle non-grid-shaped road networks. That's why recently Graph Convolutional Network (GCN) has been widely used for spatial feature extraction as its graph structure can represent the road structure more accurately.

However, traditional GCN suffers from two significant limitations. Firstly, it relies on a static adjacency matrix to represent connections between nodes, which can restrict its ability to learn the dynamic spatial properties of a road network [17, 13]. Secondly, GCN shares weights among its nodes, thereby impeding its capacity to learn the unique characteristics of each node [18]. These limitations are discussed in further detail below.

**Different Impact of Adjacent Node:** Figure 3.1 illustrates the varying influence of adjacent roads on target node 1. The adjacent nodes that have a significant influence on the target node are outlined in red. At time $t-1$, nodes 2, 4, and 5 have significant influence on the target node 1, whereas at time $t$, nodes 2, 4, and 3 are more impactful. This discrepancy demonstrates that while nodes may be adjacent to the target node, their influence can vary over time. Thus, the impact of adjacent nodes on a target node is not fixed and can differ significantly between nodes. However, the traditional GCN mechanism relies on a manually set adjacency matrix to represent influence among nodes, typically based on connectivity or distance. This results in a binary (0-1) adjacency matrix when using connectivity, implying equal influence from all upstream nodes, which does not reflect real-world scenarios and reduces prediction accuracy. Moreover, static adjacency matrices are fixed and cannot contain complete information about dynamic spatial dependency. In Fig. 3.1, we can see that the impact of adjacent nodes is changing over time. So if we assign the same weight for all adjacent nodes it will result in a suboptimal result. That's why it is important to capture the change in spatial features over time. However, GCN-based traffic forecasting models are overly simplistic and fail to capture complete spatial dependencies, leading to significant biases. Additionally, such approaches lack adaptability across different domains, limiting the effectiveness of current GCN-based models.

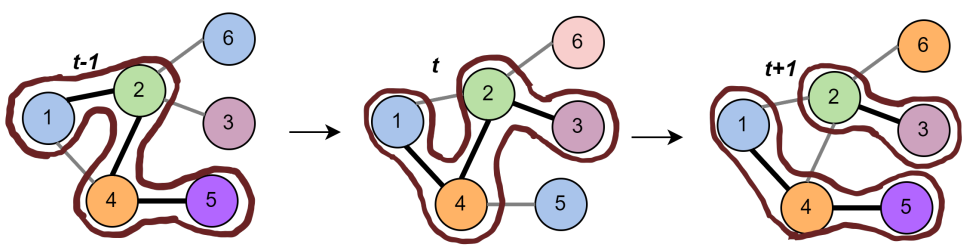



Figure 3.1: Impact of adjacent nodes over time

**Learning Node Features:** Recent deep-learning-based methods tend to be biased towards prominent and shared patterns among all traffic series. The shared parameter space in current methods hampers the accurate capture of fine-grained characteristics. Traffic series exhibit diverse patterns—ranging from similar to dissimilar and even contradictory—due to distinct attributes across various data sources. For example, nodes 1 and 6 at time $t-1$ in Fig. 3.1 are nonadjacent but can have similar traffic patterns. This can happen because of the presence of points of interest (POI) such as markets, hospitals, etc. On the contrary, the traffic patterns of two adjacent nodes may exhibit dissimilar patterns. This is because traffic series display not only close spatial correlations but also diverse patterns among different traffic series due to the dynamic nature of time series data and various influencing factors at each node [18]. Although sharing parameters is useful for learning the most prominent patterns among all nodes and significantly reducing the number of parameters, it proves suboptimal for traffic forecasting problems.

**Robustness of Model:** While deep learning models exhibit outstanding performance, they are heavily reliant on traffic data, which often suffers from issues like overfitting, noise, and data shortages [19]. Recent studies have demonstrated the vulnerability of neural networks, indicating that even slight data perturbations can significantly degrade the performance of GCN [43]. Models whose results are highly sensitive to minor changes in the dataset possess limited practical value. Implementing traffic prediction models for real-world applications requires maintaining accuracy not only with clean datasets but also when dealing with corrupted data. Developing a robust traffic prediction model remains a significant challenge. Robust prediction refers to the ability to resist noise and perturbations in the input data while predicting future traffic states [44]. In existing literature, missing values and noise are either ignored or addressed before the training process using imputation techniques. These methods require a two-step processing approach and substantial datasets and computational resources for imputation and model training. Therefore, developing an inherently robust traffic prediction model that can operate effectively without relying on complex imputation or preprocessing techniques in the presence of abnormal or missing traffic data remains a highly challenging yet crucial research problem.

To solve the problems mentioned above, in this study, a Robust Probabilistic Spatiotemporal Graph Convolutional Network (R-PST-GCN) model has been proposed that can predict in a noisy environment. A modified GCN has been used in the R-PST-GCN that solves the limitations of traditional GCN. Rather than using a static adjacency matrix in GCN, a probabilistic adjacency matrix has been used that is data-driven. It helps the model to learn the dynamic spatial characteristics of traffic networks. It also helps R-PST-GCN to learn the different influences of adjacent roads over time. Bayesian inference has been used to learn the dynamic traffic propagation patterns in the road network from historical data.

To avoid the biased result of shared parameters, node-specific learning has been used to learn the unique characteristics of the roads. These modifications in the GCN enable the R-PST-GCN to capture dynamic spatial dependencies effectively. Furthermore, they allow the R-PST-GCN model to handle missing data and noisy data without the need for imputation or preprocessing. In addressing the temporal aspect of the traffic network, GRU and an attention mechanism are incorporated. The attention mechanism proves beneficial for facilitating long-term prediction and managing sudden incidents within the R-PST-GCN. The contribution can be summed up as follows:

- A data-driven probabilistic adjacency matrix has been proposed using the Bayesian approach to tackle dynamic spatiotemporal interactions of traffic networks. The probabilistic adjacency matrix stores the propagation probability of traffic at that timestamp. Thus a dynamic probabilistic spatiotemporal graph (PST) has been created for GCN and GRU which can capture the dynamic spatiotemporal dependency.

- Probabilistic GCN model also uses node-specific learning to better capture patterns of every node of the traffic network.
- The proposed model can handle missing and noisy data inherently and can give a highly accurate prediction.

## 3.1 Robust Probabilistic Spatiotemporal Graph Convolutional Network (R-PST-GCN)

In this section, the proposed model, Robust Probabilistic Spatiotemporal Graph Convolutional Network (R-PST-GCN) for traffic prediction has been discussed. To capture the dynamic spatial dependencies, the traditional GCN has been modified. Rather than using a static adjacency matrix a probabilistic adjacency matrix has been used. Moreover, the unique characteristics of all nodes have been captured. For temporal dependencies, GRU and attention mechanisms are used. Fig. 3.2 shows the architecture of the R-PST-GCN. At first, the road network has been presented as a graph, where every road segment or sensor is a node. Then the traffic speed of every timestamp in every node has been stored in the traffic feature matrix. The traffic feature matrix has been used for calculating the probability matrix, $A^p$. Then the probabilistic adjacency matrix, $A^p$ and traffic feature matrix, $X^t$ are fed into GCN to extract spatial features. In the GCN, node-specific learning has been incorporated to better capture the node's unique features. The output of the GCN is then given to GRU to capture temporal features. For every timestamp, GCN and GRU are used separately so that the R-PST-GCN can capture spatiotemporal correlation. Finally, the attention mechanism has been utilized to handle sudden incidents. The R-PST-GCN has been further discussed in detail.

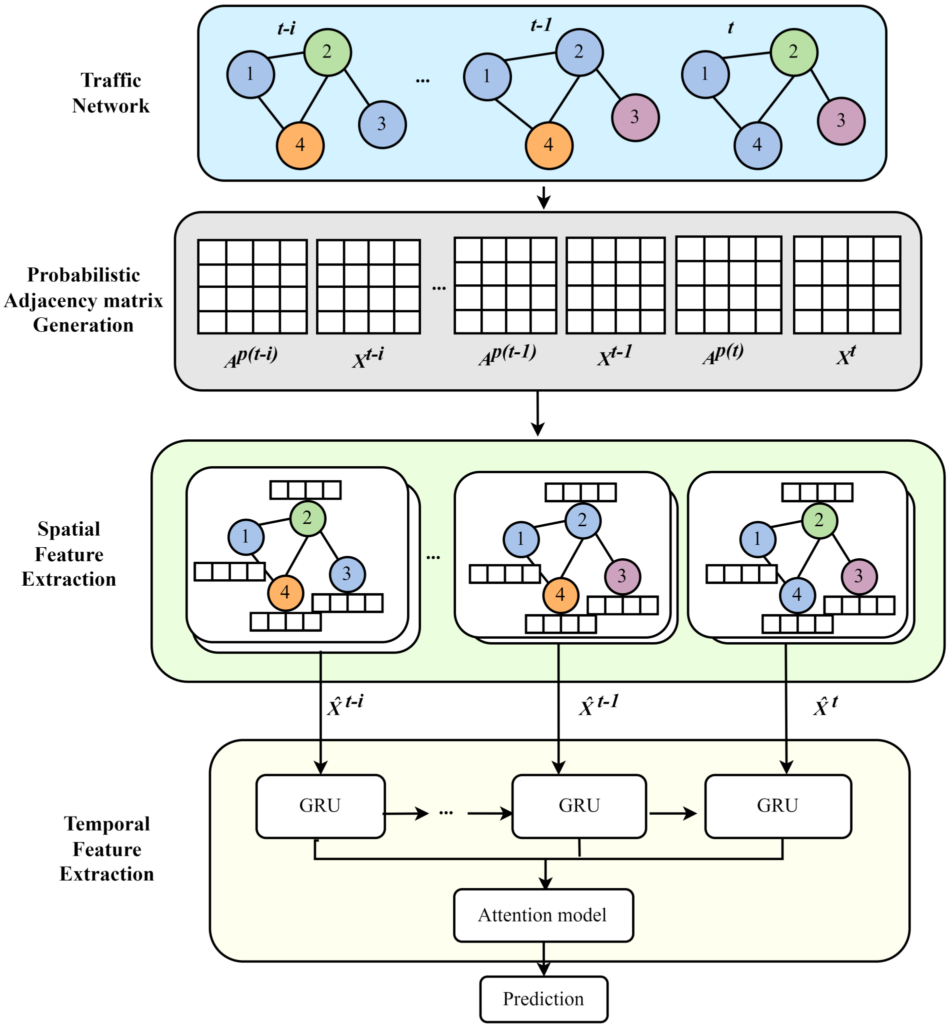



Figure 3.2: Architecture of R-PST-GCN

### 3.1.1 Spatial Feature Extraction

Traffic prediction task relies heavily on precise spatial feature extraction. GCN has been widely used for spatial feature extraction for its graph structure that can represent the traffic network and the connections between roads. That's why GCN has been employed for spatial feature extraction. At each layer of the GCN, nodes aggregate information from their neighboring nodes, through a weighted sum operation. This aggregated information is then transformed via a neural network layer to produce the node representations for the next layer. In Fig. 3.3, the architecture of traditional GCN has been presented. It stores traffic information from the road in a feature matrix and connections between the roads in an adjacency matrix. By stacking multiple such layers, GCN can capture complex

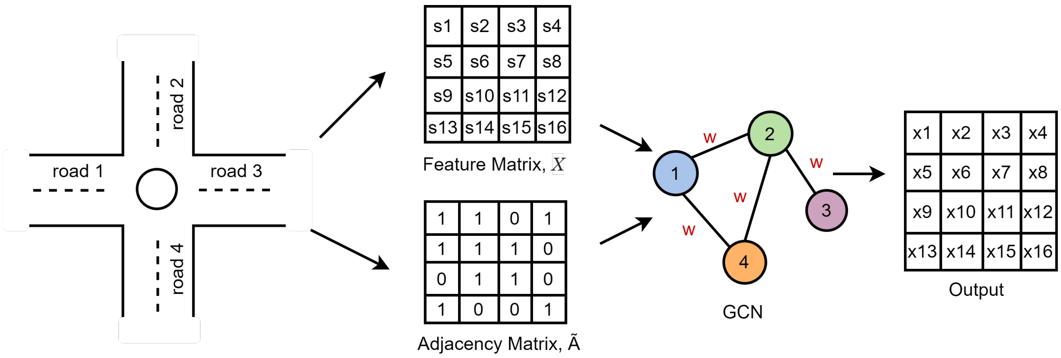



Figure 3.3: Architecture of Traditional Graph Convolutional Network

patterns and dependencies in the graph. Multilayered GCN can be represented as:

$$\tilde{X}^l = \sigma(\tilde{D}^{\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\tilde{X}^{l-1}W^{l-1} + b^{l-1}) \tag{3.1}$$

where $A$ is an adjacency matrix that represents the connection between nodes. $\tilde{A}$ is the summation of the Identity matrix, $I$ with the adjacency $A$. So that it indicates the output of a node in a hidden layer depends on itself and its neighbors. To maintain the scale of the output feature vector the adjacency matrix needs to be normalized. That's why diagonal matrix ,$\tilde{D}^{\frac{1}{2}}$ is multiplied with $\tilde{A}$. So instead of summing up itself with its neighbor, multiplying the sum with the inverse diagonal matrix, $\tilde{D}^{-\frac{1}{2}}$ sort of averages them. $W^{l-1}$ is shared weight among nodes of $l-1$ layer. $\tilde{X}^l$ indicates the output of applying GCN in the $l$ layer.

However, traditional GCN has limitations that make the model unable to capture dynamic spatial features. To address this, a probabilistic adjacency matrix and node-specific learning are integrated within the GCN framework.

#### 3.1.1.1 Probabilistic Adjacency Matrix

Traffic forecasting models that are based on traditional GCN have drawbacks because they rely on predefined adjacency matrices, $A$ for graph convolution operations. Existing methods typically compute these matrices based on distance or similarity measures. Using binary values in the adjacency matrix assumes equal influence from different upstream nodes on a downstream node, which is not representative of real-world scenarios [18]. Predefined graphs fail to capture dynamic spatial dependencies, introducing potential biases unrelated to prediction tasks. To address this, a probabilistic adjacency matrix integrating Bayesian inference has been proposed to enhance the model's ability to handle dynamic spatial interactions and provide more robust predictions.

