Network-wide Traffic Feature Learning and Forecasting Under Non-stationary Circumstances Using Advanced Deep Neural Network

Meng-Ju Tsai

A DISSERTATION
SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

UNIVERSITY OF WASHINGTON
MAY 2023

READING COMMITTEE:
YINHAI WANG, CHAIR
EDWARD MCCORMACK
QING SHEN

PROGRAM AUTHORIZED TO OFFER DEGREE:
CIVIL AND ENVIRONMENTAL ENGINEERING

Dissertation advisor: Professor Yinhai Wang, Chair Meng-Ju Tsai

Network-wide Traffic Feature Learning and Forecasting Under Non-stationary Circumstances Using Advanced Deep Neural Network

Abstract

The rapid advancement of intelligent traffic sensing and communication technologies has introduced a new era of transportation data, offering unprecedented opportunities to predict and manage urban traffic. However, traditional statistical and basic machine learning models are often inadequate in forecasting network-wide traffic states, hampered by the time-varying nature of traffic patterns and the complex geographical relationships on road networks. To compound this challenge, unexpected events like the COVID-19 pandemic have drastically altered traffic patterns, making it harder for transportation agencies to learn representative patterns from historical data. The above underscores the critical need for more advanced models that can adapt to changing conditions and deliver reliable predictions.

Building on this pressing issue, the goal of this dissertation is to **develop advanced deep learning models in both methodological and practical ways to improve traffic forecasting accuracy under non-stationary circumstances.** This dissertation aims to accomplish the goal in **six parallel perspectives**. **Firstly**, a model with the capability of capturing patterns from both short- and long-term traffic states should be developed to accommodate unexpected interventions. **Secondly**, a workflow with customized data processing and analysis

components should be designed for extracting other meaningful auxiliary information that could improve the robustness of relative long-term traffic forecasting, such as social media features. These features can then be integrated with traffic data and fed into a model with long-term prediction capability to enhance the robustness and accuracy of network-wide prolonged traffic forecasting. **Thirdly**, a model able to learn new traffic patterns without forgetting previous knowledge under continuously changing traffic conditions should be created to demonstrate how to tackle the Plasticity-stability dilemma, especially under non-stationary traffic conditions. **Fourthly**, a novel unified framework with multi-contrastive learning should be developed to improve the robustness of spatial-temporal traffic forecasting, which has a great potential to effectively handle complex and noisy data and learn fine-grained representations suitable for traffic forecasting. **Fifthly**, a real-time interactive application should be implemented to evaluate live traffic updates and predict future traffic states, enabling drivers to plan their routes more efficiently and reduce congestion on the roads. **Lastly**, a benchmark should be provided for researchers to expedite researchers to uncover more informative patterns from non-stationary data and evaluate the resilience of models in the transportation industry.

This dissertation conducts in-depth research and applications on several key technologies and steps required for building more adaptive and robust architectures. They will address several critical transportation necessities and provide tangible benefits for traffic management and optimization. Specifically, the contributions can be divided into six perspectives: **1) proposing a Multivariate Dual Long Short-term Memory model.** It considers short- and long-term traffic patterns and spatial and temporal features for network-wide traffic forecasting under interference. **2) Learning social media features in a Natural Language Process-**

ing (NLP)-joined social-aware framework to overcome the ignorance of cultural impacts and boost robustness under unexpected interventions in prediction tasks. **3) Designing an incremental learning framework** to solve catastrophic forgetting issues to build a more robust architecture given continuously changing traffic patterns. **4) Developing an innovative unified model with multi-contrastive learning and traffic representation learning** to mitigate the challenges of handling complex and noisy traffic data, enabling improved spatial-temporal traffic forecasting capabilities. From a practical standpoint, the contributions are **5) implementing a real-time traffic performance measuring platform** to assess current traffic conditions and forecast future network-wide traffic states. **6) Releasing benchmarks and a non-stationary traffic dataset** to encourage further research into developing powerful algorithms that can adapt to fluctuating traffic conditions.

In conclusion, our research highlights the need for advanced deep-learning models to improve the accuracy and adaptability of traffic forecasting under non-stationary circumstances. The proposed techniques provide promising solutions to overcome traditional modeling challenges and offer practical applications for real-time traffic management. With continued research and development in this field, we can pave the way for smarter, more efficient, and sustainable urban transportation systems.

Contents

List of Figures

List of Tables

Acronyms

AGCRN Adaptive Graph Convolutional Recurrent Network

AGC-Seq2Seq Attention graph convolutional sequence-to-sequence model

AI Artificial Intelligence

ANN Artificial Neural Networks

ARIMA Autoregressive Integrated Moving Average

ARMA Autoregressive Moving Average

BLM Black Lives Matter

CEEMDAN Complete Ensemble Empirical Mode Decomposition with Adaptive Noise

CNN Convolutional Neural Network

CPG Compacting, Picking, and Growing

CV Computer Vision

DCRNN Diffusion Convolutional Recurrent Neural Network

DL Deep Learning

DLSTM Dual LSTM

DNN Deep Neural Network

DRT Demand-responsive Transit

EWC Elastic Weight Consolidation

GAN Generative Adversarial Network

GCN Graph Convolutional Network

GEM Gradient Episodic Memory

GMAN Graph Multi-Attention Network

GNN Graph Neural Network

GRU Gated Recurrent Units

IIoT Industrial Internet of Things

ITS Intelligent Transportation Systems

KNN K-Nearest Neighbor

LSTM Long Short-term Memory

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

MAS Memory Aware Synapses

MDLSTM Multivariate Dual Long Short-Term Memory

ML Machine Learning

MLSTM Multivariate LSTM

MoCo Momentum Contrast

MSE Mean Squared Error

NCHRP National Cooperative Highway Research Program

NLP Natural Language Processing

OD Origin-destination

PCA Principal Component Analysis

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SBULSTM Stacked Bidirectional and Unidirectional LSTM

Seq2Seq Sequence-to-Sequence

SI Synaptic Intelligence

SVD Singular Value Decomposition

SVM Support Vector Machine

S-LSTM Stacked LSTM

TPS Traffic Performance Score

TRB Transportation Research Board

TTI Travel Time Index

VMT Vehicle Miles Traveled

WSDOT Washington State Department of Transportation

Acknowledgments

On this incredible learning journey, I want to express my appreciation to all the special people who have supported me along the way. First and foremost, I would like to extend a sincere thank you to The Liao Foundation. Their generous support has allowed me to pursue my dream of studying at the University of Washington. I am deeply grateful for their kindness and encouragement.

Special thanks go to Prof. Yinhai Wang, whose leadership and guidance have been invaluable throughout my Ph.D. journey. Prof. Wang's visionary approach, profound expertise, and unwavering support have played an essential role in shaping my research and fostering my academic growth. I am genuinely grateful for the opportunity to work under Prof. Wang's mentorship, and I owe much of my success to their invaluable contributions.

Next, I would like to thank my committee members, Prof. Edward McCormack, Prof. Qing Shen, and Prof. Simon Du. Their invaluable guidance and expertise have been instrumental throughout my Ph.D. journey. I am truly grateful for their presence on my committee and their impact on my work.

I would also like to acknowledge our STAR Lab members. To Dr. Zhiyong Cui, Dr. Ruimin Ke, Dr. Ziyuan Pu, Dr. Meixin Zhu, Dr. Yifan Zhuang, Dr. Wei Sun, Dr. John Ash, Mr. Hubert Chen, Mr. Shuyi Yin, Mr. Hao Yang, Mr. Chenxi Liu, Mr. Cole Kopca, Mr. Bingzhang Wang, Mr. Sam Ricord, Mr. Adam Huang, I extend my sincere gratitude. Your support and collaborative spirit made my research journey rewarding. Working alongside you in the STAR Lab has been an inspiring and fulfilling experience.

I am also grateful for the mentorship of Prof. Jen-Yu Han from the National Taiwan University. His guidance and encouragement have inspired me to push the boundaries of my knowledge and explore new frontiers.

Lastly, I want to express my gratitude to my loving family. To my father, Cheng-Jen Tsai, my mother, Chiung-Yuan Kuo, my brother, Ming-Ju Tsai, my wife, Hui-min Chuang, and my furry companion, Bourbon, your love, support, and the joy you bring have been the cornerstone of my success. Your belief in me has given me the strength to overcome challenges. I am truly blessed to have all of you by my side.

Reflecting on this transformative journey, I am filled with pride and gratitude. The road hasn't always been easy, but with the support of these exceptional individuals, I have achieved milestones and grown in ways I never thought possible. Thank you all for being an integral part of my journey!

Part I

Foundations and Context

1

Introduction

1.1 Background

1.1.1 Understanding Intelligent Transportation System for Urban Mobility

Urban mobility is essential for maintaining the efficiency of metro regions, promoting population growth by enabling citizens to access services in a timely and efficient manner, including housing and commuting[107]. According to a report from the University of Michigan's Center for Sustainable Systems[25], it is projected that 89% of Americans will reside in urban areas by 2050. As the population grows, significant challenges arise for public urban transportation networks, such as traffic congestion and increased demand for public transportation[120,135]. These issues significantly impact traditional urban mobility and may drive the adoption of new forms of transportation. Several flexible advances in urban mobility, such as Demand-responsive Transit (DRT)[2,44,132,183,207], have the potential to transform conventional urban mobility by delivering on-demand transport services. This trend highlights the need for continuous upgrades of urban mobility to keep pace with population growth. The World Bank Group, therefore, asserts that *"Urban mobility is no longer just about moving people around by motorized vehicles. What people really need is accessibility to various urban services."*[53]

As the transportation industry grapples with these complex challenges, there has been an increasing focus on strengthening transportation infrastructure. In the past decade, governments around the world have shown strong interest in researching and investing in solutions that enhance traffic management and benefit both private and public transportation[73,19,148]. As one of the most critical applications of the Industrial Internet of Things (IIoT), Intelligent Transportation Systems (ITS) has emerged as a solution for reducing urban mobility issues and improving the operation of urban transit from an information standpoint[171,179,136]. In fact, the concept of ITS is not new, but it has gained increased popularity in recent years. Implementing ITS has become essential as the transportation industry faces more severe challenges, and for a good reason – it delivers undeniable benefits to both transport operators and passengers, including efficiency and safety[172].

The remarkable increase in the volume and diversity of transportation data available through ITS can be attributed to the prompt adoption of modern traffic sensing and communication technologies. This proliferation of newly collected data has the potential to advance research and applications related to urban transportation and smart cities, including traffic control, autonomous driving, and smart city infrastructure. Traffic forecasting, as a crucial aspect of ITS [18], has also take the advantage of newly collected data and with the potential to improve roadway capacity and alleviate congestion [32]. With real-time information on traffic conditions, drivers can avoid getting stuck in traffic and estimate the time savings that could be achieved by taking alternative routes. Many road operating agencies have recognized this situation and have actively engaged in this field. For instance, the Washington State Department of Transportation (WSDOT) releases holiday travel forecasts, including the best and worst times to travel, allowing drivers to plan ahead and avoid severe congestion [205]. Additionally, the National Cooperative Highway Research Program (NCHRP) has funded projects that utilize data gathered by ITS to enhance the precision, dependability, and usefulness of traffic predictions for highway planning [49]. As exemplified by such practices, traffic forecasting plays an essential role in ITS and motivates transportation agencies to adopt sustainable policies in building smarter cities.

The next question is how to effectively analyze transportation big data and utilize computing resources. According to the usage of computing resources over time, as depicted in Figure 1.1 presented by *OpenAI*, a non-profit Artificial Intelligence (AI) research company, two distinct phases in the progress of AI systems can be identified [9]: the First Era and the Modern Era. The modern Era, beginning in 2012, demonstrates that using computing power significantly outperforms macro trends of the past half-century. The history of investment in AI computing resources is described as a period of rapid growth, during which the number of researchers participating in this field has significantly increased. A key challenge now is to effectively consume the large quantities of data collected by the widespread deployment of traffic sensors and utilize them in powerful computing machines to learn traffic patterns.

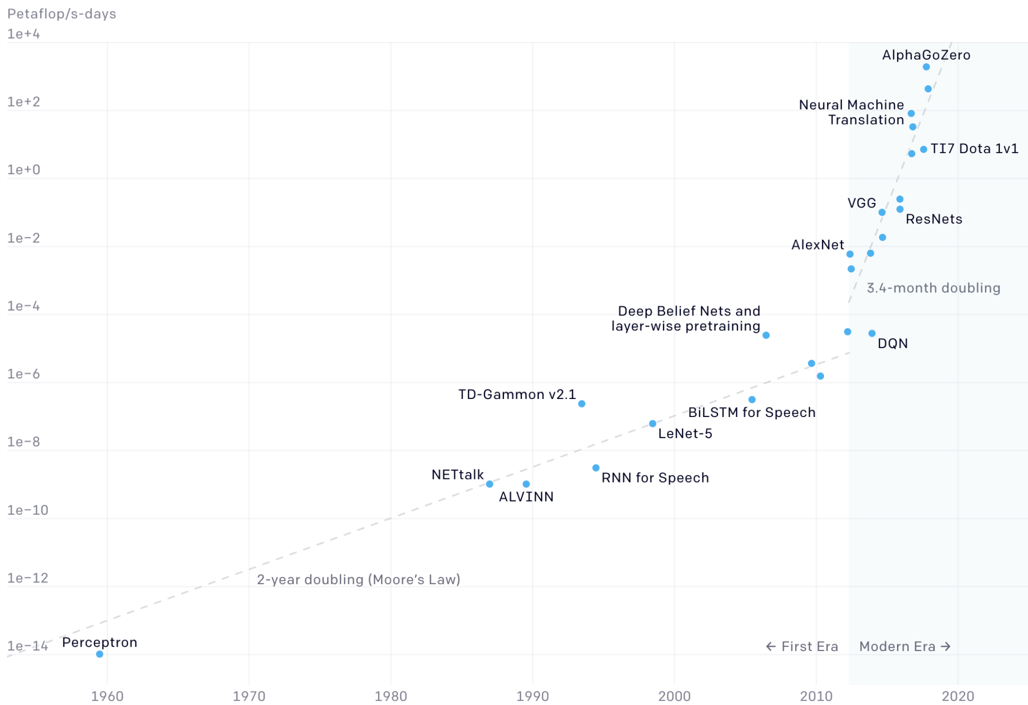


Figure 1.1: Two Distinct Eras Of Compute Usage In Training AI Systems [9]

1.1.2 Data-driven Approaches of Traffic Forecasting

Data-driven traffic forecasting, which is essential for planning and logistics, has long been a popular topic within the transportation community [142,235,216]. Given the large volume of transportation data and the rapid increase in computing capabilities in recent years, several unique data-driven approaches have been developed to address this real-world matter. According to numerous reviews of data-driven traffic forecasting[123,201,100], these algorithms can be classified into two categories: classical techniques and Artificial Neural Networks (ANN) viewpoints.

Classical techniques for traffic forecasting include models based on traffic process theory, such as state space modeling[251,150], Kalman filter modeling[250], and Origin-destination (OD) flow estimation[1]. These techniques aim to model different components of transportation data, including seasonal variations, explanatory variables, and interventions, and incorporate them into a traffic prediction model. For example, in the state space model[175], the state space vector is composed of a linearly independent collection of linear combinations from the past that is associated with future traffic states[99,168]. OD-based approaches[46] attempt to estimate traffic demand from target regions by using optimization techniques to minimize the difference between measured and estimated OD link flows. In addition to these techniques, the Autoregressive Integrated Moving Average (ARIMA) model class[105,155,208] is also commonly used in traffic forecasting. These models capture a variety of common temporal features in time series data and employ differencing approach[6], which is a widely used data transform method for making time series data stationary, to stabilize the mean of the traffic time series data by removing changes in the level of the time series and eliminating trend and seasonality. To anticipate future traffic conditions, models[209] employ auto correlations and moving averages over residual errors in the data.

While classical algorithms have been widely used for short-term traffic forecasting due to their high accuracy and relatively low computational requirements, these algorithms have particular limitations that may impact their effectiveness in real-world scenarios. For example, they often generate a large number of hypotheses and impose strict constraints on model development, which can limit their flexibility and adaptability[201]. In addition, these algorithms may be prone to failure when the forecasting data is incomplete or partially missing, which is a common occurrence in traffic forecasting[108,240]. To address the limitations of classical algorithms in traffic flow forecasting, Machine Learning (ML)-based models have been proposed as alternative approaches. These models, such as Bayesian-based models[178,56] and tree-based models[102,238], are designed to account for the complex and non-linear nature of traffic patterns and assess traffic flows between upstream and downstream road connectivity. Additionally, ML-based models have the ability to handle missing or incomplete data, making them more robust and reliable in real-world scenarios.

One trendy class of ML-based models for traffic forecasting is ANN-based algorithms, also known as Deep Learning (DL)-based algorithms. These algorithms build artificial neural networks by layering algorithms and processing neurons, and have been extensively used in the transportation field in recent decades due to their ability to effectively capture and analyze temporal patterns in sequential data[97,231,149]. Recurrent Neural Network (RNN)s, a subgroup of DL-based approaches, are particularly well-known for their ability to analyze sequential tasks, such as traffic forecasting, using internal memory units[37,162,110]. However, RNNs may struggle with the issue of vanishing gradients, which can make learning from lengthy data sequences difficult[74,153]. To address this problem, Long Short-term Memory (LSTM) models have been proposed[75,245]. These models have inbuilt gate mechanisms that can control the flow of information and determine which data in a sequence should be kept or ignored to overcome the gradient vanishing issue. As a result, many studies in the past decade have used a wide variety of LSTM-based techniques, including Gated Recurrent Units (GRU)[34], to learn a time-dependent sequence with the loop structure[124,39]. Overall, the use of ANN-based algorithms, particularly LSTM-based techniques, has emerged as a promising approach for improving the accuracy of traffic flow forecasting in real-world scenarios.

In addition to ANN-based algorithms, the ability of Convolutional Neural Network (CNN) to effectively extract meaningful representations and learn temporal patterns has garnered significant interest in the transportation field for capturing spatial dependencies in traffic networks[122]. In order to understand the spatial-temporal dependence for traffic flow forecasting, some researchers have applied CNNs to learn traffic patterns as 2D spatial images and stacked them with LSTM components to learn temporal information separately[213,15]. Some studies have even identified traffic networks as graphs with a combination of nodes and edges to extract influential patterns from roadway connectivity with various topological structures[89]. A significant number of Graph Neural Network (GNN)-based algorithms, including the widely developed Graph Convolutional Network (GCN) model, have been designed to achieve improved forecasting performance[252,224]. However, the question arises: how can longer sequences of traffic data be effectively processed while maintaining outstanding performance?

To address this challenge, the concept of attention mechanism has been introduced to the transportation field to tackle traffic forecasting tasks with complex and long sequences of time-series data[20,212]. The attention mechanism, first introduced in the Transformer architecture by *Google* in 2017[196], utilizes self-attention to help memorize long sequences and achieve more outstanding performance. Attention mechanisms allow for dependencies between source and target sequences to be established beyond the in-between distance, by constructing a representation from the current vector and the entire source input, rather than just the most recent hidden state. This idea has been particularly successful in the field of Natural Language Processing (NLP) and has the potential to bring similar benefits to transportation.

Despite the success of diverse approaches in traffic forecasting, the COVID-19 pandemic and other unforeseen events have significantly impacted existing methods unexpectedly[41,188]. The instability of social circumstances has made it even more challenging to reflect current traffic patterns accurately. In order to accommodate these fluctuating patterns, we have both designed advanced Deep Neural Network (DNN) models and incorporated meaningful auxiliary features to improve the robustness of our models in this research. Additionally, this dissertation investigates representation learning through incremental strategies and self-supervised learning to increase the generalizability of our proposed algorithms for traffic forecasting tasks.

1.2 Challenges

According to recent surveys on traffic forecasting[143,100,129], it has been observed that DNN techniques have been widely utilized in the field of traffic forecasting for the past decade, as demonstrated in Figure **??**. While a significant amount of research has sought to apply novel DL-based strategies to this task, there remain substantial challenges in the analysis of non-stationary traffic data and the implementation of approaches for real-world scenarios[94]. This dissertation aims to address these challenges from three main perspectives: (1) data; (2) model; and (3) application.

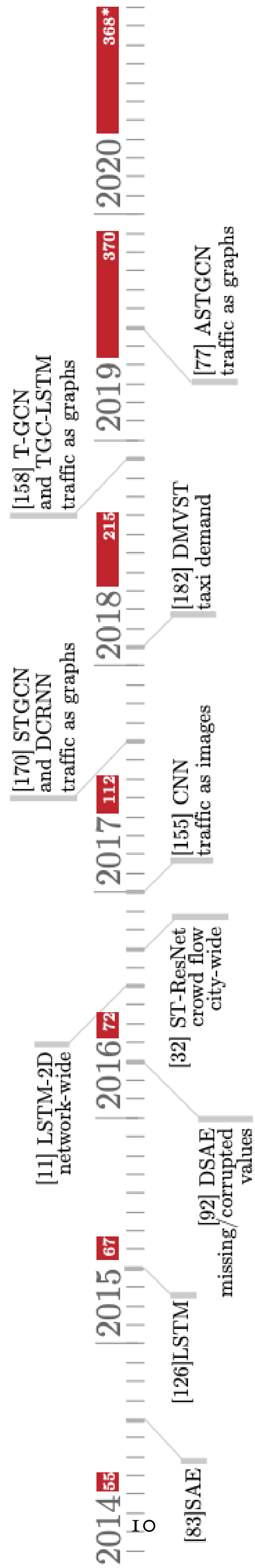


Figure 1.2: Milestones In The History Of DL-Based Short-Term Traffic Forecasting: Publications Ordered According To Their Publication Date. Horizontal Red Bars Denote The Number Of Works Published Every Year Concerning This Topic [129]

1.2.1 Data: Traffic States Under Unanticipated Events Happened

The COVID-19 pandemic has necessitated the implementation of work-from-home policies and travel restrictions to curb the spread of the virus. While these measures have effectively curtailed transmission, they have also significantly disrupted traditional traffic patterns, resulting in significant reductions in travel time and Vehicle Miles Traveled (VMT) as depicted in Figure 1.3 [41]. In response, numerous studies have been conducted to investigate the impact of COVID-19 on transportation. Organizations such as *INRIX* [84], *TomTom*, *Google* [58], and *Mapbox* [72] have also provided data and tools for COVID-19-related research. The majority of this research has focused on fluctuating traffic volume or VMT. Daniel [134] demonstrates how non-stationary traffic data can be used to inform policy decisions by visualizing residual mobility alongside new cases of COVID-19, based on data from TomTom. This work suggests that traffic data can serve as a helpful policy measure due to its temporal and spatial attributes. Another report [130] examining the impact of the COVID-19 lockdown on mobility shows that the top ten cities with the highest traffic reductions worldwide all experienced reductions of over 80%. Shi & Fang [169] also examined the time-lagged effect between outbound traffic from Wuhan, China, and the status of the COVID-19 pandemic.

In addition to COVID-19, the Black Lives Matter (BLM) events also had a significant impact on traffic patterns [10]. Figure 1.4 [188] demonstrates that the traffic pattern on June 3^{rd}, 2020 (Wednesday) in Downtown Seattle was significantly different from the same weekday in the two subsequent weeks (June 10th, 2020 and June 17th, 2020). This deviation can be attributed to BLM and Defund Seattle Police rallies that took place in the Downtown and Capitol Hill neighborhoods of Seattle on June 3^{rd}, 2020. Similar situations were observed in various regions across the United States, including in New York City, as reported by *CBS*

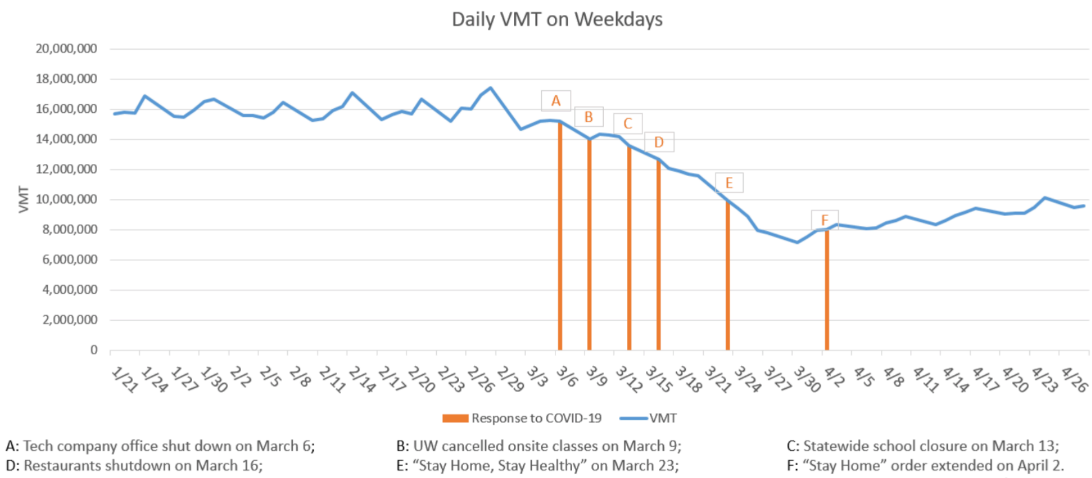


Figure 1.3: Daily VMT Changes On Weekdays[41]

News[24]. The implementation of additional road traffic controls in and around these neighborhoods likely contributed to the observed differences in traffic patterns.

To sum up, the unexpected events mentioned above have presented challenges from different perspectives for transportation planners and agencies due to the resulting fluctuations in traffic patterns. In such a dynamic environment, it can be difficult for transportation agen-

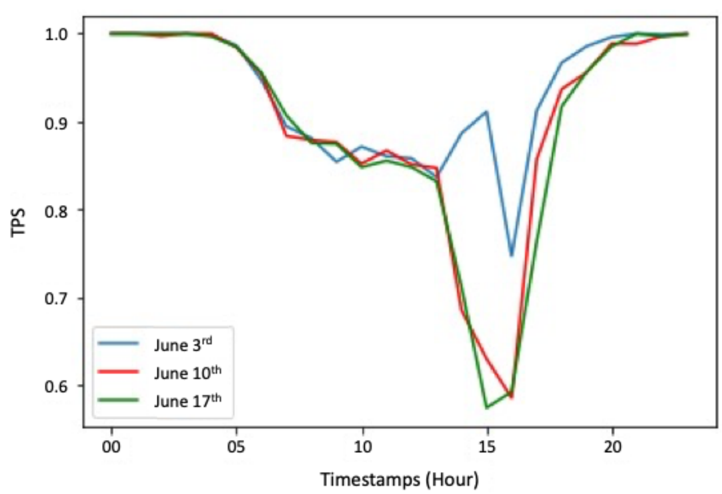


Figure 1.4: Difference Of Traffic Performance At Downtown Seattle In Three Weeks[188]

cies to learn representative traffic patterns from short-term historical data. The constantly changing circumstances also make it challenging to accurately forecast traffic conditions.

1.2.2 Model: Hurdles of Existing Traffic Forecasting Approaches

There are several apparent hurdles in the existing traffic forecasting modeling. First, how to extract meaningful representation from fluctuating traffic data? Existing studies applied a variety of LSTM-based models for traffic forecasting tasks[221,119]. However, most of the algorithms were applied to relatively stable conditions without considering dramatically unexpected events in the real world that can directly affect the statistical distribution of the experimental data. Even though AI techniques with the capability of learning complicated data structures, it is difficult for them to perform consistently under unstable circumstances if they only rely on short-term historical trends[176]. Furthermore, traffic forecasting tasks are often recognized as supervised learning questions by training models to fulfill a specific target, such as minimizing errors. For better pattern extraction, representation learning has shown outstanding performance in Computer Vision (CV)[160,57,71] and NLP[233,78,161], which can be combined with a following fine-tuning procedure to construct a promising paradigm for acquiring more representative information.

Second, existing DNN algorithms have been responsible for a number of significant breakthroughs in a wide variety of sectors. Yet, they have still afflicted with *Catastrophic Forgetting* concerns[93]. During the fine-tuning, the parameters of the trained neurons will be altered to meet the present target, which is to reduce the loss of the designed loss function. This process may lead to a different gradient direction, which is a critical reason for catastrophic forgetting[219]. Therefore, the idea of the ”Stability-Plasticity Trade-off” is arisen to strike a compromise between the adaptability of accumulating new information and the steadiness of integrating what models have previously learned[43]. Incremental learning algorithms have recently become one of the most prominent approaches to the stability-plasticity trade-off dilemma. However, most of them focused more on handwriting recognition[145] and other CV topics[47,117]. There is still a significant knowledge gap when it comes to applying incremental learning algorithms to regression problems, such as traffic forecasting tasks in the transportation field.

Third, the majority of the existing research on traffic forecasting concentrates on short-term forecasting[104]. The forecast periods are restricted to one hour or below[143]. Even though short-term traffic forecasting is essential for urban traffic signal control, relatively long-term traffic forecasting can be advantageous to the general public for trip planning in advance. It may also support the government or agencies in decision-making to further focus funding on particular roads and intersections to improve traffic conditions. However, a significant challenge for long-term forecasting is enhancing its ability to learn and adapt to meet the increasing demand for longer sequences. To fill the gap in accurate long-term traffic forecasting, we should investigate those models designed to represent long-range dependence accurately[211,249]. Furthermore, other valid auxiliary data, such as social media data, is worth exploring to extract valuable semantic information to enhance performance in long-term forecasting[156,51].

1.2.3 Application: Limitation of Fixed Online Prediction Workflow

Existing research or applications on traffic prediction focus more on building intricate DNN structures to increase the prediction accuracy, despite the fact that such improvements are, in most instances, relatively modest. However, in real-world scenarios, the traffic pattern may undergo significant changes over time, and a static model may not be able to account for various traffic patterns[186]. The traffic pattern may switch up once a month or even weekly, especially during the COVID-19 epidemic. Only a limited amount of research has tried to provide solutions to the challenge of the online traffic prediction problem. Few proposed models have been implemented and tested in real-world settings. Therefore, instead of designing a complicated DNN model, an online learning strategy is worth exploring to periodically update the prediction model to react to severe changes in traffic patterns and even obtain better prediction results.

1.3 Research Objectives

Motivated by the urgent need for robust algorithms for traffic forecasting in ITS and the above-mentioned challenges from the existing DL implementation, this dissertation aims to address this problem by focusing on learning traffic patterns while accounting for interventions, as well as handling noisy and incomplete data. Through the design of end-to-end prediction frameworks, which train complex learning systems represented by a single model, and representation learning approaches, as well as the construction of online learning strategies, this work focuses on developing algorithms that can effectively handle fluctuating traffic conditions practically and comprehensively. By achieving these objectives, this dissertation has the potential to not only provide advanced DNN models for network-wide traffic forecasting, but also to contribute valuable datasets and interactive platforms that could inspire further research into the development of robust algorithms capable of adapting to changing traffic conditions.

In summary, the research objectives of this dissertation can be divided into three perspectives:

a) Develop robust DL methods to extract non-stationary characteristics from network-wide spatial-temporal traffic patterns and perform promising accuracy in both short and long-term forecasting tasks.

b) Design representation learning-based approaches to accommodate continuously changing traffic patterns without catastrophic forgetting and extract more generalized representations from traffic networks using multi-contrastive learning techniques.

c) Release a well-organized open-source traffic dataset covering the pandemic period. It could become a standardized benchmark to evaluate model robustness in the transportation field. Besides, a real-time traffic performance measuring tool for assessing existing traffic forecasting models and predicting future traffic states is offered, which puts the network-wide traffic forecasting models into practice.

1.4 Dissertation Organization

This dissertation concentrates on advanced DNN algorithms and application for accommodating network-wide non-stationary traffic patterns. Existing approaches and challenges are comprehensively summarized, which brings up opportunities of designing robust and generalized models to learn fluctuating time-series traffic data. The contributions and the dissertation organization for the following chapters are shown in Figure 1.5.

Part 1: Foundations and Context.

Figure 1.5: Dissertation Organization And Contribution

The first part of this dissertation provides the foundations and context for this dissertation. **Chapter 1** sets the beginning of providing background information, identifying existing challenges, and presenting the objectives of this dissertation. **Chapter 2** surveys the state of the art in traffic forecasting by examining classical and deep learning methods, non-stationary traffic flow learning, and data pattern extraction with representation learning.

Part 2: Prediction with End-to-end Frameworks. The second part of this dissertation presents two novel end-to-end prediction frameworks for handling non-stationary traffic patterns during unexpected events, such as the COVID-19 pandemic. The term End-to-end Structure encompasses the concept of training a complex learning system using a single model. The implementation of work-from-home policies and travel restrictions during the COVID-19 pandemic has effectively slowed the spread of the virus. Still, it has also caused significant disruptions to traditional traffic patterns regarding reductions in travel time and vehicle miles traveled. These fluctuations in traffic patterns have posed significant challenges for transportation agencies and planners, as it can be difficult to accurately forecast traffic patterns using short-term historical data solely in such a dynamic environment. In response, we propose a Multivariate Dual Long Short-Term Memory (MDLSTM) model for network-wide traffic forecasting under interference in **Chapter 3**. This model is designed to overcome the limitations of traditional forecasting approaches and effectively learn non-stationary traffic patterns in both short- and long-term scenarios.

In addition, existing work on traffic forecasting has focused mainly on short-term prediction (e.g., under 1 hour) and has primarily relied on historical traffic patterns. To expand the forecasting horizon and provide a more comprehensive approach to traffic forecasting, in **Chapter 4**, we present Traffic-Twitter Transformer, a flexible framework for predicting physical-aware, long-term traffic conditions for network users and transportation agencies. This framework employs a unique combination of traffic and Twitter representations to enhance forecasting performance and improve robustness. To validate the effectiveness of this approach, we conducted a detailed correlation study to evaluate the relationship between these two time-series data sets. The study demonstrates the significant impact of combining these datasets on forecasting performance, ultimately resulting in more accurate predictions of long-term traffic conditions.

Part 3: Prediction with Representation Learning. The third part of this dissertation aims to explore a new perspective on forecasting by using representation learning to acquire more generic characteristics in traffic patterns. In **Chapter 5**, we address two key issues facing transportation agencies: (1) the challenge of learning representative traffic patterns given constantly changing traffic conditions, and (2) the need to determine *when* and *how* to update the forecasting model to learn new patterns without forgetting previous tasks. To address these issues, we propose an incremental learning-based framework for non-stationary data clustering and forecasting in transportation scenarios. This framework has the potential to assist government agencies and the general public in developing long-term policies and strategies for addressing changing traffic conditions.