To do that the probability of traffic propagation using Bayesian inference has been captured from one road to its adjacent road over time rather than utilizing a static adjacency matrix that defines simply the connection between roads. Congestion propagation in the network can be learned from the observed data. For calculating the probability of traffic flow propagation, we employ transitional probability with Bayesian inference. Using the Bayesian theorem, the posterior probability of a specific road segment, denoted as $P(i|X(t))$ has been computed based on observed traffic data $X(t)$ at time $t$. $P(j|i,t)$ represents the conditional probability of being on road segment $j$ at time $t$, given the traffic flow was initially on road segment $i$. By integrating conditional probabilities of being on specific road segments and transitioning between segments, the likelihood of traffic moving from one segment to another has been estimated, considering observed traffic data :

$$P(i \to j|X(t)) = P(j|i,t) \cdot P(i|X(t)) \tag{3.2}$$

Based on the $P(i \to j|X(t))$, the probabilistic adjacency matrix has been computed for every timestamp $t$ which can represent dynamic spatiotemporal inter-

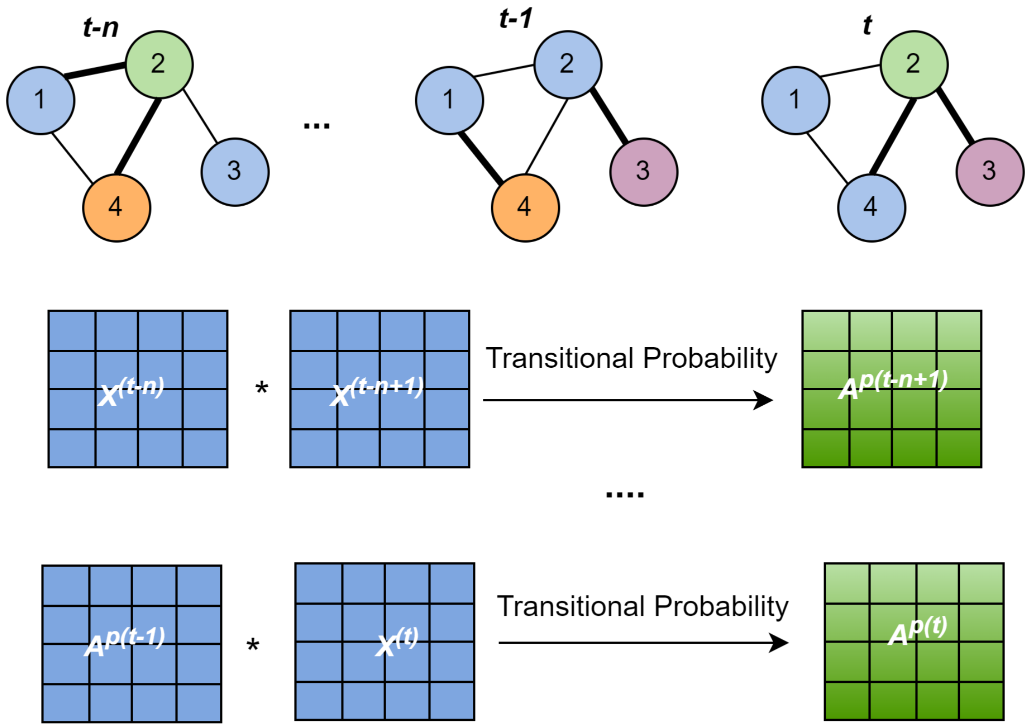



Figure 3.4: Probabilistic adjacency matrix generation process

action. In Fig. 3.4, a probabilistic adjacency matrix generation process has been presented. Using traffic features, $X(t)$ of previous timestamps, the probability of traffic congestion on the road in the next timestamp can be calculated. The probabilistic adjacency matrix can be denoted as, $A^P$. This Bayesian approach, yields a more precise estimation of the probability of traffic flow transitions, providing a refined, data-driven perspective on traffic transitions within the network. Moreover, using Bayesian inference in the adjacency matrix helps the model in handling missing data naturally and flexibly. In Bayesian inference, missing data has been interpreted as unknown parameters with a probability distribution that has been inferred using the available information and the prior beliefs. As a result, discard or imputation of the missing data is not needed, as it is in frequentist inference. Instead, Bayesian inference can handle the uncertainty and variability of the missing data, and generate plausible values for them based on the available information.

After using the probabilistic adjacency matrix, modified multilayered GCN can be presented as:

$$\tilde{X}^{l} = \sigma(\tilde{D}^{\frac{1}{2}}\tilde{A}^{P}\tilde{D}^{-\frac{1}{2}}\tilde{X}^{l-1}W^{l-1} + b^{l-1}) \tag{3.3}$$

#### 3.1.1.2 Node Specific Learning

Another issue with traditional GCN is that it uses shared weight among its nodes. Though it can reduce the number of parameters, it produces biased results. The assignment of the same weight in a layer to adjacent nodes implies equal influence on the target road for all neighboring nodes, a premise that does not align with real-world scenarios. Therefore, maintaining a distinct parameter space is necessary for every node to learn node-specific patterns; simply capturing shared patterns among all nodes is insufficient for effective traffic forecasting. To handle this problem, an embedding matrix and weight matrix are incorporated. Rather than using separate parameters for all nodes, which will result in over-fitting and time-consuming, an embedded matrix and weight matrix are used that will help in capturing node-specific characteristics. The goal of node embedding is to capture the structural and semantic properties of the nodes and their relationships within the graph in a lower-dimensional vector form. Nodes that are close or similar in the graph should have similar embedding. An embedded matrix, $E \in R^{N*e}$ where $N$ is the number of nodes and $e$ is the embedded dimension. A weight matrix, $\tilde{W} \in R^{e*l*f}$, where $l$ is the hidden layer dimension and $f$ is the feature length. Thus after incorporating the embedding matrix and weight matrix, the modified GCN formula is:

$$\hat{X}^{l} = \sigma(\tilde{D}^{\frac{1}{2}}\tilde{A}^{p}\tilde{D}^{-\frac{1}{2}}\hat{X}^{l-1}E\tilde{W}^{l-1} + Eb^{l-1}) \tag{3.4}$$

### 3.1.2 Temporal Feature Extraction

Another essential aspect of traffic data is the temporal feature, which is crucial for traffic prediction tasks. GRU and attention mechanism are employed for temporal feature extraction. GRU is good for short-term prediction however it doesn't perform much well for long-term prediction. RNN and their variants process data sequentially over time, making them well-suited for capturing short-term tendencies by remembering the most recent information [12]. However, their performance tends to decrease as the prediction horizon extends. Additionally, these models cannot differentiate the importance of various time points solely based on temporal proximity. That's why the attention mechanism has been used so that it can give importance to points that are important and capture the global trend of the traffic network.

#### 3.1.2.1 Gated Recurrent Network

As traffic data has a significant sequence structure, temporal aspects are critical for traffic prediction. To capture temporal dependencies, the RNN has been widely used. However, RNN faces a vanishing gradient problem. To overcome this, two variants of RNN are introduced: one is LSTM, and another is GRU. Both use a gating mechanism to solve the vanishing gradient problem. GRU has been used to capture the temporal dependencies as it has fewer gates, making it comparatively faster and simpler than LSTM. GRU has two gates, an update gate ($z_t$) and a reset gate ($r_t$). The update gate has been shown in equation 3.5, it takes the output from GCN, $\hat{X}$ and the previous hidden state $h_{t-1}$.

$$z_t = \sigma([\hat{X}, h_{t-1}]W_z + b_z) \tag{3.5}$$

This gate determines how much previous value needs to be considered. $W_z$ is the weight and $b_z$ is the bias of the update gate.

$$r_t = \sigma([\hat{X}, h_{t-1}]W_r + b_r) \tag{3.6}$$

The reset gate, $r_t$ decides how much previous information needs to be erased. This gate also takes the output from GCN, $\hat{X}$, and previous hidden state $h_{t-1}$. $W_r$ is the weight and $b_r$ is the bias of the reset gate.

$$\tilde{c}_t = \tanh(\hat{X}, r_t \odot h_{t-1}])W_c + b_c) \tag{3.7}$$

Current memory content, $\tilde{c}_t$ uses the reset value to determine how much information should be stored and takes output from GCN, $\hat{X}$. $W_c$ is the weight and $b_c$ is the bias of current memory content.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{c}_t \tag{3.8}$$

Finally, the current hidden state, $h_t$, is calculated using $z_t$ and $c_t$, and the previous hidden state, $h_{t-1}$. Because of the gated mechanism, GRU preserves the variation trends of prior traffic information when gathering traffic information at the current instant.

#### 3.1.2.2 Attention Mechanism

Attention mechanisms improve deep learning models by selectively focusing on important input elements, improving prediction accuracy. It also helps the R-PST-GCN in long-term prediction as it can capture the global trend. In previous subsection, GRU creates hidden states, $H = h_1, h_2, ..., h_i$ for the previous $i$ timestamp input sequence length. The weights of every hidden state, $\alpha_i$ are determined using a Softmax function where the weights and bias of the first and second layers, respectively, are denoted by $w_1$ and $b_1$ and $w_2$ and $b_2$

$$\alpha_i = softmax(w_2(w_1 * h_i + b_1) + b_2) \tag{3.9}$$

The weighted sum has been used to calculate the context vector, $c_t$ that includes global traffic variation information.

$$c_t = \sum_{i=1}^{n} \alpha_i * h_i \tag{3.10}$$

Finally, using the fully connected layer, forecasting results were given. It helps in capturing the global trend of traffic flow. Moreover, it helps the R-PST-GCN to handle sudden incidents in the traffic network.

In Algorithm 1, the R-PST-GCN's algorithm has been presented. In Line 2, the probabilistic adjacency matrix has been calculated using equation, 3.2. Then for every timestamp, GCN and GRU are used simultaneously to capture spatiotemporal dependencies. In line 4, the traffic timestamp and generated probabilistic adjacency matrix are given to GCN to capture the dynamic spatial feature. The spatial feature has been captured using equation 3.4. Then the output of GCN has been given to GRU from line 5 to line 8. The generated hidden states from GRU are given to the attention mechanism for calculating the score in lines 12-14. Then the score has been used for the prediction task in line 17 using a fully connected layer.

**Algorithm 1:** The Training phase of R-PST-GCN

**Input:** Feature matrix, $X_t, ..., X_{t-i}$

**for** $e^{th}$ *training iteration in total epoch,* $E$ **do**

$\hat{A}^p = TP(X^e_t, ..., X^e_{t-i})$;

**for** $n^{th}$ $X^e_{t-n}$ *in* $X^e_t, ..., X^e_{t-i}$, $n = 0, ..., i$ **do**

$\hat{X}^e_{t-n} = \sigma(\tilde{D}^{\frac{1}{2}} \tilde{A}^P \tilde{D}^{-\frac{1}{2}} \hat{X}^e_{t-n-1}\ EW_{t-n-1} + Eb_{t-n-1})$;

$z_{t-n} = \sigma([\hat{X}^e_{t-n}, h_{t-n-1}]W_z + b_z)$;

$r_{t-n} = \sigma([\hat{X}^e_{t-n}, h_{t-n-1}]W_r + b_r)$;

$\tilde{c}_{t-n} = \tanh(\hat{X}^e_{t-n}, r_{t-n} \odot h_{t-n-1}])W_c + b_c)$;

$h_{t-n} = z_{t-n} \odot h_{t-n-1} + (1 - z_{t-n}) \odot \tilde{c}_{t-n}$;

**end**

**return** $h_t, ....h_t - i$;

$c_t = []$;

**for** $n^{th}$ $h_{t-n}$ *in* $h_t, ....h_t - i$, $n = 0, ..., i$ **do**

$\alpha = softmax(w_2(w_1 * h_{t-n} + b_1) + b_2)$;

$c_t + = \alpha * h_{t-n}$;

**end**

**return** $c_t$;

$X^e_{t+T} = c_t W_f + b_f$;

Calculate parameters gradient by loss function and update parameters by Backpropagation algorithm.

**end**

## 3.2 Summary

To capture the dynamic spatiotemporal dependencies a robust traffic prediction model has been proposed. The R-PST-GCN model deviates from the conventional method by using a probabilistic adjacency matrix instead of a predefined one, which improves its capacity to manage uncertain incidents. Moreover it enhances the model's ability to handle noisy data without requiring extensive preprocessing or imputation. Additionally, node embedding integration helps in capturing the unique features of each node. These two crucial changes enhance the classical GCN and allow it to identify spatial patterns more accurately. In addition to spatial extraction, incorporating GRU allows R-PST-GCN to effectively capture local temporal patterns. An attention mechanism has been also included to capture the global trends of traffic networks. These enhancements make R-PST-GCN robust enough to handle noisy and missing data. Rather than resorting to laborious and sub-optimal methods to address dataset shortcomings, a common approach in many other studies, focused are given to the inherent resilience of the model itself. In the next chapter, the performance of R-PST-GCN has been demonstrated using two real-world datasets. An ablation study are carried out in next chapter to find out which part of model contribute in robustness of the model.

# Chapter 4

# Simulation and Result Analysis

This chapter presents the detailed simulation setup and result analysis for evaluating the performance of R-PST-GCN model, in robust traffic prediction. To assess the model's performance, several metrics are employed: mean absolute error (MAE), root mean square error (RMSE), accuracy, coefficient of determination ($R^2$), and variance score (var). To demonstrate the prediction performance of the proposed method, experiments on two distinct datasets are conducted: one representing city road traffic and the other representing highway traffic. The effectiveness of the model was evaluated by comparing it with several recent forecasting models. Additionally, the model's performance is investigated in scenarios involving missing data and its robustness against noisy data. Complexity and computational cost is calculated for the model. An ablation study is also carried out to see the which part of the proposed model help in handling missing data. The results and subsequent analysis provide valuable insights regarding the capabilities and performance of R-PST-GCN model.

## 4.1 Simulation Setup

To validate the improvement of the proposed method several experiments are performed. The input data for the simulation was normalized so that the training

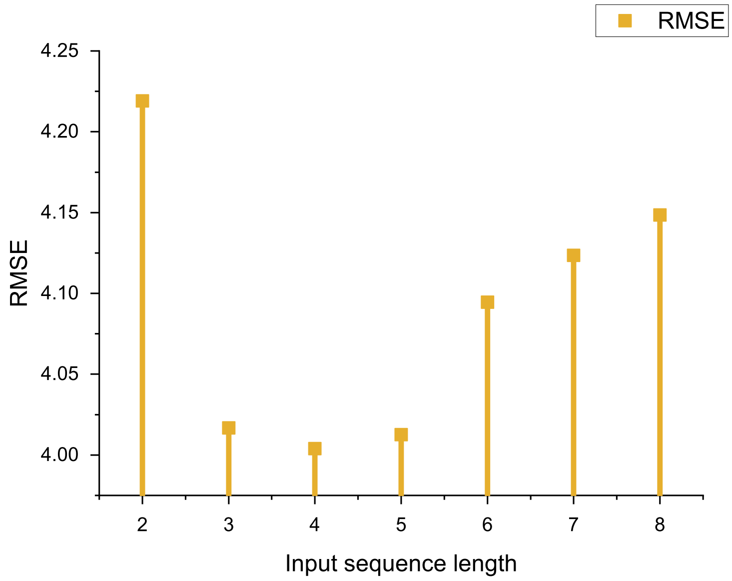



Figure 4.1: The performance of R-PST-GCN as the input sequence length changes.

process could be more stable. Furthermore, the training set comprised 80% of the data, while the testing set comprised the remaining 20%. We have got the adjacency matrix and feature matrix from the dataset, and from this, the probabilistic matrix is calculated. For the prediction task, the model requires a fixed amount of historical data as the input sequence length. Based on this sequence length, the model generates predictions for the future time horizon. As illustrated in Fig. 4.1, utilizing a sequence length of four previous timestamps for the prediction task on the SZ-taxi dataset, the R-PST-GCN model achieves the lowest RMSE. That means the previous 1-hour data helps in better prediction tasks. That's why the previous four input sequences is used to predict the speed of traffic for the next 15 minutes, 30 minutes, 45 minutes, 60 minutes, and 120 minutes for the SZ-taxi dataset. As PeMSD7 consists of 5-minute time intervals, for the last one hour 12 timestamps is needed. In Table 4.1, the hyper-meters of model is presented. As the PeMSD7 data set is large, the epoch and hidden layer are kept low compared to the SZ-taxi data set to accommodate resource limitations. All experiments are done using NVIDIA T4 GPU. Python was used as the programming language.