Besides, to overcome the challenges posed by complex and noisy traffic data and learn fine-grained representations suitable for traffic forecasting, we propose a novel unified framework for multi-contrastive learning in spatial-temporal traffic forecasting in **Chapter 6**. The proposed framework leverages multi-scale contextual information at different granularities to enhance the robustness and generalizability of traffic pattern learning. Specifically, the framework is designed to handle noisy and incomplete data, adapt to non-stationary conditions, and learn more generalized representations. Our approach enhances the accuracy and robustness of traffic forecasting models.

Part 4: Applications and Open-source Dataset. The fourth part of this dissertation focuses on the practical perspective of traffic forecasting. In **Chapter 7,** we present an interactive platform that incorporates multiple parameters for measuring traffic states in both urban and freeway network-wide traffic conditions, referred to as Traffic Performance Score (TPS). To address the challenges of network-wide online traffic prediction under fluctuating patterns, we propose a multi-step Sequence-to-Sequence (Seq2Seq)-based model with an online training and updating strategy for predicting network-wide traffic performance in real-time, similar to a weather forecast.

Moreover, we release the *TRBAI Open Data Challenge Platform* in **Chapter 8** with an open-source non-stationary traffic dataset and provide tutorials/benchmarks to guide researchers in the data cleaning, processing, and modeling process. This platform serves as an opportunity to encourage researchers to extract more informative patterns from fluctuated data and establish a benchmark for evaluating the resilience of models in the transportation industry.

Part 5: Final Remarks.

The last part of this dissertation summarizes the research contributions and practical applications presented in the preceding chapters, which focus on network-wide traffic feature learning and forecasting under non-stationary circumstances. We also highlight the importance of representation learning techniques in improving the generalizability and adaptability of traffic forecasting. Furthermore, we suggest potential future directions for research in this field.

2

Literature Review

2.1 Overview

Traffic-related issues, such as congestion, are common in most major metropolitan areas worldwide. Due to the harmful effects of traffic congestion on society, the economy, and the environment, government and transportation agencies have pursued solutions for over half a century. Some potential solutions have been raised to address traffic congestion: promoting alternative forms of transportation [12], expanding existing infrastructure [193], and controlling traffic flows [139]. However, promoting alternate modes of transportation is essentially a public policy issue. The expansion of the current infrastructure is constrained mainly by financial and topographical factors. In contrast, traffic flow management has continuously improved over the last few decades, credited to the great amount of transportation data supplied by Intelligent Transportation Systems (ITS) sensors in infrastructure and vehicles and the rapid development of the advanced technology necessary to harness that big data [152,227]. Benefiting from the increasing amount of transportation data, researchers pay more attention to modeling [97], monitoring [210], and analyzing traffic flow and occupancy [87], which facilitates the design of reliable traffic management systems to better control current traffic states and forecast further conditions.

Since the mid-1980s, data-driven traffic forecasting has started to attract researchers' attention [14,140]. The interest of the scientific community in this field, as well as the availability of data, analytical methods, and computing power, have all skyrocketed since then. The initial efforts at traffic flow prediction were time-series approaches using various classical methodologies, including the Box-Jenkins model [147] and early investigations of Kalman filtering methods [28], for short-term horizon prediction. Generally, traffic forecasting methods can be divided into two families: classical methods and Deep Learning (DL)-based methods [123]. Section 2.2 introduces classical traffic forecasting methods, including parametric and non-parametric methods for modeling the nonlinear nature of traffic flow [154]. Section 2.3 reviews existing DL-based traffic forecasting methods, which are summarized by temporal-based only and spatial temporal-based categories. To the best of our knowledge, the prediction horizon in most of the research has been kept in a short-term perspective, i.e., under one hour. Section 2.4 summarizes the existing short-term traffic forecasting works, pointing out the lack of flexible long-term traffic forecasting approaches.

Meanwhile, most algorithms are experimented with stable conditions, ignoring that extreme, unanticipated occurrences might significantly impact the statistical distribution of experimental data. In this case, Section 2.5 concerns the limited current works that take the non-stationary traffic conditions into account to anticipate future traffic network's states. Several representation learning efforts will be evaluated with the aim of gleaning more reusable information from traffic networks in order to overcome non-stationary circumstances. Both incremental learning and self-supervised learning algorithms, which are well-known research domains in representation learning in Computer Vision (CV) and Natural Language Processing (NLP) fields, are discussed in Section 2.6. This section also highlights that advanced transportation research might keep up the pace to adopt these methodologies to tackle traffic-related issues more robustly and efficiently.

2.2 Classical Methods for Traffic Forecasting

Classical traffic forecasting models are mainly statistical approaches developed based on a predefined model structure with theoretical assumptions and the parameters are calibrated using historical data. They can generally be classified into two categories: parametric and non-parametric methods[174]. The former consists of Kalman filtering methods[151], smoothing technique[111,27], linear and nonlinear regression, historical average algorithms, and autoregressive linear processes. Autoregressive Integrated Moving Average (ARIMA), among all of them, is the most well-known parametric forecasting model[105,209,234] with convinced capability in accuracy and requires less computational effort[128]. Numerous variations of ARIMA studies were conducted to enhance forecast precision, such as Autoregressive Moving Average (ARMA)[52], ARIMAX[35] (an ARIMA model with explanatory variables), KARIMA[194] (combined a Kohonen self-organizing map as an initial classifier with ARIMA), switching ARIMA model for describing the change in evolving traffic patterns, and ARIMA-GARCH[29] (combined linear ARIMA model with nonlinear GARCH model to capture both the conditional mean and conditional heteroscedasticity of traffic flow series). Even though the above-mentioned approaches perform reasonably well during normal operating conditions, they cannot respond well to traffic flow patterns with a significant seasonal pattern during peak and off-peak durations. Seasonal-based models, ARIMA fitted with seasonal components (SARIMA)[182] and Additive Seasonal Vector ARMA (A-SVARMA)[126], was proposed to model this traffic flow behavior. However, it is inevitable for them to make many hypotheses and set restrictions on developing parametric models. Moreover, many parametric methods for traffic forecasting were categorized as univariate models[201]. That is, these methods only relied on one time-dependent feature to predict future values. Nonlinear traffic data and the transportation network are too complicated for univariate parametric models to compute.

In order to deal with more complex forecasting scenarios, non-parametric models have been proposed as an alternative approach that does not require setting prior assumptions. These models can adapt to the nonlinear characteristics of traffic information, making them a suitable solution for forecasting tasks. Yu et al.[225] introduced a model using K-Nearest Neighbor (KNN) algorithm and also took spatial information, upstream and downstream road-link connectivity, into account for predicting short-term traffic conditions in an experiment site with six road links. Besides, Castro-Neto et al.[23] considered multiple atypical traffic conditions as features and integrated them into Support Vector Machine (SVM)-based models to predict short-term freeway traffic flows. Bayesian network-related models[178,56], which can statistically account for the causality between random variables, were also tested to capture the cause information for traffic flow predictions, even when the data was incomplete. To incorporate multiple sources of data, the ensemble methods of capitalizing on the benefits of tree-based models were investigated to both reduce the prediction errors and increase the robustness[158,4]. Overall, the non-parametric methods have achieved tremendous success in many aspects of traffic forecasting. However, most of them were proposed to tackle relatively less complicated traffic networks or even a simple corridor given a small amount of traffic data[91]. Furthermore, most of the aforementioned models do not adequately employ many-to-many predictions. That is, they are unsuitable for network-wide traffic state forecasting because these methods cannot process high-dimensional features and model complex spatial-temporal dependency. Therefore, many researchers have moved their attention to DL-based approaches for achieving more desired results.

2.3 Deep Learning Based Methods for Traffic Forecasting

Through advancements in methods and increased accessibility of traffic data, DL models have shown competitiveness in traffic forecasting tasks, outperforming traditional methods such as statistical and univariate models[54,21,33]. DL models are more suited to dealing with complex traffic situations since they do not need prior assumptions or additional feature engineering. Many of the models offered today are variants of Recurrent Neural Network (RNN), Long Short-term Memory (LSTM), Convolutional Neural Network (CNN), and Graph Neural Network (GNN) to capture nonlinear traffic patterns and predict future states from various angles. Based on the designed architectures and the involved data sources, the

DL-based methods can be categorized into two parts: temporal-based and spatial temporal-based traffic forecasting.

2.3.1 Temporal Based Traffic Forecasting

Due to the fact that traffic data is sequential dependency, a form of artificial neural network called RNN was developed to address difficulties with this traffic time-series data. An RNN architecture comprises multiple copies of the same network that each can store memory from the previous training stage in a hidden layer and pass a message to the successor. With the chain-like structure, RNNs become appropriate architectures to process such time-series sequence observations. Although classical RNNs exhibit a superior ability to model nonlinear time-series data, several inherent issues still need to be addressed[153]. For example, it cannot train time series data with long time lags, which is commonly seen in traffic forecasting tasks, due to gradient vanishing and exploding problems[42]. To overcome the limitations of RNNs, LSTMs models were developed based on RNN's fundamental to model relatively long-term dependencies on time series data. With the elegant multiple-gate structure, LSTM can be trained to address the vanishing and exploding issues by keeping the network error constant[77]. This advanced design has been successful in solving traffic prediction tasks while addressing the gradient vanishing issue and presenting outstanding performance[76]. Many studies have demonstrated the popularity and effectiveness of LSTM as a traffic forecasting model, with many researchers opting to use LSTMs over RNNs.

Since the capability of LSTM architecture can handle the problem of recurrent patterns, several studies modified and enhanced the original LSTM to become more robust[245,121,123] in transportation scenarios. For instance, Stacked Bidirectional and Unidirectional LSTM

(SBULSTM) network[39] was proposed to capture the forward and backward temporal dependencies in traffic data. Multivariate Dual Long Short-Term Memory (MDLSTM) was developed to integrate both the latest trends and extra historical data patterns to achieve better performance[186]. Gated Recurrent Units (GRU), that incorporates different gate operations called update gate and reset gate, was designed to improve training efficiency and reduce the required memory in short-term traffic flow forecasting[54]. These LSTM variations have been rearranged and restructured original LSTM to extract the meaningful temporal representations for traffic prediction. In addition to these modifications, different mechanisms have also been devised to aid LSTM models, including the encoder-decoder architectures[34]. The encoder-decoder structure is designed to solve sequence-to-sequence issues, which are more difficult prediction problems that accept a sequence as input and requires a multi-step prediction as output, but are better suited to traffic forecasting tasks. Tsai et al.[188] have shown that a Seq2seq-based model outperformed LSTM models among all evaluations in multi-step prediction experiments.

However, the majority of the forecasting models listed above only deal with time-series data and do not learn traffic as a matrix. That is, only a few number of them were suggested to cope with network-wide forecasting. Spatial features were often disregarded or poorly merged with time-series data, making it difficult to learn spatial-temporal patterns from multidimensional data.

2.3.2 Spatial-temporal Based Traffic Forecasting

To mitigate the gap of lacking spatial information, Li et al.[110] proposed Diffusion Convolutional Recurrent Neural Network (DCRNN) that applies diffusion convolution to capture both spatial and temporal dependencies. In this model, the geographic correlation between traffic sensors is represented by nodes and edge weights on a directed graph, allowing the network to extend beyond a single sequence to better reflect spatial relationships. Another approach proposed by Cui et al. [36] includes an adjacency matrix to better capture nonlinear spatial-temporal phenomena and a free-flow accessible matrix to integrate transportation domain knowledge.

In recent years, CNNs, which are powerful image processing algorithms that can effectively extract informative features from images, have been generalized to capture spatial relationships in traffic networks. Works from Ma et al. [122], Zhang et al. [239], and Huang et al. [81], each used CNN to detect traffic patterns from geographic figures as well as traffic time-space speed matrix and further forecast future traffic speed. Liu et al. [116], Bogaerts et al. [15], and Ma et al. [124], further combined stacked CNNs to extract spatial features with LSTM to integrate temporal information of traffic data. Zhang et al. [237] also designed a spatial-temporal feature selection algorithm as a preprocessing step to generate a two-dimensional matrix as an enhanced input for CNN to predict future traffic flow. Wu & Tan [213] exploited a hybrid architecture by incorporating a one-dimensional CNN for spatial feature extraction and LSTMs for the short-term periodicities of traffic flow mining. Deep three-dimensional convolutional networks, as upgraded CNNs, were applied to recent studies to harness the ability to extract and model spatial-temporal data without separating the interplay between spatial and temporal dimensions [96,61,63]. However, standard CNN-based approaches are incapable of dealing with various topological structures of traffic networks. To address this issue, researchers began to train traffic networks as a graph and applied GNNs to extract patterns from network-wide traffic data [110,36].

Traffic networks can be intuitively identified as graphs with a combination of nodes and edges. Average speed and volume, for example, can be derived to depict the traffic conditions at each node, which each represents a road segment. Adjacency matrices, constructed based on road network connectivity, reveal the relationship between segments. Changes in traffic conditions can cause congestion to propagate backward and forward and affect connected route segments[146]. GNNs leverage this spreading characteristic of traffic flow as a benefit to aggregate current traffic status within the neighborhood to provide more reliable prediction results. Therefore, GNN has emerged as one of the most representative methods in the transportation field to address these intractable tasks over the last five years. Recent advances[197,67] in graph-structure learning have demonstrated that graph connectivity can be used to learn effective representations of road networks. The learned graph embeddings can then be applied to a variety of downstream tasks, such as road attribute inference and traffic forecasting, which share similar backdrops. Based on the pre-defined structure, Graph Convolutional Network (GCN)-based algorithms have been widely developed in traffic forecasting tasks recently[226,228,239,36]. These works utilized various extended GCNs to capture spatial inter-dependencies to improve computational efficiency[214] and enhance prediction performance[36]. Recent efforts, such as GMAN[246] and ASTGNN[64], have applied a more complex attention mechanism to capture dynamic spatial-temporal dependency.

Although these novel algorithms achieved promising results, several complicates were left unresolved. For example, many GNN-based techniques can be viewed as a two-phase process. The first phase uses GNN to aggregate neighborhood information, while the second phase uses modified embeddings to make predictions. GNN may obscure node/edge information by aggregating neighborhood knowledge, leading to noisy representations for downstream prediction tasks[106]. In addition, GNN-based modeling has been limited by approaches that only analyze locally neighboring connections. These approaches assume that surrounding connections are the most spatially related to the target link and use their information as input for prediction algorithms. However, in practice, spatial dependency is often observed across a wider range of traffic networks. For instance, traffic patterns on two separate roads may display similarities, despite the disconnected or the huge physical distance between them. In this case, traffic data from one road may serve as a valuable indicator for forecasting traffic conditions on the other road that is not directly connected. As a result, examining spatial correlation at a local scale may result in insufficient capture of significant information from distant connections, as well as increased inaccuracy if the nearby links have no geographical influence on the target link. Furthermore, as previously mentioned in Section 2.4, most current traffic forecasting research focuses on short-term forecasting, with prediction time frames of one hour or less.

2.4 Summary of Short-term Traffic Forecasting

Table 2.1 provides a summary of representative literature that focused on traffic forecasting in the past two decades, offering a good sense of where most research has concentrated. The summary shows that the majority of work has gone towards 1) short-term forecasting and 2) spatial-temporal information integration. Recent technological advancements, as well as the widespread use of sophisticated computers and mathematical models, provide academics with a valuable opportunity to broaden their perspectives. At first, limited-scaled datasets were used in classical methods to extract temporal-only dependency. Subsequently, artificial neural networks-based approaches dominated this area with unprecedented precision by effectively learning spatial-temporal representations from traffic networks. However, existing research on traffic forecasting mostly focuses on short-term scenarios. Modern traffic management systems nowadays require predicting longer sequence to alleviate congestion in advance.

Since Artificial Intelligence (AI) methods have evolved significantly in NLP, the Transformer[196], a model with a multi-head attention mechanism, was proposed to help memorize long source sentences in neural machine translation and achieved maximal performance. In other words, the Transformer can effectively solve forecasting tasks with a long time-series sequence. To leverage long-term temporal dependencies in traffic forecasting, Cai et al.[20] developed a Traffic Transformer. Yan et al.[220] further enhanced the performance by learning the dynamic and hierarchical structure of traffic flow. However, due to the COVID-19 pandemic and several other unexpected culture-related events, the existing traffic prediction models are affected without consideration of related semantic information.

2.5 Non-stationary Traffic Flow Learning

Stationary data is defined as data that is come from the same distribution[141]. However, this assumption does not hold for most real-world data, such as traffic data, which is considered as time series data and more likely to exhibit a fluctuating characteristic. Given historical traffic data, there are irregular traffic flows caused by planned events or real-time unexpected interventions, such as crashes or constructions, leading to non-stationary issues[94]. Unfortunately, the non-stationary challenge in past decades is not taken seriously in academia since it is more difficult to deal with. Most publications, as described in above sections, treated traffic data as stationary time series rather than non-stationary ones. While the forecasting

Table 2.1: Literature Of Short-term Traffic Forecasting

Author(s)	Year	Predict	Step	Max. Horizon	ST
Chen & Grant-Muller [30]	2001	Volume	30 mins	30 mins	✗
Williams & Hoel [209]	2003	Volume	15 mins.	15 mins.	✗
Sun et al. [177]	2003	Volume	5 mins.	30 mins.	✗
Alecsandru & Ishak [5]	2004	Speed	5 mins.	20 mins.	✗
Cetin & Comert [26]	2006	Speed	1 min	1 min	✗
Wang et al. [204]	2006	Travel Time	1 min	20 mins.	✗
Li et al. [111]	2008	Travel Time	5 mins.	20 mins.	✗
Min et al. [138]	2010	Volume	15 mins.	15 mins.	✓
Boto-Giralda et al. [17]	2010	Volume	5 mins.	10 mins.	✗
Kamarianakis et al. [90]	2012	Speed	5 mins.	1 hour	✗
Haworth & Cheng [69]	2012	Travel Time	5 mins.	5 mins.	✓
Wang & Shi [202]	2013	Travel Time	1 min	1 min	✓
Ma et al. [123]	2015	Speed	2 mins.	2 mins.	✗
Liu et al. [114]	2015	Speed	2 mins.	10 mins.	✗
Cai et al. [21]	2016	Speed	5 mins.	1 hour	✓
Wu & Tan [213]	2016	Speed	5 mins.	5 mins.	✓
Jeon & Hong [88]	2016	Speed	5 mins.	5 mins.	✗
Xia et al. [215]	2016	Volume	5 mins.	5 mins.	✓
Zhao et al. [244]	2017	Volume	5 mins.	1 hour	✓
Yu et al. [229]	2017	Speed	5 mins.	1 hour	✗
Yu et al. [226]	2017	Speed	5 mins.	45 mins.	✓
Yu et al. [228]	2017	Speed	2 mins.	30 mins.	✓
Cui et al. [38]	2018	Speed	5 mins.	5 mins.	✓
Li et al. [110]	2018	Speed	15 mins.	1 hour	✓
Luo et al. [121]	2019	Volume	5 mins.	5 mins.	✓
Wu et al. [214]	2019	Speed	5 mins.	1 hour	✓
Guo et al. [63]	2019	Volume	6 mins.	1 hour	✓
Cui et al. [39]	2020	Speed	5 mins.	5 mins.	✓
Huang et al. [80]	2020	Speed	5 mins.	1 hour	✓
Zheng et al. [246]	2020	Volume/Speed	5 mins.	1 hour	✓
Zheng et al. [247]	2020	Volume	5 mins.	1 hour	✓
Guo et al. [64]	2021	Volume	5 mins.	1 hour	✓

ST: Spatial-temporal

performance may be excellent when the data is simplified in a stationary format, it may not be reliable enough for practical use in modern traffic management systems under fluctuated conditions.

Few studies have attempted to account for temporal fluctuation in traffic circumstances. Zheng & Su[248] and Stathopoulos & Karlaftis[175] both designed a pre-processing phase to divide time periods into several pieces based on domain experience to represent the dynamic characteristics of the traffic. Specifically, one-day data was split into six time periods to model the variability of traffic flow data (period 1: midnight-6:30 a.m.; period 2: 6:30-10:00; period 3: 10:00-13:30; period 4:13:30-17:00; period 5: 17:00-20:30; and period 6: 20:30-midnight). Guo et al.[62] applied temporal convolution to learn the complicated non-stationary temporal dependence for multiple-step forecasting. Zhao[243] implemented a two-stage algorithm to 1) decompose non-stationary traffic data into sub-components by the theory of Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) to reduce the complexity of raw traffic data, 2) apply LSTM to learn meaningful patterns from low-dimensional signals. Tsirigotis et al.[189] and Xiao et al.[217] investigated the impact of weather conditions and fused meteorological data with non-stationary traffic data to take this impact into account with regard to traffic forecasting. Overall, limited studies have focused on non-stationary traffic pattern learning. Despite the fact that most of these studies have relied on pre-processing blocks to address the complexity of traffic data, non-stationary traffic pattern learning has received increasing attention, particularly during the COVID-19 epidemic[169], as the traffic pattern may change every month or even every week. Therefore, there is still a need to fill a gap in theoretically advanced and robust non-stationary traffic pattern learning methods to address this issue, such as representation learning.

2.6 Extract Critical Data Patterns with Representation Learning

Machine Learning (ML) questions, including traffic forecasting tasks, rely significantly on data representation. Most of the work in implementing ML algorithms is spent on designing preparation pipelines and data transformations for extracting critical patterns from complex and multivariate raw datasets. Such a time-consuming procedure highlights the limitation of organizing and extracting meaningful data with existing learning algorithms. To overcome this barrier, it is inevitable to develop novel ML learning algorithms that are less dependent on feature engineering but move closer to AI learning techniques, which have been shown to identify the underlying explanatory factors hidden in high-dimensional raw sensory data.

While AI models handled the spatial-temporal component of traffic prediction as mentioned in previous sections, they have yet to offer solutions for addressing following questions:

a) Catastrophic forgetting is a typical problem that occurs while training AI models with multiple tasks, and it becomes even more critical when dealing with constantly changing spatial-temporal mobility patterns.

b) Supervised learning approaches tend to prioritize achieving specific goals over learning broadly applicable background representations. This limitation becomes particularly evident in traffic prediction, where the constantly fluctuating environment requires adaptable and versatile models. To achieve robust traffic forecasting, it is crucial to learn generalize and applicable background representations that can be either reused for several downstream activities or utilized to deal with incomplete/complex data scenarios, rather than solely focusing on specific goals.

Recently, many researchers have paid more attention to incremental learning to address catastrophic forgetting issues[118,93]. Incremental learning is present as a technique to balance the flexibility of acquiring new knowledge and the stability of consolidating what models have already learned. Self-supervised representation learning, on the other hand, can extract informative low-dimensional representations from raw time series by leveraging the data's inherent structure, without the need for explicit supervision[185]. This approach has been extensively researched in fields such as CV and natural language processing, but it has received little attention for time series applications, particularly in transportation engineering. To this end, this section will investigate these two effective strategies to address the challenges in existing traffic forecasting approaches.

2.6.1 Incremental Learning with Continuously Changing Data

Plasticity is required for models to integrate new knowledge, while stability is needed to solidify what they have previously learned. Therefore, a great variety of algorithms have been devised to deal with this crucial problem. The existing incremental learning algorithms can be divided into three main strategies: 1) Selective Synaptic Plasticity, 2) Additional Neural Resource Allocation, and 3) Memory Rehearsal. Each strategy will be detailly described below.

Review on Selective Synaptic Plasticity algorithms

This approach is also known as the Regularization-based Approach, which is the most well-developed way to solve the catastrophic forgetting issue. This category can also be divided into two subgroups: 1) weight-constrained approaches and 2) gradient-constrained approaches.

Elastic Weight Consolidation (EWC)[93] is known as an iconic model in this subgroup. By restricting weights to stay close to their learned values as new tasks are encountered, EWC maintains the integrity of connections required for previously learned tasks. A regularization-based method is utilized for important parameter selection. And the Fisher information Matrix is applied to estimate the importance of neurons in EWC. Overall, EWC is an approach to figure out a generalized parameter set that can properly solve all tasks with acceptable errors.

Some variations have been proposed: Memory Aware Synapses (MAS)[7] adds a heuristic measure of output sensitivity, whereas Synaptic Intelligence (SI)[232] seeks to explain the decrease in loss during training to specific parameters. These three techniques have combined to generate a slew of new regularization-based approaches.

Gradient Episodic Memory (GEM)[118] is then categorized into the type of gradient-constrained. Specifically, the gradients of the new model should follow the same or similar path as the gradients of the prior model. The new model is unlikely to forget about existing classes under this requirement. Theoretically, we need to minimize a quadratic function that is subjected that the inner product of the current and the previous gradient direction should be equal to or larger than zero. As expected, solving the gradient-constrained approach for each violating gradient prior to updating the model weights is time-consuming. Aljundi et al.[8] speed up the process by sampling a representative subset of the gradient restrictions.

Review on Additional Neural Resource Allocation algorithms

Another sort of algorithm aims to prevent forgetting by altering network architecture. The Progressive Neural Networks[164] was proposed by DeepMind, an artificial intelligence subsidiary of Alphabet Inc., to include past information at each layer of the feature hierarchy, reuse existing computations, and learn new ones. Throughout the training process, progressive networks keep a pool of pre-trained models and discover lateral connections to extract important characteristics for new tasks. The progressive learning technique allows for richer compositionality and the integration of past information at each tier of the feature hierarchy.

Inspired by the concept of model extension from Progressive Networks, Hung et al. [82] presented a novel architecture: Compacting, Picking, and Growing (CPG), which integrates the ideas of weights pruning, important weights selection, and network extension together. This design is flexible for model extension while maintaining model compactness when dealing with sequential learning tasks.

However, the downside of these models is the massive amount of parameters. The number of parameters would gradually increase for the original progressive networks since the number of new tasks increases. Although the CPG model involved the idea of model compact, we still need to initialize a large model to accommodate the upcoming new tasks.

Review on Memory Rehearsal algorithms

The memory rehearsal-based category is a new strategy that addresses forgetting problems in a totally different way. Shin et al. [170] presented a dual-model architecture that included a deep generative model as well as a task solver model. In this technique, training data from previously learned tasks may be sampled and fused with information from new tasks using produced pseudo-data. As a result, there is no need to explicitly modify prior training samples for experience replay, lowering the working memory requirements.

Various memory rehearsal-based models were later proposed to solve the incremental learning tasks by generating pseudo-data from Generative Adversarial Network (GAN) or auto-encoder[48]. Still, all the approaches mentioned above were evaluated on a simple dataset, MNIST. As a result, it's unclear if this generative technique can handle more complicated domains. Furthermore, the additional structure, a deep generative model, would also suffer from catastrophic forgetting issues.

To sum up, the existing research related to incremental learning has concentrated on classification problems rather than regression tasks, where the catastrophic forgetting issue frequently arises. Besides, previous knowledge about new tasks is provided in their experimental setting, implying that the dataset was manually split before processing. Furthermore, incremental learning techniques are not applied to traffic forecasting tasks yet. It has great potential to become a suitable solution to tackle the difficulties of traffic forecasting in non-stationary patterns.

2.6.2 Self-supervised techniques for Representation Learning

In traffic forecasting scenarios, real-world time-series data is often high-dimensional and complex, which brings many obstacles to data modeling, even deep learning-based models. In addition, the data collected from various sensors may be inaccurate or missing due to communication issues. Currently, the approaches applied in traffic forecasting are primarily trained in a supervised way, meaning the model parameters are all trained to fulfill a specific target. These approaches may lead to an excellent performance in a particular task rather than learning generalized knowledge that can be transferred to other downstream tasks that share the same backdrop, such as the same relative layouts of buildings and land use. Therefore, self-supervised learning, which uses raw data as its supervision, has recently gained a surge of interest as the difficulty of supervised models in generalizing beyond their training data has become evident. It is worth investigating whether contrastive learning, a type of self-supervised approach that has recently demonstrated outstanding performance in a variety of research fields (e.g., CV and NLP), can replicate the success of learning more generalizable background knowledge to improve traffic forecasting performance under non-stationary conditions.

The basic idea behind contrastive learning is to enhance agreement between representations with similar semantics, known as positive pairs, while decreasing agreement between representations with unrelated semantic information, known as negative pairings. Momentum Contrast (MoCo) model[71], which is one of the most popular designs in contrastive learning, can be simplified as a dictionary look-up problem. Given a reference picture I, it will be augmented into two views, query and key. The query token must match positive pairs over a collection of sampled negative pairs from other images. Chen et al.[31] summarized the main component of a contrastive learning framework that can be categorized into three parts as showed in Figure 2.1: 1) A data augmentation module transforms raw data I_i to various perspectives as positive pairs, denoted as $I_i^+ = \{I_1, I_2, ..., I_n\}$; 2) An encoder module that extracts features from raw complex data by mapping it to a low-dimensional space; 3) A project head, such as non-linear projection, which further maps extracted representations into a normalized space to evaluate the contrastive loss.

Recently, various graph contrastive learning-based algorithms with excellent performance were proposed. These approaches produced generalizable representations that may be utilized in subsequent tasks. For example, You et al.[223] developed four types of graph data augmentations in contrastive learning to address the challenge of data heterogeneity in previous

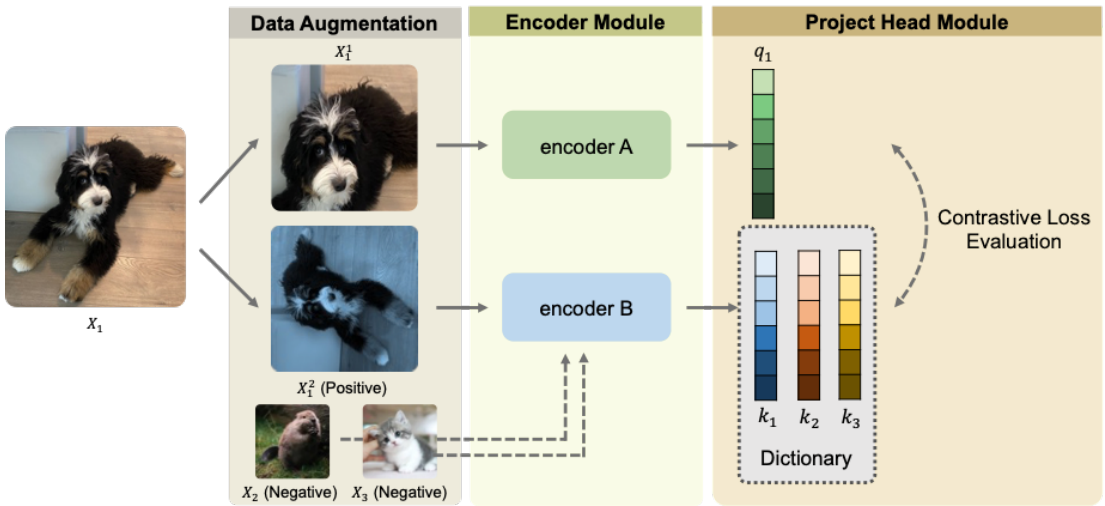


Figure 2.1: A Basic Constrastive Learning Framework

graph-based research. The representations were applied in downstream graph classification tasks and achieved state-of-the-art performance. Veličković et al. [198] proposed a general approach to learn node representations in a self-supervised way to also cope with node classification problems, hence addressing the real-world issue of most graph data being unlabeled. However, these approaches target neither time-series data nor the transportation field. They cannot be directly applied to traffic forecasting until solving several significant drawbacks:

a) Most of them focused on learning coarse-grained representation, which are suitable for instance-level anomaly detection and node classification, rather than fine-grained representations for traffic forecasting.

b) Most of them ignored the multi-scale contextual information at various granularities. Features with multiple scales may provide rich semantics and enhance the capacity to learn generalized representations.

c) Most of them neglected the importance of defining generalized spatial and temporal positive/negative pairs, which impedes the effectiveness of contrastive learning in spatial-temporal traffic forecasting task.

2.7 Chapter Conclusion

This section has reviewed the development process of traffic forecasting, starting from classical methods to deep learning approaches to learn more complex traffic patterns. Many researchers have dived into this field to propose advanced algorithms to achieve state-of-the-art performance. However, existing research on traffic forecasting mostly focuses on short-term forecasting. They are also impacted by COVID-19 epidemic and a number of other unanticipated culture-related phenomena due to the ignorance of non-stationary traffic conditions. In order to provide trustworthy forecasting results for practical usage in modern traffic management systems, appropriate and robust algorithms that tolerate fluctuating patterns and learn generalized representations must be developed. To summarize, this section highlights the shortcomings of existing works in dealing with these difficulties. The following sections will propose novel and practical deep learning approaches for traffic representation learning and forecasting under non-stationary circumstances.

Part II

Prediction with End-to-end Frameworks

3

Multivariate Dual LSTM-Based Network for Traffic Forecasting Under Interference

3.1 Overview

The impacted traffic patterns from the unexpected event brings challenges to the U.S. Department of Transportation and transportation planners. With fluctuated traffic conditions, it is difficult for transportation agencies to learn representative traffic patterns from short-term historical data. Therefore, we propose a Multivariate Dual Long Short-Term Memory (MDLSTM) model for network-wide traffic forecasting under interference. Both spatial and temporal features were included to forecast the influenced traffic behavior during the COVID-19 period. Multi-dimensional spatial-temporal features were fused into historical matrices as the model input, which enables the proposed structure to accommodate intervention from unexpected events. Thorough experiments were conducted using loop detector data collected in the Greater Seattle Area from 2020 to early 2021. The proposed model showed competitiveness against other state-of-art algorithms in all experiment time frames, from pre-COVID-19 to COVID-19-relieving period.

3.1.1 Background

In March 2020, the small and medium business group conducted a comprehensive survey of over 500 businesses and discovered that the personal service and retail sectors were among those most heavily impacted by the COVID-19 pandemic[159]. Local small businesses faced financial challenges as a result of lockdown orders implemented by authorities. Many stores saw a decline in profits as customer traffic decreased. By using the Traffic Performance Score (TPS), a novel evaluation parameter proposed by Cui et al.[41], and incorporating land use layers in the Greater Seattle area (as shown in Figure 3.1), we were able to observe that industrial areas were relatively unaffected in terms of traffic patterns due to the continued need for human operations to maintain daily work. However, two areas stood out as seeing the largest changes in TPS: the route from Northgate to downtown Seattle and the route from Newcastle to Bellevue. Traffic from urban residential areas to intensive urban areas significantly

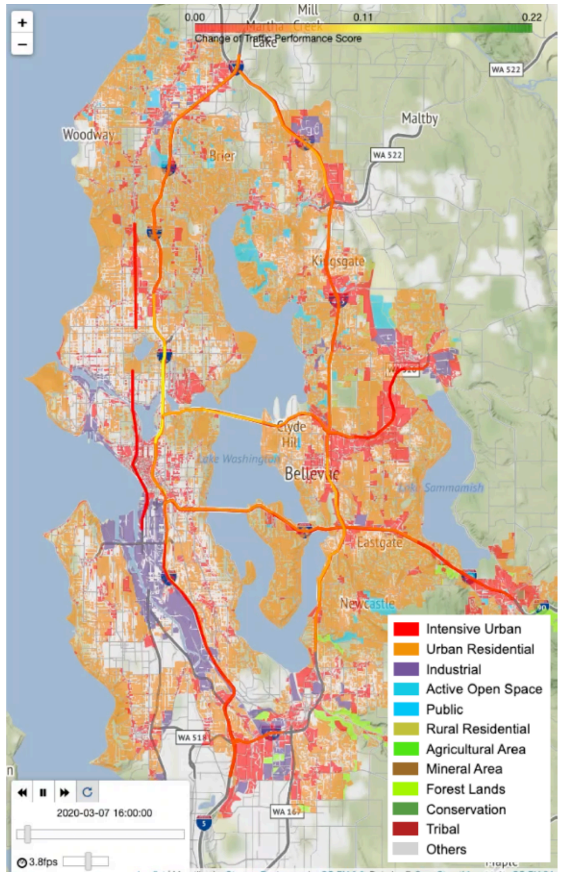


Figure 3.1: Visualization Of Traffic Performance Score And Land Use Layers

decreased, likely due to the implementation of work-from-home policies and the resulting reduction in commuting as people stayed home more often and companies began to discontinue renting offices.

While the visualization of TPS demonstrates past history and status quo of traffic conditions, predicting future changes is essential for local businesses and transportation planners to adjust to upcoming challenges. The goal of Intelligent Transportation Systems (ITS) is to improve traffic performance and efficiency with data-driven approaches. Short-term forecasting of traffic state is an essential part of traffic control and operation. By estimating future traffic, transportation system may be managed in a more efficient way and further mitigate congestion. With accurate and timely traffic prediction, it is expected to lead to an overall travel time reduction and traffic safety increment[123]. To achieve a desired forecasting result, a considerable number of data is required to learn historical traffic patterns. With the wide deployment of traffic sensors as well as surveillance cameras nowadays, a massive amount of traffic data is collected. And one of the most critical subject is to develop a robust pipeline to consume these data sources, transform them into valuable information, and utilize them in traffic forecasting and planning.

Given an increasing number of data and rapid development of computational capability in recent years, plenty of algorithms have been proposed for traffic forecasting. As indicated in previous literatures[123,201,91], forecasting algorithms can be categorized into traditional approaches and Artificial Neural Networks (ANN)s perspectives. The former consists of Autoregressive Integrated Moving Average (ARIMA) family models[105,209], which have convinced capability in accuracy and require less computational effort[128]. However, it is inevitable for them to make many hypotheses and set restrictions on developing models[201].

Over the past decades, ANN-related algorithms emerged as one of the most representative methods in the transportation field. One of the subdivisions, Recurrent Neural Network (RNN)s, has become a competitive model regarding traffic forecasting. Many researchers leveraged its capability of processing arbitrary sequences of time-series inputs with internal memory units, to achieve reliable prediction results[195]. However, it suffers from vanishing and exploding gradients in the training stage, which hampers the feature learning of long data. To solve this problem, Long Short-term Memory (LSTM) networks, a critical framework with gate structures, was proposed. It promptly became one of the most well-known traffic forecasting methods in recent years. However, the basic LSTM-based models showed a dependence on recurrence data and could not accommodate interference well. During the coronavirus pandemic, traffic patterns changes continuously based on work-form-home policy and state regulations. The unsteadiness of social circumstances makes traffic forecasting a difficult task to reflect the status quo.

3.1.2 Contributions and Organization

To solve the aforementioned challenges, we propose a MDLSTM in this chapter for network-wide traffic forecasting under intervention. The integration of a long-term LSTM and a short-term LSTM architecture allows the model to capture up-to-date trends with extra historical data input. Meanwhile, it is capable of taking additional multi-dimension features into consideration and making the architecture more robust and flexible. A sequence of experiments covered the COIVD-19 pandemic, the most fluctuating period, reveals the competitiveness of the proposed method against baseline algorithms. The proper length of traced-back training data and the training efficiency are further analyzed to investigate how these factors could influence the model training result. To summarize, the contributions of this chapter are listed as follows:

1. Propose an integrated dual LSTM-based traffic forecasting model that can accommodate interference and incidents in recurrence data. The model is evaluated by a large-scale network-wide non-stationary dataset.

2. Consider spatial and temporal features that could become crucial, especially when unexpected occurrences happen, such as day of the week, hour of the day, and land use type, as inputs to improve the robustness of the entire model.

3. Construct an attention-based learning component to leverage both long-term and short-term representations, which provides flexibility to balance the contribution from each side.

The rest of this chapter is organized as follows: In Section 3.2, we define the problem and describe the design of each component in the model, including the details of the proposed model: Multivariate Dual Long Short-Term Memory. In Section 3.3, experimental settings are initially introduced, followed by evaluation results and ablation studies, respectively. The visualized comparison between the prediction result and the ground truth is presented as well. In Section 3.4, we conclude with some remarks and point out potential future works to expand concepts to accommodate non-stationary traffic patterns.

3.2 Methodology

3.2.1 Problem Statement

Traffic congestion is a common spatial-temporal pattern in road networks. It is propagated to not only forwarding direction but also backward and all the preceding segments [146]. The traffic data matrices are composed of three elements: temporal data, spatial data analysis, and other features. Temporal data is one of the most important component in the problem. The traffic trend from time to time is the main characteristic that the model needs to observe. Spatial data, such as land use, also provides decision-useful information. Depending on each location's own geographical attributes, travel pattern may vary. For instance, residential areas have higher outbound traffic volume in morning peak hours whereas industrial areas have higher inbound traffic volume. Locations with similar combination of geographical attributes have similar travel patterns. Other features include time of day and day of week, which are categorical indicators of peak and off-peak periods. Below shows how the three elements are integrated into the proposed model.

Temporal data contains historical data collected from segment sensors within the entire network. The short-term historical data matrices X_S consists of data within N timestamps before the targeted time t_c in all M segments:

$$X^S = \begin{bmatrix} x^1_{t_c-N} & x^2_{t_c-N} & \dots & x^M_{t_c-N} \\ x^1_{t_c-N-1} & x^2_{t_c-N-1} & \dots & x^M_{t_c-N-1} \\ \vdots & \vdots & \ddots & \vdots \\ x^1_{t_c-1} & x^2_{t_c-1} & \dots & x^M_{t_c-1} \end{bmatrix} \tag{3.1}$$

Each element x_t^m symbolizes the traffic condition in m^{th} segment at t^{th} timestamp. To accommodate unexpected events that could directly influence traffic conditions, such as the COVID-19 outbreak, we also considered relatively long-term historical traffic data matrices X_L, which is composed of data with N timestamps before a specific past timestamp t_p in all M segments:

$$X^L = \begin{bmatrix} x^1_{t_p-N} & x^2_{t_p-N} & \dots & x^M_{t_p-N} \\ x^1_{t_p-N-1} & x^2_{t_p-N-1} & \dots & x^M_{t_p-N-1} \\ \vdots & \vdots & \ddots & \vdots \\ x^1_{t_p-1} & x^2_{t_p-1} & \dots & x^M_{t_p-1} \end{bmatrix} \tag{3.2}$$

In this study, the past timestamp t_p was set to be one week ahead the targeted time t_c. That is, we also considered the traffic performance one week before the prediction was made. The historical traffic parameters include average speed, average volume on general purpose lanes, and TPS. TPS is a novel traffic index that incorporates multiple parameters for measuring network-wide traffic performance, which can be calculated using the following equation:

$$TPS_t = \frac{\sum_{i=1}^{n} V_t^i \cdot Q_t^i \cdot L^i}{V_f \cdot \sum_{i=1}^{n} Q_t^i \cdot L^i} \times 100\% \tag{3.3}$$

where V_t^i and Q_t^i denote the speed and volume of each road segment i at time t respectively. L_i is the length of the detector on segment i and V_f represents the free flow speed. In this case, the range of TPS is from 0% to 100%. 100% means the best traffic condition without congestion and 0% presents the worst network-wide traffic condition. TPS was also the parameter that our proposed model attempts to forecast.

By integrating long-term historical matrices X_L and short-term historical traffic data X_S, we obtained a more robust spatial-temporal pattern to adapt unforeseen events. Meanwhile, the non-time series feature matrix D, including hour of day D_{Hour}, day of week D_{Day}, and type of land use D_{Land}, was also taken into consideration in this study.

$$D = \{D_{Hour}, D_{Day}, D_{Land}\} \tag{3.4}$$

We sought to predict the network-wide TPS value $\hat{Y}^{t_c}$ for all the segments at the targeted timestamp t_c. The procedure of forecasting network-wide traffic conditions can be written as:

$$\hat{Y}_{t_c} = G(X_S, X_L, D) \tag{3.5}$$

where G is MDLSTM, the main algorithm that we proposed in this study. The objective is to minimize the difference between the sequence of predicted TPS $\hat{Y}_{t_c}$ and the ground truth

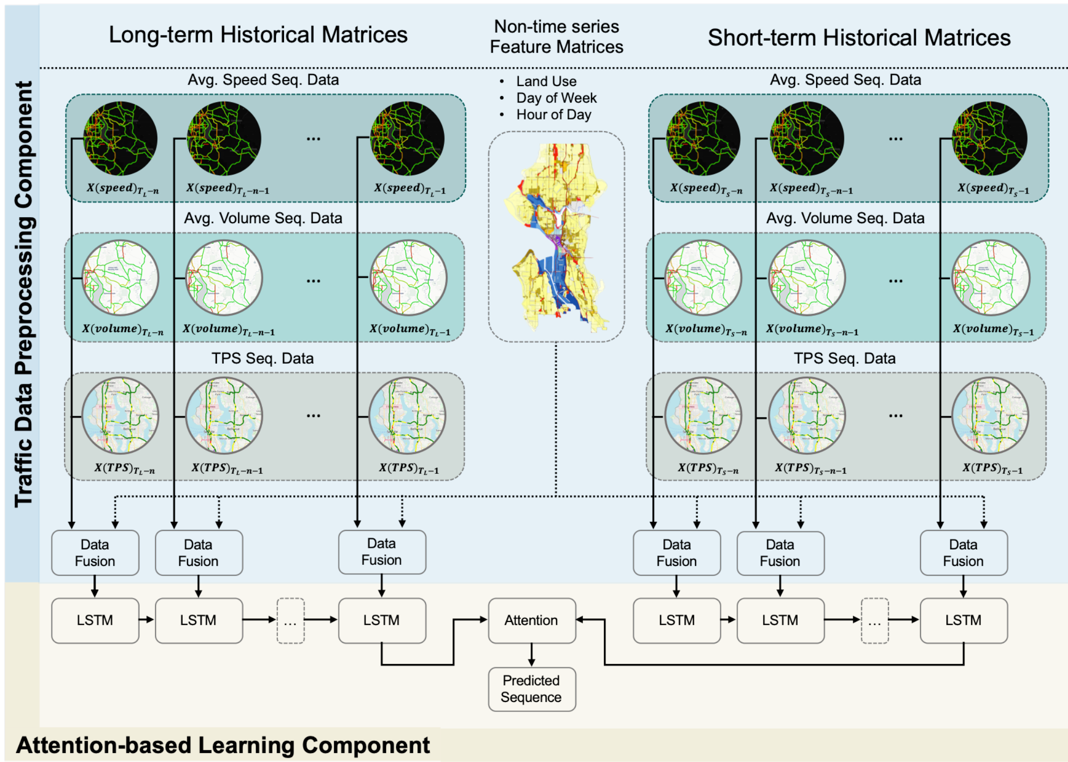


Figure 3.2: Multivariate Dual Long Short-Term Memory Model Architecture

data Y_{t_c}.

3.2.2 Multivariate Dual LSTM Model Description

The MDLSTM model, as shown in Figure 3.2, consists of two main components: two traffic data processing component and an attention-based learning component. Each of them is described as follows.

Traffic Data Processing Component

According to previous research[127], we summarized that speed and volume are two essential factors in traffic conditions that could directly reflect potential and existing congestion. Along with TPS, the three traffic-related data resources have time-series characteristics, which means that they are continuously collected at multiple timestamps and ordered chronologically. In this study, we selected two time slots of time-series data with all three aforementioned sources as parts of the model input: long-term sequence historical matrices T_L and short-term historical matrices T_S. By learning from both long-term and short-term perspectives, the model became more robust to accommodate external interference.

Non-time series data, the hour of day, the day of week, and the type of land use, were also included to help forecast traffic conditions. Daily commuters with high repetitive travel activities would bring up morning and evening peak traffic period in weekdays[191]. The type of land use could also become a critical feature, especially when unexpected events happened. For example, during the COVID-19 pandemic, the vehicle miles traveled record showed an expected drop due to the work-from-home policy. However, the necessary trips in specific land use, such as industrial, for supporting essential economic operations remained consistent with the pattern before the outbreak of COVID-19[41].

In the traffic data processing component, we implemented data fusion processors to merge time series data and non-time series data as the input for the training model. An embedding approach[137] was executed to transform the categorical factors into neural network sequence data. An embedding procedure is widely used in Natural Language Processing (NLP) for capturing the semantics by placing similar inputs close together in the embedding space. Specifically, we embedded non-time-series matrices to transform categorical features and concatenated them with time-series matrices as the final inputs X_{F_S} and X_{F_L} for the following training. The final inputs can be formulated as follows:

$$X_{F_S} = X_S \oplus E_{Day_S}(D_{Day}) \oplus E_{Hour_S}(D_{Hour}) \oplus E_{Land_S}(D_{Land}) \tag{3.6}$$

$$X_{F_L} = X_L \oplus E_{Day_L}(D_{Day}) \oplus E_{Hour_L}(D_{Hour}) \oplus E_{Land_L}(D_{Land}) \tag{3.7}$$

where D_{Day}, D_{Hour}, and D_{Land} are raw categorical data. E_{Day}, E_{Hour}, and E_{Land} are trained embedding matrices. $\oplus$ represents the concatenation process. The time slots of X_{F_S} and X_{F_L} would be $[T_c - N, T_c - N + 1, \cdots, T_c - 1]$ and $[T_p - N, T_p - N + 1, \cdots, T_p - 1]$ respectively. The length of two inputs are both N.

Attention-Based Learning Component

After executing the traffic data processing component, we fed X_{F_S} and X_{F_L} into two LSTM models separately. Each LSTM cell takes cell state c_{t_s-1} and hidden state h_{t_s-1} from the previous time step as well as the current input vector $[x_{t_s}^1, x_{t_s}^2, \cdots, x_{t_s}^M]$ as its input at timestamp t_s. They are controlled by a three-gate structure, the forget gate f_{t_s}, the input gate i_{t_s}, and the output gate o_{t_s}, to learn the pattern and transform the sequential input into the predicted sequential values. The functionalities of each controllers are shown as follows:

$$f_{t_s} = \sigma(W_f[h_{t_s-1}, X_{t_s}^M] + b_f) \tag{3.8}$$

$$i_{t_s} = \sigma(W_i[h_{t_s-1}, X_{t_s}^M] + b_i) \tag{3.9}$$

$$o_{t_s} = \sigma(W_o[h_{t_s-1}, X_{t_s}^M] + b_o) \tag{3.10}$$

$$c_{t_s} = f_{t_s} \odot c_{t_s-1} + i_{t_s} \odot tanh(W_c[h_{t_s-1}, X_{t_s}^M] + b_c) \tag{3.11}$$

$$h_{t_s} = o_{t_s} \odot tanh(c_{t_s}) \tag{3.12}$$

where b_f, b_i, b_o, and b_c are bias vectors; W_f, W_i, W_o, and W_c are weight mapping matrices; σ and $tanh$ are sigmoid and hyperbolic tangent activation functions, respectively. $\odot$ denotes the element-wise product process.

After two chain-like LSTM models, the final hidden states from both components, h_{t_c-1} and h_{t_p-1} were passed into a linear attention component. The component was in charge of managing and quantifying the interdependence between the input and output elements:

$$\hat{Y}_{t_c} = tanh(W_L h_{t_p-1} + W_S h_{t_c-1}) \tag{3.13}$$

where W_L and W_S are the weight matrices for preceding hidden state outputs and $\hat{Y} t_c$ is the final output. We used $tanh$ as the activation function since TPS ranges from 0 to 1. The detailed spatial-temporal data is described in Section 3.3.

Table 3.1: Model Structure And Component Comparison

Model	S-LSTM	L-LSTM	Other features
LSTM	✓		
DLSTM	✓	✓	
MLSTM	✓		✓
MDLSTM	✓	✓	✓

DLSTM: Dual LSTM; MLSTM: Multivariate LSTM;

3.3 Experiments

As suggested in multiple studies, neural network outperforms classical statistic models for traffic forecasting and conventional methods often cannot handle multidimensional predictions. Thus, traditional time series analysis methods, such as ARIMA, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM)[236], are not included in this paper. In this section, we compared the proposed multivariate dual LSTM-based network with state-of-art baseline, LSTM, on network-wide traffic forecasting. Table 3.1 shows the structure and components of the four models evaluated in this study.

3.3.1 Experiment Setting

The TPS Dataset is composed of traffic parameters developed by STAR Lab TPS[i]. It contains TPS, traffic speed, and traffic volume data on four freeways (I-5, I-90, I-405, and SR-520), as shown in Figure 3.3, at a 15-minute interval in the greater Seattle area in 2020 and 2021. The

[i] UW STAR Lab TPS Website: http://tps.uwstarlab.org/

land use data was labeled based on the Seattle GeoData [ii]. By plotting the surrounding land use layers along with the road network, we labeled the land use for each segment.

In this study, we utilized the data from January, 2020 to January, 2021. Mean Squared Error (MSE) was utilized to evaluate the model accuracy, which can be calculated using the equation:

$$MSE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|^2}{n} \tag{3.14}$$

where y_i and $\hat{y}_i$ represents the ground truth and predicted value respectively.

In order to evaluate the models using data with and without external interference, we used one month, two months, and three months data as training and validation data to predict the TPS value for the coming month. This study contained 30 experiment sets, which are demonstrated in Table 3.2. Based on Covid-19 pandemic, the experiments include four scenarios: 1) before coronavirus (January 2020 to March 2020), 2) coronavirus outbreak (March 2020 to June 2020), 3) gradually recovering (June 2020 to January 2021), and 4) pandemic deteriorate (November 2020). There are 87 segments in the network and we take three hour data as our input for one LSTM layer.

3.3.2 Performance Comparison

Additional Features And Structures In LSTM

In this section, we determined whether adding additional features or implementing the dual LSTM structure would have a better performance. We fixed the training data time range as three months and the input data range as 12 timesteps ahead. While King County announced

[ii] City of Seattle Geo Data: https://data-seattlecitygis.opendata.arcgis.com/

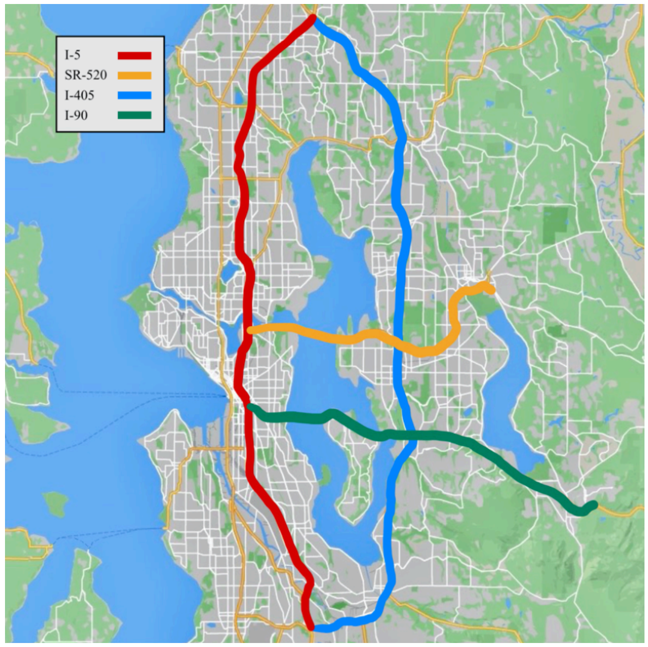


Figure 3.3: Study Area: The Greater Seattle Area

the stay home, stay healthy order on March 23, 2020, an increasing amount of businesses started to work from home. The phase 2 reopen order was announced on June 19, 2020 and the outbreak deteriorated again in November, 2020. Figure 3.4 and Table 3.3 show the MSE of four models using data within different time periods.

They demonstrate an accuracy drop in the third, sixth and ninth model respectively, which

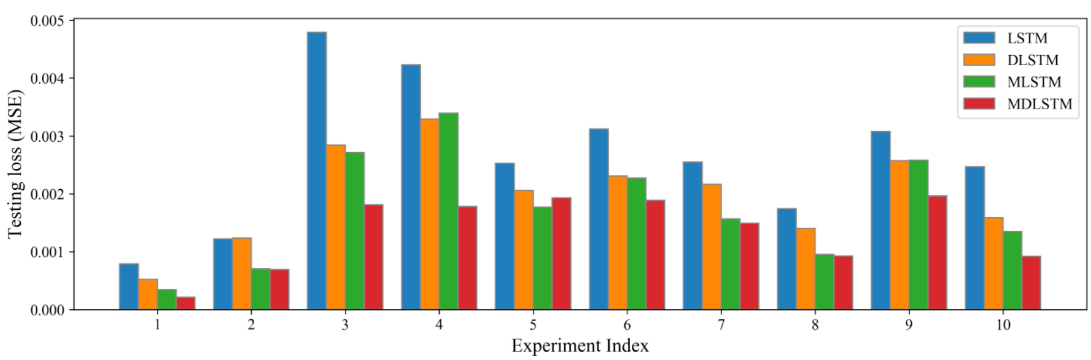


Figure 3.4: Testing Mean Squared Error Of All Experiments

Table 3.2: Experiment Time Frames

Index	Training (3 month)	Training (2 month)	Training (1 month)	Testing
1	01/2020 → 03/2020	02/2020 → 03/2020	03/2020 → 03/2020	04/2020
2	02/2020 → 04/2020	03/2020 → 04/2020	04/2020 → 04/2020	05/2020
3	03/2020 → 05/2020	04/2020 → 05/2020	05/2020 → 05/2020	06/2020
4	04/2020 → 06/2020	05/2020 → 06/2020	06/2020 → 06/2020	07/2020
5	05/2020 → 07/2020	06/2020 → 07/2020	07/2020 → 07/2020	08/2020
6	06/2020 → 08/2020	07/2020 → 08/2020	08/2020 → 08/2020	09/2020
7	07/2020 → 09/2020	08/2020 → 09/2020	09/2020 → 09/2020	10/2020
8	08/2020 → 10/2020	09/2020 → 10/2020	10/2020 → 10/2020	11/2020
9	09/2020 → 11/2020	10/2020 → 11/2020	11/2020 → 11/2020	12/2020
10	10/2020 → 12/2020	11/2020 → 12/2020	12/2020 → 12/2020	01/2021

corresponds to the the three time points since the traffic pattern as well as travel behaviors changed significantly due to different policies.

From the results shown in Table 3.3, we can see the competitiveness of the proposed method against baseline algorithms. The MDLSTM model has the highest accuracy among the nine out of ten testing time frames. In terms of additional features, the model accuracy increases when time of day, day of week, and land use information are included in the model input. It is shown that Multivariate LSTM (MLSTM) and MDLSTM both have lower testing errors in all the experiment cases. Extra informative variables not only improves the performance during regular situations, but also mitigates the impact of external interference for traffic

Table 3.3: Testing Mean Squared Error Of All Experiments

Index	LSTM	DLSTM	MLSTM	MDLSTM
1	0.00079	0.00052	0.00035	**0.00021**
2	0.00122	0.00124	0.00071	**0.00069**
3	0.00479	0.00284	0.00272	**0.00182**
4	0.00423	0.00329	0.00339	**0.00178**
5	0.00253	0.00206	**0.00177**	0.00193
6	0.00312	0.00231	0.00227	**0.00189**
7	0.00255	0.00217	0.00157	**0.00149**
8	0.00175	0.00217	0.00157	**0.00093**
9	0.00308	0.00257	0.00258	**0.00196**
10	0.00247	0.00159	0.00135	**0.00092**

* Models with the least MSE are marked in bold font

forecasting.

As for the proposed Dual LSTM (DLSTM) structure, we found it successfully accommodates intervention. The MDLSTM models have a close performance compared to the MLSTM models using data from time frames 1, 2, 7, and 8, which are considered as status before coronavirus and gradually recovering. In other words, they function similarly when there is no interference. However, at the time announcing stay-at-home order and phase 2 reopen order, or even encountering the pandemic deteriorated, MDLSTM performs better in terms of testing loss, proving the capability of accommodating external disturbance.

Figure 3.5 visualizes the prediction results of two randomly picked times and segments: Northbound I-405 from June 1, 2020 to June 7, 2020 and Northbound I-5 from September 2, 2020 to September 8, 2020 Northbound I-5. In Figure 3.5(a), we used the historical data from March 2020 to May 2020 as training data. According to the announcement dates for

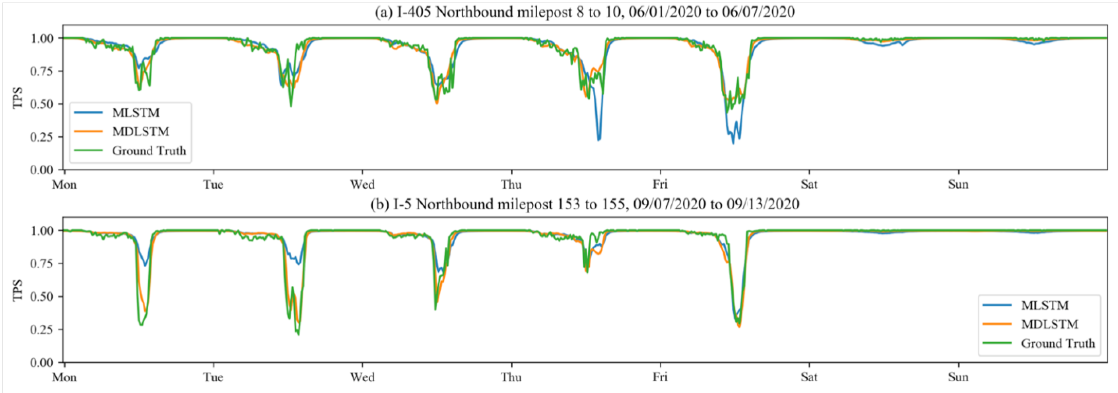


Figure 3.5: Comparison Between Ground Truth And Prediction Results From Proposed Models (a) At I-405 Northbound Milepost 8 To 10, (b) I-5 Northboumd Milepost 153 To 155

several policies related to the outbreak of COVID-19, the training data were undoubtedly under interference, which means traffic patterns in training data would fluctuate. The prediction results of MDLSTM demonstrate the competitiveness that normal daytime and peak hours could be detected effectively and aligned to the ground truth data. But the result of MLSTM diverged from the ground truth at peak hours. In Figure 3.5(b), we selected the historical data from June 2020 to August 2020 as training data. Even though some unexpected events happened, such as the phase 2 reopen order on June 19, 2020, the MDLSTM model remains robust and exhibits the capability to adapt interference and follow a longer trend.

Different Training Data Duration

In this section, we aimed to determine how long the training data should trace back. Training data is a crucial element in machine learning process. The quality of the training data will influence the model training result. In most cases, including more data for training can improve the model performance. However, during the COVID-19 pandemic, the longer the duration of training data, the more external events occurred. Therefore, we used one

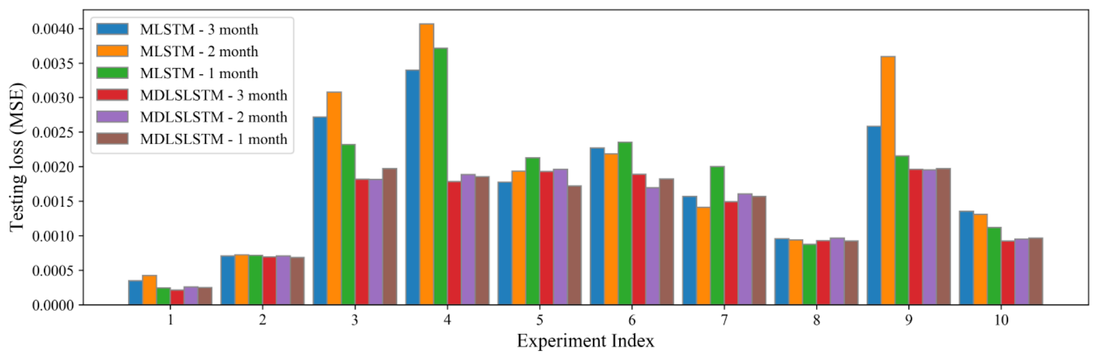


Figure 3.6: Testing Mean Squared Error Using Different Duration Of Training Data

month, two months, and three months data as training and validation data to predict the TPS value for the coming month. MLSTM and MDLSTM were implemented to evaluate the aforementioned issue. Figure 3.6 and Table 3.4 show the MSE of the two models in all the experiment cases. They also demonstrate an error increment in the third (and fourth), sixth and ninth model, which is the same as the previous section. Several policies and orders were announced during these experiment time frames, causing the change of traffic conditions.

In terms of MLSTM, Figure 3.6 demonstrates that using 2 months data to train the model performs badly when an incident occurs. It has less input data than MLSTM-3 and the training data does not align to the testing data in comparison to MLSTM-1. That is, it does not have enough data and cannot observe the most recent change in traffic pattern. The MLSTM models have similar performance when experiments have less interventions within their duration.

As for MDLSTM models, they have steady performance regardless of the amount of training data. The mean squared errors of using one month, two months, or three months training data do not exceed 0.002, which is a relatively low value. It indicates that MDLSTM has the capability of achieving higher precision while using less data. It is able to capture traffic patterns under interference with fewer samples.

Training Efficiency

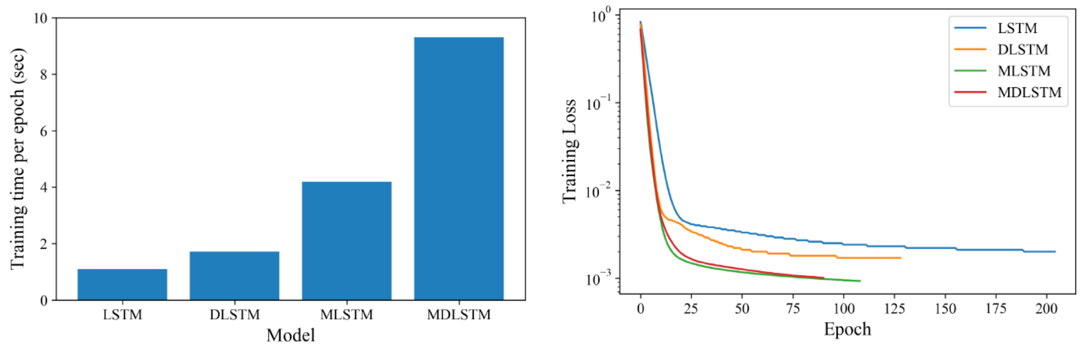
To adjust to the most recent changes or incidents, using the same model is no longer feasible. We may need to regenerate the model frequently to reflect the trend of the status quo. Training time and efficiency are essential to ensure that the service can be updated frequently in an efficient way. In this subsection, we compare the training efficiency of the proposed model with other LSTM-based models. The models are developed using PyTorch and the experiments were conducted on a computer with 2.3 GHz Dual-Core Intel Core I5 Processor.

The efficiency of each model was assessed by the training time required per epoch as shown in Figure 3.7a. The LSTM model is the simplest structure and thus does not need as much computation time as the other models whereas MDLSTM require the most training time

Table 3.4: Testing Mean Squared Error Using Different Duration Of Training Data

Index	MLSTM-3	MLSTM-2	MLSTM-1	MDLSTM-3	MDLSTM-2	MDLSTM-1
1	0.00035	0.00042	0.00024	**0.00021**	0.00026	0.00025
2	0.00071	0.00072	0.00072	0.00069	0.00070	**0.00068**
3	0.00272	0.00308	0.00232	0.00182	**0.00181**	0.00197
4	0.00339	0.00407	0.00371	**0.00178**	0.00188	0.00185
5	0.00177	0.00193	0.00213	0.00193	0.00196	**0.00172**
6	0.00227	0.00218	0.00235	0.00189	**0.00169**	0.00182
7	0.00157	0.00141	0.00200	**0.00149**	0.00160	0.00157
8	0.00096	0.00094	**0.00087**	0.00093	0.00096	0.00092
9	0.00258	0.00359	0.00216	0.00196	**0.00195**	0.00197
10	0.00135	0.00131	0.00112	**0.00092**	0.00095	0.00097

* Model-N: The model is trained using N month of data



(a) Training Time Per Epoch

(b) Training Loss (Experiment 6 Using 3 Months Data)

Figure 3.7: Training Efficiency Analysis

due to its complexity. DLSTM spends less time compared with MLSTM since it utilized less features and has fewer parameters to train. Since the early-stopping mechanism terminated the training process when the validation loss have not improved for over 10 epochs, this study did not evaluate in the number of epochs trained. However, we can observe from Figure 3.7b that LSTM converges the slowest while DLSTM, MLSTM, and MDLSTM have similar speed in terms of reducing training error. It proves that the multivariate and dual structure indeed improve the baseline model and contribute to capturing obscure patterns.

3.4 Chapter Conclusion

In this study, we propose a multivariate LSTM-based network to forecast traffic parameters under interference. The improvements and contributions of this study focus on four aspects:

1. Propose an integrated dual LSTM structure to accommodate external interventions.

2. Apply a real-world dataset and incident to evaluate the proposed method.

3. Fuse both spatial and temporal features as input to improve the model robustness.

4. Construct an attention-based learning component to leverage both long-term and short-term prediction results.

Experiment results indicate that including the day of the week, the hour of the day, and land use information in the MDLSTM model has the best performance during the COVID-19 period. MDLSTM can effectively handle the interference and capture the pattern in terms of a longer trend. Additional features can enhance the model's performance under fluctuating circumstances. And the dual LSTM structure ensures that the model takes external interventions into consideration. In addition, MDLSTM has a more lenient requirement on the size of training data. Unlike most machine learning models, taking only one month of data for training can achieve a satisfying result. When any incidents occur in the future, we will be able to spend lower computation costs to retrain a forecasting model and accurately reflect the status quo promptly. Consequently, it is proved that the proposed model is more capable of accommodating random events than basic LSTM models. Future studies may discuss the extension of the method as well as parameter optimization. The number of timesteps to input and the time of the historical data are not investigated in this paper. Predicting multiple time steps for network-wide traffic data is a common subject in traffic forecasting.