To implement R-PST-GCN model, various Python libraries are utilized, including Numpy, Pandas, TensorFlow, and Keras. These libraries provided essential functionalities for data manipulation, deep learning model creation, and training. Other state-of-the-art traffic flow prediction models are executed in the same experimental setup. This comparison enabled us to evaluate the performance of R-PST-GCN model relative to these models.

Table 4.1: Learning Parameters

| Parameter | Sz-taxi | PeMSD7 |
|---|---|---|
| Learning rate | 0.001 | 0.001 |
| Batch size | 64 | 32 |
| Embedding dimensions | 10 | 10 |
| Training epochs | 3000 | 500 |
| Hidden layers | 100 | 64 |
| Input sequence length | 4 | 12 |
| Optimizer | Adam | Adam |

## 4.2 Dataset Description

Simulations are conducted on two publicly available traffic datasets to demonstrate the efficacy of the proposed model. Specifically, the SZ-taxi dataset is utilized for city road traffic and the PeMSD7 dataset for highway traffic. The SZ-taxi dataset [11], encompasses traffic data from city roads, while the PeMSD7 dataset [45], comprises traffic data from highways.

City road and highway traffic networks exhibit distinct traffic patterns, posing a significant challenge in capturing both types effectively. The selection of these datasets was deliberate to evaluate whether R-PST-GCN can adeptly capture the unique characteristics inherent to both city and highway traffic data. These datasets are extensively utilized in traffic forecasting research and have been employed in numerous previous studies for performance comparison.

**SZ-taxi:** The taxi trajectory data of Shenzhen city, China from January 1 to January 31, 2015, are stored in this dataset. The dataset contains the traffic data for 156 roadways. There are two parts in this dataset, the first part shows the connection between the roads as a 156*156 adjacency matrix. If there is any connection between roads then it would be 1 otherwise it will be 0 in the adjacency matrix. Another one is a 2976*156 feature matrix, where 2976 is the total timestamp with traffic speed for 15-minute time intervals. The feature matrix shows each road's traffic speed over time.

**PeMSD7:** The datasets were collected by the Caltrans Performance Measurement System (PeMS) [45] in real-time at 30-second intervals. The raw traffic data are aggregated into 5-minute intervals. The system includes over 39,000 detectors deployed on highways in major metropolitan areas across California. The datasets also record geographic information about the sensor stations. PeMSD7 is traffic data from 228 sensors measuring the speed of traffic in District 7 of California for the period of May to June 2012 (only weekdays). The size of the feature matrix is 12671*228, where 12671 total timestamp that shows the speed of traffic in 228 sensors. The adjacency matrix is 228*228 which shows the connection between the sensors. This matrix is constructed using Gaussian kernel threshold.

## 4.3 Performance metrics

To evaluate the performance of the proposed traffic prediction model, five metrics are employed: mean absolute error (MAE), root mean square error (RMSE), accuracy, Coefficient of Determination (R-squared), and Variance Score. Each of these metrics offers unique insights into the model's predictive capabilities and overall performance.

The average difference between the actual and predicted data values is represented by the RMSE value. The model and its predictions perform better when the RMSE is smaller.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.1}$$

Here, $y_i$ denotes the actual value, $\hat{y}_i$ the predicted value, and n the total number of observations. RMSE penalizes larger errors more significantly due to the squaring of differences, making it sensitive to outliers. A lower RMSE indicates better model performance.

MAE, another metric for measuring average prediction error, differs from RMSE by not squaring the errors, which makes it less sensitive to extreme values.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)| \tag{4.2}$$

MAE provides a straightforward interpretation of the average magnitude of errors between predicted and actual values.

In the context of traffic prediction, accuracy is defined as the ratio of correctly predicted instances to the total number of instances.

$$Accuracy = 1 - \frac{||Y - \hat{Y}||_F}{||Y||_F} \tag{4.3}$$

Here, $||.||_F$ denotes the Frobenius norm. Accuracy in this sense measures the overall correctness of the model's predictions.

The Coefficient of Determination, or R-squared, assesses the proportion of variance in the dependent variable (traffic condition) that is explained by the independent variables in the model. It is expressed as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{Y})^2} \tag{4.4}$$

In the equation, $\bar{Y}$ represents the mean of the actual values. A higher R-squared value indicates a better fit of the model to the observed data, suggesting that the model explains a large proportion of the variance.

The Variance Score quantifies the proportion of variance in the dependent variable that is captured by the model, providing insights into the model's ability to account for variability in the actual traffic conditions. It is calculated as follows:

$$var = 1 - \frac{var(Y - \hat{Y})}{var(Y)} \tag{4.5}$$

In this equation, $var(.)$ denotes the variance. A higher variance score indicates that the model successfully captures the variability of the traffic data.

## 4.4 Compared methods

To evaluate the effectiveness of R-PST-GCN model for traffic prediction, it is compared with several recent traffic prediction models which are listed below.

1. GCN [7], is a class of neural networks designed to operate on graph-structured data. They extend the concept of convolution from traditional grid-based data, such as images, to irregular domains like graphs. GCN leverages the graph's connectivity structure to aggregate and transform node features, enabling the capture of local and global patterns. This makes them particularly suitable for tasks such as node classification, link prediction, and traffic flow forecasting, where relationships between entities are complex and interdependent. GCN has shown significant promise in effectively modeling spatial dependencies in various applications, including transportation networks.

2. GRU [9], a deep learning model, has proven effective in identifying dependencies in sequential data. It features gating mechanisms, specifically the reset and update gates, which regulate the flow of information and preserve long-term dependencies. The update gate determines how much of the past information needs to be passed to the future, while the reset gate controls the incorporation of new input with the previous memory. GRU is computationally efficient and have shown competitive performance in sequence modeling tasks, such as time series prediction

3. TGCN [11], is a novel traffic forecasting method designed for urban road networks. It integrates a Graph Convolutional Network (GCN) to capture the spatial dependencies within the road network's complex topology and a Gated Recurrent Unit (GRU) to model the temporal dynamics of traffic data. This combination enables T-GCN to effectively handle spatio-temporal forecasting tasks. Evaluations using SZ-taxi speed data and Los-loop datasets demonstrate that T-GCN is suitable for both short-term and long-term traffic forecasting.

4. A3TGCN [12], use the TGCN model for local spatial and temporal features extraction. The attention mechanism is employed to capture the global trend of the traffic flow. It also assumes the road network as a static graph and thus uses a fixed adjacency matrix.

5. AGCRN [18], utilizes a dynamic adjacency matrix. The Node Adaptive Parameter Learning (NAPL) module captures node-specific patterns, while the Data Adaptive Graph Generation (DAGG) module automatically infers inter-dependencies among different traffic series. Traditional pre-defined graphs may lack comprehensive spatial dependency information and are not directly related to prediction tasks, potentially introducing significant biases. To address this issue, AGCRN employs an adaptive adjacency matrix. These modules are integrated into AGCRN, which leverages recurrent networks and the adaptive modules to automatically capture spatial and temporal correlations within traffic series data

6. DCGCN [13], is specifically designed for fine-grained prediction at the lane level. It utilizes GCN with a data-driven adjacency matrix to capture spatial features, treating individual lanes as nodes. The proposed data-driven adjacency matrix integrates a distance-based adjacency matrix with correlation matrix, enabling adaptation to spatio-temporal changes. Temporal features

are extracted using a GRU. A gating mechanism is employed to fuse spatial and temporal features, resulting in the final spatio-temporal features used for lane-level traffic prediction.

7. ASTGCN [38], used attention-based spatial and temporal convolutional network. This model comprises three distinct components, each designed to capture different temporal properties of traffic data, including recent, daily-periodic, and weekly-periodic dependencies. Each component employs a spatial-temporal attention mechanism to effectively capture dynamic correlations. Additionally, graph convolutions for spatial pattern extraction and standard convolutions for temporal feature modeling are used. The outputs from these components are weighted and fused to generate the final prediction results.

8. DCRNN [22], introduces bidirectional random walks in directed graphs for this purpose. This model integrates diffusion convolution into the GRU architecture, resulting in the Diffusion Convolutional Gated Recurrent Unit (DCGRU). For temporal modeling, DCRNN employs a Sequence-to-Sequence (Seq2Seq) architecture, with both the encoder and decoder composed of DC-GRUs. During training, historical time series data is input into the encoder, and the encoder's final states are used to initialize the decoder, which subsequently generates predictions based on previous ground truth observations.

9. USTGCN [39], is designed to comprehensively capture both spatial and temporal dependencies in traffic data. Unlike traditional methods that separately process spatial and temporal information, USTGCN integrates these components using a spatio-temporal adjacency matrix, enabling simultaneous aggregation. This model also incorporates historical daily patterns, analyzing traffic data from the past week alongside current-day patterns to enhance prediction accuracy. USTGCN demonstrates performance in traffic

forecasting tasks on three publicly available datasets from the Performance Measurement System (PeMS).

Some traffic prediction models are also added that are designed to handle missing data using various imputation techniques.

1. LSTM-M [42]: LSTM-M utilizing LSTM performs traffic prediction with missing data. It employs a multi-scale temporal smoothing technique to infer missing values.

2. GRU-D [46]: GRU-D, derived from GRU, performs prediction tasks by integrating masking information and incorporating the time intervals of missing values as input.

3. GMN [19]: GMN considers traffic networks as a graph Markov process and infers missing values step by step.

## 4.5 Prediction with Normal Data

The performance of R-PST-GCN and baseline models is summarized in Table 4.2. R-PST-GCN outperforms all baseline methods across all metrics in the SZ-taxi dataset. For the PeMSD7 dataset, R-PST-GCN achieves superior results for the 15-minute and 30-minute prediction intervals. However, for longer prediction horizons (45, 60, and 120 minutes), the USTGCN model exhibits better performance due to its utilization of both current and historical data of the past week. Although this enhances long-term prediction accuracy, it comes with a significant increase in computational costs, as illustrated in Table 4.4. GCN consistently exhibits the highest RMSE and lowest accuracy across both datasets and all time horizons, which is expected given its limitation in capturing temporal features.

In the SZ-taxi dataset, GRU follows GCN with high RMSE and low accuracy, primarily because it overlooks the spatial dynamics of traffic flow. As the

Table 4.2: The overall prediction results of baseline methods and R-PST-GCN in the SZ-taxi dataset and PeMSD7 dataset