4

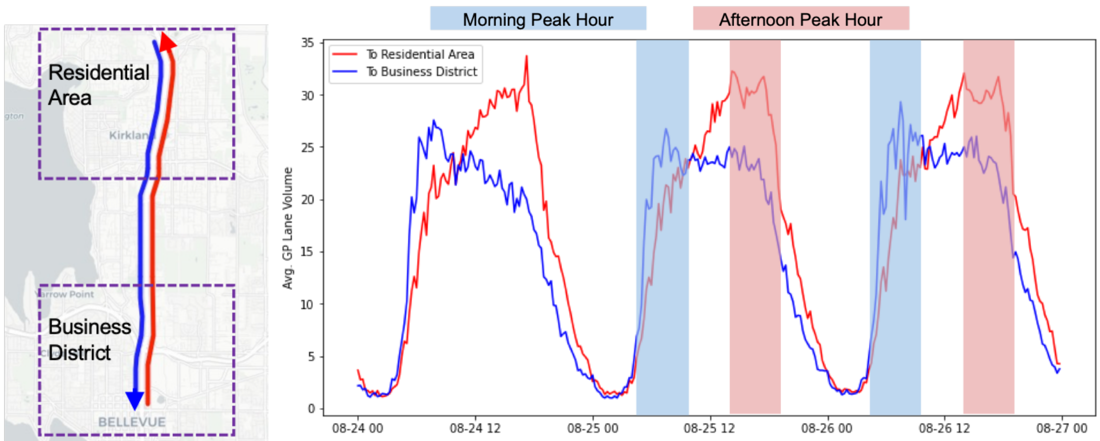
Traffic-Twitter Transformer: A Nature Language Processing-joined Framework for Network-wide Traffic Forecasting

4.1 Overview

With accurate and timely traffic forecasting, adverse traffic conditions can be proactively predicted to guide agencies and network users to respond appropriately. However, existing works on traffic forecasting have primarily relied on historical traffic patterns, confining to short-term prediction (e.g., under 1 hour). To better manage future roadway capacity and accommodate social and human impacts, proposing a flexible and comprehensive framework to predict physical-aware, long-term traffic conditions for network users and transportation agencies is crucial. In this paper, the gap in robust long-term traffic forecasting was bridged by including social media features. A correlation study and a linear regression model were implemented to evaluate the significance of the correlation between two time-series data sets, (1) traffic intensity and (2) Twitter data intensity. These datasets were then fed into a proposed social-aware framework, named the Traffic-Twitter Transformer, which integrated Nature Language representations into time-series records for long-term traffic prediction. Experimental results in the Great Seattle Area showed that the proposed model outperformed baseline models in all evaluation matrices. This Natural Language Processing (NLP)-joined social-aware framework promises to become a valuable tool for network-wide traffic prediction and management for traffic agencies.

4.1.1 Background

Nowadays, user demand (for both people and goods) has increased significantly. According to a report from Deloitte[45], urban freight delivery will surge by more than 40% by 2050. Cities are growing, and with this growth, are becoming more congested; driving the need for intelligent solutions to address forthcoming traffic-related challenges. Intelligent Transportation Systems (ITS) offers data-driven approaches to improve traffic performance and efficiency. Reliable and accurate traffic forecasting, one facet of ITS[18], is a critical topic that has the potential to increase roadway capacity and alleviate congestion[32]. Traffic forecast-



(a) Different land use locations

(b) Traffic pattern variations under different land-use scenarios over morning/evening-peak hours

Figure 4.1: Geographical Attributes And Mutual Relationship

ing is a spatial-temporal sequence prediction problem as shown in Figure 4.1, which involves both location adjacency and temporal impact. With regard to geographic attributes and social activities (Figure 4.1a), traffic patterns might vary significantly (Figure 4.1b). Multiple algorithms have taken these key points into consideration for predicting traffic parameters, including speed and volume[201,112,214,36,246,64]. These algorithms have made notable advancements in various aspects of traffic forecasting, including improving model accuracy and reducing computation time.

Through advancements in methods and increased accessibility of traffic data, deep learning models have outperformed conventional methods in traffic forecasting tasks[21,33]. Several of these new methods include Recurrent Neural Network (RNN), Long Short-term Memory (LSTM), Convolutional Neural Network (CNN), and Graph Neural Network (GNN). RNNs have received significant attention in handling sequential forecasting problems. However, with a deep network structure, the gradients being back-propagated have to go through consequent matrix multiplications based on the chain rule. In this case, RNN models learn to recognize sequential patterns, but the gradient tends to either explode or vanish [13]. LSTM models have the capability to address these gradient vanishing and explosion issues [76]. Therefore, many studies in the last decade have leveraged LSTM [184,199,79,125,216,186] to maintain a time-related sequence and internal memory with the loop structure. CNNs, which are powerful image processing algorithms that can effectively extract informative features, have been generalized to capture spatial relationships in traffic networks. Works from Ma et al. [122], Zhang et al. [239], and Huang et al. [81], each used CNN to detect traffic patterns from geographic figures as well as traffic time-space speed matrix and further forecast future speed. Liu et al. [116], Bogaerts et al. [15], and Ma et al. [124], further combined stacked CNNs to extract spatial features with LSTM to integrate temporal information of traffic data. However, standard CNN-based approaches are incapable of dealing with various topological structures of traffic networks. To address this issue, researchers began to train traffic networks as a graph and applied GNNs to extract patterns from graph-structure data [110,36]. Most of the aforementioned methods focused on short-term forecasting, with time periods ranging from 5 minutes to 1 hour ahead. Moreover, unanticipated complications, such as crashes and the COVID-19 pandemic, significantly affect prediction results. In this case, other meaningful auxiliary data, such as social media data, were examined to fill the gap for robust long-term forecasting.

Social media has evolved considerably in the past decade and is now extensively used to share user-generated information, sentiments, and other forms of expression [22]. Consequently, Twitter has become a powerful tool for gathering information from a reasonably large and diverse user pool. Since tweets can be retrieved with a relatively small building and maintenance costs, this data source can be treated as another type of sensor, such as loop detectors, for traffic conditions. A Seattle Mariners game, for example, may impact traffic near the T-mobile park (the stadium where the Mariners play). Those attending the game may post tweets about proximate traffic, which indirectly contributes information that has the potential to help with traffic forecasting.

Various studies have attempted to integrate social media data into transportation research. He et al.[70] examined the possibility of using rich information in online social media to improve traffic prediction. The authors analyzed the correlation between traffic volume and tweet counts with various granularities. An optimization framework was also proposed to extract traffic indicators based on transformed tweet semantics. A recent work[222], dedicated to traffic forecasting with multi-source data features, considered leveraged machine learning models with tweet semantics to predict morning traffic conditions. The experimental results showed the capability of robustly learning traffic patterns from tweets semantics when compared to other algorithms with the proposed approaches. Other studies[241,60] attempted to examine the relationships between words in tweets and traffic crashes. They identified important key terms in tweets to determine traffic incidents through tweet data mining.

Based on the previously described research, some inherent limitations are summarized to give incentive for innovation:

- **Ignore the culture impacts in prediction tasks.** Most of the previous research took common features, such as speed, volume, weather conditions, and roadway geometry, into consideration for traffic forecasting. Yet, cultural-related events (e.g., Black Lives Matter (BLM) rallies), which can severely influence traffic conditions, should also be considered.

- **Social impacts are not considered in the predicting values.** In some of the previous research, social information is considered as part of the input. However, the final output still mirrors classical traffic patterns (i.e., volume and speed), with less consideration of physical attributes combination. New interpretable traffic matrices need to be proposed and further integrated with human activities.

- **Lack of a flexible long-term traffic forecasting approach.** Prior studies mainly focused on short-term forecasting. It is hard to model long-term spatial and temporal trends even using graph-based algorithms. The graph-structure connectivity also constrains the flexibility of data aggregation for each road segment. To tackle this problem, a new, flexible comprehensive model, that incorporates temporal-spatial dependence and tweet information, is required.

4.1.2 Contributions and Organization

Inspired by the previously discussed research and their limitations, this study expands the temporal scale from predicting morning-peak traffic [222] to the whole day network-wide traffic performance. This work proposes a NLP-joined social-aware traffic forecasting model, originated from Transformer [196], with a temporal encoder as opposed to a positional encoder, and includes social media features. Tweet data and traffic data can each be fed into the proposed model to predict an accurate long-term network-wide traffic performance. Unlike previous studies, this work utilizes a more interpretable and inclusive matrix, called the Traffic Performance Score (TPS) [41], to evaluate traffic conditions. Due to the integration of social media features, which contain personal opinions, the TPS (which ranges from 0% to 100%), is a superior explanatory and comprehensive matrix to assess network-wide traffic states, rather than predicting classical metrics, such as speed or volume.

In summary, the contributions of this paper are listed as follows:

1. A correlation study is conducted to prove that it is meaningful to involve social media features in the model, with a strong correlation between traffic data and Twitter data.

2. An NLP-joined social-aware framework, Traffic-Twitter Transformer is proposed to increase robustness under various unexpected events.

3. The forecasting results rely on historical traffic patterns as well as varied social media features to improve the model's flexibility and robustness.

4. A time encoder is applied to replace the positional encoder in the originally developed Transformer. The time encoder allows the model to keep the recurrent characteristic from the sequential time-series input.

5. An ablation study shows the causality of the proposed model and how each Twitter feature impacts traffic forecasting.

The remaining section of this chapter is organized as follows: In Section 4.2, two selected datasets that capture the spatial-temporal correlations will first be introduced: the TPS dataset and the Tweet dataset. A correlation Study will also be conducted to demonstrate that tweet features could significantly contribute to traffic forecasting tasks. In Section 4.3, we propose an end-to-end architecture, Traffic-Twitter Transformer, to deal with original traffic and tweet data. Several innovation claims will be described in detail in this section. Section 4.4 presents the experimental results and related analysis. Many state-of-the-art baseline models will be implemented and compared with our proposed model among the various widely-used evaluation metrics. This chapter is concluded in Section 4.5, which summarizes the contributions of the proposed model and identifies future directions.

4.2 Data Preprocessing

A combination of the TPS dataset and the tweets dataset from May 1st, 2020 to August 31st, 2020, are analyzed.

4.2.1 Traffic and Tweet Data Collection

TPS Dataset

The TPS dataset[41] is collected from roughly 8000 inductive loop detectors deployed on the freeway network in the northwestern region of Washington State. The freeway network mainly includes several major freeways, such as I-5, I-90, I-405, and SR-520. It assigns the freeway to 106 segments and measures the average volume and average speed of each segment with 15-minute intervals for calculating TPS. TPS, the prediction target of this research, is an indicator to measure the traffic performance of the traffic network. It is a value ranging from 0% to 100%. Overall network-wide traffic condition is the best when the TPS is 100% and worsens when TPS closes to 0%.

Tweet Dataset

The tweets dataset is collected based on the location of the traffic network from the TPS dataset and Seattle's population distribution. Specifically, tweets from 14 sites are acquired, using a 5 kilometers buffer around the center of each segment from the TPS dataset, as shown in Figure 4.2. These tweets are then assigned to their respective segments, which are then

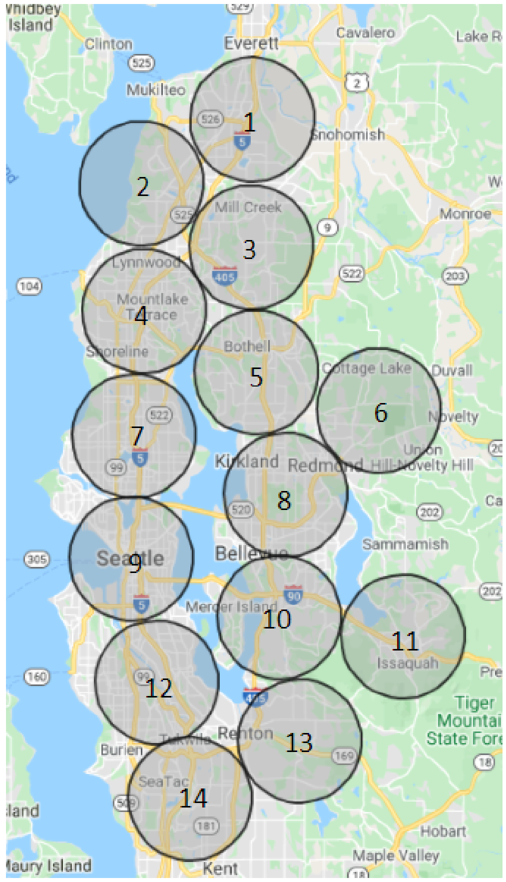


Figure 4.2: Location Segments Of Collected Tweets

applied to construct tweet feature matrices. The time interval is aligned with the 15-minute interval of the TPS dataset. Because tweets are clustered together, aggregation of the features from each tweet is required.

4.2.2 Tweet Text Processing

Three semantic features are extracted from the Tweet dataset: term frequency features, traffic-related features, and event-related features.

Firstly, term frequency reflects specific traffic conditions. Specific terms, such as "congestion" and "roadblock," may correspond to traffic states, which is valuable information for traffic forecasting. In order to convert tweet terms to a numerical matrix, we transformed them into a high-dimensional matrix of token counts. The document-term frequency matrix showed that the vocabulary size was 91812 (91812 unique tokens.) When a threshold of three was established to filter out those words with low counting frequency, the dimensions of the frequency matrix were too large and computationally burdensome. Therefore, truncated Singular Value Decomposition (SVD), a dimension reduction approach, was applied for our following computations. Truncated SVD is similar to traditional SVD methods, yet works well on sparse matrices like count and TF-IDF matrices[65]. The desired dimension for the frequency matrix was identified as $k = 100$, which explains approximately 80% of the variation as shown in Figure 4.3. To indicate the final term frequency for a segment at a certain time, all the transformed document-term frequencies were simply summed as a single value.

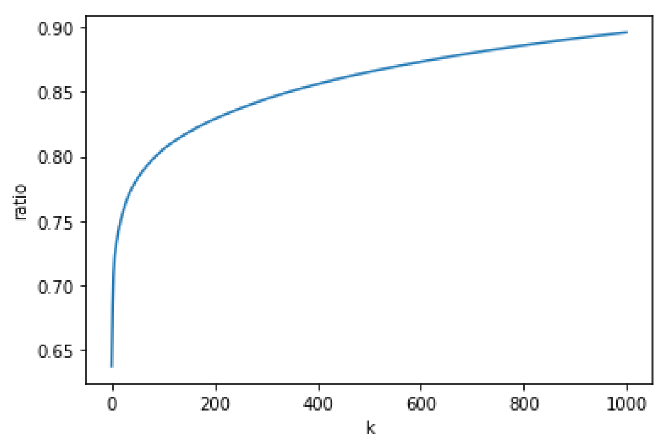


Figure 4.3: Explained Variance Ratio From K = 1 To 1000 With `TruncatedSVD`

Second, traffic crashes can directly affect transportation operations by breaking down segment-wise traffic. Therefore, traffic-related features require extraction to better represent traffic

Table 4.1: Traffic-related Keywords

police, accident, traffic, crash, road, car, vehicle, highway, driver, county, injured, injuries, scene, hospital, died, patrol, morning, happened, dead, driving, department, involved, vehicles, passenger, hit, truck, monday, left, lane, killed, struck, closed, investigation

conditions. Specifically, the number of traffic-related tweets was counted based on traffic-related keywords, listed in Table 4.1, as proposed by Zhang et al. [241].

Third, event-related features were extracted as, they too, have the potential to greatly alter traffic conditions. For example, Figure 4.4 demonstrates that June 3rd, 2020 (Wednesday) had a significantly different pattern than the same day of the two subsequent weeks (June 10th, 2020 and June 17th, 2020) in Downtown Seattle. Certain high-frequency words were discovered from Twitter data, as shown in Table 4.2. These words are closely tied to *Black Live Matters* and *Defund Seattle Police* rallies held in the Downtown and Capitol Hill neighborhoods of Seattle on June 3rd, 2020. As a result, there were additional, and unanticipated, road traffic controls implemented in and around these neighborhoods. Event-related features in this study were derived by counting the number of tweets containing event-related keywords.

Table 4.2: Event-related Keywords

blm, BlackLivesMatter, Ahmaud Arbery, Breonna Taylor, George Floyd, Jacob Blake, AllLivesMatter, protest, privilege, police, Seattlepd, Durkan, durkanresign, Anderson, mayorjenny, realdonaldtrump, seattlepd, hard, capitol, privilege

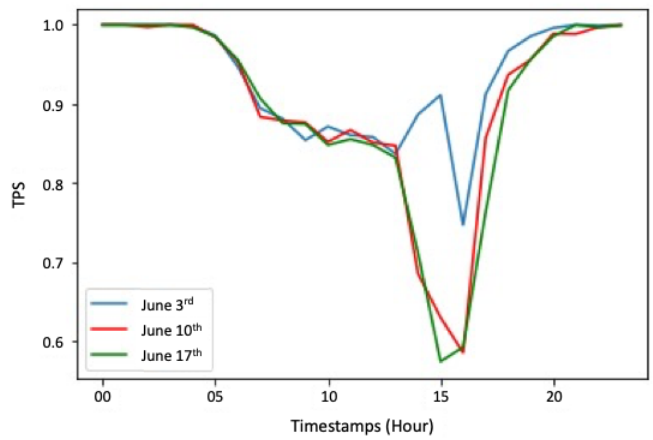


Figure 4.4: Difference Of TPS At Downtown Seattle In 3 Weeks

4.2.3 Traffic-Tweet Correlation Study

In order to confidently include tweet features as inputs to predict traffic states, it is essential to confirm whether there is a statistically significant correlation between tweets and traffic features. Before the correlation analysis, it is practical to detrend time-series data with evident periodic fluctuations.

Data Detrending

The periodic characteristic of traffic and social media patterns can be shown in Figure 4.5a and Figure 4.5b. These seasonal fluctuations can strongly affect correlation studies and need to be removed. To detrend these patterns in time-series data, the process outlined in[70] is followed. To exclude these patterns in the subsequent correlation analysis, the variation was estimated and then subtracted from both traffic and tweet data to get the detrended version. with regard to the traffic data, the variation component $s_{h,d}$ can be formulated as:

$$s_{h,d} = \frac{\sum_{\{t|(h,d)\}} \mathbf{v}^t}{|\{t|(h,d)\}|} \tag{4.1}$$

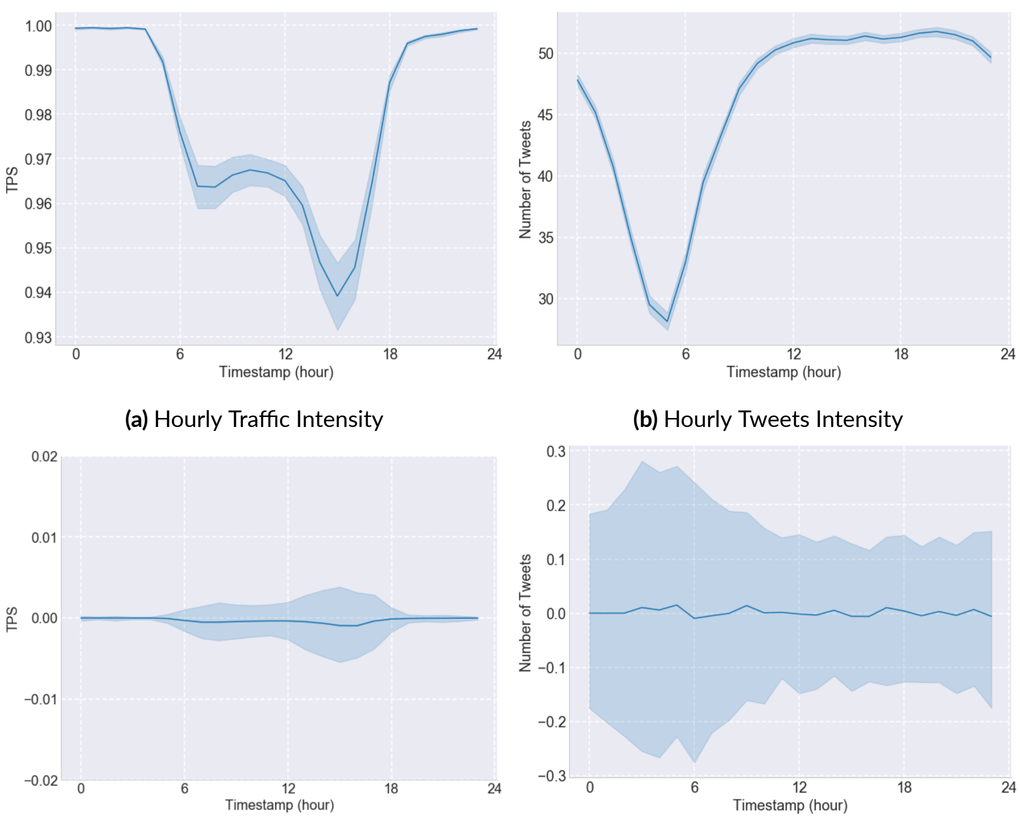
where $\mathbf{v} \in \mathbb{R}^T$ denotes the TPS data, T is the total number of time stamps, and its t^{th} element $\mathbf{v}^t$ is the TPS averaged over all detectors in timestamp t. Besides, $|\{\cdot\}|$ denotes the number of elements in the set and pair (h, d) consists of $h = 0, ..., 23$ and $d = 0, ..., 6$.

Thus, the detrend TPS $\mathbf{v}' \in \mathbb{R}^T$ can be defined as followed:

$$\mathbf{v}' = \mathbf{v}^t - s_{h,d}. \tag{4.2}$$

The Twitter data $\mathbf{c} \in \mathbb{R}^T$ follows the same procedure to access the detrended version $\mathbf{c}'$.

Figure 4.5 shows the comparison of the original data and the detrended data for both traffic



(a) Hourly Traffic Intensity

(b) Hourly Tweets Intensity

(c) Detrended Hourly Traffic Intensity

(d) Detrended Hourly Tweets Intensity

Figure 4.5: Traffic And Twitter Data Detrending Process

and tweet intensities. The recurrent patterns in Figure 4.5a and Figure 4.5b can be clearly observed. After detrending by Equation 4.2, both their original periodic patterns, as shown in Figure 4.5c and Figure 4.5d.

Correlation Analysis

As a first step towards predicting traffic intensity using Twitter data, the correlation between social activity and the traffic intensity measure is investigated. The relationship between average TPS and average tweets counts across the days of the week is illustrated in Figure 4.6. The Pearson correlation between TPS and tweet counts is -0.223, indicating a negative relationship between the two variables, with TPS increasing as tweet count drops.

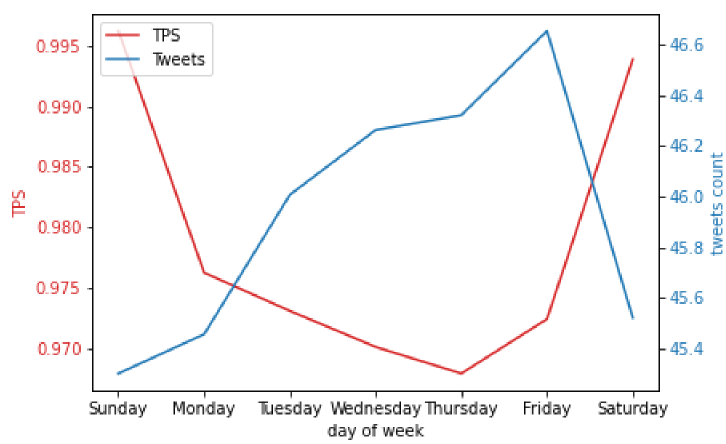


Figure 4.6: Traffic And Tweets Intensity Visualization By Day Of Week

The correlation analysis took the time lag impact of TPS and tweet counts into account, which may be identified using historical data to predict the current state. Figure 4.7 shows the cross-correlation results between the current detrended traffic intensity $\mathbf{v}'$ and the detrended tweets activity intensity $\mathbf{c}'$ over the previous 24 hours with a time resolutions of 1 hour. The

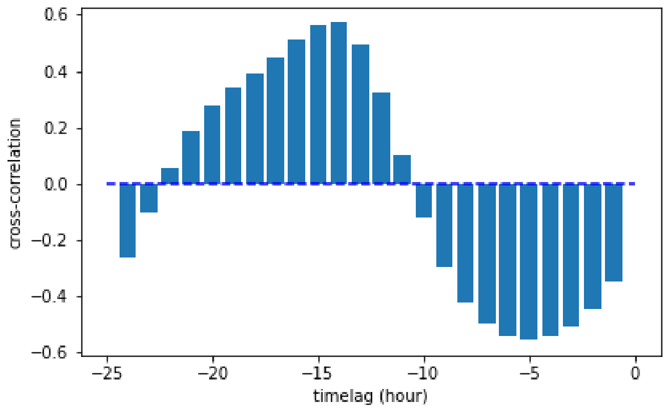


Figure 4.7: Cross Correlation Between Traffic Tweets Intensity

height of the blue bar at time lag δt represents the correlation between $\mathbf{v}'^{t}$ and $\mathbf{c}'^{t-\delta t}$. The cross-correlation is negative when the time lag is within 10 hours and shows a sine wave over time. This implies that when Twitter activity is lower than usual, traffic performance on the road network is higher (better) than the average in the near future, assuming a 10-hour time lag.

Further, the significance of the correlation between the two time-series data sets is evaluated by combining historical traffic intensity ($\mathbf{v}'^{t-\delta t}$), and current and lagged tweet intensity ($\mathbf{c}'$ and $\mathbf{c}'^{t-\delta t}$), to the auto-regressive model for current traffic intensity prediction [173]. Specifically, current traffic intensity $\mathbf{v}'$ can be predicted by the following linear regression model.

$$\mathbf{v}'^{t} = \alpha + \beta_1 \mathbf{v}'^{t-1} + \beta_2 \mathbf{c}'^{t} + \beta_3 \mathbf{c}'^{t-1} \tag{4.3}$$

where α is the intercept, $\beta_1, \beta_2, \beta_3$ are coefficients associated with traffic and Twitter data with various lags. The results are shown in Table 4.3 with an R^2 as 0.776. The detrended tweets count with a one-hour time lag ($\mathbf{c}'^{t-1}$) prove to be statistically significant with a p-

Table 4.3: Results Of Linear Regression For Correlation Study

Coefficient	Value	Std. Error	p-value
α	-0.0013	0.000	0.90
β_1	0.8809	0.009	0.00*
β_2	-0.0640	0.004	0.095
β_3	-0.0844	0.041	0.041*

value of 0.041. The negative correlation between TPS and tweets data is consistent with the previous analysis.

The results of correlation analysis in Table 4.3 confirm a statistically significant relationship between TPS and tweet characteristics. As a result, it is meaningful to incorporate twitter features into the proposed model for network-wide traffic forecasting.

4.3 Methodology

Inspired by a well-known NLP model, Transformer[196], the original architecture has been modified for the task of long-term network-wide traffic forecasting. The proposed Traffic-Twitter Transformer model, as shown in Figure 6.1, consists of three main blocks: Data fusion block, Traffic-Twitter Transformer encoder block, and decoder block. Before diving into the detail of the methodology, we first describe the preliminaries as follows.

4.3.1 Preliminaries

Changes in traffic conditions for any reason, can propagate (shockwave) congestion backward and forward and potentially impact connecting road segments[146]. Thus, to consider

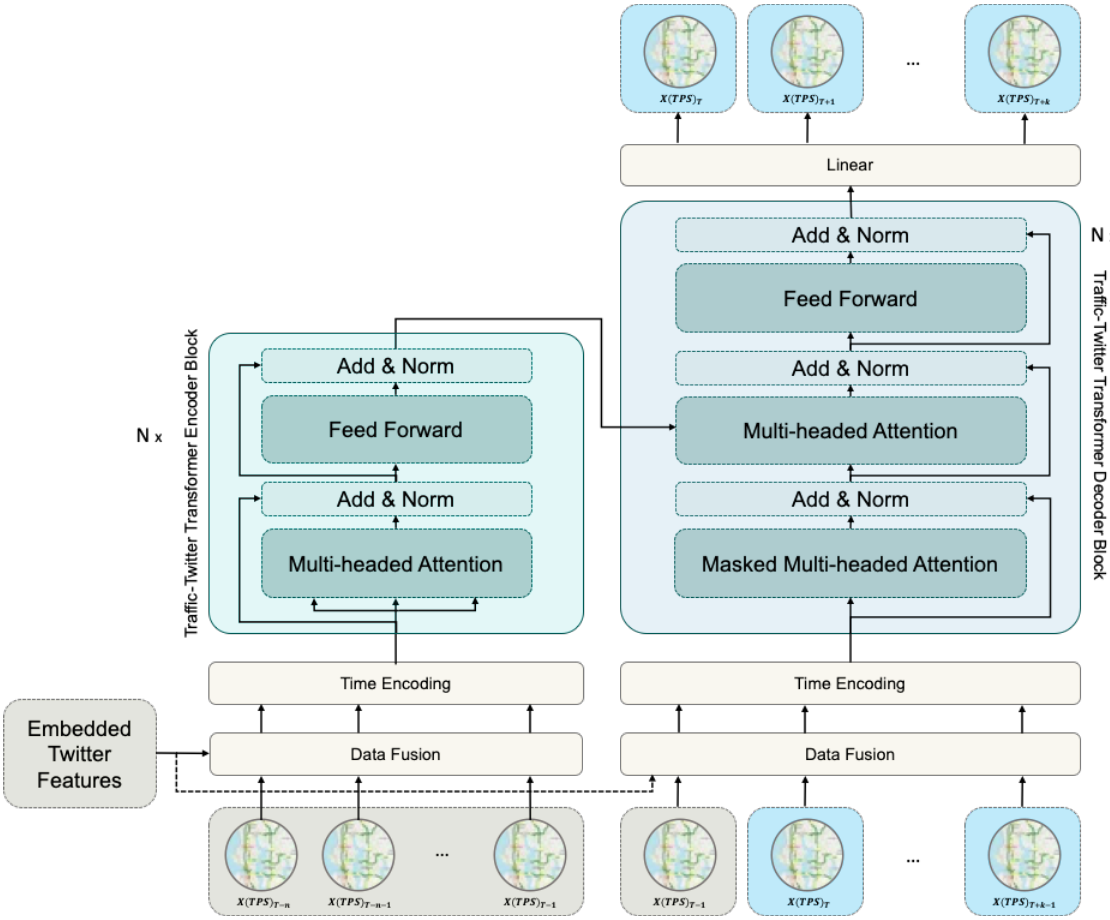


Figure 4.8: Traffic-Twitter Transformer architecture

robust traffic forecasting, network-wide traffic data matrices must be utilized[39]. For this work, a traffic data matrix is constructed through historical data collected from segment sensors across the entire network. The matrix consists of N previous timestamps before time T in M segments. Each element x_t^m symbolizes the traffic condition in m^{th} segment at t^{th} timestamp:

$$X_M^T = \begin{bmatrix} x_1^{T-N} & x_2^{T-N} & ... & x_M^{T-N} \\ x_1^{T-N-1} & x_2^{T-N-1} & ... & x_M^{T-N-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{T-1} & x_2^{T-1} & ... & x_M^{T-1} \end{bmatrix} \tag{4.4}$$

Meanwhile, the Twitter feature matrix is constructed using tweets from May 1, 2020, to August 31, 2020. First, tweets are collected and assigned to the nearest segment based on the distance between the center of the bounding circle and the segment. Then the remaining tweets are filtered for traffic- and event-related keywords. Ultimately, of the 687,803 original tweets, 150,736 traffic-related tweets remained, and 46,276 event-related tweets remained. The matrix $C \in \mathbb{R}^{N\times(K\times M)}$ can be defined as the following equation:

$$C_{K\times M}^T = \begin{bmatrix} C_{1M}^T & C_{2M}^T & ... & C_{KM}^T \end{bmatrix} \tag{4.5}$$

where $C_{1M}^T, C_{2M}^T, ..., C_{KM}^T \in \mathbb{R}_M^N$. Each of the submatrices represents one type of tweet count matrix consisting of N previous timestamps before time T in M segments. More specifically, traffic-related and event-related tweets are collected in the Greater Seattle area; therefore, N should be 2 to describe two types of features that were included in this study.

The research problem can be described as: given the historical traffic data matrices and Twitter feature matrices with time series $[T-N, T-N+1, , T-1]$, can the network-wide traffic condition $\hat{Y}_M^T$ in M segments at timestamp T be predicted? The procedure developed in this work to forecast network-wide traffic conditions can be presented as:

$$\hat{Y}_M^T = G(X_M^T, C) \tag{4.6}$$

where G is the Traffic-Twitter transformer, the primary algorithm proposed in this study. The objective is to minimize the difference between the sequence of predicted TPS $\hat{Y}_T^M$ and the ground truth data Y_T^M.

4.3.2 Data Fusion and Time Encoder Component

Based on the designed architecture as shown in Figure 6.1, traffic-related features and Twitter features are fused together at the beginning. Here column-wise concatenation is used to transform them into a long vector $\bar{X}_t$ in the data fusion block, as follows:

$$\bar{X}_t = X_t \oplus C_1 \oplus C_2 \oplus C_3 \tag{4.7}$$

where C_1 is the tweet term frequency feature, C_2 is the traffic-related tweet feature, C_3 is the event-related tweet feature, and $\oplus$ represents the fusion process (here use concatenation process as an example). Each aggregated long vector represents multiple features in a specific timestamp, such as a TPS score, traffic volume, average speed, and corresponding Twitter features.

The sequence of fused data is then encoded with the time encoder. This series of data is entered into the Traffic-Twitter Transformer model concurrently, unlike the RNN-based model. The input elements in the original transformer model only notify the model about the input order depending on their index in the sequence. To account for the order of the traffic information in the input sequence and also maintain the temporal recurrent characteristic[3], a time-encoded vector is added to each input data. These vectors are generated from the time encoder, which replaces the positional encoder in the original Transformer architecture.

The time encoder generator can be separated into two parts. The first requires the generation of the original positional encoder features as formulated:

$$\tau_t(k) = \begin{cases} sin(pos/10000^{k/d_\tau}) & \text{, k is even} \\ cos((pos/10000^{(k-1)/d_\tau}) & \text{, k is odd} \end{cases} \quad (4.8)$$

where $\tau_t(k)$ is the k^{th} feature, pos is the position in the sequence and d_τ is the dimension of the encoder features. For the 2D positional encoding matrix, the size of the row is the length of the input sequence, and the number of columns would be equal to the number of input features. For example, 12-steps of historical data are used with 64 different features to predict future traffic conditions. The shape of the positional encoding matrix would be $[12 \times 64]$.

4.3.3 Traffic-Twitter Encoder-decoder Architecture

Meanwhile, the second part of the time encoder requires the extraction of seven-dimensional time features and combining them into the time encoder. The normalized seven-dimensional time features are *minute*, *hour*, *dayofweek*, *day*, *dayofyear*, *month*, and *weekofyear*. Specifically, the input embedding sequence is concatenated with the positional encoder features and the additional time features as followed:

$$\tilde{X}_t = W(\bar{X}_t \oplus \tau_t \oplus T_t) \quad (4.9)$$

where $W \in \mathbb{R}^{d_\tau \times (2d_\tau + Dim_t)}$ is the weight matrix, Dim_t is the temporal feature dimension, T_t is the time feature, and $\oplus$ is the symbol of concatenation. Finally, the processed input sequences are fed into the encoder block to further extract inherent traffic states.

Then, the input vectors are sent to a Multi-headed Attention cell in the encoder block, which accesses all input sequences for hints that might aid in better encoding traffic semantics. It is an approach that inherently can capture meaningful representation from complex vectors. Different learnable neural networks can project the encoded sequence of input into three matrices (Query (Q), Key (K), and Value (V)) in the attention mechanism:

$$Q = \tilde{X}_t W_Q \tag{4.10}$$

$$K = \tilde{X}_t W_K \tag{4.11}$$

$$V = \tilde{X}_t W_V \tag{4.12}$$

Specifically, the key matrix comprises representations of road networks. And the dot product of the query and key matrix in Equation 4.13 is essentially a matrix of similarity scores. The dot computation may be thought of as an aggregation procedure for combining spatial-temporal information. Besides, the final attention weights provide the flexibility to integrate spatial-temporal representations:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4.13}$$

where $Q, K, V \in \mathbb{R}^{N \times d_k}$, and d_k can be utilized as a scaling factor which provides a more stable gradient to the model[196]. It can be further expanded to multi-headed attention. By repeating the attention mechanism h times, the multi-headed output can be attained as:

$$\begin{aligned} MultiHeaded(Q, K, V) &= Concat(head_1, ..., head_h)W^O \\ head_i &= \mathit{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{4.14}$$

where $d_k = d_v = d_{model}/h$, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d_k}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. In this study, the number of heads (h) is set as 8, adhering to the recommended configuration proposed by previous research [196].

After processing through a multi-headed attention cell, the processed data is passed to a position-wise fully connected feed-forward network. The representation data generated by the encoder block is sent to the decoder block as input data.

The decoder block takes the representation data received from the encoder and the last timestamp of traffic-related data X_{T-1} as inputs to perform a step-wise prediction multiple steps ahead for future traffic conditions. A masked attention mechanism is applied to mask future positions to prevent data leakage and only allow attending the earlier positions in the output sequence. A linear feed-forward network is assigned as the last step to improve the expressiveness of the model and reshape the output to be the same as the sequence of label data. The outcome of each step is fed to the bottom decoder in the next step, and the decoder uses the inputs to predict the next timestamp results.

After developing the aforementioned technical approaches, three widely-used metrics in traffic forecasting (Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE)) were selected to evaluate the model accuracy, which can be calculated using the following equations:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.15}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.16}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{4.17}$$

where y_i and $\hat{y}_i$ represent the ground truth and predicted value respectively.

4.4 Experiments

4.4.1 Experiment Setting

Each data set was collected at a 15-minute interval in the Greater Seattle area from May 2020 to August 2020. Three months of data were used as the training dataset, and 15 days were used as the validation and the testing dataset respectively. In order to achieve the goal of long-term forecasting, methods suggested by Essien et al. [50] were used to set 12-steps ahead as a target for multi-step forecasting. Conclusively, an end-to-end long-term network-wide forecasting was executed, predicting the upcoming 12-steps ahead (3 hours ahead) given 12 historical sequences of data.

4.4.2 Experiment Results Analysis and Comparison

Based on the description of the model in Section 4.3, each input sequence $X(TPS)_i$ is a 3D matrix with the shape $B \times 1 \times (Num\_of\_Segment \cdot 3)$ where B is the batch size and $Num\_of\_Segment$ is the total number of the segments in the network. It is a matrix with 12 historical sequences of data. The last dimension $Num\_of\_Segment \cdot 3$, consists of each segment's TPS, volume, and average speed. The Twitter data were fed into the data fusion block to fuse with the traffic 3D matrix. Because the prediction model is many-to-many, the length of the output is 12. Therefore, the shape of the final output is $[B, 12, Num\_of\_Segment]$.

Comparison with Baseline Models for Network-wide Traffic Condition Prediction

Five baseline models that have the ability to predict the traffic condition of the whole network were selected to compare with the Traffic-Twitter Transformer model developed in this work:

1. **S-LSTM**: Stacked LSTM (S-LSTM)[39] is a networks with several stacked LSTM hidden layers. With the stacked-layers mechanism, the forecasting performance can enhance significantly. In the experiment, the number of layers was set to be 2, which means it is a two-layer LSTM structure.

2. **AGC-Seq2Seq**: Attention graph convolutional sequence-to-sequence model (AGC-Seq2Seq)[242] approach, unlike the LSTM model, has an encoder receiving the sequence of input and generating an encoded vector. It also has a decoder to accept this encoded vector and generate the output. It incorporates Graph Convolutional Network (GCN) and RNN modules to capture the spatial-temporal dependency.

3. **AGCRN**: Adaptive Graph Convolutional Recurrent Network (AGCRN), which combines Node Adaptive Parameter Learning module and Data Adaptive Graph Generation module with recurrent networks for the multi-step traffic prediction task[11].

4. **Original Transformer**: Transformer[196] has first developed in NLP-based research then was applied to various aspects, including transportation engineering. With the attention mechanism, the model can learn longer time-series data and perform a promising forecasting result. In the experiment, the number of heads in the multi-head attention process was set as 8; besides, the number of layers in the encoder and decoder is 6. All these hyperparameters are the same as the original setting in[196].

5. **GMAN**: Graph Multi-Attention Network (GMAN) employs an encoder-decoder architecture, in which both the encoder and the decoder are made up of numerous spatio-temporal attention blocks, to model the traffic states[246].

We applied the default settings for the above baseline models from their original studies. From a model perspective, it is important to note that, without the data fusion block (Equation 4.7), the baseline models cannot combine traffic and twitter features. Some baseline models do not support multivariate input data either. The proposed model, Traffic-Twitter, is the only model with the ability to benefit from this component in the following experiments.

Table 4.4 reflects the network-wide prediction from each of the six different models (the model proposed in this work and the five comparison models). The S-LSTM model can be considered the baseline for all deep learning-based approaches. Given a more complex architecture with an encoder-decoder structure, the AGC-Seq2Seq model performed better than the S-LSTM in all three metrics. The Transformer achieved a better performance than the AGC-Seq2Seq, and the GMAN model exhibits performance comparable to Transformer. This suggests the attention mechanism is capable of learning long time-series patterns and of achieving compelling results. The Traffic-Twitter Transformer presents the best score across

Table 4.4: Network-wide Overall Performance Comparison Of The Proposed Model With Baseline Models

Model	MSE	MAE	MAPE
S-LSTM	0.0028	0.0207	2.9168%
AGC-Seq2Seq	0.0025	0.0202	2.7929%
AGCRN	0.0026	0.0201	2.7358%
Original Transformer	0.0021	0.0160	2.3315%
GMAN	0.0021	0.0151	2.2527%
Our Model*	**0.0019**	**0.0135**	**2.0141%**

* Our Model: Traffic-Twitter Transformer

all evaluation metrics, suggesting that it is more suitable for application in comparatively non-stationary traffic conditions.

The step-wise prediction results were further investigated to compare the performance in time steps ahead as follows: 15 minutes, 60 minutes, 120 minutes, and 180 minutes ahead. The results are shown in Table 4.5. The S-LSTM model displayed a competitive prediction result for the 15-minute ahead task; however, its prediction decreased significantly when additional timestep prediction processing was applied. This was anticipated because the S-LSTM can only access the latest former hidden state and cell state, which presents embedding information of previous steps. Thus, the prediction performance predictably decreases when processing a long-term prediction. The AGC-Seq2Seq model produced a similar finding but with slightly better outcomes. Its architecture, with a more reliable encoder-decoder structure, was likely the primary driver of the improvement.

Surprisingly, AGCRN outperformed all other models in a 15-minute prediction chal-

Table 4.5: Step-wise Performance Comparison Of The Proposed Model With Baseline Models

Model	15 min			60 min			120 min			180 min		
	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE	MSE	MAE	MAPE
S-LSTM	0.0026	0.0195	2.0137%	0.0028	0.0225	2.2571%	0.0033	0.0232	2.5342%	0.0085	0.0238	5.0719%
AGC-Seq2Seq	0.0023	0.0190	2.0212%	0.0025	0.0199	2.2250%	0.0026	0.0217	2.3297%	0.0032	0.0228	2.8438%
AGCRN	**0.0012**	0.0136	1.5375%	0.0023	0.0190	2.2901%	0.0029	0.0221	2.5123%	0.0033	0.0233	3.2968%
Transformer	0.0015	0.0138	1.6095%	0.0019	0.0143	1.6610%	0.0021	0.0169	1.8039%	0.0024	0.0175	2.0876%
GMAN	0.0016	0.0141	1.7313%	0.0019	0.0158	1.8033%	0.0021	0.0159	1.8035%	0.0023	0.0164	2.0276%
Our Model*	0.0013	**0.0089**	**1.3878%**	**0.0017**	**0.0105**	**1.4299%**	**0.0018**	**0.0132**	**1.5645%**	**0.0020**	**0.0136**	**1.7078%**

* Our Model: Traffic-Twitter Transformer

lenge regarding MSE performance. For long-term occupations, however, its forecast accuracy steadily decreased. In contrast, the Transformer model supplies all historical data concurrently and determines which timestamp data is crucial by giving those inputs weights, thus generating a more competitive result. With a Transformer-like architecture that includes an encoder-decoder structure and a multi-attention mechanism, GMAN produced a comparable result. The model proposed in this work, the Traffic-Twitter Transformer, delivered the best performance across each of the four time steps. The proposed enhancement in the architecture, taking both traffic data and Twitter features into consideration, shows an effective result in complicated network-wide long-term traffic forecasting.

Ablation Study of Twitter Features and Time Encoder

In this experiment, each Twitter feature and time encoder are removed sequentially and the performance decrements are computed to determine how much each of them contributed to the model prediction. The results are shown in Table 4.6.

The performance of the proposed model, the Traffic-Twitter Transformer, suffers the most significant MSE decrement from 0.00186 to 0.00203 when the traffic-related characteristics were disregarded. The removal of the time encoder also significantly impacts overall MSE performance. Contrarily, the MSE suffers a relatively small decrement when either the frequency term or event-related features are ignored. These results suggest the importance of the inclusion of social media data, particularly tweets with traffic-related semantics. They also validate the replacement of the original positional encoder with the time encoder. There are three critical reasons for this: (1) The frequency-term features reflect a pattern of daily activity that combines all types of tasks, including current traffic states, none of which has a strong correlation to traffic conditions; (2) event-related features have time-sensitive temporal and spatial characteristics, which have the potential to help improve particular cases, but less than the traffic-related features, which mainly focus on traffic states in the network; and (3) the temporal features provide more crucial information in traffic forecasting tasks than positional indices.

Table 4.6: Ablation Results Of Removing Particular Feature

Description	MSE
Remove event-related feature	0.00195
Remove document-term frequency	0.00199
Remove traffic-related feature	0.00203
Remove time encoder	0.00204

* Proposed Traffic-Twitter Transformer MSE: 0.00186

4.5 Chapter Conclusion

A novel NLP-joined social-aware transformer model, Traffic-Twitter Transformer, is proposed in this paper for network-wide traffic forecasting. The contributions concentrate on five aspects:

1. A traffic and Twitter data integrated structure is proposed to accommodate external interventions.
2. Real-world datasets were used to evaluate the proposed method.
3. Both spatial and temporal features are fused in an end-to-end architecture to improve the model robustness.

4. A time encoder is designed to replace the positional encoder to retain time dependency from data with strong temporal characteristics.

5. Twitter features are summarized with great potential to help improve traffic condition prediction.

According to the correlation analysis, it is clear that Twitter features can be included as important factors that can impact traffic conditions, especially when unforeseen events happen. The experiment results reveal that Traffic-Twitter Transformer is the most accurate model in predicting the overall network-wide traffic performance as compared to five competitive comparison models. The proposed model also delivers superior results in all timestep-ahead predictions, which suggests the Traffic-Twitter Transformer can accommodate complicated spatial dependency and expand the ability to model long-term temporal dependence. A potential negative impact of this research is the time-sensitive tweet data. If some unanticipated events might occur in a short time period with less discussion on Twitter society, the model may not benefit from these additional tweet features.

Future works can be divided into three aspects: (1) Twitter dataset can be further investigated to extract meaningful semantics, (e.g., local sports events); (2) more traffic network data sets can be used to evaluate the proposed model and validate its generalizability; and (3) self-supervised learning and other representation learning algorithms may be used to learn meaningful embeddings, which can help with a variety of downstream tasks, including traffic forecasting.

Part III

Prediction with Representation Learning

5

An Incremental Learning-based Framework for Non-stationary Traffic Representations Clustering and Forecasting

5.1 Overview

To curb the growth of COVID-19, many rules, including a work-from-home policy, were issued in 2020. While these limits successfully prevented the virus's transmission, they completely altered original mobility patterns, resulting in considerable reductions in travel time and vehicle miles traveled. Under this non-stationary data stream, the US Department of Transportation struggled to anticipate future traffic conditions. Obviously, two essential challenges need to be addressed immediately: 1) it is challenging for transportation agencies to learn representative traffic patterns from the continually changing traffic circumstances. And 2) when and how should the forecasting model be updated to learn new patterns without forgetting previous tasks? We proposed an incremental learning-based framework for non-stationary data clustering and forecasting in transportation scenarios to tackle the issues mentioned above. It is a dual-module architecture that includes a Temporal Neighborhood Clustering module and an Incremental Learning module. The objective of the first component is to dynamically detect the optimal boundary for clustering statistically similar neighbors by enlarging both the in-group similarity and between-group dissimilarity. The second module applies the online-Elastic Weight Consolidation (EWC) approach, which is commonly used in image classification tasks but rarely in regression models, to learn new tasks and avoid catastrophic forgetting, which is a typical occurrence while training neural networks with multiple tasks. Experiments on the Greater Seattle Area employed loop detector data collected in 2020 yielded reliable prediction performance in both robustness and accuracy. The dual-module framework can generate promising results from pre-COVID-19 to post-COVID-19 time frames. This framework would aid government agencies and the general public in developing long-term policies and strategies for post-pandemic intelligent transportation systems.

5.1.1 Background

Multiple affecting factors, including accidents and special events, significantly impact traffic states[188]. The COVID-19 pandemic is one of the most influential phenomenons that reshapes overall urban mobility patterns. Traffic performance in the major cities worldwide can hardly return to its original levels even in the current post-pandemic era[55]. Accordingly, the constantly changeable spatial-temporal mobility patterns aggravate the difficulty of traffic forecasting[186]. Accommodating non-stationary data streams and learning when and how to update the trained model has become a critical topic for transportation agencies.

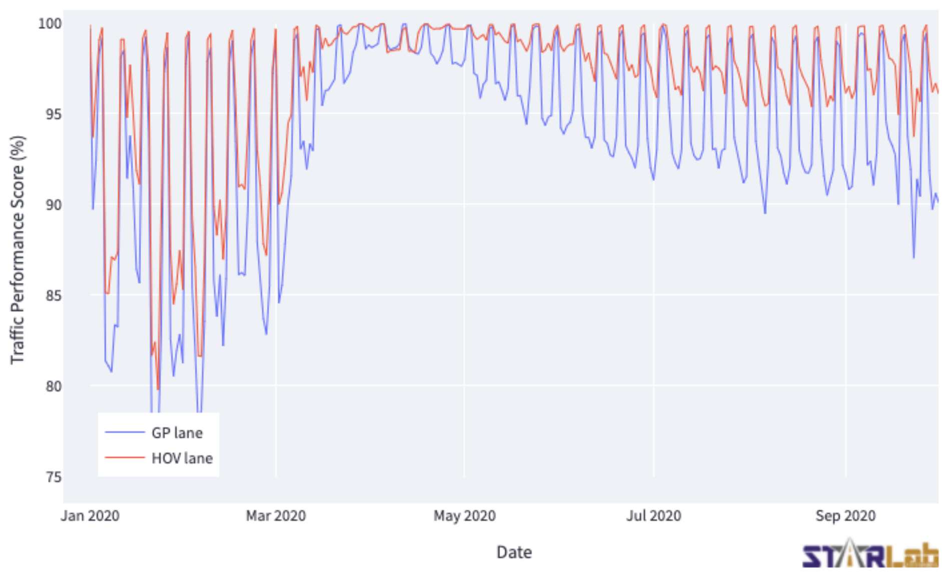


Figure 5.1: Non-stationary Traffic Patterns In Seattle During Pandemic[192]

Traffic prediction has become an active research topic over decades, notably in recent years with the growth of artificial intelligence[123]. Traffic prediction approaches have been gradually changing from conventional statistical models to data-driven machine learning-based methods as the volume of traffic data and computing capabilities have increased exponentially[201]. However, most of the algorithms were applied to relatively stable conditions without considering dramatically unexpected events in the real world that can directly affect the statistical distribution of the experimental data. Few researchers have proposed fine-tuning-based approaches to deal with shifting patterns[40]. They may still have a high probability of suffering from catastrophic forgetting problems.

Deep learning algorithms have yielded several advancements in a variety of fields but are still plagued by catastrophic forgetting issues. During the fine-tuning process, parameters in trained neurons will be adjusted to fulfill the current objective: minimize the loss of the given loss function, which is one of the primary causes of catastrophic forgetting[157]. Simply said, the direction of the gradient while calculating backward-propagation on "task B" might be totally opposite to the one from the original model that trained on "task A"[118]. The model will gradually forget the previously trained tasks throughout the retrained process. In this case, "Stability-Plasticity Trade-off" is presented to balance the flexibility of acquiring new knowledge and the stability of consolidating what models have already learned[133].

Over the past five years, incremental learning algorithms emerged as one of the most representative methods in solving the problem of stability-plasticity trade-off[93,7,232]. Still, most of them paid more attention to image classification fields, including handwriting recognition[145]. There is still a considerable gap in applying incremental learning algorithms to regression tasks, especially in transportation scenarios. This vacancy might become the primary constraint for real-world regression applications, such as traffic forecasting. Besides, the series of learning tasks were well-defined and provided in the previous incremental learning research, which is impractical in regression tasks because the time-series data is collected continually without splitting.

5.1.2 Contributions and Organization

Traffic forecasting is a classical regression task that attempt to predict future traffic conditions. Multiple variables influence traffic performance, including accidents, weather, and special events. The pandemic of COVID-19 is an example of a event that has a significant impact on travel patterns. This makes it harder to comprehend the ever-changing nature of network-wide traffic and to foresee how it will behave throughout the epidemic. To solve the challenges mentioned earlier in regression tasks and accommodate non-stationary traffic, we propose a dual-module architecture, which combines a Temporal Neighborhood Clustering module and an Incremental Learning component, in this paper for dynamically detecting the appropriate splitting points also learned the sequence of tasks without forgetting. Meanwhile, the framework is capable of taking multivariate into account to make the design more robust and versatile. To sum up, the contributions of this paper are listed as follows:

1. A novel clustering method is proposed to split the whole non-stationary dataset into several pieces properly.

2. The applied incremental learning model considers spatial-temporal traffic features to learn patterns in each task without forgetting.

3. The incremental learning-based technique is successfully utilized to solve the regression tasks in traffic forecasting problems.

The remainder of this chapter is structured as follows: In Section 5.2, we explain the problem and present the dual-module architecture, which includes a temporal neighborhood clustering module and an incremental learning module. Experimental setups are presented in Section 5.3, followed by assessment outcomes and prediction performance comparison among all baseline frameworks. We end with some insights and point out possible future studies to develop ideas to suit non-stationary traffic patterns in Section 5.4.

5.2 Methodology

5.2.1 Problem Statement

The collected sequential non-stationary data is presented as $X_{T_K}^M$ in M segments since the current timestamp is T:

$$X_{T_K}^M = \begin{bmatrix} x_{T-K}^1 & x_{T-K}^2 & \dots & x_{T-K}^M \\ x_{T-K+1}^1 & x_{T-K+1}^2 & \dots & x_{T-K+1}^M \\ \vdots & \vdots & \ddots & \vdots \\ x_{T-1}^1 & x_{T-1}^2 & \dots & x_{T-1}^M \end{bmatrix} \tag{5.1}$$

Each element $x_{t_k}^m$ represents the traffic state in m^{th} segment at t_k^{th} timestamp. Here the traffic state represents the average traffic volume and it can be further concatenated by average speed, and Traffic Performance Score (TPS)[41]. The number of rows in $X_{T_K}^M$ can be extremely large depending on when the data started to collect.

Due to unexpected events (i.e. COVID-19), the original long-term traffic patterns would be non-stationary and the data distribution can be shifted significantly. To keep the similar data distribution between training and testing data, we will feed the original traffic states $X_{T_K}^M$ as an input into the Temporal Neighborhood Clustering module G. This module can help split the original dataset into several subdatasets (i.e. sub-groups) $(D_1, D_2, ..., D_N)$ to enlarge the in-group similarity and the between-group dissimilarity. Specifically, we will use

KL-divergence to measure the divergence of one probability distribution from another:

$$\{D_1, D_2, ..., D_N\} = G(X_{T_K}^M) \tag{5.2}$$

The subdatasets can be recognized as a series of different tasks. They will be utilized sequentially to train the incremental learning-based model I. Meanwhile, an Elastic Weight Consolidation (ewc) module will also be involved to store the diagonal Fisher information from previous tasks. The incremental learning-based model can be initialized as a model (I_0) that is well-suited to time series data, such as Recurrent Neural Network (RNN) or Long Short-term Memory (LSTM). And the initialized EWC module (ewc_0) can be assigned as $None$. Therefore,

$$I_i = I_{i-1}(D_i, ewc_{i-1}) \tag{5.3}$$

$$ewc_i = ewc_{i-1}(I_i) \tag{5.4}$$

where $i \in \{1, 2, ..., N\}$. Both I and ewc would be updated according to the current new task and the previously learned tasks. The mathematical equation will be described in the following section. After learning from each task, the final incremental learning model I_N is capable of predicting traffic states in each task (D_i):

$$\hat{Y}_{T_{D_i}}^M = I_N(D_i) \tag{5.5}$$

Overall, the objective of the proposed framework is to split the original dataset properly and minimize the difference between the predicted traffic states $\hat{Y}_{T_{D_i}}^M$ in each subdataset D_i and the ground truth data $Y_{T_{D_i}}^M$.

5.2.2 Dual-module Architecture Description

The proposed framework, as shown in Figure 6.1, consists of two main components: the Temporal Neighborhood Clustering module and the Incremental Learning component. Each of them is well-described as follows.

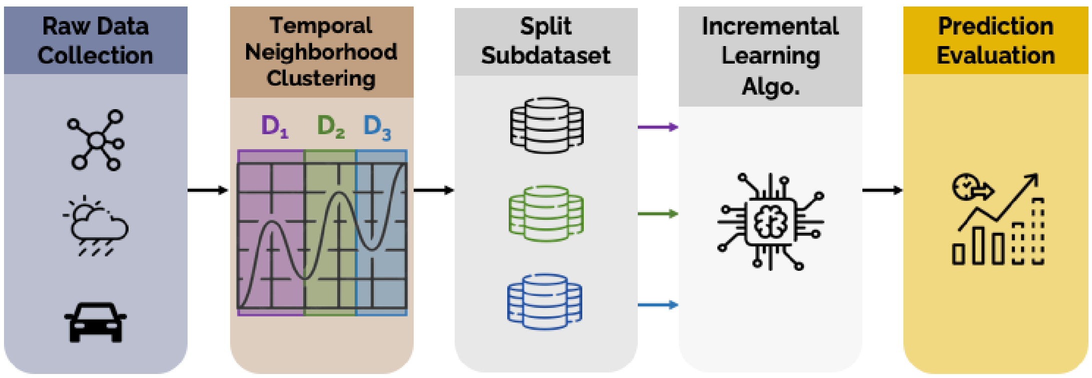


Figure 5.2: Incremental Learning-based Architecture For Traffic Forecasting

Temporal Neighborhood Clustering Module

It is vital to train a high-performing deep learning model and test it against data that come from the same target distribution. However, the original dataset gathered in the real world seldom fits the norm. We divide the non-stationary dataset into N sections to meet this fundamental condition as much as possible. The value of N here will be set based on the domain knowledge. We observed that there were approximately five different traffic phases in 2021. This hyperparameter will be investigated to find an intelligent way to assign it for future work. The optimization issue can be expressed as determining how to partition the time-series data into N sections while maximum between-group dissimilarity:

$$max \sum_{1 \leq i \neq j \leq N} D_{KL}(D_i, D_j) \tag{5.6}$$

where D_{KL} is a function to calculate KL-divergence between two samples. Other distance estimating approaches can further replace it. In Equation 5.6, N is a hyperparameter to define the number of subdatasets. In future work, the number of subdatasets can also be solved in the dynamic programming process. To prevent the subdataset from becoming too small, we also pre-split the original dataset. The length of each pre-split dataset is half a month. It would be the minimum length of each final-split dataset. Specifically, we solve this optimization problem with a dynamic programming approach. The candidate with the maximum distance between-group will be selected as splitting points to divide the original dataset into N pieces. We can get all combinations of splitting points and the sum of the distances. For instance, N is assigned as 3, which means the original dataset will be split into three parts. Given the length of half a month as the pre-split length, the total number of pre-split datasets is set as 10. After processing through the Temporal Neighborhood Cluster, there are two candidate splitting-points sets: [0, 2, 7, 10] and [0, 3, 6, 10] with the sum of the distance between groups 300 and 100, respectively. We will select the first candidate set as a splitting point set because of the larger distance.

Incremental Learning Module

According the original EWC paper[93], the loss function that we need to minimize in EWC is:

$$\mathscr{L}_N(\theta^*) = \mathscr{L}_N(\theta) + \frac{1}{2} \sum_i \big(\sum_{t<N} \lambda_t F_{t,i} \big) (\theta_i - \theta^*_{N-1,i})^2 \tag{5.7}$$

where $\mathscr{L}_N(\theta)$ is the current lost for task N only, λ_t is a task-specific hyperparameter for task t, i represents each i^{th} parameters in the model, and $F_{t,i}$ is the diagonal Fisher information with respect to the $i^t h$ parameter, which is calculated by previous subdatasets. Overall, EWC approach can sort out a solution for all tasks. For example, in Figure 5.3, the area of each ellipse-shaped area is a optimum solution θ_i with acceptable errors for task i. The red-circled area, which is overlapped by every ellipse, is the optimal goal to deal with all tasks with acceptable performance.

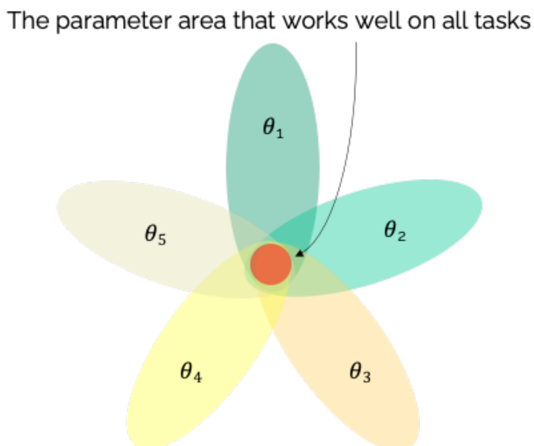


Figure 5.3: General Solution Among All Tasks[192]

However, the whole previous tasks need to be stored to calculate the relevant F_i, which is really memory-consuming. In the Equation 5.7, the F is calculated by the sequence of sum $\lambda \cdot F_n$ where $n \in \{1, 2, ..., N-1\}$. Fortunately, the online-EWC[167] was proposed to slightly revise the Equation 5.7 as follows:

$$\mathscr{L}_N(\theta^*) = \mathscr{L}_N(\theta) + \frac{1}{2}\sum_i \lambda F^*_{t-1,i}(\theta_i - \theta^*_{N-1,i})^2 \tag{5.8}$$

where the fisher information is initialized as $F_1 = \gamma \cdot F_1$ and $F^*_N = \gamma F_{N-1} + F_N$ for task N, $\gamma(<1)$ is a hyperparameter help retain associated term from the previous learned tasks $N-1$. With a slightly revision from in Equation 5.7 to Equation 5.8, the less memory is required to store the diagonal Fisher information with respect to previous tasks and still provide a promising result according to Schwarz et al. [167].

5.3 Experiments

5.3.1 Experiment Setting

The TPS Dataset[41], which is hosted by STAR Lab at the University of Washington, covers TPS, traffic speed, and traffic volume for four highways (I-5, I-90, I-405, and SR-520) in the greater Seattle region at 15-minute intervals from 2020. Here we selected the whole-year data in 2020. The Traffic Performance Score is an evaluation index that considers each segment's length, volume, and speed, ranging between 0%(0) to 100%(1). 0 means the worst case of traffic states (congestion), and 1 represents a segment with free-flow speed. It could become a more interpretable matrix for public users. And it is also selected as a target parameter that will be predicted in this study.

5.3.2 Result of Temporal Neighborhood Cluster Module

According to the module description in the Sec. 5.2, a whole-year original data is pre-split into 24 parts, which we defined the minimum length of the final-split would be half of a month. We believe that a significant change in the transportation field would become a trend if it remained for at least two weeks. The pre-defined value can also help us prevent an over-splitting issue. In the experiment, the number of the subdataset N is set as 5. Figure 5.4 shows a reasonable result that successfully splits the original dataset properly. For instance, the first

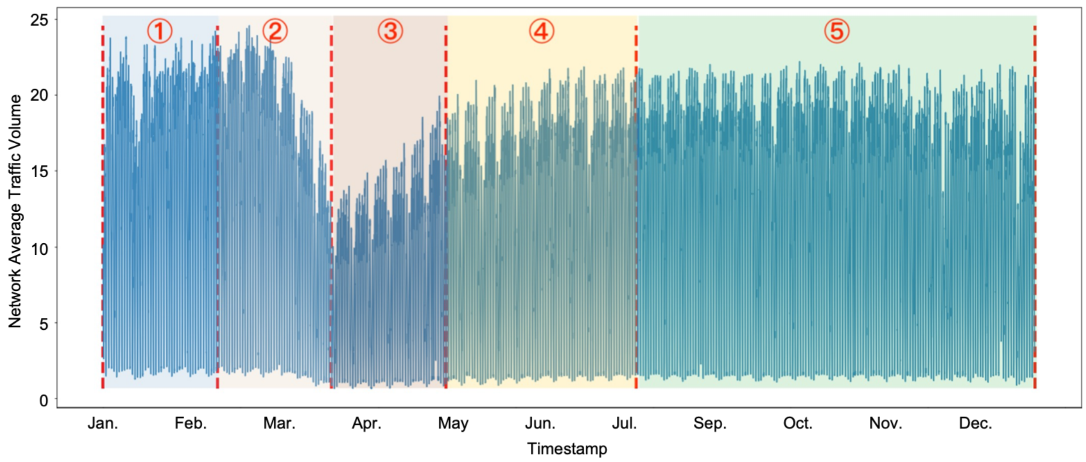
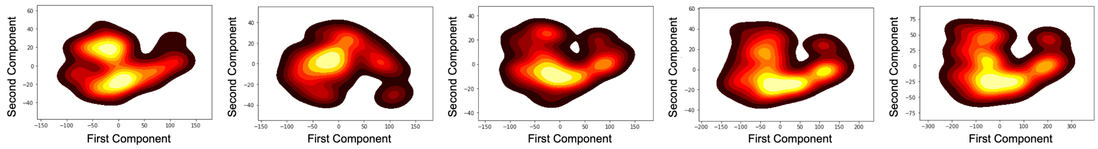


Figure 5.4: Split Non-stationary Traffic Patterns In Seattle During Pandemic Into Five Parts[192]

splitting point is around Feb. 16th, 2020, when some tech companies started a work-from-home policy in Washington State. The second splitting point is at the end of March when the Washington state government announces the "Stay Home, Stay Healthy" order. The third splitting point is on May 11th, close to the reopening of the restricted 25% capacity limit indoor dining. The fourth splitting point is on Jul. 22th, which is the period all counties already moved to Phase 4 reopening policy.

After processing through the Temporal Neighborhood Clustering module, the original non-stationary is then split into five subdatasets. They can be recognized as a series of tasks with different data distributions. Figure 5.5 illustrates how the data distribution gradually changes over time. The X and Y-axis represent the first two principal components extracted by Principal Component Analysis (PCA). These two elements can explain 79% overall variance. From Figure 5.5a to 5.5e, the position with the higher density change from two centers in Figure 5.5a to one lower center in Figure 5.5e. It indicates that the distributions between groups are significantly different and have a periodic pattern. The results conclude that the forecasting model needs to be updated since the data distribution changes gradually.



(a) From Subdataset 1 **(b)** From Subdataset 2 **(c)** From Subdataset 3 **(d)** From Subdataset 4 **(e)** From Subdataset 5

Figure 5.5: The Change Of Data Distribution From Each Subdataset (X And Y Represent The First Two Principal Components From The Whole Network Features Extracted By PCA. Five Subdatasets Are The Results Of The Temporal Neighborhood Clustering Module, Which Can Match To The Visualized Result In Figure 5.4)

5.3.3 Prediction Performance Comparison

In this section, we compare four models to evaluate their performance on five test subdatasets, which are the results of Temporal Neighborhood Clustering.

1. **Naive Model**: We reimplemented a Stacked-LSTM model[39] with 2 layers. This model was only trained by the first subdataset. In this case, it can be presented as a baseline model (lower bound model).