| | | **SZ-taxi** | | | | | **PeMSD7** | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Time | Model | RMSE | MAE | Accuracy | $R^2$ | var | RMSE | MAE | Accuracy | $R^2$ | var |
| 15min | GCN | 6.0141 | 4.4683 | 0.5811 | 0.6681 | 0.6682 | 12.1832 | 8.9170 | 0.7960 | 0.2189 | 0.2212 |
| | GRU | 4.3054 | 2.8171 | 0.7000 | 0.8299 | 0.8299 | 3.7234 | 2.1488 | 0.9376 | 0.9270 | 0.9271 |
| | TGCN | 4.1094 | 2.7812 | 0.7137 | 0.8450 | 0.8452 | 9.9978 | 6.8702 | 0.8326 | 0.4740 | 0.4741 |
| | A3TGCN | 4.0883 | 2.7768 | 0.7148 | 0.8462 | 0.8463 | 9.8651 | 6.6965 | 0.8348 | 0.4879 | 0.4891 |
| | AGCRN | 4.0840 | 2.7600 | 0.7155 | 0.8469 | 0.8470 | 3.4139 | 2.0732 | 0.9428 | 0.9386 | 0.9386 |
| | DCGCN | 4.2861 | 2.8866 | 0.7014 | 0.8314 | 0.8314 | 3.6369 | **2.0073** | 0.9394 | 0.9312 | 0.9312 |
| | ASTGCN | 4.0678 | 2.7081 | 0.7166 | 0.8482 | 0.8485 | 4.1703 | 2.1873 | 0.9302 | 0.9085 | 0.9091 |
| | DCRNN | 4.0501 | 2.6931 | 0.7105 | 0.8492 | 0.8498 | 4.1032 | 2.1273 | 0.9312 | 0.9185 | 0.9191 |
| | USTGCN | 4.0302 | 2.6696 | 0.7189 | 0.8259 | 0.8331 | 3.4838 | 2.0413 | 0.9399 | 0.9212 | 0.9242 |
| | R-PST-GCN | **4.0055** | **2.6660** | **0.7210** | **0.8530** | **0.8532** | **3.4033** | 2.0273 | **0.9430** | **0.9390** | **0.9391** |
| 30min | GCN | 6.0126 | 4.472 | 0.5811 | 0.6683 | 0.6684 | 12.1834 | 8.9311 | 0.7960 | 0.2193 | 0.2220 |
| | GRU | 4.3602 | 2.8485 | 0.6963 | 0.8256 | 0.8256 | 5.0128 | 2.8016 | 0.9160 | 0.8678 | 0.8680 |
| | TGCN | 4.1429 | 2.8225 | 0.7114 | 0.8425 | 0.8426 | 9.7613 | 6.6534 | 0.8365 | 0.4988 | 0.50009 |
| | A3TGCN | 4.0792 | 2.7396 | 0.7158 | 0.8473 | 0.8474 | 9.5860 | 6.5601 | 0.8394 | 0.5167 | 0.5169 |
| | AGCRN | 4.1398 | 2.8084 | 0.7116 | 0.8428 | 0.8429 | 4.6203 | 2.8452 | 0.9226 | 0.8877 | 0.8878 |
| | DCGCN | 4.3217 | 2.8879 | 0.6989 | 0.8286 | 0.8286 | 4.94032 | 2.6954 | 0.9172 | 0.8716 | 0.8736 |
| | ASTGCN | 4.0587 | 2.6848 | 0.7143 | 0.8489 | 0.8490 | 4.5005 | **2.4326** | 0.9246 | 0.8935 | 0.8940 |
| | DCRNN | 4.0411 | 2.6754 | 0.7153 | 0.8499 | 0.8498 | 4.4915 | 2.4426 | 0.9247 | 0.8937 | 0.8940 |
| | USTGCN | 4.0441 | 2.6785 | 0.7166 | 0.8501 | 0.8502 | 4.4830 | 2.5732 | 0.9238 | 0.8935 | 0.8953 |
| | R-PST-GCN | **4.0257** | **2.6686** | **0.7196** | **0.8513** | **0.8514** | **4.4765** | 2.5451 | **0.9250** | **0.8946** | **0.8948** |
| 45min | GCN | 6.0300 | 4.4872 | 0.5799 | 0.6664 | 0.6665 | 12.1883 | 8.9190 | 0.7959 | 0.2191 | 0.2215 |
| | GRU | 4.38689 | 2.9148 | 0.6944 | 0.8234 | 0.8234 | 5.9451 | 3.3000 | 0.9004 | 0.8142 | 0.8142 |
| | TGCN | 4.1757 | 2.8472 | 0.7091 | 0.8400 | 0.8401 | 9.8751 | 6.8739 | 0.8346 | 0.4873 | 0.4884 |
| | A3TGCN | 4.1026 | 2.7394 | 0.7142 | 0.8456 | 0.8457 | 10.3508 | 7.0555 | 0.8266 | 0.4365 | 0.4388 |
| | AGCRN | 4.1470 | 2.8226 | 0.7111 | 0.8422 | 0.8425 | 5.3805 | 3.3711 | 0.9099 | 0.8478 | 0.8480 |
| | DCGCN | 4.3628 | 2.9148 | 0.6960 | 0.825 | 0.8254 | 5.7484 | 3.1644 | 0.9037 | 0.8263 | 0.8291 |
| | ASTGCN | 4.0988 | 2.7897 | 0.7145 | 0.8459 | 0.8459 | 6.8598 | 3.4272 | 0.8851 | 0.7526 | 0.7530 |
| | DCRNN | 4.0898 | 2.7287 | 0.7155 | 0.8460 | 0.8469 | 6.7508 | 3.3272 | 0.8951 | 0.7626 | 0.7670 |
| | USTGCN | 4.4517 | 2.9218 | 0.6863 | 0.8216 | 0.8219 | **5.1539** | 3.1602 | **0.9138** | **0.8691** | **0.8691** |
| | R-PST-GCN | **4.0469** | **2.7033** | **0.7181** | **0.8497** | **0.8500** | 5.2171 | **3.1271** | 0.9126 | 0.8569 | 0.8575 |
| 60min | GCN | 6.0476 | 4.5016 | 0.5787 | 0.6645 | 0.6646 | 12.2024 | 8.9210 | 0.7956 | 0.2177 | 0.2199 |
| | GRU | 4.4044 | 2.8599 | 0.6932 | 0.8220 | 0.8226 | 6.70139 | 3.7495 | 0.8877 | 0.7640 | 0.7642 |
| | TGCN | 4.1894 | 2.8647 | 0.7081 | 0.8390 | 0.8393 | 10.0478 | 6.9842 | 0.8317 | 0.4695 | 0.4701 |
| | A3TGCN | 4.1298 | 2.7866 | 0.7123 | 0.8435 | 0.8439 | 10.6701 | 7.3645 | 0.8213 | 0.4018 | 0.4020 |
| | AGCRN | 4.1860 | 2.8551 | 0.7084 | 0.8393 | 0.8393 | 5.7984 | 3.6053 | 0.9028 | 0.8233 | 0.8246 |
| | DCGCN | 4.4317 | 2.9901 | 0.6912 | 0.8198 | 0.8199 | 6.4646 | 3.5409 | 0.8917 | 0.7804 | 0.7834 |
| | ASTGCN | 4.0990 | 2.7022 | 0.7166 | 0.8482 | 0.8478 | 6.8648 | 3.5369 | 0.8851 | 0.7524 | 0.7527 |
| | DCRNN | 4.0901 | 2.6922 | 0.7168 | 0.8485 | 0.8480 | 6.7658 | 3.4969 | 0.8878 | 0.7704 | 0.7713 |
| | USTGCN | 7.9399 | 4.4331 | 0.4435 | 0.4221 | 0.4770 | **5.6172** | **3.2244** | 0.9043 | **0.8435** | **0.8435** |
| | R-PST-GCN | **4.0473** | **2.6950** | **0.7174** | **0.8491** | **0.8493** | 5.6699 | 3.3784 | **0.9050** | 0.8311 | 0.8315 |
| 120min | GCN | 6.0959 | 4.5414 | 0.5753 | 0.6545 | 0.6593 | 12.3912 | 9.2704 | 0.7924 | 0.1942 | 0.1994 |
| | GRU | 4.4244 | 2.9323 | 0.6918 | 0.8204 | 0.8204 | 8.7432 | 5.0310 | 0.8535 | 0.5988 | 0.6063 |
| | TGCN | 4.2241 | 2.8824 | 0.7057 | 0.8364 | 0.8364 | 10.7370 | 7.7221 | 0.8201 | 0.3950 | 0.4009 |
| | A3TGCN | 4.1487 | 2.8069 | 0.7109 | 0.8422 | 0.8424 | 10.7414 | 7.5372 | 0.8201 | 0.3945 | 0.3958 |
| | AGCRN | 4.2209 | 2.8769 | 0.7059 | 0.8366 | 0.8371 | 6.9623 | 4.3482 | 0.8740 | 0.7043 | 0.7039 |
| | DCGCN | 4.5106 | 3.0622 | 0.6857 | 0.8135 | 0.8135 | 7.9928 | 4.6833 | 0.8661 | 0.6647 | 0.6704 |
| | ASTGCN | 4.0961 | 2.7429 | 0.7147 | 0.8462 | 0.8468 | 7.6764 | 4.1701 | 0.8715 | 0.6908 | 0.6912 |
| | DCRNN | 4.0861 | 2.7089 | 0.7167 | 0.8469 | 0.8470 | 7.5764 | 4.1023 | 0.8745 | 0.6968 | 0.6972 |
| | USTGCN | 6.3269 | 3.5680 | 0.5566 | 0.6331 | 0.6452 | **6.4278** | **3.8428** | **0.8898** | **0.7988** | **0.7996** |
| | R-PST-GCN | **4.0751** | **2.7049** | **0.7161** | **0.8478** | **0.8480** | 6.8359 | 4.0016 | 0.8855 | 0.7547 | 0.7551 |

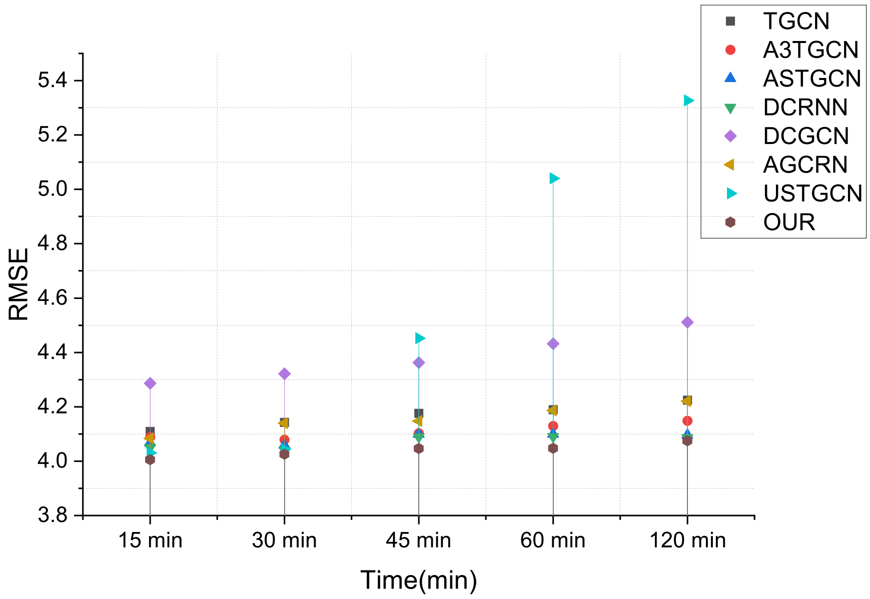



Figure 4.2: RMSE value of all methods under different horizons in the SZ-taxi dataset.

A3TGCN and ASTGCN models used attention mechanisms, it has lower RMSE and higher accuracy than TGCN. Though DCGCN used a dynamic correlation matrix to better capture the spatial feature, it still has comparatively higher RMSE. While USTGCN demonstrates strong performance in the PeMSD7 dataset, its effectiveness diminishes when applied to the SZ-taxi dataset. This disparity is attributable to the SZ-taxi dataset's longer time intervals, where historical data becomes less impactful for accurate predictions. Consequently, integrating historical data confuses the model, leading to diminished performance, particularly for longer prediction horizons. Among baseline models, the DCRNN model notably excelled because of its bidirectional random walks in graph. However, R-PST-GCN managed to decrease RMSE by 1.11% while increasing accuracy by 1.45% compared to DCRNN.

Moreover, R-PST-GCN offers a computational advantage over DCRNN, which incurs higher computational costs. In Figure 4.2, the RMSE comparison of all models across different horizons for the SZ-taxi dataset is presented. As illustrated in Figure 4.2, it's evident that R-PST-GCN consistently achieves the lowest RMSE

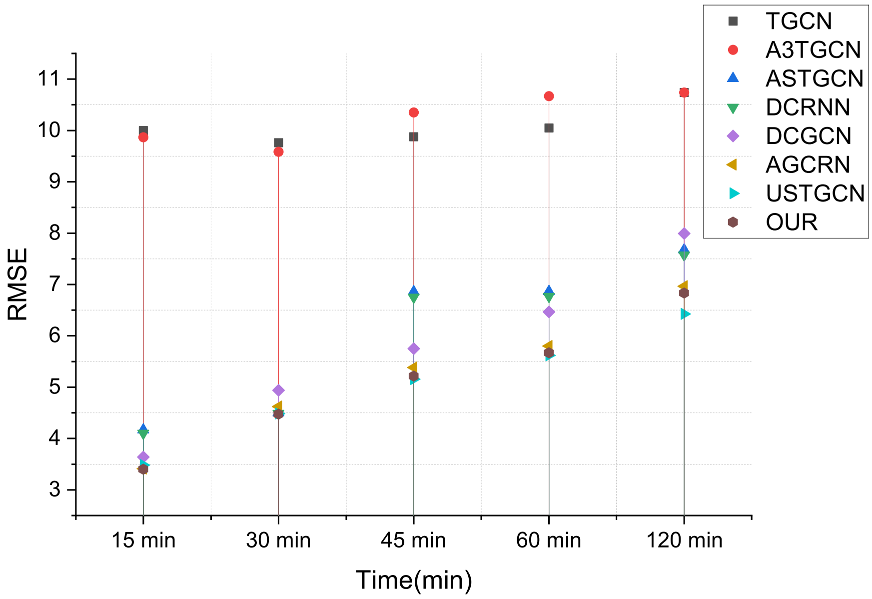



Figure 4.3: RMSE value of all methods under different horizons in the PeMSD7 dataset

compared to other baseline models.

In the PeMSD7 dataset, we can see that TGCN, A3TGCN, ASTGCN and DCRNN worked poorly compared to how those models did in the SZ-taxi dataset. When the time interval is low means high temporal resolution data, TGCN, A3TGCN, ASTGCN, DCRNN and GCN models have high RMSE as these models use static adjacency matrix. The static adjacency matrix performs worst when the time interval is very low as it contains high variability making it hard to predict accurately [47]. Data that have a lower temporal resolution like the SZ-taxi dataset are smoother due to aggregating traffic measurements of a relatively longer horizon; as a result, TGCN and A3TGCN performed well on the SZ-taxi dataset compared to the PeMSD7 dataset. On the other hand, AGCRN, DCGCN, USTGCN and R-PST-GCN use a data-driven dynamic adjacency matrix for that reason these models can better capture the traffic flow even when the time interval is low.

We can see that it is crucial to use a dynamic adjacency matrix as it can represent dynamic spatial traffic patterns rather than a predefined adjacency matrix.

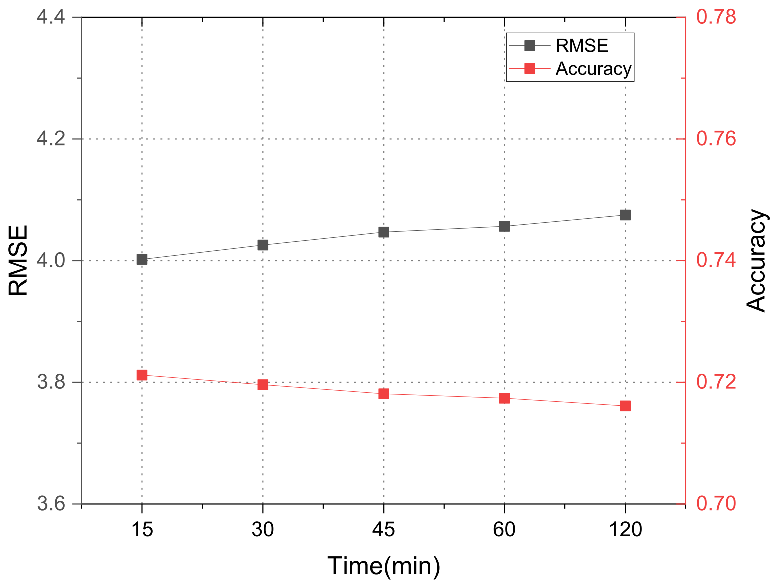



Figure 4.4: RMSE and Accuracy of R-PST-GCN under the different horizons in the SZ-taxi dataset

R-PST-GCN works better than AGCRN and DCGCN since it uses a probabilistic spatiotemporal graph. As for USTGCN, it considers more historical data that helps in gaining lower RMSE than R-PST-GCN for long-term prediction. However that cost in higher computational training time of USTGCN. This makes the model infeasible for real time traffic prediction. In real time traffic prediction, models need to have lower computational time to give fast traffic prediction. In Fig. 4.3 the RMSE of all models for the different horizons in the PeMSD7 dataset is plotted.

The R-PST-GCN model is capable of being utilized for both short-term and long-term prediction tasks. From Fig. 4.4 and Fig. 4.5, we can see that no matter how much the horizon changes RMSE values are consistent. We have predicted for 15-minute, 30-minute, 45-minute, 60-minute, and 120-minute time horizons to check the short-term and long-term forecasting capability. For the SZ-taxi datasets, we can see that in Fig. 4.4 the RMSE and accuracy don't fluctuate much. Although PeMSD7's larger size facilitates better learning for deep learning

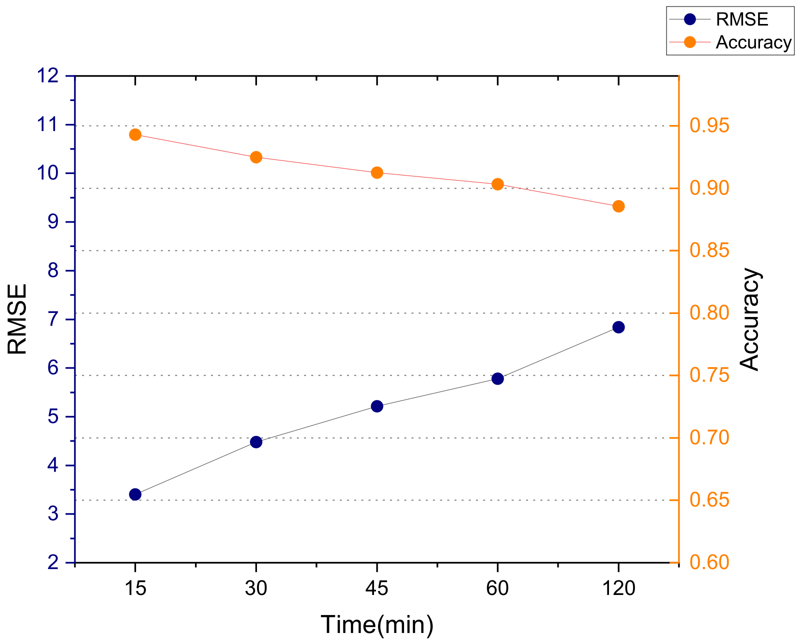



Figure 4.5: RMSE and Accuracy of R-PST-GCN under the different horizons in the PeMSD7 dataset

models compared to SZ-taxi, its 5-minute intervals pose challenges for predicting longer horizons like 60 minutes and 120 minutes, leading to a slight increase in RMSE for these intervals (Figure 4.5). Nonetheless, R-PST-GCN maintains lower RMSE values compared to other models while offering reduced computational costs compared to USTGCN, which excels in long-term predictions for PeMSD7. Thus, R-PST-GCN strikes a balance between precision and efficiency by delivering better performance with lower computational overhead.

## 4.6 Prediction with Random Missing Data

Traffic prediction is conducted for 15 minutes using one-hour historical traffic data from the SZ-taxi dataset under various missing rates. For the PeMSD7 dataset, the previous 10 timestamps are used to predict the next timestamp. In Table. 4.2 the result for both datasets are displayed. We can see that under random missing perturbation, R-PST-GCN outperforms baseline models. R-PST-GCN has the lowest RMSE compared to other models for different missing rates. LSTM-M and GMN showed better performance on the PeMSD7 dataset compared to the SZ-taxi dataset due to the smaller size of the SZ-taxi dataset. However, GRU-D didn't perform well on both datasets.

Table 4.3: Prediction results of baseline models and R-PST-GCN for various missing rates in the SZ-taxi dataset and the PeMSD7 dataset.

| Missing rate | Model | SZ-taxi | | | | | PeMSD7 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RMSE | MAE | Accuracy | $R^2$ | var | RMSE | MAE | Accuracy | $R^2$ | var |
| 10% | GCN | 6.1379 | 4.5645 | 0.5724 | 0.6541 | 0.6542 | 12.2355 | 8.9008 | 0.7953 | 0.2102 | 0.2112 |
| | GRU | 4.9357 | 3.2604 | 0.6561 | 0.7763 | 0.7778 | 8.3511 | 5.9427 | 0.8603 | 0.6321 | 0.6327 |
| | TGCN | 4.2558 | 2.9380 | 0.7035 | 0.8337 | 0.8337 | 10.6125 | 7.2897 | 0.8224 | 0.4058 | 0.4079 |
| | A3TGCN | 4.2408 | 2.8865 | 0.7046 | 0.8349 | 0.8349 | 10.3493 | 7.1063 | 0.8269 | 0.4350 | 0.4358 |
| | AGCRN | 4.2684 | 2.9312 | 0.7026 | 0.8327 | 0.8327 | 4.1401 | 2.6431 | 0.9307 | 0.9095 | 0.9105 |
| | DCGCN | 4.9136 | 3.3604 | 0.6577 | 0.7783 | 0.7789 | 3.9407 | 2.2971 | 0.9341 | 0.9181 | 0.9182 |
| | LSTM-M | 6.9719 | 4.6873 | 0.6145 | 0.6884 | 0.6908 | 5.5901 | 3.4368 | 0.9070 | 0.8353 | 0.8357 |
| | GRU-D | 7.7023 | 5.4650 | 0.5147 | 0.4643 | 0.4699 | 12.9795 | 8.9977 | 0.7840 | 0.1122 | 0.1195 |
| | GMN | 7.3423 | 4.9671 | 0.6017 | 0.6169 | 0.6338 | 5.4431 | 3.5720 | 0.9094 | 0.8439 | 0.8441 |
| | R-PST-GCN | **4.0981** | **2.7426** | **0.7145** | **0.8458** | **0.8460** | **2.8697** | **1.8523** | **0.9524** | **0.9573** | **0.9579** |
| 20% | GCN | 6.2466 | 4.6383 | 0.5648 | 0.6417 | 0.6420 | 12.2606 | 8.9405 | 0.7949 | 0.2070 | 0.2083 |
| | GRU | 5.4116 | 3.7098 | 0.6230 | 0.7311 | 0.7319 | 9.8438 | 7.1996 | 0.8353 | 0.4888 | 0.4895 |
| | TGCN | 4.2969 | 2.9761 | 0.7006 | 0.8305 | 0.8305 | 10.6767 | 7.3622 | 0.8214 | 0.3987 | 0.3990 |
| | A3TGCN | 4.3117 | 2.9890 | 0.6996 | 0.8293 | 0.8295 | 10.8265 | 7.3679 | 0.8189 | 0.3817 | 0.3840 |
| | AGCRN | 4.3117 | 2.9890 | 0.6996 | 0.8293 | 0.8295 | 4.55279 | 2.8768 | 0.9238 | 0.89065 | 0.8907 |
| | DCGCN | 4.9894 | 3.5872 | 0.6524 | 0.7714 | 0.7717 | 4.2467 | 2.5555 | 0.9290 | 0.9049 | 0.9050 |
| | LSTM-M | 7.2885 | 4.9308 | 0.6552 | 0.6692 | 0.6704 | 6.2296 | 3.7713 | 0.8965 | 0.7888 | 0.7890 |
| | GRU-D | 7.7842 | 5.3798 | 0.3987 | 0.4081 | 0.4280 | 12.7319 | 9.2290 | 0.7885 | 0.1180 | 0.1180 |
| | GMN | 7.5934 | 5.1103 | 0.4981 | 0.5900 | 0.6091 | 5.5580 | 3.7318 | 0.9077 | 0.8319 | 0.8322 |
| | R-PST-GCN | **4.1474** | **2.7830** | **0.7111** | **0.8421** | **0.8423** | **3.1132** | **2.0112** | **0.9479** | **0.9489** | **0.9502** |
| 40% | GCN | 6.6705 | 4.9104 | 0.5353 | 0.5915 | 0.5918 | 12.2739 | 8.8665 | 0.7947 | 0.2053 | 0.2055 |
| | GRU | 6.4535 | 4.6824 | 0.5504 | 0.6176 | 0.6176 | 11.6429 | 8.6497 | 0.8052 | 0.2849 | 0.2857 |
| | TGCN | 4.4080 | 3.1549 | 0.6929 | 0.8216 | 0.8229 | 10.8774 | 7.3501 | 0.8180 | 0.3758 | 0.3816 |
| | A3TGCN | 4.3852 | 3.0946 | 0.6945 | 0.8234 | 0.8250 | 10.7750 | 7.2574 | 0.8197 | 0.3875 | 0.3890 |
| | AGCRN | 4.4013 | 3.0287 | 0.6934 | 0.8221 | 0.8222 | 5.6731 | 3.6947 | 0.9050 | 0.8302 | 0.8303 |
| | DCGCN | 5.5758 | 4.1052 | 0.6115 | 0.7146 | 0.7150 | 5.6163 | 3.3714 | 0.9060 | 0.8336 | 0.8336 |
| | LSTM-M | 8.1514 | 5.5630 | 0.5502 | 0.6150 | 0.6166 | 7.1740 | 4.3324 | 0.8806 | 0.7288 | 0.7288 |
| | GRU-D | 7.8258 | 5.4817 | 0.4437 | 0.4778 | 0.4810 | 13.0664 | 8.9794 | 0.7825 | 0.1002 | 0.1184 |
| | GMN | 8.2432 | 5.5723 | 0.4983 | 0.5178 | 0.5466 | 5.7168 | 4.0210 | 0.9049 | 0.8278 | 0.8278 |
| | R-PST-GCN | **4.1541** | **2.7810** | **0.7106** | **0.8416** | **0.8418** | **3.8334** | **2.4361** | **0.9359** | **0.9225** | **0.9230** |

Although GMN showed consistent results, it has a higher RMSE than R-PST-GCN. DCGCN performed well on prediction tasks but it can not handle missing data well. The reason is that DCGCN depends on node correlation thus if the missing rate goes higher its error rate also goes up. Conversely, R-PST-GCN's

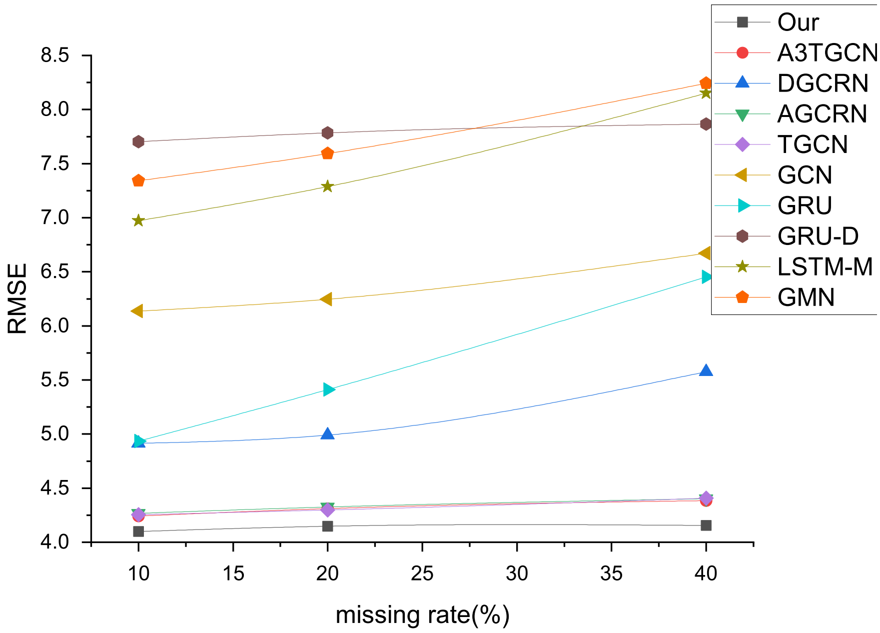



Figure 4.6: Traffic forecasting using various missing rates in SZ-taxi dataset

probabilistic adjacency matrix mitigates the impact of missing values, resulting in stable performance even with increasing missing rates. This is because Bayesian inference incorporates uncertainty into the analysis to deal with missing data. In Figure 4.6, the RMSE of all baseline models and R-PST-GCN for various missing rate in DZ-taxi dataset is plotted. R-PST-GCN achieved the lowest RMSE compared to the baseline models for different missing rates. It proves the models capability of handling missing data inherently without any preprocess or imputation technique.

The proposed probabilistic adjacency matrix treats missing data as additional parameters, estimating their distribution based on observed data. It refines the estimation using iterative updating, thus producing a comprehensive understanding of both observed and missing values. This flexibility provides a more realistic representation of uncertainty and contributes to robust decision-making in the presence of missing/incomplete information.

## 4.7 Prediction with Noisy Data

To evaluate the robustness of the proposed approach in eliminating noise, experiments with two distinct types of noise are conducted on the SZ-taxi and PeMSD7 datasets. The first type of noise is Gaussian noise and the second type is Poisson noise. Gaussian noise is characterized by a Gaussian distribution $N(0, \sigma^2)$, where $\sigma$ takes on values from the set $\{0.2, 0.4, 0.8, 1, 2\}$. Gaussian noise is commonly used to simulate random variability in data, resembling the random fluctuations typically observed in real-world measurements.

In Figure 4.7 presents the outcomes when Gaussian noise is introduced in SZ-taxi dataset. The y-axis represents various evaluation metrics depicted in different colors, while the x-axis indicates the different values of $\sigma$. In Figure 4.8 presents

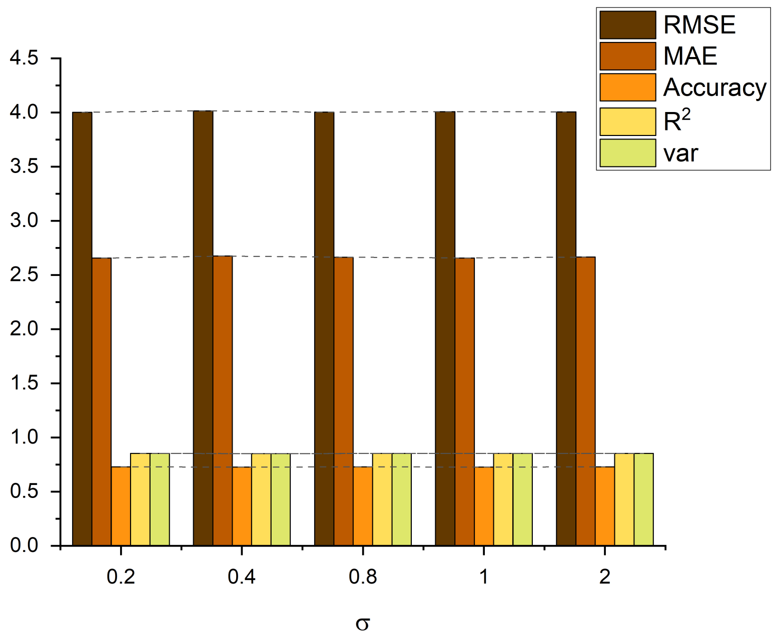



Figure 4.7: Gaussian Perturbation on SZ-taxi dataset

the outcomes when Gaussian noise is introduced in PeMSD7 dataset. The y-axis represents various evaluation metrics depicted in different colors, while the x-axis indicates the different values of $\sigma$.

The second type of noise is Poisson noise, characterized by a Poisson distri-

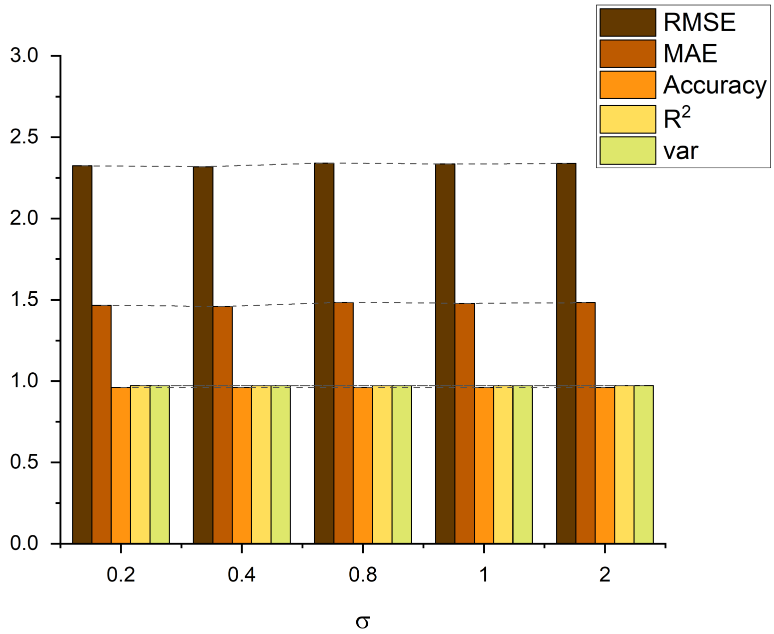



Figure 4.8: Gaussian Perturbation on PeMSD7 dataset

bution $P(\lambda)$, where $\lambda$ takes on values from the set $\{1, 2, 4, 8, 16\}$. Poisson noise is particularly relevant for count-based data, modeling the variability observed in discrete event counts over a fixed interval.