2. **Naive Model-whole**: In this model, we combine the whole subdataset and use the same architecture as the first Stacked-LSTM model. The Naive Model-whole model can be viewed as a upper bound among all models because all kinds of data have been seen and learned. Besides, memory usage is a side-effect that we need to deal with while training this model.

3. **Retrain Model**: It is also a 2-layers Stacked-LSTM model. This model is trained by the first subdataset then sequentially retrained by the following four subdatasets. This model may perform well in the last test dataset but forget what it has learned from the previous tasks. We can later evaluate if the catastrophic forgetting issue happens in this model.

4. **Online-EWC Model** (our proposed framework): Online-EWC Model is what we introduced in the previous section. The architecture would be the same as the last three models, 2-layers Stacked-LSTM. Besides, the online-EWC technique is also applied in this model to measure how it can overcome the catastrophic forgetting issue.

Here we select Mean Squared Error (MSE) as a evaluation metrics to measure the performance of each model:

$$MSE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|^2}{n} \tag{5.9}$$

where y_i and $\hat{y}_i$ represents the ground truth and predicted value respectively.

Table 5.1 indicates the comparison of the performance in this study to evaluate the impact of forgetting issues and check if the proposed incremental learning framework can match the upper-bound model. D_1 to D_5 represents the five subdataset generated from the Temporal Neighborhood Clustering module. From the results shown in Table 5.1, we first see the **Naive model**, as the baseline mode, which shows the competitiveness in the D_1 test data com-

Table 5.1: Performance Comparison Of The Proposed Model With Baseline Models

Model	Mean Square Error (E-02)				
	D_1	D_2	D_3	D_4	D_5
Naive	1.802	0.255	0.284	0.569	0.809
Naive-whole	**0.846**	**0.041**	**0.060**	0.133	0.249
Retrain	1.922	0.098	0.173	0.117	0.233
Online-EWC	0.874	0.043	0.066	**0.108**	**0.217**

pared to the Retrained model; it is because this model only trained on D_1's training dataset, which has a similar data distribution to D_1 test data. But the comparative performance drops dramatically from D_2 to D_4. The reason is the pandemic completely disrupted the typical mobility patterns. The Naive model can barely accommodate the non-stationary data.

The **Naive Model-whole model** is represented as an upper-bound model because it already learned all conditions given the whole original dataset. It can be also identified as a multi-tasks learning approach. As imagined, this model achieved the three best performances among all five testing subdatasets. The performance of this model testing on each test data can also be viewed as a result without forgetting.

The **Retrain model** shows a great result with a small MSE in the last subdataset because the parameters in the model are adjusted to fulfill the traffic circumstance in D_5 at the end. However, this model slightly forgot the learned tasks in D_1, which is even worse than the first Naive model. Surprisingly, we can see that the forecasting performance is excellent in D_4 and D_5, which are better than the upper-bound model. It reveals that the learned information will be forget in some cases, but retraining from new tasks can sometimes improved the performance in the previous task. It can be further investigate in the future work.

Lastly, the applied **Online-EWC model** shows the capability to overcome forgetting problems and maintain relatively outstanding performance. For tasks 1 to 3, the model retains the memory learned from these tasks and presents a comparable performance to the upper-bound model. Besides, it is remarkable that the Online-EWC model beat all models in the last two cases. Furthermore, the Online-EWC model exhibits the lowest mean MSE (average of MSE of five test subdatasets) among all baseline models (Naive: 0.744, Naive-whole: 0.265, Retrain: 0.509, Online-EWC: 0.261, with unit E0-2). The Online-EWC model shows a promising result and prevents memory-storage issues, which might happen in the Naive-whole model because the entire five subdatasets are fed into the model.

5.4 Chapter Conclusion

This study proposes an incremental learning-based network to learn the latest task and remember the previously known tasks given a non-stationary time-series data stream. The critical contribution of this research can be concluded into four aspects:

1. We propose an incremental learning-based model into a regression task, especially in transportation scenarios, and utilized its capacity to learn the gradually changing traffic patterns.
2. A dynamic splitting-point detector is proposed and provides an excellent ability to deal with the real-world non-stationary data.
3. The proposed framework is compared with the well-known strategies, such as retrain approach and multi-tasks learning, to show the advantages of incremental learning.
4. We employed a real-world shifting dataset to evaluate the proposed approach and other baseline models.

Experiment results indicate that the Temporal Neighborhood Clustering module can detect the proper splitting points, matching to the dates that real-world policies released. The comparison results show the proposed framework can align with the upper-bound model. The Online-EWC also has the advantage of requiring less memory than the upper-bound model. Therefore, it reveals that the proposed framework is more capable of accommodating non-stationary datasets.

Future studies will improve the Temporal Neighborhood Clustering to solve the optimum N, the number of subdatasets, dynamically without manually setting. Besides, the non-stationary data stream can be further utilized as an input of a classification task, classifying the current type of mobility pattern and assigning a proper forecasting model to address the current traffic condition.

6

Unified Framework for Multi-Contrastive Learning in Spatial-Temporal Traffic Forecasting

6.1 Overview

Addressing urban mobility and traffic congestion challenges requires accurate traffic forecasting methods. Although deep learning-based models in Intelligent Transportation Systems (ITS) have shown promising performance, certain limitations still exist, such as difficulty in handling complex and noisy data and the inability to learn fine-grained representations suitable for traffic forecasting. In this study, a novel unified framework is introduced that combines traffic representation learning and multi-contrastive learning to address these challenges. By considering multi-scale contextual information, designing multiple perspectives of contrastive learning, and introducing generalized definitions for spatial-temporal positive/negative pairs, the framework demonstrates improved performance in spatial-temporal traffic forecasting. Experimental results reveal that the unified framework enhances the accuracy of various base models across all evaluation metrics. For instance, at the 180-minute prediction horizon, which represents a relatively long forecasting time frame, significant improvements were observed in these metrics. Additionally, an ablation study highlights the importance of integrating multiple contrastive learning techniques to achieve robust and effective traffic forecasting. Overall, the findings emphasize the potential of the proposed framework in advancing the state-of-the-art in spatial-temporal traffic forecasting, contributing to the development of more efficient and generalizable transportation systems.

6.1.1 Background

Urban mobility is an increasingly pressing concern in our modern society, with cities growing rapidly and traffic congestion becoming a significant problem. ITS have emerged as a promising solution, allowing us to collect and analyze vast amounts of traffic data to gain insights into traffic patterns and optimize transportation management[98]. Accurate traffic flow prediction is a crucial aspect of this effort, enabling traffic managers to make informed decisions and take proactive measures to reduce congestion and improve traffic flow. However, traditional statistical approaches to transportation traffic forecasting have limitations when dealing with complex temporal-spatial data patterns[200]. To address these challenges, researchers have proposed deep learning-based models that utilize algorithms such as Long Short-term Memory (LSTM)[206], Convolutional Neural Network (CNN)[213], Graph Neural Network (GNN)[89], and Transformer[163]. These models can capture complex spatial-temporal patterns in traffic data and extract relevant features without requiring manual feature engineering.

To further improve the accuracy and applicability of deep learning-based models for transportation traffic forecasting, researchers have been exploring various techniques. For example, some researchers have proposed hybrid models that combine different types of deep learning algorithms, leveraging their individual strengths to produce more accurate predictions[15,203,20]. Other approaches include incorporating external data sources, such as weather[165] and social media data[188], to capture additional context and improve the models' performance. Additionally, novel techniques for handling high-dimensional data, such as dimensionality reduction[16] and representation learning[187], have been proposed. As the field of deep learning for transportation traffic forecasting continues to evolve, ongoing research into these and other techniques is needed to further advance the accuracy and applicability of these models.

Recent research has explored the potential of contrastive learning-based algorithms to enhance the robustness of existing models[218]. In various fields, including Computer Vision (CV) and Natural Language Processing (NLP), contrastive learning has been successfully used to produce generalizable representations that can be applied to subsequent tasks. For example, in CV, contrastive learning has been used to learn visual representations that generalize well to different datasets and downstream tasks, by training on pairs of similar and dissimilar images[86]. Similarly, in NLP, contrastive learning has been used to learn contextualized word embeddings by contrasting different views of the same sentence[103]. However, the application of contrastive learning to spatial-temporal traffic forecasting poses unique challenges, as existing contrastive learning techniques typically focus on learning coarse-grained representations that are more suitable for instance-level anomaly detection[115,95] and node classification[68], rather than fine-grained representations suitable for traffic forecasting.

Another major challenge in integrating contrastive learning to spatial-temporal traffic forecasting is the lack of a generalized definition for spatial and temporal positive/negative pairs. Recapping the key points, representations that exhibit similar semantics are recognized as positive pairs, while representations with unrelated semantics are defined as negative pairs. This has limited existing contrastive research and impeded the proper training of contrastive components in spatial-temporal traffic forecasting models. While positive pairs in other fields are typically established by applying data augmentations to generate two views of the same input (anchor) and forming negative pairs between the anchor and all other inputs' views within a batch, defining positive/negative pairs for spatial-temporal time series data requires considerations of both temporal dependencies and topological connections between roads. Should we consider representations generated from two connected segments as negative pairs? Similarly, should we consider representations from two different time frames with similar traffic patterns as negative pairs? Therefore, further research is needed to address this challenge and enable the proper application of contrastive learning to spatial-temporal traffic forecasting.

After reviewing the previous research, we have identified certain inherent limitations that have limited contrastive learning direct applicability to traffic forecasting until the following significant drawbacks are addressed:

1. The majority of existing studies have focused on learning coarse-grained representations that are more suitable for anomaly detection and node classification at the instance level, rather than fine-grained representations that are better suited for traffic forecasting.

2. Most of these studies have overlooked multi-scale contextual information at different granularities. Features with multiple scales can provide rich semantics and enhance the ability to learn generalized representations.

3. The lack of a generalized definition for spatial and temporal positive/negative pairs has hindered existing contrastive research, which has limited the proper training of contrastive components in spatial-temporal traffic forecasting models.

6.1.2 Contributions and Organization

To address the challenges mentioned above, we propose a unified framework that enhances spatial-temporal traffic forecasting through joint traffic representation learning and multi-contrastive learning. The traffic data augmentation approach enables the model to handle noisy and incomplete data and adapt to non-stationary conditions. Additionally, the jointly learning architecture allows the model to capture the latest trends with historical traffic data input and learn the generalized representation, which empowers it to acquire more robust and generalized patterns. To summarize, the contributions of this research can be listed as follows:

1. **Development of a unified framework:** A unified framework is developed that combines traffic representation learning and contrastive learning, which is designed to han-

dle noisy and incomplete traffic data and adapt to non-stationary traffic conditions.

2. **Consideration of multi-scale contextual information:** In this research, the use of multi-scale contextual information is carefully considered in temporal-wise contrastive learning to improve the model's capacity to learn generalized representations and to further achieve better performance.

3. **Design of multiple perspectives of contrastive learning:** Multiple perspectives of contrastive learning are designed to explore the impact of contrastive learning on different feature representations.

4. **Proposal of a generalized definition for spatial-temporal positive/negative pairs:** This definition is designed to provide a consistent and effective way of selecting positive and negative samples in spatial and temporal contrastive learning.

The rest of this chapter is organized as follows: Section 6.2 outlines the problem and describes the design of each component in the model, including the details of spatial-temporal data augmentation, traffic representation encoder/decoder, multi-contrastive learning components, and the generalized definition for spatial-temporal positive/negative pairs. In Section 6.3.1, we present the experimental settings and evaluate the results, including ablation studies and a visualized comparison between the prediction result and the ground truth. Finally, Section 6.4 concludes with remarks and points out potential future works to expand concepts and leverage the advantages of contrastive learning applied to traffic forecasting.

6.2 Methodology

6.2.1 Problem Statement

In this unified framework for improving spatial-temporal traffic forecasting with contrastive learning, the primary objective is to develop a robust and accurate traffic forecasting model that effectively incorporates both the spatial and temporal dependencies present in traffic data. The framework comprises two main components, namely Traffic Data Augmentation and Joint Representation Learning, as illustrated in Figure 6.1.

The processed spatial-temporal traffic data will undergo augmentation using the data augmentation component. Subsequently, the augmented data and the original data will both be fed into the traffic representation encoder to learn meaningful traffic patterns. The prediction and the multi-contrastive learning will then be conducted simultaneously. The aggregated loss, which combines multiple contrastive losses along with the original prediction loss, will serve as the objective to be minimized during the training process. Each of these components will be described in detail in the subsequent sections.

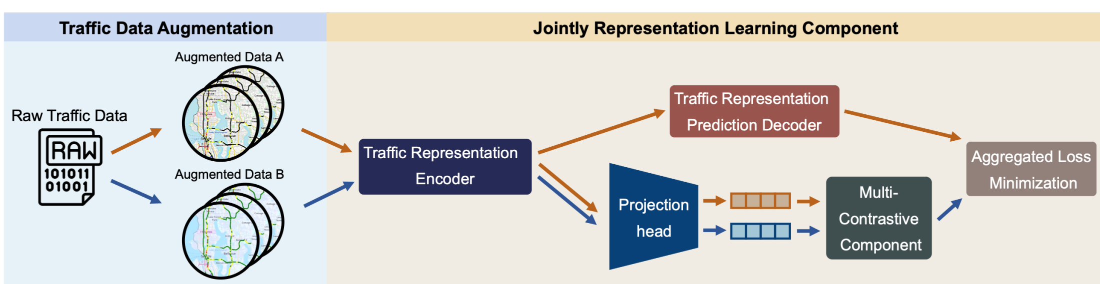


Figure 6.1: A Unified Framework For Spatial-temporal Traffic Forecasting With Contrastive Learning

6.2.2 Spatial-temporal Traffic Network Data

A traffic network is typically represented as an undirected graph comprising interconnected nodes and edges, denoted as $G = (V, E)$. Here, V represents road segment matrix with the features related to each road segment that include its location, length, width, and other relevant attributes. The shape of V is $N_v \times d_v$ with N_v representing the number of nodes and d_v is the number of features associated with each node. And E records the relationship between each road segments, such as connectivity. The shape of E is $N_e \times d_e$, where N_e represents the number of edges, and d_e is the number of features associated with each edge. We further transform E to an adjacency matrix $A \in \mathbb{R}^{N_v \times N_v}$. Specifically, $A_{ij} = 1$ if there is an edge between node i and node j, and $A_{ij} = 0$ otherwise.

To incorporate temporal information into the traffic network representation, we extend the graph to a spatial-temporal network. For each timestamp t, a spatial-temporal matrix $X_t = \{x_t^1, x_t^2, \cdots, x_t^{N_v}\}$ is collected where $x_t^v \in \mathbb{R}^{d_v}$. This matrix can be expanded to a generalized structure of a spatial-temporal traffic data matrix with a corresponding A, which assumes a fixed graph topology in this frame. X_T^V consists of N previous timestamps before time T in V segments.

$$X_T^V = \left(\begin{bmatrix} x_{T-N}^1 & x_{T-N}^2 & \dots & x_{T-N}^V \\ x_{T-N-1}^1 & x_{T-N-1}^2 & \dots & x_{T-N-1}^V \\ \vdots & \vdots & \ddots & \vdots \\ x_{T-1}^1 & x_{T-1}^2 & \dots & x_{T-1}^V \end{bmatrix}, A \right) \tag{6.1}$$

Each element x_t^v represents the traffic condition in v^{th} segment at t^{th} timestamp.

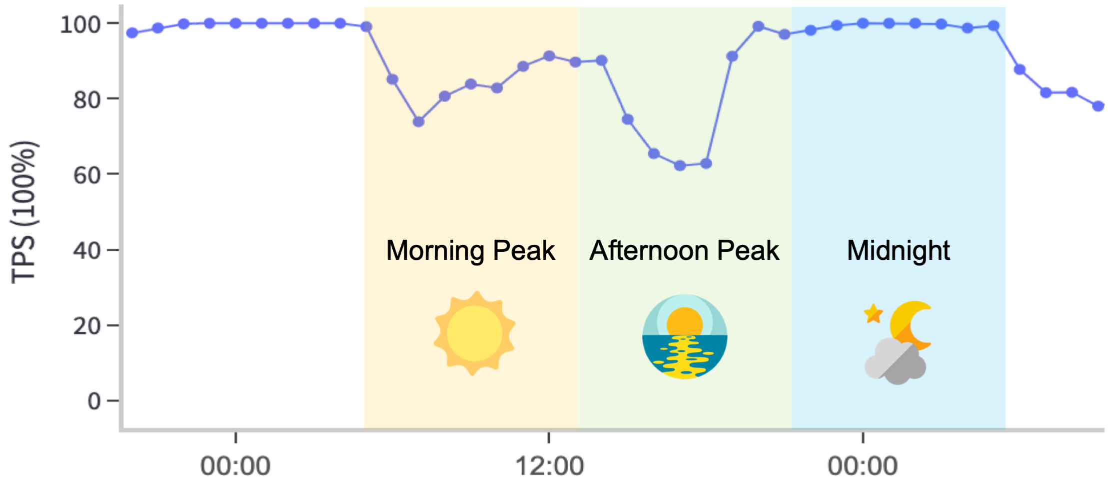


Figure 6.2: Temporal Pattern Inconsistency In Nearby Time Frames In An I-5 Segment On A Normal Weekday

6.2.3 Spatial-temporal Traffic Data Augmentation

Data augmentation is a crucial process in contrastive learning, as it helps to create semantically similar pairs and enables the model to learn invariant representations under different types and levels of perturbations. Although there are many data augmentation in various domains, such as rotation and cropping in CV, there has been less attention to its application in spatial-temporal traffic data. For example, the current approaches in this field assume that the subsequent time frame should be a positive pair [185], which may not hold in fluctuating traffic conditions, as demonstrated in Figure 6.2. Therefore, it is essential to propose new data augmentation strategies that can better account for the unique properties of spatial-temporal traffic data.

Spatial-temporal Masking

To simulate non-stationary traffic conditions and improve the robustness of our proposed framework, we utilize a masking mechanism to randomly mask certain attributes of the spatial-temporal traffic data in Equation 6.1. This is accomplished by using the following equation, which allows us to maintain the overall semantic structure while introducing variability:

$$X_t^{v\prime} = X_t^v \cdot [r > T_{st}] \tag{6.2}$$

where $X_t^{v\prime}$ and X_t^v represent the element in the t-th row and v-th column of matrices X' and X, respectively. The variable r represents a random value generated from a uniform distribution over $[0, 1]$, and T_{st} is the threshold value for spatial-temporal masking. The notation $[r > T_{st}]$ denotes the Iverson bracket notation, which takes the value 1 if r is greater than T_{st} and 0 otherwise.

Graph Topological Masking

To enhance the generaliability of the proposed unified framework, a graph topological masking approach is also utilized to randomly remove connections between segments. Specifically, the original adjacency matrix A is converted to A' by following formula:

$$A'_{i,j} = A_{i,j} \cdot [r > T_a] \tag{6.3}$$

where $A_{i,j}$ is defined as the connectivity between segment i and j, r represents a random value drawn from a uniform distribution over $[0, 1]$, and T_a denotes the threshold value for the graph topological masking technique.

6.2.4 Traffic Data Representation Encoder & Decoder

The current approaches for traffic forecasting can be summarized into an encoder and decoder structure. The backbone encoder receives the input spatial-temporal traffic data and learns traffic patterns from different perspectives, depending on the model architecture. According to the framework architecture (Figure 6.1), we now have two sets of data: X_T^V and $X_T^{V}{}'$ after processing through the traffic data augmentation component. Both are fed into the encoder, but only the representations generated from X_T^V are later fed into the decoder. Therefore, here we use X_T^V as an example of input data for the following description.

In an RNN-based model, the encoder applies a series of recurrent layers to process the input data, with each layer computing a hidden state vector h_t based on the input at time t, denoted by x_t, which is the t^{th} row in X_T^V, for example, and the previous hidden state h_{t-1}. The hidden state is updated using the following equation:

$$h_t = f(h_{t-1}, x_t) \tag{6.4}$$

where f is a non-linear function that combines the previous hidden state and the current input.

In a GNN-based model, the encoder applies graph-based layers (e.g., Graph Convolutional Network (GCN)) to the input data, which are represented as a graph with nodes and edges. Each node in the graph corresponds to a road segment, and each edge corresponds to the connectivity or similarity between roads. The graph-based layer updates the hidden state h_v of each node v based on the hidden states of its neighbors and its own previous hidden state, using the following equation:

$$h_v^{(l)} = \sigma\left(\sum_{u\in N(v)} \frac{1}{c_{u,v}} W^{(l)} h_u^{(l-1)} + W^{(l)} h_v^{(l-1)}\right) \tag{6.5}$$

where $N(v)$ denotes the set of neighboring nodes of v, $c_{u,v}$ is a normalization constant, $W^{(l)}$ is a learnable weight matrix for the l-th layer, and σ is a non-linear activation function.

In a Transformer-based model, the encoder applies a series of self-attention layers to the input data, allowing each element in the input sequence to attend to all other elements. The hidden state h_t of each element is updated based on a weighted sum of all elements in the input sequence, using the following equation:

$$h_t = \sum_{i=1}^{n} \alpha_{t,i}(W^{(1)} x_i) \tag{6.6}$$

where n is the length of the input timestamp, x_i is the i-th timestamp in the input sequence (the i-th row in X_T^V), $W^{(1)}$ is a learnable weight matrix, and $\alpha_{t,i}$ is the attention weight of the i-th element for the t-th element.

In general, the learned representations or hidden states generated by the different types of encoding structures mentioned above are then used as inputs to the decoder part of the model. The decoder part decodes these representations using regression heads, which are commonly implemented by a series of fully-connected layers, activation functions, and normalization layers, to project the high-dimensional representation to a low-dimensional result, which is the final prediction. The generalized prediction result can be expressed as:

$$\hat{Y} = Decoder(h) \tag{6.7}$$

where h is the learned representation or hidden states from the various types of encoders, and

$Decoder(\cdot)$ is the generalized decoder function with regression head. Finally, the predicted result is compared to the ground truth data to evaluate the performance of how well the encoder-decoder structure can learn the spatial-temporal traffic patterns. The prediction loss ($\mathscr{L}_{pred}$) is then minimized to improve the performance of the model.

6.2.5 Multi-contrastive Component and Generalized Definition for Spatial-temporal Positive/Negative Pairs

In contrastive learning, positive pairs (P) and negative pairs (N) are utilized to calculate the contrastive loss. For temporal-wise, spatial-wise, and batch-wise contrastive learning, P consists of pairs of representations from the same region, neighborhood, or batch, respectively, while N consists of pairs of representations from different regions, non-neighborhoods, or batches, respectively. The generic contrastive loss can be calculated using the following equation:

$$\text{contrastive\_loss}(P, N) = -\frac{1}{|P|} \sum_{(x_i, x_j) \in P} \log \frac{\exp(\text{sim}(x_i, x_j))}{\sum_{(x'_i, x'_j) \in N} \exp(\text{sim}(x'_i, x'_j))} \tag{6.8}$$

where $|P|$ is the number of positive pairs in the mini-batch, (x_i, x_j) is one of the positive pairs, $sim(x_i, x_j)$ is the similarity between x_i and x_j (the similarity function is a simple dot product) and (x'_i, x'_j) is one of the negative pairs. This loss function is minimized during training to learn feature representations that can distinguish between positive and negative examples.

Based on the basic contrastive loss function defined in Equation 6.8, our proposed multi-contrastive component represents an innovative extension that expands the scope of contrastive learning from multiple angles. It incorporates three new perspectives: temporal-wise contrastive learning, spatial-wise contrastive learning, and batch-wise contrastive learning. By considering different perspectives and levels of granularity, the multi-contrastive component is able to generate a more comprehensive representation that captures spatial and temporal dependencies in the traffic data. Specifically, temporal-wise contrastive learning encourages the model to learn representations that capture the temporal dependencies in the data, while spatial-wise contrastive learning emphasizes the importance of the spatial relationships between different regions. Batch-wise contrastive learning, on the other hand, enables the model to learn from multiple traffic sequences simultaneously. Together, these extensions enable the model to learn a more robust and generalized representation of the traffic data, leading to improved robustness and forecasting performance.

Temporal-wise Contrastive Learning

To design our temporal-wise contrastive loss, we adopt a hierarchical structure inspired from Yue et al.[230] as shown in Figure 6.3. Let $X \in \mathbb{R}^{B\times T\times N\times D}$ denote the input data and X' denote its augmented version with the same size. We obtain representations H and H' by passing X and X', respectively, through the spatial-temporal representation encoder component. We consider pairs of representations (h_i, h'_i) for which the corresponding nodes i have the same timestamp as positive pairs, and the remaining pairs as negative pairs. We then apply a max-pooling operation to the resulting representations H and H' along the temporal dimension, reducing their sizes to $\mathbb{R}^{T/2\times N\times D}$.

Algorithm 1: Temporal-wise Contrastive Learning

Input: Representation H and H' with dimension (B, T, N, D)
Output: Temporal-wise contrastive loss $\mathscr{L}_{temp}$
Function `TemporalContrastive(`H`)`**:**
    $\mathscr{L}_{temp} \leftarrow 0$;
    $T_{cur} \leftarrow T$;
    while $T_{cur} > 1$ **do**
        $P \leftarrow$ positive pairs in (H, H') with the same timestamp;
        $N \leftarrow$ negative pairs in (H, H') with different timestamp;
        $\mathscr{L}_{curr} \leftarrow \text{contrastive\_loss}(P, N)$;
        $\mathscr{L}_{temp} \leftarrow \mathscr{L}_{temp} + \mathscr{L}_{curr}$;
        $T_{cur} \leftarrow T_{cur}/2$;
        $H \leftarrow \text{max\_pool}(H, \text{kernel\_size} = 2)$;
        $H' \leftarrow \text{max\_pool}(H', \text{kernel\_size} = 2)$;
    end
    $\mathscr{L}_{temp} \leftarrow \mathscr{L}_{temp}/(\log_2 T)$;
    return $\mathscr{L}_{temp}$;

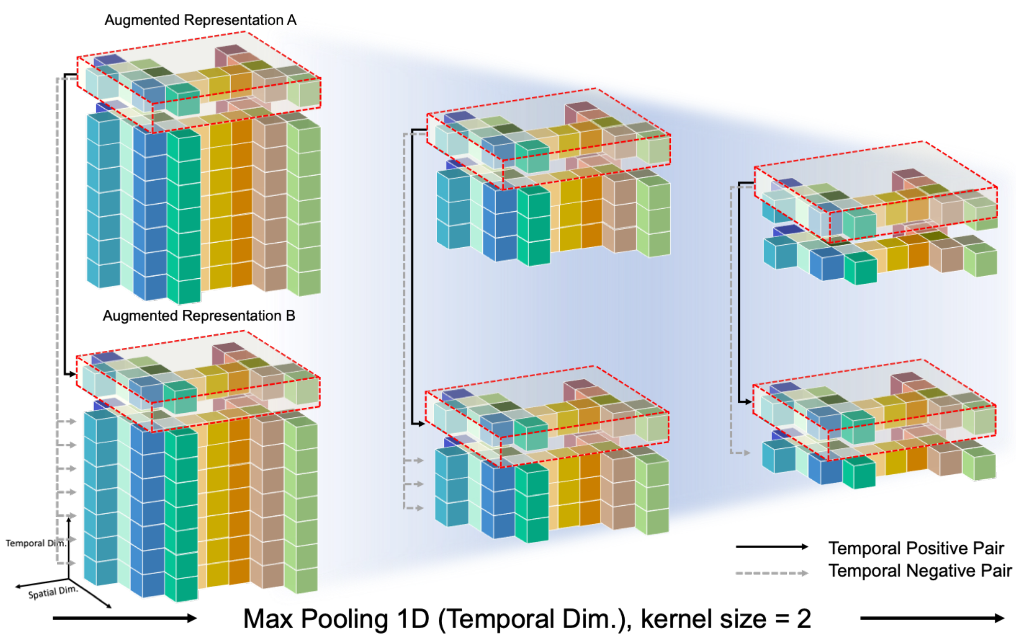


Figure 6.3: Temporal-wise Contrastive Learning

We repeat this max-pooling and contrastive loss calculation until $T/m = 2$, accumulating the temporal-wise contrastive loss at each iteration. Finally, we take the mean of the accumulated loss as the final temporal-wise contrastive loss. This hierarchical structure is summarized in Algorithm 1.

Spatial-wise Contrastive Learning

In the spatial-wise contrastive calculation, we use a readout function, which is a commonly used approach in graph to summarize an entire graph representation, to summarize the temporal dimension of the representations H and H', compressing the temporal dimension and resulting in representations with dimensions $\mathbb{R}^{B \times N \times D}$. The outputs of the readout function are denoted as H_r and H'_r.

To define the neighborhood and non-neighborhood regions of each node in the graph, we use the adjacency matrix A. In order to define the neighborhood zone, we consider the k-hop neighborhood of a node as the region of interest, where k is a predefined value. For example, a 2-hop neighborhood of a node includes all the nodes that can be reached from the node within two hops in the graph, as shown in Figure 6.4.

The positive pairs in the augmented representation H'_r for a specific node N are the nodes within the neighborhood region, as defined by the adjacency matrix A. The nodes outside of the neighborhood region are considered as negative pairs for node N. This neighborhood-based approach allows us to capture local contextual information around each node, which is useful for learning representations that are sensitive to the graph structure. This approach is implemented in Algorithm 2.

Algorithm 2: Spatial-wise Contrastive Learning

Input: Representations H_r, H'_r, and adjacency matrix A
Output: Spatial-wise contrastive loss $\mathscr{L}_{spat}$
Function `SpatialContrastive(`H_r, H'_r, A`)`:
  $\mathscr{L}_{spat} \leftarrow 0$;
  for $N \leftarrow 1$ *to* N **do**
    $P \leftarrow$ positive pairs in (H_r, H'_r) within neighborhood of node N defined by A;
    $N \leftarrow$ negative pairs in (H_r, H'_r) outside of neighborhood of node N defined by A;
    $\mathscr{L}_{curr} \leftarrow \text{contrastive\_loss}(P, N)$;
    $\mathscr{L}_{spat} \leftarrow \mathscr{L}_{spat} + \mathscr{L}_{curr}$;
  end
  $\mathscr{L}_{spat} \leftarrow \mathscr{L}_{spat}/N$;
  return $\mathscr{L}_{spat}$;

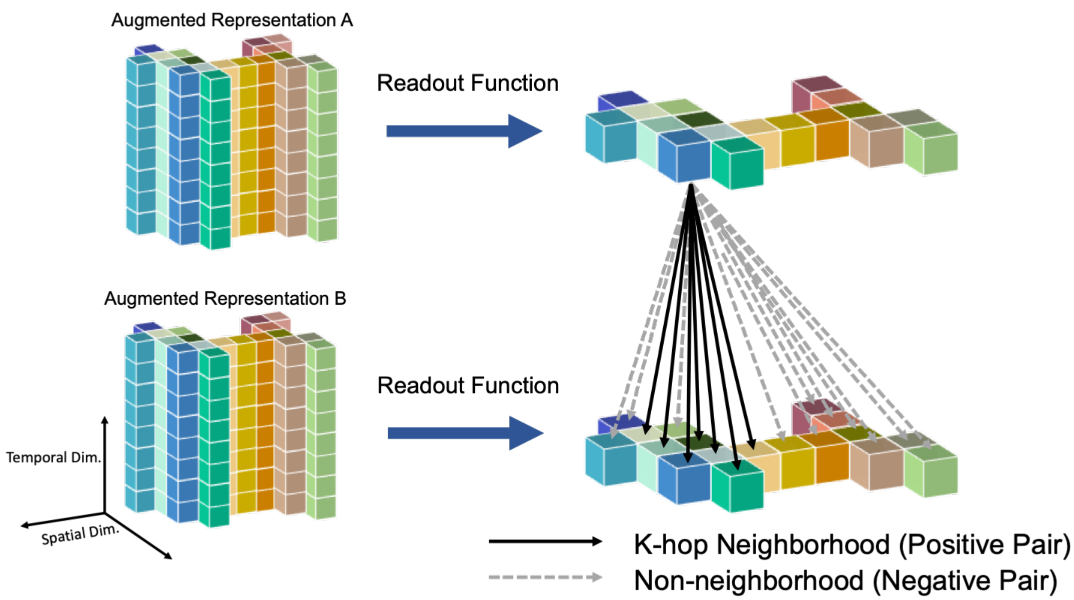


Figure 6.4: Spatial-wise Contrastive Learning

Batch-wise Contrastive Learning

In batch-wise contrastive learning (Figure 6.5), the Pearson Correlation Coefficient is calculated within the batch dimension using representations H and H'. The resulting representations are then fed into the readout function to obtain H_r and H'_r, which are used to compute the batch-wise contrastive loss. Positive and negative pairs are determined based on a threshold T_b using the calculated Pearson Correlation Coefficient, where representations with similarity scores above the threshold are categorized as positive pairs and those below it as negative pairs. The average of the batch-wise contrastive losses is then computed to obtain the final batch-wise contrastive loss. Implementation details are presented in Algorithm 3.

Algorithm 3: Batch-wise Contrastive Learning

Input: Representations H_r, $H'r$, batch-wise Pearson Correlation Coefficient C, and threshold T

Output: Batch-wise contrastive loss $\mathscr{L}batch$

Function `BatchContrastive(`$H_r, H'r, C, Tb$`)`**:**

$\mathscr{L}batch \leftarrow 0$; **for** $i \leftarrow 1$ *to* B **do**

$P \leftarrow$ positive pairs in $(H_r, H'r)$ with $Ci, j \geq Tb$; $N \leftarrow$ negative pairs in $(H_r, H'r)$ with $Ci, j < T_b$; $\mathscr{L}curr \leftarrow$ contrastive$_{\text{l}}$oss(P, N);

$\mathscr{L}batch \leftarrow \mathscr{L}batch + \mathscr{L}curr$;

end

$\mathscr{L}batch \leftarrow \mathscr{L}batch/B$; **return** $\mathscr{L}_{batch}$;

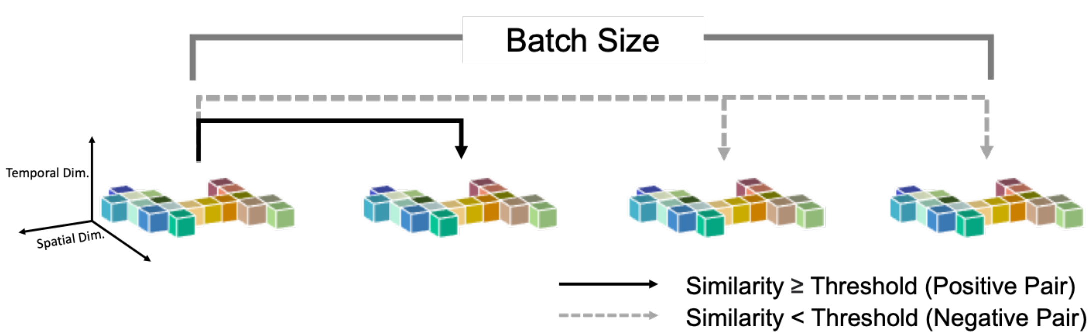


Figure 6.5: Batch-wise Contrastive Learning

6.2.6 Aggregated Loss Minimization

In order to enhance the performance and generalizability of the proposed framework, we combine multiple contrastive losses as auxiliary regularization terms. These contrastive losses, including temporal-wise , spatial-wise , and batch-wise contrastive loss, are integrated with the original prediction loss ($\mathscr{L}_{pred}$) to form an aggregated loss. This approach helps to prevent overfitting and improve the model's ability to capture incomplete and complex spatial and temporal dependencies in the traffic data.

$$\mathscr{L} = \mathscr{L}_{pred} + \alpha \cdot \mathscr{L}_{temp} + \beta \cdot \mathscr{L}_{spat} + \gamma \cdot \mathscr{L}_{batch} \tag{6.9}$$

The tunable coefficients α, β, and γ control the relative importance of each contrastive loss term, allowing for fine-grained adjustment of their contribution to the overall loss function in training process.