In Figure 4.9 shows the prediction outcomes under Poisson noise in SZ-taxi dataset. The y-axis represents various evaluation metrics depicted in different colors, while the x-axis indicates the different values of $\lambda$.

Similarly, Figure 4.10 shows the prediction outcomes under Poisson noise in PeMSD7 dataset. From these figures, it is evident that despite variations in $\sigma$ and $\lambda$, the evaluation metrics remain stable. This indicates that R-PST-GCN is capable of maintaining its predictive accuracy even when data is noisy.

These experiments collectively demonstrate that R-PST-GCN can effectively handle high levels of noise in both the SZ-taxi and PeMSD7 datasets, showcasing its robustness and reliability in noisy environments. This robustness is crucial for real-world traffic prediction tasks where data can often be noisy and imperfect.

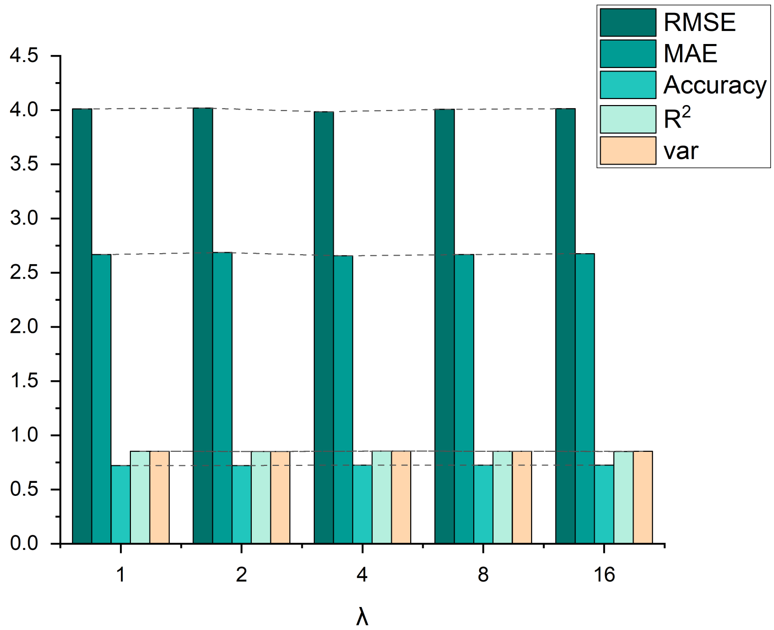



Figure 4.9: Poisson Perturbation on SZ-taxi dataset

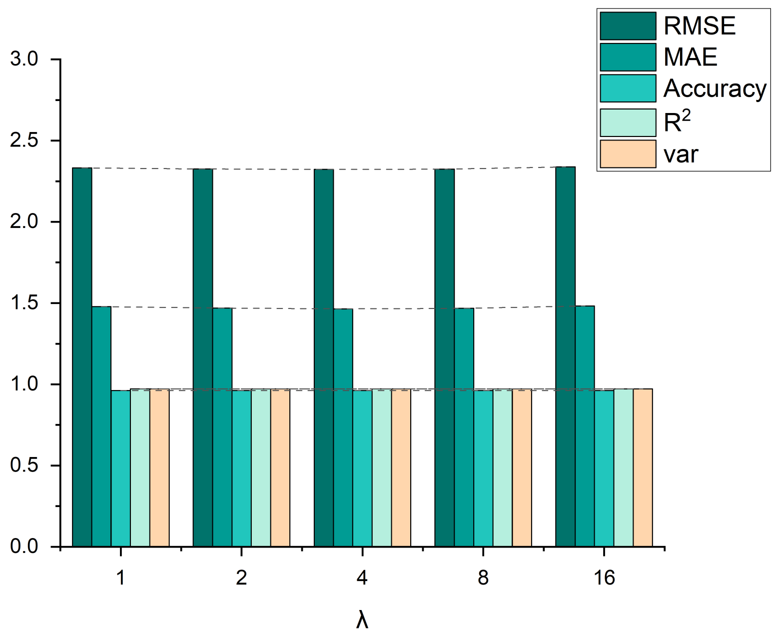



Figure 4.10: Poisson Perturbation on PeMSD7 dataset

## 4.8 Computational Cost

To evaluate the computational cost, the R-PST-GCN model is compared with baseline models, as shown in Table 4.4. The training times of these models are calculated for the prediction task on the PeMSD7 dataset. Although the TGCN and A3TGCN models require minimal training time, their RMSE values are relatively high. The training time of the DGCRN model is comparable to that of R-PST-GCN, but its RMSE is higher. Moreover, the performance of DGCRN degrades significantly more than R-PST-GCN as the prediction horizon increases.

As depicted in Figure 4.3, the AGCRN model exhibits the second lowest RMSE among the baseline models; however, it requires a substantial amount of training time, approximately five times that of R-PST-GCN. The DCRNN and ASTGCN models employ iterative methods for prediction tasks, which increases their computational time. The USTGCN model, which considers traffic data from the previous week, has a significantly higher training time (nearly 100 times that of R-PST-GCN), whereas all other models only consider current time data.

Table 4.4: The Computational Cost on PeMSD7 Dataset

| Model | Training Time (s/epoch) | RMSE |
|---|---|---|
| TGCN | 5.58 | 9.9978 |
| A3TGCN | 5.93 | 9.8615 |
| ASTGCN | 35.12 | 4.1703 |
| AGCRN | 35.21 | 3.4139 |
| DCRNN | 36.01 | 4.2532 |
| USTGCN | 100.91 | 3.4828 |
| DGCRN | 8.05 | 3.6312 |
| R-PST-GCN | 8.10 | 3.4033 |

Real-time traffic prediction requires fast computation and high accuracy. However, the ASTGCN, AGCRN, DCRNN, and USTGCN models have very high training times, making them infeasible for real-time traffic prediction tasks. In contrast, R-PST-GCN has significantly lower training times and lower RMSE, making it suitable for real-time traffic prediction tasks for both short-term and long-term predictions.

In comparison to the baseline models, it is evident that the computational cost of R-PST-GCN is reasonable, especially when considering its significant performance improvement.

## 4.9 Complexity Analysis

A thorough analysis of the time complexity for R-PST-GCN and baseline models is conducted, summarizing the findings in Table 4.5. In analysis, various parameters are considered such as the time sequence length denoted by $T$, the feature length denoted by $F$, and the number of nodes denoted by $N$. Additionally, the embedding dimension is defined as $E$, the number of layers as $L$, and the filter dimension for convolution as $K$.

For the training complexity of R-PST-GCN, as shown in Algorithm 1, the complexity of adjacency matrix calculation (step 2) is $O(ITN^2)$. The complexity of GCN and GRU in steps 4 to 9 is $O(ITEF) + O(ITNEF^2) + O(ITH^2) + O(ITEFN^2)$. The complexity of the attention mechanism (shown in steps 12 to 14) is $O(IT^2NH)$ and the complexity of step 15 is $O(NF) + O(N)$. Thus, the training complexity is $O(ITN^2) + O(ITEF) + O(ITNEF^2) + O(ITH^2) + O(ITEFN^2) + O(IT^2NH) + O(NF) + O(N)$.

Considered $I >> N > F > H > E, T$, the training complexity could be simplified as $O(ITN^2) + O(ITNEF^2)$. For the test complexity of R-PST-GCN, as the test phase only carries out steps 2 to 16, the test complexity of $O(TN^2) + O(TFEN^2)$, is far less than the training complexity of R-PST-GCN.

To present simplified versions of their complexities, the dominant components of the models are focused on. From Table 4.5, it's evident that TGCN and A3TGCN models exhibit lower complexity compared to others due to their utilization of static adjacency matrices. Next, DGCRN and R-PST-GCNs have

Table 4.5: The Comparison of Computational Complexity

| Model | Complexity |
|---|---|
| **TGCN** | $O(TFHN^2)$ |
| A3TGCN | $O(TFHN^2) + O(T^2NF)$ |
| ASTGCN | $O(LNTF^2) + O(TNHF^2) + O(KN^2H^2)$ |
| AGCRN | $O(TN^2E) + O(TEF^2N^2)$ |
| DCRNN | $O(TKHN^2) + O(TNFH^2) + O(TNH^2)$ |
| **USTGCN** | $O(TN^3) + O(N^2T^2HF)$ |
| DGCRN | $O(TN^2) + O(TFN^2)$ |
| R-PST-GCN | $O(TN^2) + O(TEFN^2)$ |

relatively higher complexities compare to TGCN and A3TGCN because of the use of dynamic adjacency matrices. After that, AGCRN's complexity is higher due to its incorporation of a learnable node embedding matrix. DCRNN and ASTGCN, which employ complex diffusion convolution and attention mechanisms, respectively, incur increased computational costs. Moreover, DCRNN and ASTGCN adopt iterative prediction approaches, impacting their speed, as shown in Table 4.4. Finally, USTGCN's complexity is highest among the comparative methods as it aggregates data from the previous week, leading to larger input matrices and longer processing times.

## 4.10 Ablation Study

To gain a deeper understanding of the impact of the probabilistic adjacency matrix and node-specific learning on handling missing data, an ablation study was conducted. This study evaluated the prediction performance of the R-PST-GCN model under different conditions on the PeMSD7 dataset. Specifically, the model's performance was assessed without the probabilistic adjacency matrix and without node-specific learning across various missing data rates.

As illustrated in Figure 4.11, the use of a static adjacency matrix, as opposed to a probabilistic adjacency matrix, resulted in a substantial increase in RMSE. This indicates that the probabilistic adjacency matrix plays a crucial role in managing

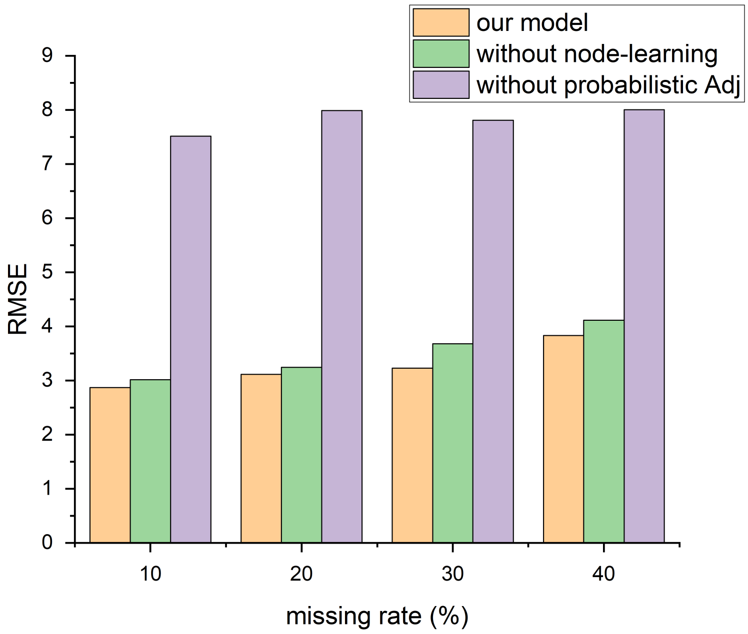



Figure 4.11: Ablation study on PeMSD7 dataset.

missing data. When the node-specific learning module was removed, the RMSE also increased, although the effect was less pronounced than the removal of the probabilistic adjacency matrix. This suggests that while both components are important, the probabilistic adjacency matrix has a more significant impact on the model's ability to handle missing data. These findings underscore the importance of the probabilistic adjacency matrix in improving the robustness and accuracy of the R-PST-GCN model in the presence of missing data. The node-specific learning module also contributes to the model's performance, reducing error to an extent. Overall, the ablation study highlights the critical components of the R-PST-GCN model that enhance its predictive capabilities under challenging conditions.

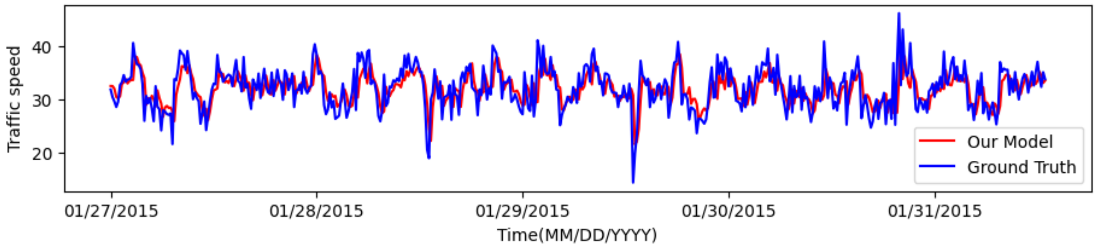
## 4.11 Visualization

To better understand the predictive capability of R-PST-GCN, the actual and predicted traffic data are analyzed for a specific road segment from the SZ-taxi dataset at various time horizons: 15 minutes, 30 minutes, 45 minutes, 60 minutes, and 120 minutes. Figure 4.12 presents these predictions for one of the 156 roads in the dataset. Similarly, for the PeMSD7 dataset, predictions for a selected sensor among the 228 sensors are visualized at the same time horizons in Figure 4.13. Analysis reveals that R-PST-GCN is capable of accurately capturing traffic patterns, including peak traffic periods, across both datasets. Traditional GCN often yields suboptimal predictions due to their inherent smoothing effects, which can obscure critical traffic dynamics and peaks [11]. In contrast, the R-PST-GCN model addresses this limitation by incorporating a modified GCN architecture that effectively learns the unique characteristics of individual nodes and captures dynamic spatial dependencies. This enhancement enables the model to maintain high fidelity in reflecting traffic peaks and overall trends.

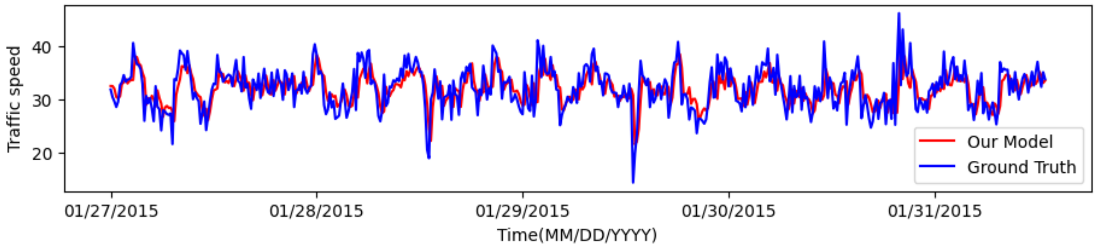
The modifications in the GCN component of R-PST-GCN are crucial for its improved performance. By adapting to the unique characteristics of the nodes, the model can dynamically adjust to the varying traffic conditions, leading to more accurate and responsive predictions. This ability to capture the detailed and fluctuating nature of traffic data is essential for real-time traffic forecasting applications.

Further analysis was conducted using noisy data to evaluate the robustness of R-PST-GCN under different noise conditions. Figure 4.14a presents the prediction results under Gaussian noise with $\sigma = 0.4$, while Figure 4.14b shows the predictions under Poisson noise with $\lambda = 4$. In both cases, R-PST-GCN maintained accurate performance, effectively capturing the traffic flow despite the noise.