6.3 Experiments

In this section, several well-known RNN-based, GNN-based, and Transformer-based traffic forecasting models are applied to our unified framework as the backbone encoder and decoder to investigate how the jointly multi-contrastive learning structure can improve traffic forecasting accuracy.

6.3.1 Dataset

The Traffic Performance Score (TPS) Dataset[41], hosted by STAR Lab at the University of Washington, provides data including TPS, traffic speed, and traffic volume for four highways (I-5, I-90, I-405, and SR-520) in the greater Seattle region at 15-minute intervals from March to June in 2020. The TPS is an evaluation index that considers each segment's length, volume, and speed, ranging between 0%(0) to 100%(1), where 0 represents the worst case of traffic states (congestion), and 1 represents a segment with free-flow speed. It serves as a target parameter for prediction in this study, as it is a more interpretable matrix for public users. For training, the first three months of data were selected, with half a month each for validation and testing. The dataset consists of 87 segments in the network, and three-hour data are used as input to predict the traffic conditions for the next three hours.

6.3.2 Experimental Base Models

Three base models, including RNN-based, GNN-based, and Transformer-based models, known for their ability to predict traffic conditions across the entire network, were selected to be applied within our unified framework for joint training with traffic prediction and multi-contrastive learning.

1. a LSTM-Sequence-to-Sequence (**LSTM-Seq2Seq**) model consists of LSTM-based encoder and a decoder, which allow the model to capture long-term dependencies in the input and output sequences, making it suitable for modeling time series data with complex temporal patterns [180]. It is categorized as a RNN-based representative in this study;

2. an Adaptive Graph Convolutional Recurrent Network (**AGCRN**), which combines Node Adaptive Parameter Learning module and Data Adaptive Graph Generation module with recurrent networks for the multi-step traffic prediction task [11]. It is categorized as a GNN-based representative in this study;

3. a Graph multi-attention network (**GMAN**) employs an encoder-decoder architecture, in which both the encoder and the decoder are made up of numerous spatio-temporal attention blocks, to model the traffic states [246]. It is categorized as a Transformer-based representative in this study.

The default settings of the above-mentioned baseline models were applied, as per their original studies. Our investigation focused on exploring how the performance of these base models can be enhanced by incorporating the proposed framework for improving spatial-temporal traffic forecasting with contrastive learning.

6.3.3 Experiment Setting

In this study, we evaluated the accuracy of the models using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). MAE and RMSE can be calculated using the following equations:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - \hat{y}i|}{n} \tag{6.10}$$

$$RMSE = \sqrt{\frac{\sum i = 1^n (y_i - \hat{y}_i)^2}{n}} \tag{6.11}$$

where y_i and $\hat{y}i$ represent the ground truth and predicted values, respectively. Additionally, we also used MAPE, which is a percentage-based metric, given by:

$$MAPE = \frac{100}{n} \sum i = 1^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{6.12}$$

Some detailed experimental settings utilized in this study are presented as follows. The

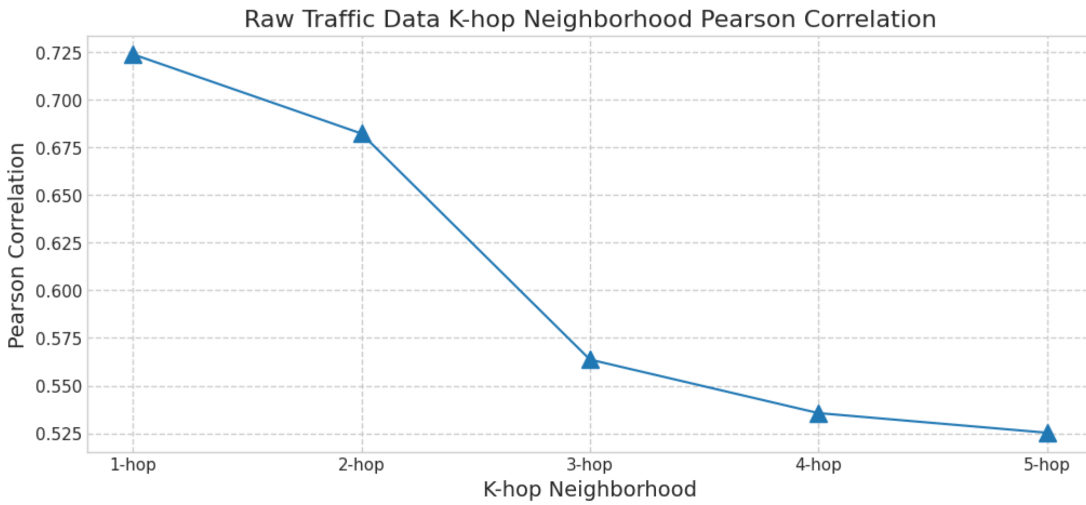


Figure 6.6: Raw Traffic Data K-hop Neighborhood Pearson Correlation

value of T_{st} was determined through spatial-temporal masking, while T_a was set based on graph topological masking. To perform spatial-wise contrastive learning, a k-hop neighborhood region was defined and a 2-hop neighborhood was selected based on the calculation of k-hop neighborhood from raw traffic data using Pearson Correlation Coefficient (as illustrated in Figure 6.6). For batch-wise contrastive learning, a matrix C was utilized to record the batch-wise Pearson Correlation Coefficient, and the threshold T_b was set to 0.80. Moreover, in the aggregated loss minimization part, α, β, and γ were set to 0.1 after the tuning process. For optimization in all experiments, we selected the Adam optimization method to minimize the aggregated loss. We also implemented an early stopping mechanism to prevent over-fitting. If the model improvement, which was evaluated by the decrease of the validation loss, failed to exceed a threshold (0.00001 of MAE) for ten consecutive epochs, the training process was terminated.

6.3.4 Experiment Results Analysis and Comparison

Table 6.1 presents the overall performance comparison of the Unified Framework with three base models. Compared to the original LSTM-Seq2Seq model, the Unified Framework achieved lower values for MAE (0.0133 vs 0.0141), MAPE (2.1730% vs 2.3184%), and RMSE (0.0444 vs 0.0465). Similar improvements in performance were observed for the GMAN and AGCRN models when applied to the proposed unified framework for joint training of traffic prediction and multi-contrastive learning. The findings from Table 6.1 emphasize the promising potential of the Unified Framework in enhancing the performance of the different types of base models, including RNN-based, GNN-based, and Transformer-based models, as evidenced by the lower values of MAE, MAPE, and RMSE achieved with the Unified Framework compared to the base models alone.

The experimental results presented in Table 6.2 provide further evidence that the proposed Unified Framework outperforms the base models across all evaluation metrics and time inter-

Table 6.1: Overall Performance Comparison Of The Unified Framework With Base Models

Model	MAE	MAPE	RMSE
LSTM-Seq2Seq	0.0141	2.3184%	0.0465
w/ the Unified Framework	**0.0133**	**2.1730%**	**0.0444**
GMAN	0.0143	2.3882%	0.0491
w/ the Unified Framework	**0.0135**	**2.3118%**	**0.0478**
AGCRN	0.0136	2.2046%	0.0449
w/ the Unified Framework	**0.0125**	**2.0082%**	**0.0425**

vals. These statistically significant improvements demonstrate the framework's effectiveness in enhancing traffic forecasting accuracy.

Specifically, the Unified Framework reduced the MAE, MAPE, and RMSE for LSTM-Seq2Seq by 9.3% (from 0.0172 to 0.0156), 6.8% (from 2.6662 to 2.4859), and 5.3% (from 0.0532 to 0.0504), respectively, for the 180-minute prediction horizon. Similar improvements were observed for the other two base models (GMAN and AGCRN). Notably, the magnitude of improvements varied across different time intervals, with the largest improvements in MAE, MAPE, and RMSE observed at the 120-minute interval in LSTM-Seq2Seq with the Unified Framework. This finding is particularly significant as it is widely recognized that the farthest time stamp is the most challenging to predict accurately. Overall, the AGCRN model performed the best among the base models across the three evaluation metrics, and its performance was further improved after integrating it into the proposed unified framework.

The visualization in Figure 6.7 compares the prediction results of the base Model and the model with the Unified Framework, using the AGCRN as an example. A random segment and time from the testing dataset were chosen for this illustration. Both models exhibited satisfactory performance during normal daytime conditions. However, the AGCRN with the Unified Framework demonstrated its competitiveness during peak hours when substantial traffic pattern changes occur. The TPS value experienced a significant drop at noon, and the proposed Unified Framework accurately forecasts this situation. Conversely, the base model detected the peak hour condition but did not provide a response of comparable quality. This outcome indicates that the proposed Unified Framework is more capable of handling fluctuating and noisy temporal-spatial traffic patterns than the base model.

Table 6.2: Step-wise Performance Comparison Of The Unified Framework With Base Models

Model	15 min			60 min			120 min			180 min		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
LSTM-Seq2Seq	0.0120	2.0070%	0.0405	0.0130	2.1455%	0.0443	0.0146	2.3848%	0.0475	0.0172	2.6662%	0.0532
w/ the Unified Framework	**0.0116**	**1.9136%**	**0.0399**	**0.0124**	**2.0578%**	**0.0423**	**0.0137**	**2.2329%**	**0.0453**	**0.0156**	**2.4859%**	**0.0504**
GMAN	0.0130	2.2324%	0.0463	0.0132	2.2603%	0.0469	0.0146	2.4287%	0.0496	0.0167	2.6691%	0.0530
w/ the Unified Framework	**0.0128**	**2.2320%**	**0.0460**	**0.0128**	**2.2380%**	**0.0456**	**0.0137**	**2.3329%**	**0.0476**	**0.0161**	**2.5969%**	**0.0524**
AGCRN	0.0100	1.5649%	0.0325	0.0130	2.1367%	0.0416	0.0142	2.3408%	0.0474	0.0162	2.5998%	0.0517
w/ the Unified Framework	**0.0096**	**1.5147%**	**0.0322**	**0.0113**	**1.8140%**	**0.0386**	**0.0133**	**2.1436%**	**0.0444**	**0.0155**	**2.4818%**	**0.0502**

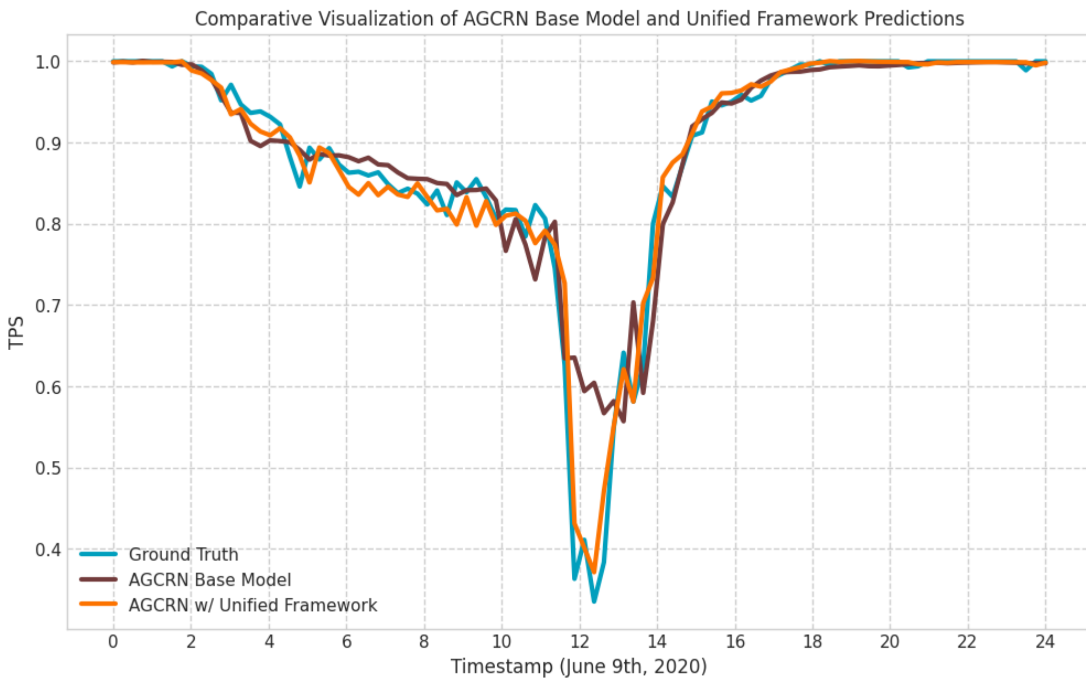


Figure 6.7: Comparative Visualization Of AGCRN Base Model And Unified Framework Predictions

6.3.5 Ablation Studies

Multi-contrastive Learning Sensitivity Analysis

This section analyzes the impact of removing spatial, temporal, and batch-wise contrastive learning from the proposed unified framework. We investigated how the performance would be impacted for three base models that were applied to the proposed framework. As shown in Table 6.3, the proposed unified framework consistently outperforms the base models across all evaluation metrics. When removing individual contrastive learning component, the impact on performance varies across different base models, indicating that the importance of each technique depends on the architecture of the base model.

Although the impact of removing the batch-wise contrastive learning component is rel-

Table 6.3: Ablation Results Of Removing Particular Contrastive Learning In The Unified Framework

Model	MAE	MAPE	RMSE
Basic LSTM-Seq2Seq	0.0141	2.3184%	0.0465
Only Remove Spatial CL	0.0138	2.3049%	0.0463
Only Remove Temporal CL	0.0137	2.2817%	0.0463
Only Remove Batch CL	0.0136	2.1871%	0.0459
w/ the Unified Framework	**0.0133**	**2.1730%**	**0.0444**
Basic GMAN	0.0143	2.3882%	0.0491
Only Remove Spatial CL	0.0140	2.3808%	0.0490
Only Remove Temporal CL	0.0141	2.3430%	0.0480
Only Remove Batch CL	0.0138	2.3569%	0.0484
w/ the Unified Framework	**0.0135**	**2.3118%**	**0.0478**
Basic AGCRN	0.0136	2.2046%	0.0449
Only Remove Spatial CL	0.0131	2.0945%	0.0438
Only Remove Temporal CL	0.0129	2.0477%	0.0432
Only Remove Batch CL	0.0127	2.0810%	0.0438
w/ the Unified Framework	**0.0125**	**2.0082%**	**0.0425**

atively small compared to spatial and temporal contrastive learning components, it consistently affects the performance of all base models. Based on this result, it can be inferred that batch-wise information, such as accounting for correlations among sequences of traffic data in a batch, can contribute to the accuracy of the traffic forecasting model. Furthermore, the unified framework integrating all three contrastive learning methods outperforms all base models and models with individual contrastive learning components removed. This finding suggests combining different contrastive learning methods can provide a more robust and practical approach to traffic forecasting.

Unified Framework Robustness Analysis

In this experiment, we analyzed the robustness of our proposed unified framework with multi-contrastive learning by introducing fluctuating/incomplete data. Specifically, we randomly masked 5% of the input data to simulate incomplete or unreliable data. We compared the MAE performance degradation of the base model and the model with the unified framework when tested on masked data relative to their performance on the original data, as shown

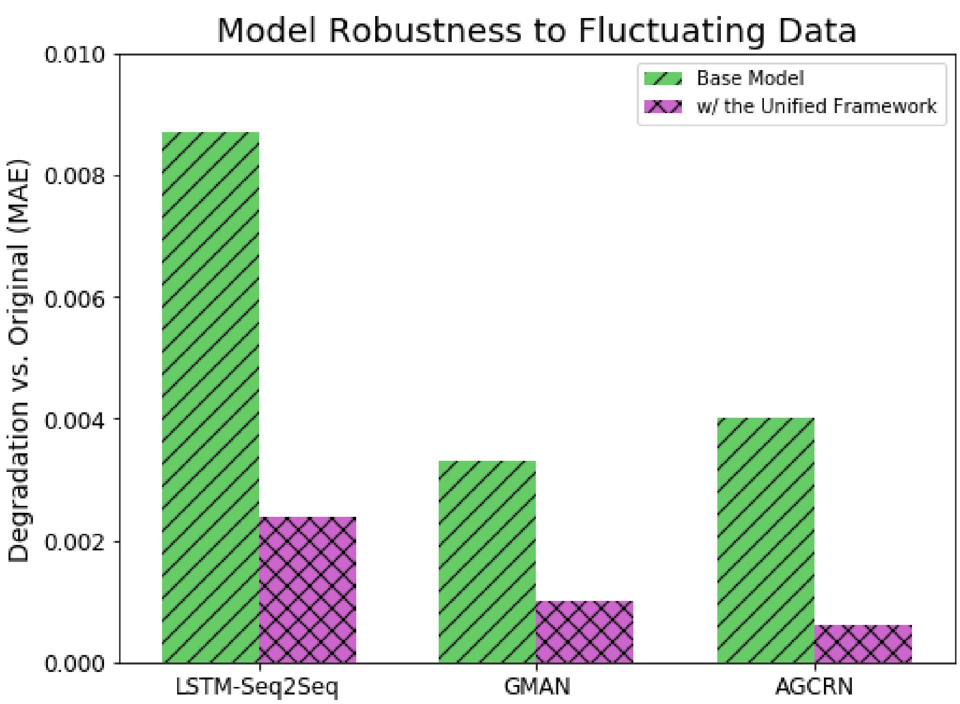


Figure 6.8: Model Robustness To Fluctuating Data

in Figure 6.8.

The result indicated the proposed unified framework with multi-contrastive learning demonstrated significantly greater robustness to fluctuating data than the base model. Specifically, the unified framework's performance degradation was considerably smaller than the base model compared to the original data. For the three groups (LSTM-Seq2Seq, GMAN, and AGCRN), the performance drop for the unified framework was [0.0024, 0.0010, 0.0006] (Purple Bars w/ cross texture), while the base model encountered a more significant degradation of [0.0087, 0.0033, 0.0040] (Green Bars w/ slash texture). These findings provide strong evidence for the effectiveness of the multi-contrastive learning component in the proposed framework and its ability to handle incomplete data.

Additionally, this ablation study result has crucial implications for real-world applications because data quality may be inconsistent or incomplete for various reasons, including technical communication challenges. By demonstrating that the proposed framework is more robust to fluctuating data than relying solely on a base model, this robustness analysis highlights the potential of the proposed framework to provide reliable results even in challenging environments.

6.4 Chapter Conclusion

In conclusion, our research presents a novel unified framework combining joint traffic representation and multi-contrastive learning to enhance spatial-temporal traffic forecasting. The proposed framework considers multi-scale contextual information at different granularities and is designed to handle noisy and incomplete data, adapt to non-stationary conditions, and learn generalized representations. Our research has made the following contributions:

- Developed a unified framework that combines traffic representation learning and multi-contrastive learning to enhance spatial-temporal traffic forecasting.
- Proposed a generalized definition for spatial-temporal positive/negative pairs to provide a consistent and effective way of selecting positive and negative samples in spatial, temporal, and batch-wise contrastive learning
- Considered multi-scale contextual information at different granularities to handle noisy and incomplete data, adapt to non-stationary conditions, and learn generalized representations.
- Demonstrated the effectiveness of the proposed unified framework in enhancing traffic forecasting accuracy compared to all base models and showed that the combination of different contrastive learning methods could provide a more robust and effective approach to traffic forecasting than individual methods alone.
- Conducted a robustness analysis to evaluate the proposed unified framework's ability to handle incomplete or unreliable data and found that it outperformed the base model for performance degradation on masked data. These results show the potential of the proposed framework to provide more reliable traffic forecasting in challenging real-world environments where data quality may be inconsistent/incomplete.

Moving forward, there are several potential areas for future research that can make contributions based on this study. One promising avenue is the exploration of more generic or intelligent approaches to assigning threshold values in filtering positive/negative pairs. This could potentially enhance the effectiveness of the proposed framework in handling noisy and incomplete data, adapting to non-stationary conditions, and learning generalized representations.

Another area is the potential of training the contrastive learning component separately to learn the traffic network's representation. By identifying common patterns or features in the road network, such an approach could potentially be used as a warm start to improve the generalizability of the model across different experimental regions or contexts.

Further research can also explore the extension of the proposed framework to incorporate additional data sources, such as weather and social media data, to enhance its applicability in real-world traffic forecasting scenarios.

Part IV

Applications and Open-source Dataset

7

Measure Urban Mobility and Online Predict Near-term Traffic like Weather Forecast

7.1 Overview

Measuring traffic performance is critical for public agencies managing traffic and individuals planning trips, especially when special events like the long-lasting COVID-19 pandemic happen. However, most existing traffic performance metrics narrowly focused on one aspect of the impacts but not comprehensive changes to the network. Further, urban traffic patterns and traveler trip planning have been dramatically affected since the pandemic breakout. Therefore, network-wide online traffic prediction becomes an urgent but more complicated task. To overcome these challenges, this study proposes a Traffic Performance Score (TPS) incorporating multiple parameters for measuring both urban and freeway network-wide traffic performance. The TPS is compared with other metrics to show its superiority. This study also presents a multi-step Sequence-to-Sequence (Seq2Seq)-based model with an online training and updating technique to predict network-wide traffic performance similar to a weather forecast to handle the complex problem of network-wide traffic prediction. Experimental results indicate that the proposed model with the online learning strategy outperforms existing methods regarding prediction accuracy and learning efficiency. In addition, the TPS measurement and its related online prediction functions are implemented on a publicly accessible platform and applied in real practice, which is another contribution of this work.

7.1.1 Background

The measurement of traffic performance is highly valuable for both transportation agencies, who utilize it to inform their operations and management strategies, and the general public, as traffic patterns significantly impact daily life. Traditional traffic performance metrics, such as speed, volume, travel time, etc., provide insight into the traffic conditions of specific roadways or corridors. However, they are unable to provide a comprehensive understanding of network-wide traffic performance, especially in large cities. The reason is that these metrics often only measure one aspect of traffic and struggle to differentiate between complex traffic scenarios. Moreover, freeway and urban traffic patterns are typically measured using different data sources, making it difficult to compare and understand traffic performance across both types of networks. To address these limitations, this study aims to develop a more comprehensive metric for measuring network-wide traffic performance for both urban and freeway traffic networks. The COVID-19 pandemic, as a long-lasting event, has drastically changed traffic patterns and made understanding dynamic network-wide traffic and predicting performance during the pandemic a more challenging task. To tackle this issue, this study also seeks to develop novel and robust prediction methodologies to foresee network-wide traffic performance, accounting for various influential factors such as incidents, weather, and special events.

Traffic prediction, a subject of active research and development for over 40 years, has gained significant attention in recent years with the advancement of artificial intelligence (AI)[101]. It is crucial for both transportation management and travel/trip planning. With the rise in traffic data and computational power, traffic forecasting methods have moved from traditional statistical models to data-driven machine learning approaches[201]. In recent years, deep learning research has spurred significant advances in traffic forecasting. Various deep neural network models, such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Generative Adversarial Network (GAN), have been applied in traffic forecasting studies and have achieved outstanding prediction results by exploiting the spatial-temporal patterns in vast data resources.

Predicting network-wide traffic performance like weather forecasting can be very helpful for the public. It can assist travelers in their trip planning by presenting the performance prediction as a simple score or visualized map. Nevertheless, current deep learning-based traffic prediction models primarily focus on developing novel neural network structures or combining existing neural network components to achieve modest accuracy improvements. Very few studies have successfully implemented the proposed models for public use. This is partly due to the challenges posed by online prediction tasks. An effective traffic prediction model should take into account both short-term traffic fluctuations and tiny long-term traffic changes, and be able to adapt and improve performance accordingly. Therefore, this study, aimed at applying traffic performance in practice, will also propose effective learning strategies to tackle the challenges of online traffic prediction.

7.1.2 Contributions and Organization

To solve all the aforementioned challenges, this study proposes a traffic performance measurement metric suitable for both urban and freeway traffic networks. To apply the proposed metric to real applications and benefit public users, traffic prediction models for online prediction tasks will be devised and integrated with the performance metric applications. To sum up, the contribution of this study can be summarized as follows:

1. A traffic performance score for measuring network-wide traffic is proposed. The mobility performance is measured by both freeway TPS and urban traffic TPS. The TPS is also implemented on a publicly accessible traffic performance measurement platform.

2. This study proposes a multi-step sequence-to-sequence-based model to predict network-wide traffic performance. To accommodate the online prediction task, this study also offers an online learning strategy with the periodical model training and updating process.

3. The proposed model with the online learning strategy outperforms existing methods in terms of prediction accuracy and learning efficiency. The TPS online prediction model is also taken into practice by implementing it on the traffic performance measurement platform. To the best of our knowledge, this study is the first proposed online traffic prediction model with real applications.

In the following sections, this research introduces the proposed TPS and describes how the freeway and urban TPS can be adopted to measure urban mobility in Section 7.2. Then, the Seq2Seq-based traffic prediction model and its online learning strategy are presented in Section 7.3. The numerical studies are exhibited in Section 7.4. Finally, Section 7.5 summarized this work and shed light on the future research directions.

7.2 Traffic Performance Score For Urban Mobility Measurement

Performance monitoring is critical for roadway operations, including real-time applications, operational planning, and transportation planning. According to a report from Federal Highway Administration [190], travel time is the basis for defining mobility-based performance measures. To that end, many performance measures have been designed, such as average travel speed, travel time, travel rate index, and delay per Vehicle Miles Traveled (VMT). However, most existing performance measures are road-segment or trip-based. These existing metrics cannot measure performance over a complicated road network. This section outlines a traffic performance score to measure traffic performance from the road network perspective.

7.2.1 Preliminaries

Traffic performance can be estimated from various data sources. Since, in real cases, only a portion of vehicles' trajectories can be collected, fixed-location-based sensors deployed in the whole traffic network are more robust to collecting traffic data to capture accurate network-wide traffic performance. Thus, this study chooses the freeway traffic data collected by loop inductive detector sensors deployed on freeways and urban traffic data collected by magnetic sensors mined at urban road intersections to measure the overall traffic network performance.

The freeway traffic data [i] is collected from roughly 8000 inductive loop detectors deployed on the freeway network in the northwest region of Washington State, including major freeways I-5, I-90, I-405, SR-520, etc. Representative detectors are shown by blue dots in the right map in Figure 7.1. The raw data contains lane-wise speed, volume, and occupancy (density) information collected by each loop detector. Based on the consecutive detectors' location information, freeways can be separated into tiny road segments, each containing one loop detector per lane. A road segment's length is then considered as the corresponding detector's covered length.

The Sensys [ii] magnetic sensors collect the urban traffic data mined at 45 key intersections in the Seattle Downtown area, consisting of volume and occupancy information of each lane. The intersections are illustrated by the green drops in the left map in Figure 7.1. The urban and freeway data contain different spatial and temporal resolutions; thus, their performance metrics are defined separately.

[i] Washington State Department of Transportation Traffic Data: http://data.wsdot.wa.gov/traffic/
[ii] Sensys Data: https://www.sensysnetworks.com/

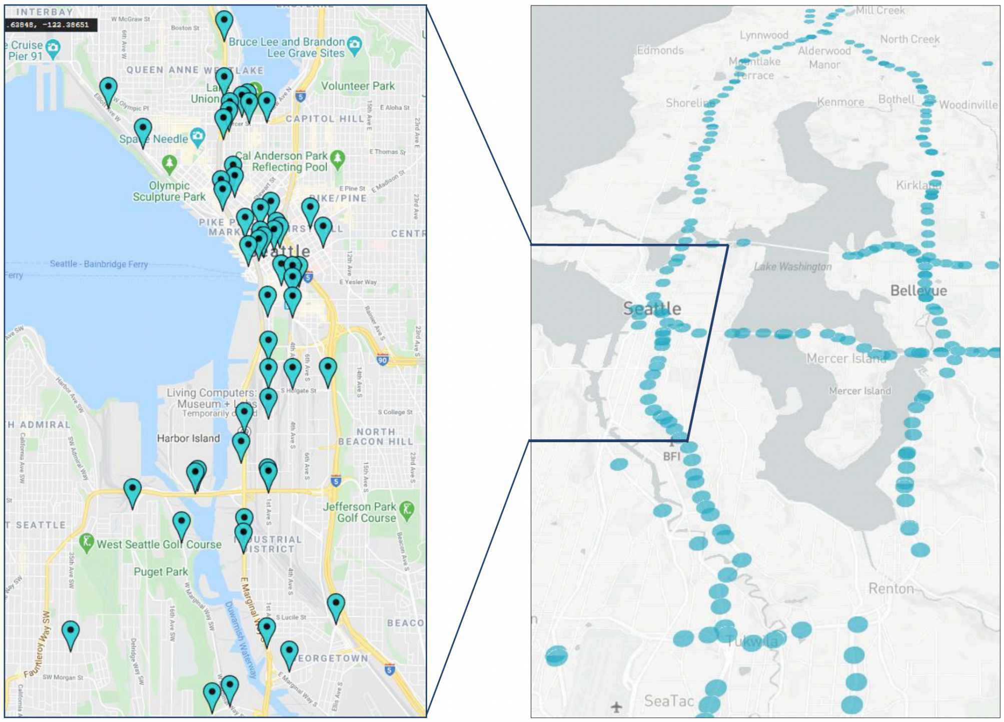


Figure 7.1: The Geospatial Locations Of The Sensors In The Urban Traffic Network (Green Drops On The Left) And The Freeway Network (Blue Dots On The Right) In The Greater Seattle Area

7.2.2 Freeway Traffic Performance Score

The physical qualities of road segments are normally constant, but the traffic parameters on each segment, such as lane-level volume (Q), speed (V), and density (K), are constantly changing. These traffic parameters for each road segment should be considered when measuring the traffic performance of the traffic network. Since these three characteristics are related to each other, one can be inferred from the other two. As a result, it is sufficient to incorporate two of them to design the traffic metric. In this study, volume and speed, which are more general and intuitive parameters, are adopted in the design of TPS. The length (L) of the road segment is also taken into consideration by multiplying L with the volume Q, which $(Q \cdot L)$ basically represents the VMT of the road segment. Then, the TPS is defined as:

$$TPS_t^F = \frac{\sum_{i=1}^{n} V_t^i \cdot Q_t^i \cdot L^i}{\sum_{i=1}^{n} V_f \cdot Q_t^i \cdot L^i} \times 100\% \tag{7.1}$$

where V_t^i and Q_t^i represent the speed and volume of each road lane i at time t. L^i is the length of i-th detector's covered road segment. V_f is the default free-flow speed. In this way, the TPS is a value ranging from 0% to 100%. Overall network-wide traffic condition is best when the TPS is 100% and worst when TPS is 0%.

7.2.3 Urban Traffic Performance Score

Comparing to freeway traffic networks, the urban traffic network is more complicated whose traffic data is more difficult to be collected. The unobservable parameters, such as the speed, will make the TPS equation defined in Equation 7.3 incomplete. To overcome this challenge, the urban TPS inherits the freeway TPS definition and attempts to infer the unobserved parameters to complete. In our urban traffic setting, the observable parameters include volume and occupancy. The average speed is $\bar{\mu}$ inferred based on traffic stream characteristics[66]:

$$\bar{\mu} = \frac{Q \cdot c_k}{Occupancy} \tag{7.2}$$

where c_k is the constant of proportionality between occupancy and density under certain simplifying assumptions. In addition, since urban traffic data is collected by single-location sensors, the length of the road segment no longer affects the measurement of road segment traffic performance. Therefore, the lengths of urban road segments are assumed the same. In this way, the urban TPS is defined as:

$$TPS_t^U = \frac{\sum_{i=1}^{n} \bar{\mu}_t^i \cdot Q_t^i \cdot \cancel{L^i}}{\sum_{i=1}^{n} V_f \cdot Q_t^i \cdot \cancel{L^i}} \times 100\% = \frac{\sum_{i=1}^{n} \bar{\mu}_t^i \cdot Q_t^i}{\sum_{i=1}^{n} V_f \cdot Q_t^i} \times 100\% \tag{7.3}$$

Hence, compared to freeway TPS, urban TPS is simplified by getting rid of L and requires more calculation of average speed.

7.2.4 Performance Metric Comparison

This section compares the TPS with existing traffic performance metrics to show its superiority. One of the widely used performance metrics is the Travel Time Index (TTI). Taking the freeway TPS as an example, the difference between TPS and TTI is analyzed in this section.

Definition of Travel Time Index

The TTI is defined in various formats by the Urban Mobility Report [iii], SHRP2 Project L03 [131], and the Urban Congestion Report [iv]. Based on the Urban Congestion Report, the TTI is the ratio of the peak-period travel time to the free-flow travel time calculated during the AM peak period (6 am to 9 am) and PM peak period (4 pm to 7 pm) on weekdays. It is averaged across urban areas, road sections, and time periods are weighted by VMT using volume. Thus, the VMT weighted averaged travel time can be represented by $\frac{\sum_{i=1}^{n}(L_i/V_i)\cdot(Q_i\cdot L_i)}{\sum_{i=1}^{n} Q_i\cdot L_i}$. Then, divided by the free-flow travel time, the TTI can be calculated as:

$$TTI = \frac{\sum_{i=1}^{n}(L_i/V_i) \cdot (Q_i \cdot L_i)}{\sum_{i=1}^{n}(L_i/V_f) \cdot (\sum_{i=1}^{n} Q_i \cdot L_i)} \tag{7.4}$$

[iii] Urban Mobility Report: https://mobility.tamu.edu/umr/report/

[iv] Urban Congestion Report: https://ops.fhwa.dot.gov/perf_measurement/ucr/

where the parameters are the same as those of TPS defined in Equation 7.3. In this way, TTI is a value with the range $[1.0, \infty)$. If TTI and TPS are calculated based on the data of only one road segment, i.e $n = 1$, the product of TPS and TTI should equal to one. If the road network is considered, i.e. $n > 1$, the product is assumed to be close to one. Another TTI definition that was not weighted by VMT can be represented as:

$$TTI^{NoW} = \frac{V_f}{\sum_{i=1}^{n} V_i / n} \tag{7.5}$$

The product of TPS and non-weighted TTI, TTI^{NoW}, is represented by:

$$TPS \times TTI^{NoW} = \frac{n \sum_{i=1}^{n} V_i \cdot Q_i \cdot L_i}{\sum_{i=1}^{n} V_i \cdot \sum_{i=1}^{n} Q_i \cdot L_i} \tag{7.6}$$

Measuring whether the product is close to one helps to validate the influence of the weight, e.g. $VMT_i = Qi \cdot Li$, in the definition of TPS.