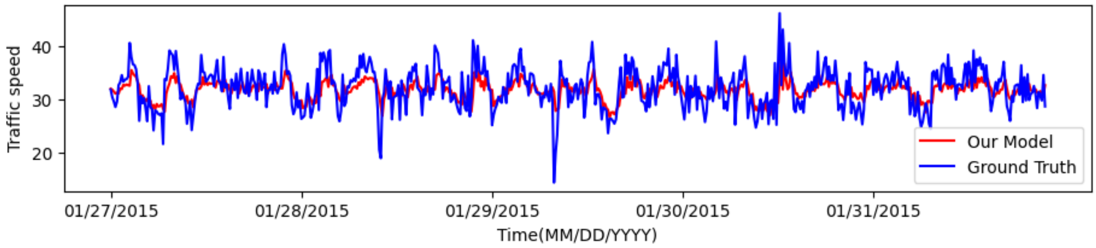
The model's performance was also tested with varying rates of missing data. The predictions with a 10% missing rate are visualized in Figure 4.14c, and those with a 50% missing rate are shown in Figure 4.14d for SZ-taxi dataset. While R-PST-GCN performed well with a 10% missing rate, its performance declined with a 50% missing rate. This decrease in accuracy is attributed to the relatively small size of the SZ-taxi dataset, where the removal of 50% of the data leaves insuffi-



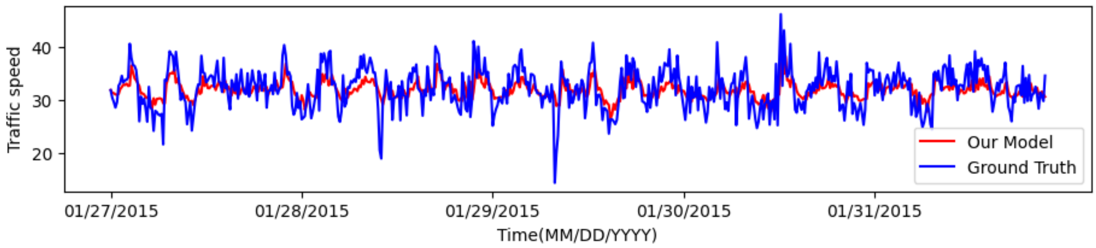

(a) 15 minute



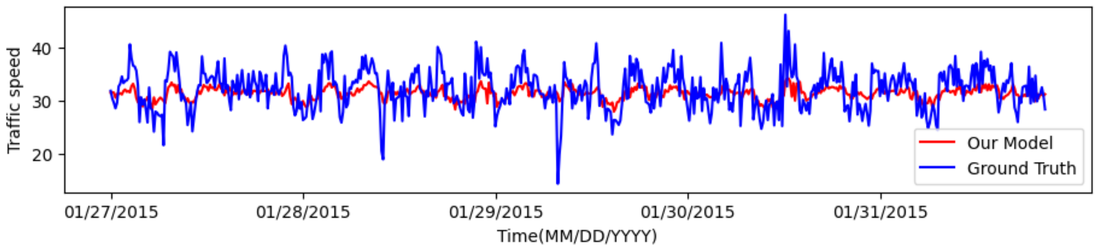

(b) 30 minute




(c) 45 minute

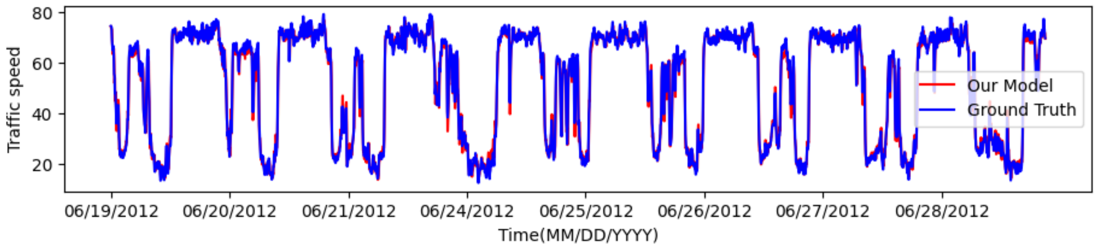



(d) 60 minute

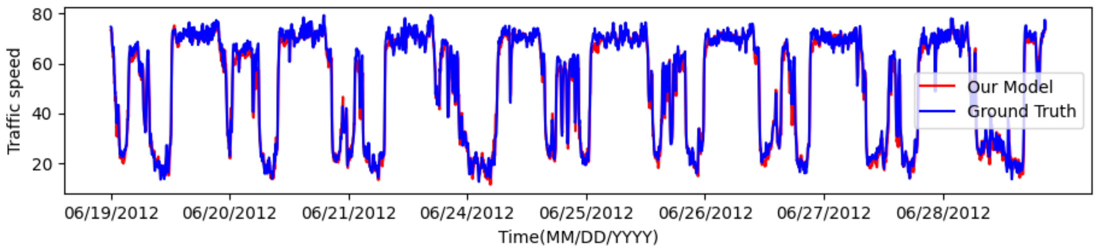



(e) 120minute

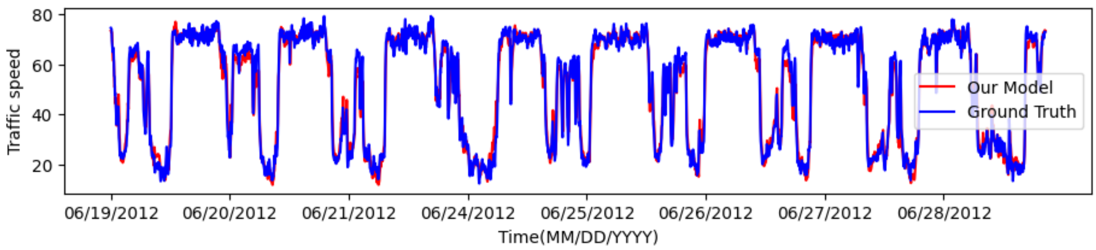
Figure 4.12: The visualization results for prediction horizon of SZ-taxi dataset



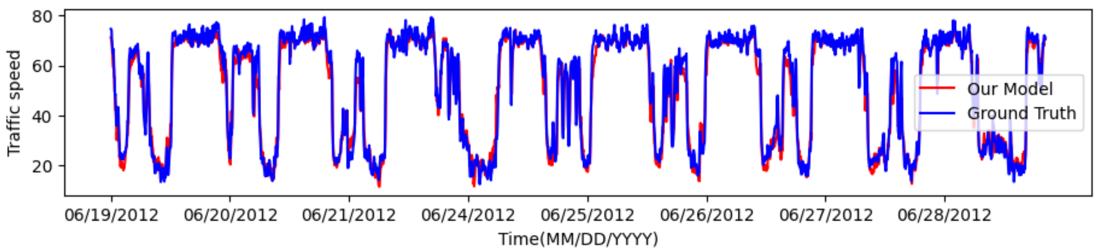

(a) 15minute



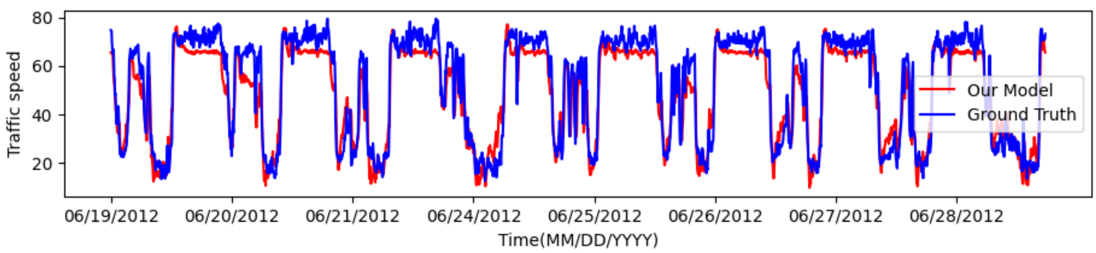

(b) 30 minute




(c) 45 minute



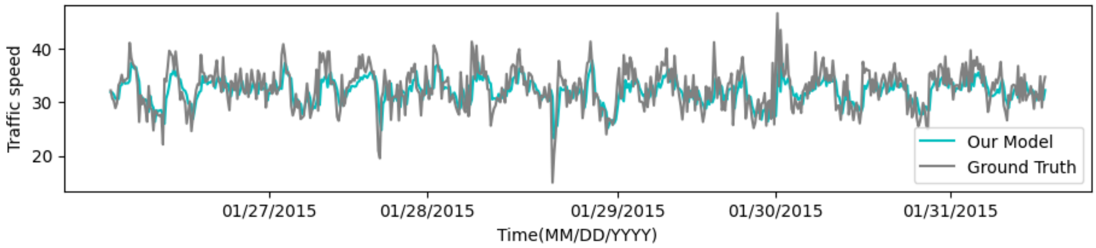

(d) 60 minute



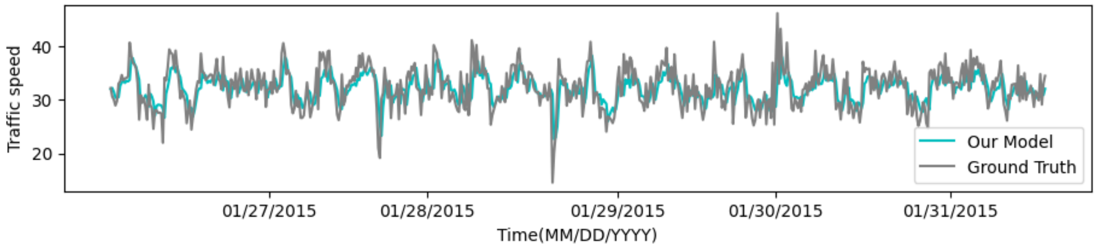

(e) 120 minute

Figure 4.13: The visualization results for prediction horizon of PeMSD7 dataset

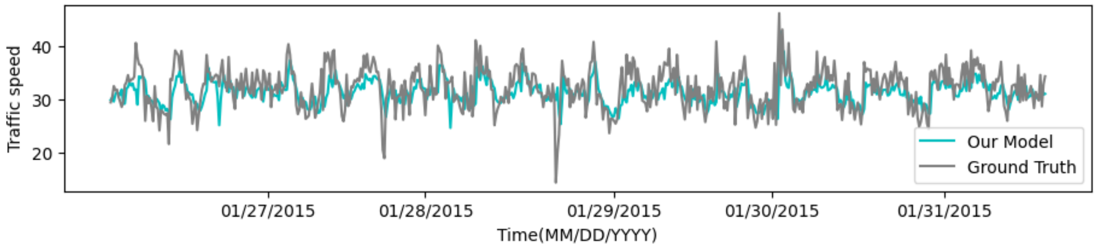
cient information for the model to learn effectively, preventing it from accurately capturing the traffic trends.



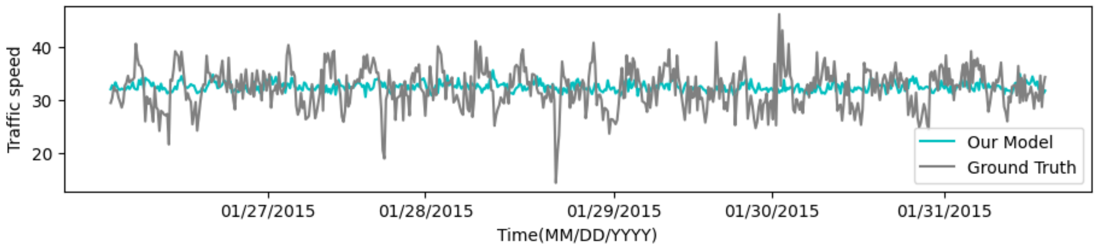

(a) Gaussian noisy perturbation ($\sigma = 0.4$)




(b) Poisson noisy perturbation ($\lambda$=4)




(c) Missing rate 10%

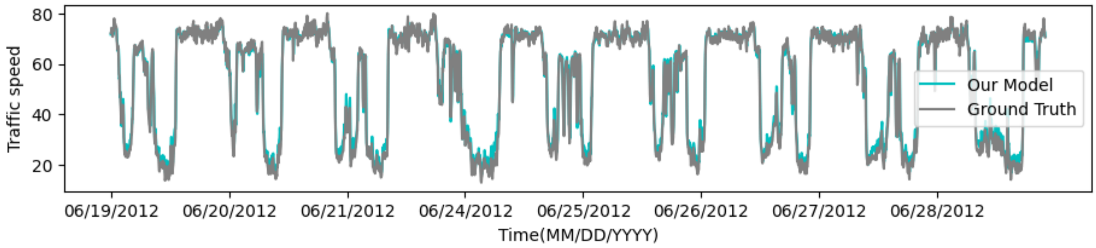



(d) Missing rate 50%

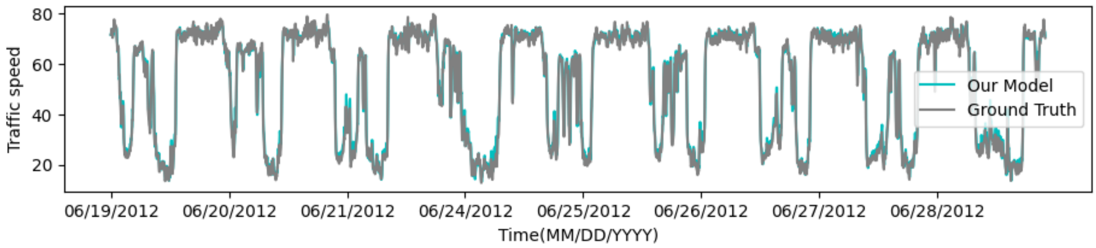
Figure 4.14: Visualization of predictions on the SZ-taxi dataset with noise and missing data.

A similar visualization process was applied to the PeMSD7 dataset, as shown in Figure 4.15. Predictions under Gaussian noise ($\sigma = 0.4$) are visualized in Figure 4.15a, and those under Poisson noise ($\lambda = 4$) are shown in Figure 4.15b.

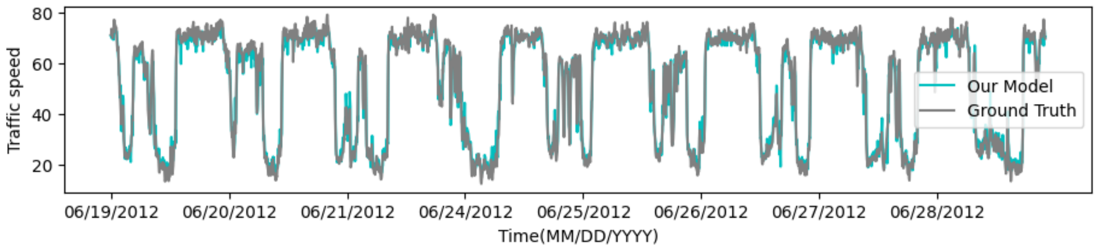
In both scenarios, R-PST-GCN consistently captured the traffic flow accu-



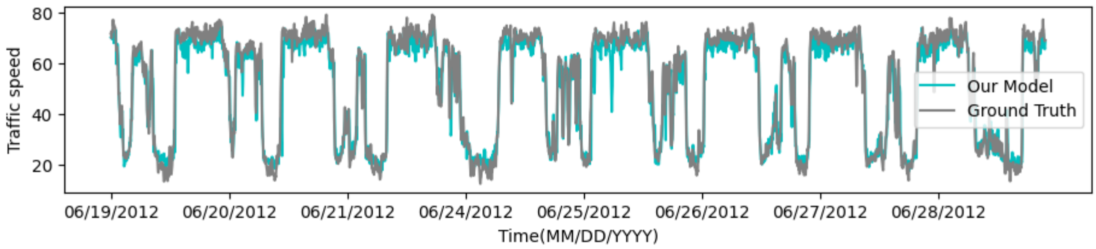

(a) Gaussian noisy perturbation ($\sigma = 0.4$)




(b) Poisson noisy perturbation ($\lambda$=4)




(c) Missing rate 10%




(d) Missing rate 50%

Figure 4.15: Visualization of predictions on the PeMSD7 dataset with noise and missing data.

rately. Performance with a 10% missing rate is depicted in Figure 4.15c, and with a 50% missing rate in Figure 4.15d. Unlike the SZ-taxi dataset, R-PST-GCN maintained its performance even with a 50% missing rate in the PeMSD7 dataset. This is due to the larger size of the PeMSD7 dataset, which ensures that enough data remains available for the model to learn the traffic trends effectively, demonstrating the robustness of R-PST-GCN in handling significant data loss.

Overall, these results confirm the robustness and effectiveness of R-PST-GCN in traffic prediction accurately, even under conditions of noise and missing data. This highlights the model's potential for real-time traffic prediction applications, where data quality and completeness can often be compromised.

## 4.12 Summary

This chapter presents an in-depth simulation setup and comprehensive result analysis to evaluate the performance of the R-PST-GCN model in robust traffic prediction. Several performance metrics are employed, including mean absolute error (MAE), root mean square error (RMSE), accuracy, coefficient of determination ($R^2$), and variance score (var). The experiments are conducted on two distinct datasets representing city road traffic and highway traffic, demonstrating the model's capability for both short-term and long-term prediction tasks.

The results indicate that R-PST-GCN maintains consistent RMSE values across different time horizons, with predictions made for 15-minute, 30-minute, 45-minute, 60-minute, and 120-minute intervals. For the SZ-taxi dataset, RMSE and accuracy remain stable, while the PeMSD7 dataset poses challenges due to its 5-minute intervals, leading to a slight RMSE increase for longer horizons. Nevertheless, R-PST-GCN achieves lower RMSE values compared to other models and offers reduced computational costs, striking a balance between precision and efficiency.

An ablation study further clarifies the importance of the probabilistic adjacency matrix and node-specific learning in managing missing data. The use of a static adjacency matrix results in a substantial RMSE increase, highlighting the importance of the probabilistic adjacency matrix. Removing the node-specific learning module also increases RMSE, though to a lesser extent. This indicates that the probabilistic adjacency matrix is crucial for managing missing data effectively.

R-PST-GCN's ability to accurately capture traffic patterns, including peak periods, is confirmed across both datasets. Traditional GCN models often produce suboptimal predictions due to their smoothing effects, but R-PST-GCN overcomes this limitation through a modified GCN architecture that learns the unique characteristics of nodes and captures dynamic spatial dependencies. This modification ensures high fidelity in reflecting traffic peaks and trends.

The chapter concludes that R-PST-GCN demonstrates robustness and effectiveness in traffic prediction, even under conditions of noise and incomplete data. This combination of accuracy and reduced computational overhead positions R-PST-GCN as a promising model for real-time traffic prediction applications, particularly in scenarios where data quality and completeness are compromised.

# Chapter 5

# Conclusion

Accurate traffic prediction is crucial for the effective functioning of intelligent transportation systems (ITS). It helps in urban traffic planning, traffic management, and traffic control. However, traffic prediction remains a challenging task due to its complex spatiotemporal relationships, sudden incidents, inaccurate/noisy traffic data, and other factors [48]. In recent decades, there has been a shift in prediction techniques owing to advancements in big data and computational tools. As traffic data expands and computational power improves, forecasting methods are transitioning from traditional statistical approaches to deep learning models. Several deep learning-based methods have been proposed in recent years to predict traffic information [6]. Recently, Graph Convolutional Network (GCN) has attracted researchers' attention as it can better represent graph-shaped road networks and extract spatial features of traffic. Many studies employed GCN with GRU [11, 12, 22, 13] or GCN with LSTM [14, 16, 17, 15] to simultaneously capture spatial-temporal features. However, traditional GCN has some drawbacks since it uses a static adjacency matrix which is unable to capture the time-varying features of traffic propagation. Sharing parameters in the model leads to biased results [18]. Moreover, GCN is highly sensitive to disturbances in data [49].In real-world scenarios, data are often corrupted. Therefore, there is a need for a traffic prediction model that can provide accurate predictions while being robust enough to handle data imperfections. In this context, the current thesis investigates the influence of dynamic adjacency matrices and node-specific learning within GCN to address its limitations. Next, it proposes a robust traffic prediction model is proposed that can satisfy the need. This chapter provides a summary of the thesis, highlighting its contributions and achievements. Lastly, it is concluded with possible future research directions.

## 5.1 Robust Probabilistic Spatiotemporal Graph Convolutional Network

This thesis proposes an enhancement to the traditional graph convolutional network by introducing a probabilistic adjacency matrix and node-specific learning, enabling the dynamic spatial correlations extraction from data. Using a probabilistic adjacency matrix using Bayesian inference enhances the model's ability to handle noisy data without requiring extensive preprocessing or imputation. For temporal feature extraction, GRU and attention mechanisms are utilized so that the model can capture both local and global trends of traffic networks. Furthermore, the attention mechanism helps in capturing sudden incidents.

Evaluation of real-world datasets, including city roads and highways, demonstrates the superior performance of our model compared to baseline approaches. The proposed model showcases consistent performance across various time horizons, indicating its reliability regardless of the prediction horizon making it suitable for both short-term and long-term prediction. Experiments show the importance of employing a dynamic adjacency matrix in GCN. The results highlight how the proposed model strikes a balance between prediction accuracy and computational cost. Its simple structure contributes to cost reduction compared to more complex models [37, 22, 39, 18]. This makes our proposed model suitable for real-time traffic prediction as its computational cost is comparatively low and accuracy is high. The inherent resilience of the model is focused on, rather than resorting to sub-optimal methods to address dataset shortcomings, a common approach in many other studies. The proposed model performed well compared to baseline approaches under noisy or missing data. Importantly, the proposed model exhibits consistent performance across varying levels of data imperfections, highlighting its robustness and practical utility in real-world scenarios.

## 5.2 Future Work

In this thesis, a robust traffic prediction model is proposed that can capture the dynamic spatiotemporal dependencies of traffic data. This research can be further explored based on the following directions:

- **Generalization:** This work modified the GCN for the traffic prediction task. However, this approach can be developed as a generalized form of GCN that can also be adopted in other fields of work.

- **Optimization:** Although the training time of the proposed model is lower compared to some advanced deep learning models, it remains relatively high. Future work should explore model compression techniques and efficient inference algorithms to reduce latency, thereby enhancing the model's suitability for real-time applications.

# Bibliography


[1] I. Lana, J. Del Ser, and I. I. Olabarrieta, "Understanding daily mobility patterns in urban road networks using traffic flow analytics," in *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 1157–1162, IEEE, 2016.

[2] M. M. Rahman, A. R. M. Jamil, and N. Nower, "Uncertainty-aware traffic prediction using attention-based deep hybrid network with bayesian inference," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.

[3] D. Rukmana, "Rapid urbanization and the need for sustainable transportation policies in jakarta," in *IOP conference series: earth and environmental science*, vol. 124, p. 012017, IOP Publishing, 2018.

[4] M. M. Rahman and N. Nower, "Attention based deep hybrid networks for traffic flow prediction using google maps data," in *Proceedings of the 2023 8th International Conference on Machine Learning Technologies*, pp. 74–81, 2023.

[5] A. A. Haider, "Traffic jam: The ugly side of dhaka's development," *The Daily Star*, vol. 13, 2018.

[6] H. Yi, H. Jung, and S. Bae, "Deep neural networks for traffic flow prediction," in *2017 IEEE international conference on big data and smart computing (BigComp)*, pp. 328–331, IEEE, 2017.

[7] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[8] Y. Song and J. Lu, "Rnn-based traffic flow prediction for dynamic reversible lane control decision," in *Data Science and Knowledge Engineering for Sensing Decision Support: Proceedings of the 13th International FLINS Conference (FLINS 2018)*, pp. 323–330, World Scientific, 2018.

[9] J. Guo, Z. Wang, and H. Chen, "On-line multi-step prediction of short term traffic flow based on gru neural network," in *Proceedings of the 2nd International Conference on Intelligent Information Processing*, pp. 1–6, 2017.

[10] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with lstm recurrent neural network," in *2017 IEEE 20th international conference on intelligent transportation systems (ITSC)*, pp. 1–6, IEEE, 2017.

[11] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[12] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, "A3t-gcn: Attention temporal graph convolutional network for traffic forecasting," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, p. 485, 2021.

[13] B. Li, Q. Yang, J. Chen, D. Yu, D. Wang, F. Wan, *et al.*, "A dynamic spatio-temporal deep learning model for lane-level traffic prediction," *Journal of Advanced Transportation*, vol. 2023, 2023.

[14] F. Hou, Y. Zhang, X. Fu, L. Jiao, and W. Zheng, "The prediction of multi-step traffic flow based on ast-gcn-lstm," *Journal of Advanced Transportation*, vol. 2021, pp. 1–10, 2021.

[15] Z. Chen, Z. Lu, Q. Chen, H. Zhong, Y. Zhang, J. Xue, and C. Wu, "Spatial–temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism," *Information Sciences*, vol. 611, pp. 522–539, 2022.

[16] J. Gu, Z. Jia, T. Cai, X. Song, and A. Mahmood, "Dynamic correlation adjacency-matrix-based graph neural networks for traffic flow prediction," *Sensors*, vol. 23, no. 6, p. 2897, 2023.

[17] Z. Chen, Z. Lu, Q. Chen, H. Zhong, Y. Zhang, J. Xue, and C. Wu, "Spatial–temporal short-term traffic flow prediction model based on dynamical-learning graph convolution mechanism," *Information Sciences*, vol. 611, pp. 522–539, 2022.

[18] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17804–17815, 2020.

[19] Z. Cui, L. Lin, Z. Pu, and Y. Wang, "Graph markov network for traffic forecasting with missing data," *Transportation Research Part C: Emerging Technologies*, vol. 117, p. 102671, 2020.

[20] Y. Liu, S. Rasouli, M. Wong, T. Feng, and T. Huang, "Rt-gcn: Gaussian-based spatiotemporal graph convolutional network for robust traffic prediction," *Information Fusion*, vol. 102, p. 102078, 2024.

[21] S. Chakraborty, "Traffic congestion in dhaka city and its economic impact," *Dhaka University Journal of Business Studies*, vol. 1, no. 1, pp. 44–68, 2016.

[22] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[23] L. Ge, S. Li, Y. Wang, F. Chang, and K. Wu, "Global spatial-temporal graph convolutional network for urban traffic speed prediction," *Applied Sciences*, vol. 10, no. 4, p. 1509, 2020.

[24] T. Alghamdi, K. Elgazzar, M. Bayoumi, T. Sharaf, and S. Shah, "Forecasting traffic congestion using arima modeling," in *2019 15th international wireless communications & mobile computing conference (IWCMC)*, pp. 1227–1232, IEEE, 2019.

[25] S. Shahriari, M. Ghasri, S. Sisson, and T. Rashidi, "Ensemble of arima: combining parametric and bootstrapping technique for traffic flow prediction," *Transportmetrica A: Transport Science*, vol. 16, no. 3, pp. 1552–1573, 2020.

[26] S. V. Kumar, "Traffic flow prediction using kalman filtering technique," *Procedia Engineering*, vol. 187, pp. 582–587, 2017.

[27] Y. Liu and H. Wu, "Prediction of road traffic congestion based on random forest," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, pp. 361–364, IEEE, 2017.

[28] X. Feng, X. Ling, H. Zheng, Z. Chen, and Y. Xu, "Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2001–2013, 2018.

[29] W. Chen, J. An, R. Li, L. Fu, G. Xie, M. Z. A. Bhuiyan, and K. Li, "A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features," *Future generation computer systems*, vol. 89, pp. 78–88, 2018.

[30] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, "An improved bayesian combination model for short-term traffic prediction with deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1332–1342, 2019.

[31] L. Yang, Q. Yang, Y. Li, and Y. 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)*, pp. 27–30, IEEE, 2019.

[32] Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," *arXiv preprint arXiv:1612.01022*, 2016.

[33] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, "Improved deep hybrid networks for urban traffic flow prediction using trajectory data," *Ieee Access*, vol. 6, pp. 31820–31827, 2018.

[34] L. Liu, J. Zhen, G. Li, G. Zhan, Z. He, B. Du, and L. Lin, "Dynamic spatial-temporal representation learning for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7169–7183, 2020.

[35] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, 2016.

[36] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[37] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, "Ast-gcn: Attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[38] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 922–929, 2019.

[39] A. Roy, K. K. Roy, A. A. Ali, M. A. Amin, and A. M. Rahman, "Unified spatio-temporal modeling for traffic forecasting using graph neural network," in *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2021.

[40] A. Liu, C. Li, W. Yue, and X. Zhou, "Real-time traffic prediction: A novel imputation optimization algorithm with missing data," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, IEEE, 2018.

[41] X. Chen, Z. He, and L. Sun, "A bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transportation research part C: emerging technologies*, vol. 98, pp. 73–84, 2019.

[42] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "Lstm-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[43] J. Dai, W. Zhu, and X. Luo, "A targeted universal attack on graph convolutional network," *arXiv preprint arXiv:2011.14365*, 2020.

[44] Z. Zhou, Z. Yang, Y. Zhang, Y. Huang, H. Chen, and Z. Yu, "A comprehensive study of speed prediction in transportation system: From vehicle to traffic," *Iscience*, vol. 25, no. 3, 2022.

[45] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation research record*, vol. 1748, no. 1, pp. 96–102, 2001.

[46] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.

[47] P. Fafoutellis and E. I. Vlahogianni, "Unlocking the full potential of deep learning in traffic forecasting through road network representations: A critical review," *Data Science for Transportation*, vol. 5, no. 3, pp. 1–21, 2023.

[48] I. Laña, I. I. Olabarrieta, M. Vélez, and J. Del Ser, "On the imputation of missing data for road traffic forecasting: New insights and novel techniques," *Transportation research part C: emerging technologies*, vol. 90, pp. 18–33, 2018.

[49] J. Dai, W. Zhu, and X. Luo, "A targeted universal attack on graph convolutional network," *arXiv preprint arXiv:2011.14365*, 2020.