Weighting Impact on TPS

The distribution of $TPS \times TTI$ of all segments over one month from 2020-01-21 to 2020-02-28 before COVID-19 is shown in Figure 7.2. It is obvious most of the products locate around 1.0. Then, the minor cases whose products are less than 0.9 or larger than 1.1 are analyzed. The data samples with a product less than 0.9 are shown in Figure 7.3. The three subplots with different patterns display the volume-speed pairs collected from three specific road segments located on I-5 GP lanes in the Seattle area. Although the three distributions are different from each other, it is found that the three road segments are all located at intersecting areas on the freeway system, implying the traffic patterns at these segments vary dramatically.

Since the freeway-TPS is somehow equivalent to lane-level VMT-weighted normalized averaged network speed, the different lane patterns at one site contribute to the $TPS \times TTI$ deviating from one. Thus, the difference between TPS and TTI can be analogous to the difference between average speed and weighted average speed. Thus, TPS is superior to average speed and other simple metrics because it considers the VMT (volume and road segment length). The definition of TPS is similar to TTI to some extent if TTI is weighted by volume and other factors.

7.2.5 Urban Mobility Measurement with Freeway and Urban TPS

Given the two versions of TPS defined, the urban and freeway traffic mobility patterns can be measured respectively. Figure 7.4 shows the freeway and urban TPS analysis results generated by the TPS platform, where Figure 7.4a-7.4c show the freeway TPS analysis and Figures 7.4d-7.4f display the urban TPS analysis results, respectively. As shown in Figure 7.4a and 7.4d, the freeway TPS is illustrated on a 2D colored map, where the Urban TPS is demonstrated by 3D

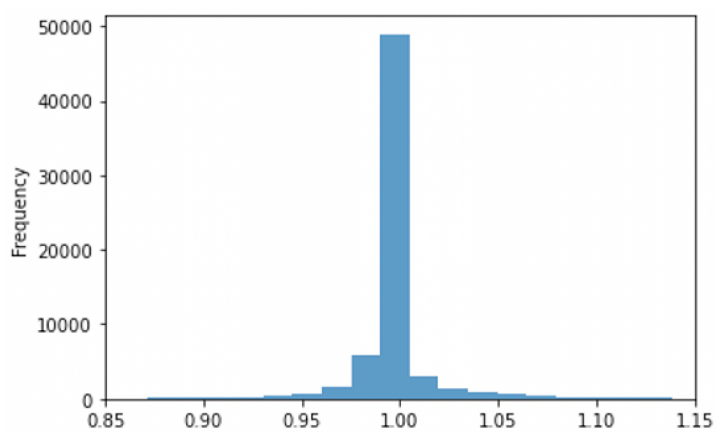


Figure 7.2: Distribution Of The Products Of TPS And TTI Over One Month In The Seattle Area

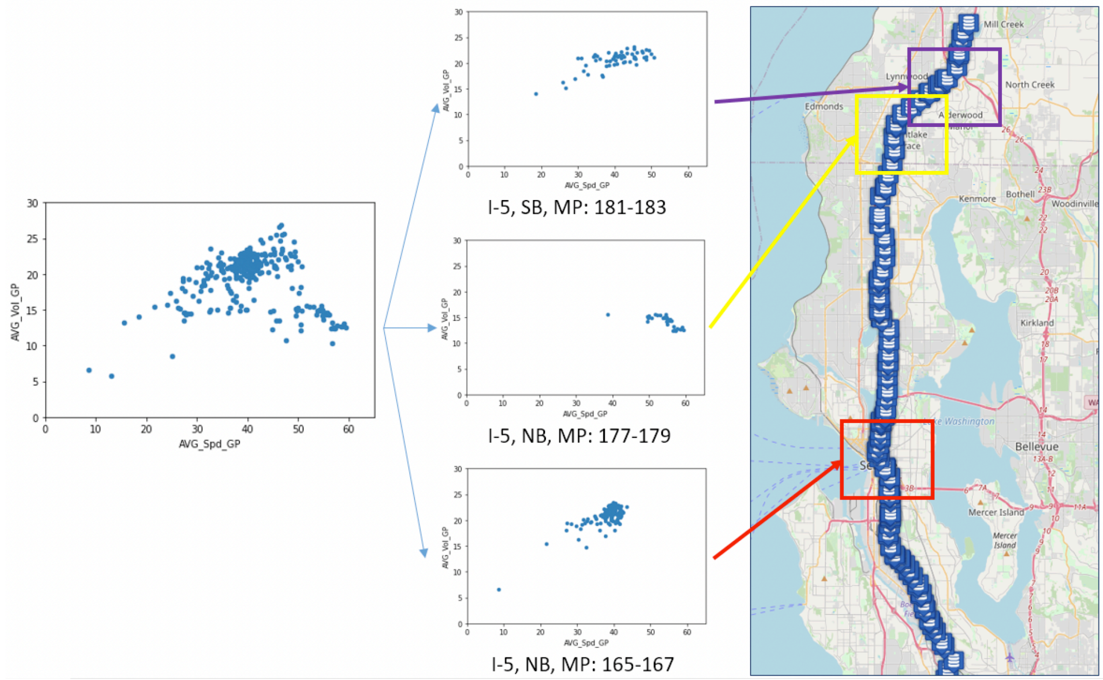
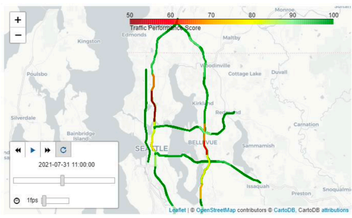


Figure 7.3: Visualization Of The Spatial Distribution Of Sample Data When $TPS \times TTI < 0.9$

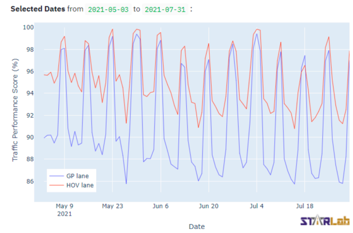
histograms on the map. The Freeway TPS can distinguish the TPS of general-purpose (GP) lanes and HOV lanes as shown in Figure 7.4b. Figure 7.4c also indicates the freeway VMT variation over the past two years where an apparent valley on the VMT curve can be observed indicating the impact of COVID-19 on traffic. Figures 7.4e and 7.4f display the daily volume and occupancy variations at urban intersections that provides us intuitive awareness of the urban traffic operations.

7.3 Traffic Performance Score Prediction Like Weather Forecast

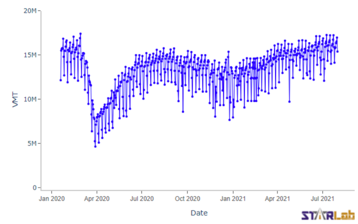
TPS can measure traffic performance for both the traffic network and road segments, and thus, it has the great potential to assist public users in planning their travel activities. To



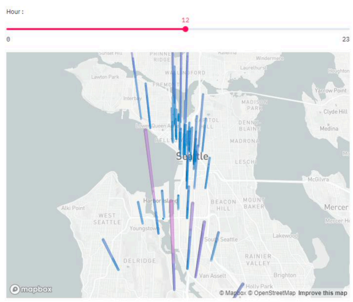
(a) Freeway TPS Map



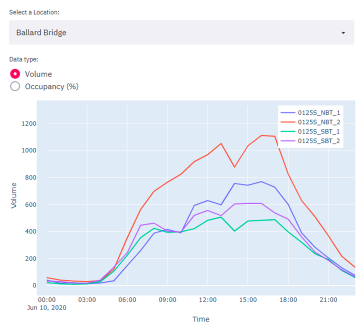
(b) Freeway TPS Of GP And HOV Lanes



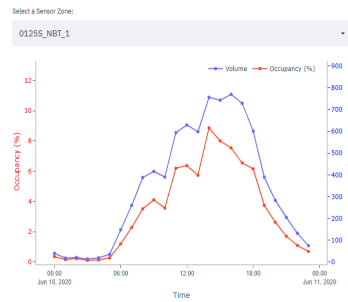
(c) Freeway VMT Over Years



(d) Urban TPS 3D Map



(e) Daily Lane-level Volume Of An Intersection



(f) Lane-level Volume And Occupancy

Figure 7.4: Freeway And Urban TPS Analysis Results Provided By The TPS Platform

provide informative future traffic status, this study also proposes a real-time TPS prediction model and implements it on the TPS platform like the weather forecast. Since the freeway TPS data forms a well-structured network-level spatial-temporal traffic data, the TPS prediction module is mainly designed for the freeway TPS. To fulfill this weather forecasting-like task, we propose a Seq2Seq-based TPS prediction model. We further propose an online learning TPS prediction strategy to enhance the model's robustness to deal with the varying network-wide traffic patterns, especially during special events like the pandemic.

7.3.1 Preliminaries

A traffic network normally consists of multiple roadway links. The traffic forecasting task targets to predict future traffic states of all (road) links or sensor stations in the traffic network based on historical traffic state data. The collected spatial-temporal traffic state data of a traffic network with S segments can be characterized as a T-step sequence $[x_1, x_2, \cdots, x_T] \in \mathbb{R}^{T \times S}$, in which $x_t \in \mathbb{R}^S$ demonstrates the traffic states, i.e. TPS in this study, of all S links at the t-th step. The traffic state of the s-th link at time t is represented by x_t^s. In this study, the superscript of a traffic state represents the spatial dimension and the subscript denotes the temporal dimension. The short-term traffic forecasting problem can be formulated as, based on T-step historical traffic state data, learning a function $F(\cdot)$ to generate the traffic states at future time steps as follows:

$$F([x_1, x_2, \cdots, x_T]) = [x_{T+1}, x_{T+2}, \cdots, x_{T+N}] \tag{7.7}$$

7.3.2 Seq2Seq-based Prediction Model

It has been shown that Long Short-term Memory (LSTM) models work well on sequence-based tasks with long-term dependencies. Therefore, the LSTM is adopted as the basic structure of the Seq2Seq based prediction. Different from LSTM, this Seq2Seq structure has an encoder receiving the input $[x_1, x_2, \cdots, x_T]$ and generating an encoded vector c and a decoder accepting c as input and generate the output $[x_{T+1}, x_{T+2}, \cdots, x_{T+N}]$. The Seq2Seq structure is depicted in Figure 7.5. The encoder and decoder are both LSTM networks. In the decoder, the output of the current step will be taken as the input of the next step to generate the whole output sequence. Note that the first step of the decoder may not have input, and the encoder output c is taken as the decoder's first input in this study.

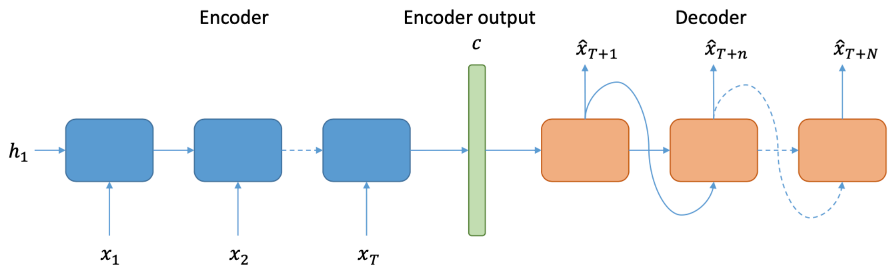


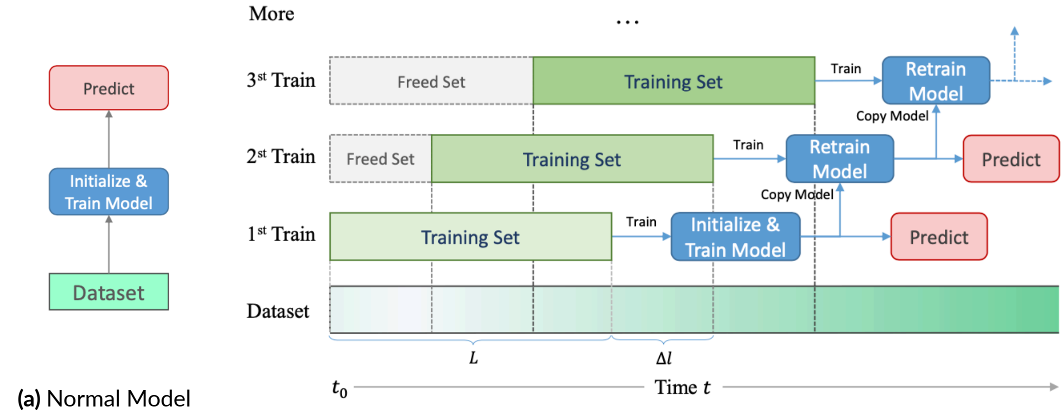
Figure 7.5: Seq2Seq Structure With The Encoder And Decoder For TPS Prediction. Encoder And Decoder Both Take LSTM As The Backbone.

7.3.3 Online Learning Strategy for TPS Prediction

The application scenarios are significant differences between the proposed TPS online learning model and other existing traffic prediction models. Existing traffic prediction studies mainly focus on designing complicated neural network structures to improve prediction accuracy, though the improvement is very limited in most cases. However, in real-world cases, the traffic pattern may change dramatically over time and a static model may not be able to deal with all traffic patterns. Especially during the COVID-19 pandemic, the traffic pattern may change every month or even every week. In this study, we target to design a novel online learning strategy to periodically update the prediction model to adapt to drastic traffic pattern changes and achieve even better prediction results.

Traditional model training and testing strategy can be described as a tandem structure as shown in Figure 7.6a that the dataset is inputted to the model for training and the well-trained model will be used to complete the prediction or classification task. However, it does not fit for on-line tasks, especially when time goes by the new data samples have different patterns/distributions.

To deal with this challenge, we propose an online training and model update strategy,



(a) Normal Model Training Prediction

(b) Online Training And Model Update Strategy

Figure 7.6: Proposed Online Learning Strategy For The Training And Updating Of The TPS Prediction Model

shown in Figure 7.6b. We consider the model to be running in a system, and the system continuously receives spatial-temporal data from outside sources. In the beginning, when the time steps of the dataset are long enough with the length of L, the dataset will be taken to the initialized model to conduct the first training. Then, after Δl time steps, the training dataset will be updated by keeping the most recent L steps of data. The early collected data will be freed and no longer be used. In this way, the training set keeps replacing a portion of the set. Once the new training set is generated, the model trained by the last dataset will be re-trained again without parameter initialization. Thus, during each Δl time step, a new model will be trained online based on previously accumulated datasets and trained parameters and be utilized for the prediction task. In this way, the newly trained model can learn the traffic pattern from the most recently collected data to enhance the prediction accuracy of the coming future time steps.

7.4 Experiments

This section compares the proposed approach with existing traffic forecasting models. The freeway TPS data is employed to test the models. Hyper-parameters, software, and hardware used in the experiments are also introduced.

7.4.1 Experiment Setting

Datasets:

In this study, we conducted experiments on freeway segment-based TPS datasets for a period of one year from 2020-07-01 to 2021-06-30. Four freeways, including I-5, I-90, I-405, and SR-520, were selected, comprising a total of 91 road segments. Each segment at a specific time has a TPS value ranging from 0 to 100%. The raw data for the freeways has a temporal resolution of one minute. In order to fulfill the online daily TPS prediction task, the temporal resolution of the TPS data was aggregated to one hour.

Hardware:

In this study, the experiments were conducted on a computer with an Intel i7-7700 CPU @ 4.2GHz processor and 32GB of memory. All the neural network-based models were trained and evaluated on a single NVIDIA GeForece GTX 1080 Ti with 11GB memory.

Baseline Models:

This study compare two widely used models:

- **GRU**[34]: Gated Recurrent Units (GRU) referring to gated recurrent units is a type of RNN. GRU can be considered as a simplified LSTM.
- **LSTM**[76]: LSTM is a powerful variant of RNN, which can overcome the gradients exploding or vanishing problem. It is suitable for being a model's basic structure for traffic forecasting.
- **Seq2Seq**: the proposed method without online training strategies.

Parameters:

The neural network models are implemented by PyTorch 1.4.0. In the training process, we use the mini-batch training strategy. The parameters of the dataset, the proposed model, and the proposed online training and learning strategy are listed as follows:

- **Dataset parameters**: The dataset contains one-year hourly TPS data with 8000 data samples. The input of the forecasting models is a 3-D vector $X \in \mathbb{R}^{b \times T \times S}$. The batch size b is set as 32 and $S = 91$ is the number of segments in the freeway TPS dataset. The length of the input sequence is set as 36 covering one and a half days, and the output sequence is set as 12 covering a half day. The samples are randomized and divided into the training, validation, and test set with the ratio 7:2:1.
- **Model parameters**: All the RNN-based models are trained by minimizing the Mean Squared Error (MSE) using the Adam optimization method[92]. The early stopping mechanism is used to avoid over-fitting. If the model improvement, i.e. the decrease of the validation loss, cannot exceed a threshold, set as 0.00001 (MSE), in 10 consecutive epochs, the training process will be terminated. We also design a learning rate decay

mechanism for the training process to speed up the models' convergence. Learning rate decay mechanism is a technique used in training modern neural networks that starts with a large learning rate and gradually reduces it until a local minimum is achieved to ensure optimal convergence and performance. It is empirically observed to help both optimization and generalization. The initial learning rate of all models is set as 10^{-4}.

- **Online training parameters**: For the sake of simplicity in this numerical test, the dataset size L for the online training strategy is set as 6000, and the updating interval Δl is set as 1000.

Evaluation Metrics

To measure the effectiveness of different traffic state prediction algorithms, widely used traffic prediction metrics [109], including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE), are computed using the following equations:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7.8}$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{7.9}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|^2}{n}} \tag{7.10}$$

where y_i is the TPS label data, and $\hat{y}_i$ is the predicted TPS.

7.4.2 Performance Comparison

The results of the prediction tests on the TPS dataset are presented in Table 7.1. It is evident that both GRU and LSTM, which are derived from vanilla recurrent neural networks, face challenges in effectively encoding sequences and generating multi-step values. On the other hand, the Seq2Seq-based model, comprising encoder and decoder modules in the form of RNNs, exhibits superior performance in multi-step traffic state prediction. This is likely due to the model's ability to dynamically adjust its weights through online learning, as demonstrated by the Seq2Seq-Online model, which was trained three times. In conclusion, the proposed Seq2Seq model with an online training strategy demonstrates a clear advantage in all three metrics.

Table 7.1: Experimental Results Of The Proposed And Baseline Models

Model	MAE	MAPE	RMSE
GRU	0.057	0.126	10.865%
LSTM	0.052	0.103	9.874%
Seq2Seq	0.04	0.094	7.172%
Seq2Seq-Online	0.034	0.083	5.720%

The effectiveness of the online training and learning strategy was also evaluated by comparing the validation losses of Seq2Seq models with and without this strategy. As depicted in Figure 7.7, the Seq2Seq model without online learning (Seq2Seq-Normal) exhibits a higher validation loss. It is notable that the first round of training for the Seq2Seq-Online model resulted in a lower validation loss, leading to improved prediction performance. This may be attributed to the smaller dataset size used for each training of the Seq2Seq-Online model in comparison to the Seq2Seq-Normal model. Furthermore, all three training processes for the Seq2Seq-Online model resulted in lower losses than the Seq2Seq-Normal model, suggesting that the online learning strategy allows for dynamic adjustment of the model to fit future changes and enhance prediction performance. It is also worth mentioning that the total number of epochs for the second and third training was significantly lower than that of the first training. This suggests that the iterative training strategy can effectively train the model based on existing training results and minimize computation in future steps.

7.4.3 TPS Prediction Model Deployment

To put the traffic prediction model into practice, we have implemented an online freeway TPS prediction model on a publicly accessible web-based traffic performance score platform. To the best of our knowledge, this is the first instance of a proposed traffic prediction model

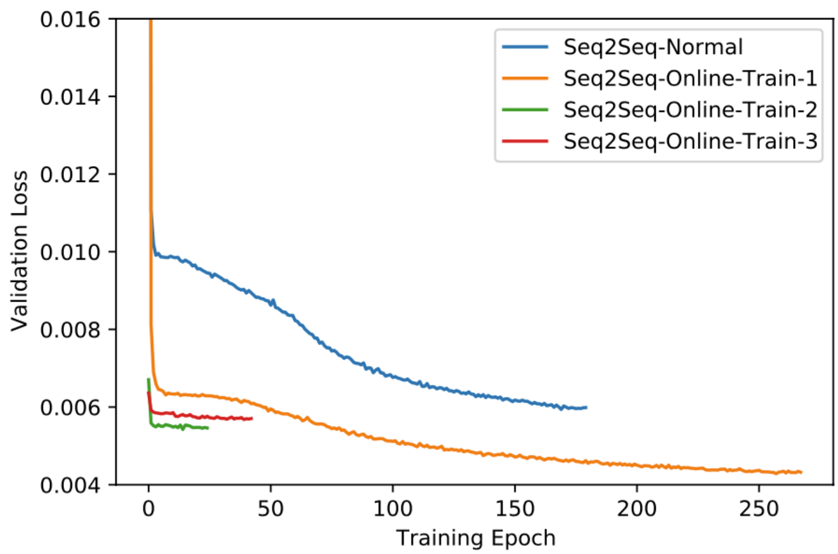


Figure 7.7: Validation Loss Of The Proposed Seq2Seq Prediction Models With The Normal Training Strategy And The Online Training-learning Strategy (The Loss Of The First Three Online Training Are Displayed)

being implemented into a real traffic data platform for actual usage. The architecture of the TPS platform, comprising three layers that integrate real-time data streaming, data source pre-processing, data storage, data analysis, data prediction, and data visualization. For more information about the TPS online prediction functions, please refer to the platform [v].

7.5 Chapter Conclusion

In this study, we propose the use of traffic performance scores as a metric for evaluating network-wide traffic performance. Using real-time loop detector data and other sources, we have developed a traffic performance score platform that displays both network-level and segment-level freeway traffic performance in the Greater Seattle area. By comparing the TPS with TTI, we demonstrate the superiority of the TPS in capturing dynamic patterns of road segments, taking into account the VMT. As traffic patterns can vary significantly over time, particularly during prolonged periods such as the ongoing pandemic, the ability to forecast traffic like weather forecasts is beneficial for public use in trip planning and other purposes. To address this challenging task of network-wide online traffic prediction, we propose a Seq2Seq-based online learning method. Our experimental results show that this method outperforms existing methods regarding prediction accuracy and learning efficiency. In addition, we have implemented the TPS measurement and its related online prediction functions on a publicly accessible platform, representing a significant contribution to our work. Network-wide traffic performance can now be predicted in a manner analogous to weather forecasts.

In the future, we plan to further investigate the advantages of the TPS as compared to

[v] TPS Platform: http://tps.uwstarlab.org/

other metrics in terms of representing traffic status during the ongoing pandemic period. We will also work to improve the proposed Seq2Seq prediction model and the online learning strategy in order to enhance the online prediction accuracy on the TPS platform.

8

Release Non-stationary Traffic Dataset and Benchmark Platform

8.1 Overview

The increasing volume and diversity of transportation data collected by modern traffic sensing and Artificial Intelligence (AI) technologies have opened up new opportunities for transportation agencies and researchers to address complicated transportation problems. In recent years, Deep Learning (DL)-based models have shown advantages in providing robust solutions to complex transportation problems that classic statistical methods cannot easily solve. However, the design of these models needs to be customized depending on specific issues and datasets. Most of the existing DL-based models are not designed initially for transportation problems.

To address the aforementioned issues, a new platform [i] has been introduced that collaborates with the Transportation Research Board (TRB) Artificial Intelligence and Advanced Computing Committee to present a series of challenges to generate AI-driven solutions for transportation problems. The platform provides benchmarked datasets and models to guide participants in data cleaning, processing, and modeling, along with an online evaluation system that ranks the modeling results. Two datasets for traffic flow prediction and pavement distress detection have been provided, respectively. Tutorials for each challenge enable transportation data scientists and engineers to evaluate state-of-the-art AI models. Researchers are encouraged to develop multiple customized models and compare their performance on these datasets. Following this standard procedure will facilitate an easy comparison of results, aiding in selecting the most effective models for further study or implementation. Additionally, the platform will save researchers valuable time by eliminating the need for reimplementing existing ideas. Instead, they could focus on different advancements based on state-of-the-art algorithms.

Overall, this platform represents a significant contribution to Intelligent Transportation Systems (ITS) and can potentially drive substantial advancements in transportation research and practice.

[i] TRBAI Open Data Challenge Platform: https://trbaiac.web.app/

8.1.1 Background

In recent years, traffic congestion has become a considerable challenge in urban environments. For example, drivers in New York City lost 117 hours on average in congestion in 2022[85]. Such delays impact individual drivers, national/local authorities, and logistics companies[113,166]. Therefore, there is a growing need for accurate traffic prediction to enable better planning and management of transportation systems.

Numerous cities have adopted ITS over the last two decades to improve urban transportation network planning and traffic management[83]. These systems utilize current and historical traffic data to enhance transport efficiency and safety by informing users of road conditions and adjusting infrastructure, such as street lights. In addition, the logistics industry relies on accurate traffic conditions to optimize scheduling, route planning, and overall performance.

Various Machine Learning (ML) and DL techniques have been employed to achieve accurate traffic predictions, capable of processing vast quantities of historical and real-time data[102,37]. DL methods, in particular, have demonstrated superior effectiveness in predicting road traffic. Despite these advancements, the design of these DL-based models remains a challenge due to the need for customization depending on specific issues and datasets. Most existing traffic datasets involve relatively stable conditions without accounting for unexpected real-world events that may significantly impact the statistical distribution of experimental data, such as COVID-19. To our best knowledge, there is a notable absence of datasets that consider non-stationary circumstances, which is a hurdle in developing robust traffic forecasting models to accommodate fluctuated conditions.

8.1.2 Contributions and Organization

To solve the aforementioned challenge, a new platform with open data challenges has been established, encouraging transportation researchers to join and solve the non-stationary issue in traffic prediction. A non-stationary Traffic Performance Score (TPS) dataset is also released[192], covering six-month data from January to June 2020, which is the initial outbreak of COVID-19. In summary, the contributions of this study can be articulated as follows:

1. Introducing a novel platform with evaluation systems to assess AI-driven solutions for addressing open transportation challenges, including non-stationary traffic forecasting and pavement distress detection.

2. Releasing a non-stationary TPS dataset to encourage researchers to address this complex issue in traffic prediction.

3. Providing several benchmark models as fundamental algorithms to allow participants more focus on advancing generalizable and robust algorithms.

4. Offering comprehensive tutorials that encompass data preprocessing, model development, and model training to guide participants through a standardized procedure, enabling seamless participation in this research/application field.

5. Ensuring all submitted models are open-source, enabling free access with proper citations for individuals to utilize in their applications.

8.2 Transportation Open Data Challenge Platform

The TRBAI Open Data Challenge Platform is a platform aimed at promoting innovation and collaboration among transportation researchers to solve open and complex questions. The platform features two data challenges - the Traffic Forecasting Challenge (Figure 8.1) and the Pavement Evaluation Challenge (Figure 8.2).

The Traffic Forecasting Challenge requires participants to develop models to predict network-wide traffic patterns accurately given non-stationary historical traffic data. The Pavement Evaluation Challenge, on the other hand, requires participants to detect and classify different types of pavement distress present in images captured from multiple sources and under different conditions.

The frontend interface of the TRBAI Open Data Challenge Platform is developed using React.js [ii], providing a user-friendly and interactive experience for participants. Account authentication is powered by Google Firebase [iii], ensuring secure access to the platform.

In order to participate in the data challenges, participants are required to sign up for an account. Upon submission of their results, the backend server, developed using Python [iv], will evaluate the results based on several evaluation metrics. It is also required for participants to provide a link to their developed model (GitHub repository [v] or a Colab link [vi]), with public access so that others may access and re-implement the models.

[ii] React.js: https://react.dev/

[iii] Google Firebase: https://firebase.google.com/

[iv] Python: https://www.python.org/

[v] Github: https://github.com/

[vi] Colab: https://colab.research.google.com/

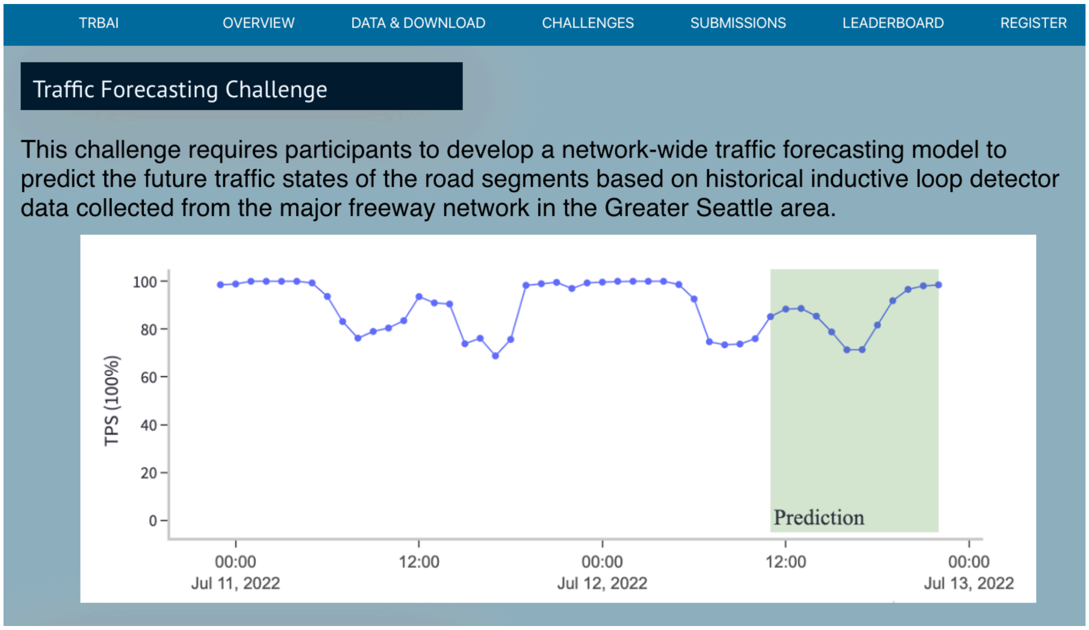


Figure 8.1: Traffic Forecasting Challenge Overview

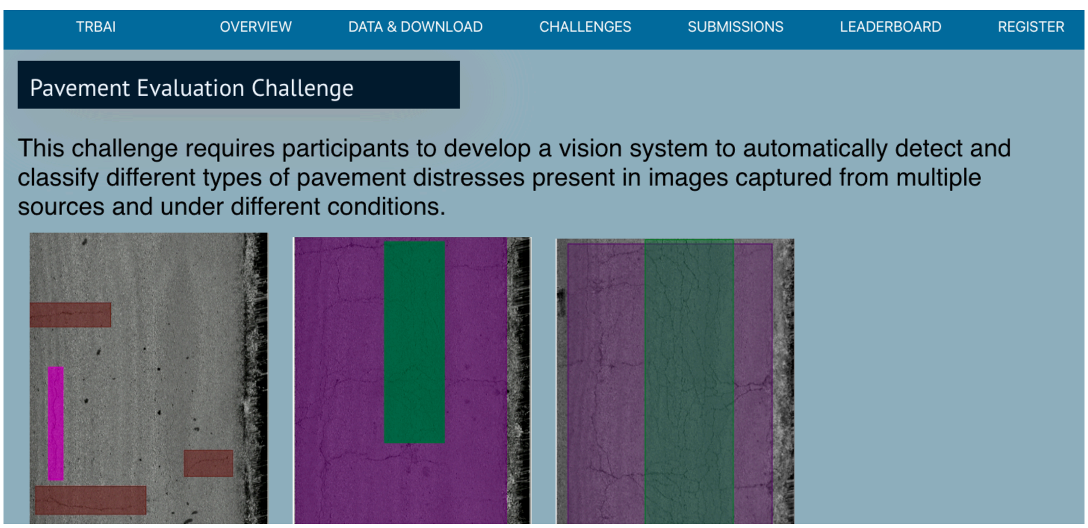


Figure 8.2: Pavement Evaluation Challenge Overview

8.3 Open-source TPS Dataset

The TRBAI Open Data Challenge Platform is a collaborative effort between our organization and an assistant professor, Yaw Adu-Gyamfi [vii], at the University of Missouri. Our organization is responsible for the Traffic Forecasting Challenge, while the professor and their team at the University of Missouri are responsible for the Pavement Evaluation Challenge. Therefore, this section will focus on introducing the non-stationary TPS dataset, which serves as the foundation for the Traffic Forecasting Challenge.

The non-stationary TPS dataset utilized in this platform was collected through the use of inductive loop detectors deployed on freeways in the Seattle metropolitan area. The freeways included in the dataset are I-5, I-405, I-90, and SR-520, as illustrated in the accompanying Figure 8.3.

This dataset represents a comprehensive collection of traffic performance data, providing information on spatiotemporal speed and volume patterns in the freeway system. The data is recorded in 15-minute increments, offering a highly granular representation of traffic patterns over time. Specifically, participants in the challenge can access a comprehensive training dataset covering the period from January to May of 2020, with fluctuating traffic patterns as shown in Figure 8.4. Additionally, the testing dataset consisting of 15 distinct time slots is provided as shown in Table 8.1. This testing dataset includes both weekdays and weekends, as well as morning and afternoon peak hours, providing a comprehensive evaluation of the models' performance under different traffic conditions.

In addition to the non-stationary TPS dataset, a well-organized adjacency matrix is also provided as shown in Figure 8.5. This matrix represents the relationships between differ-

[vii] Faculty Profile: https://engineering.missouri.edu/faculty/yaw-adu-gyamfi/

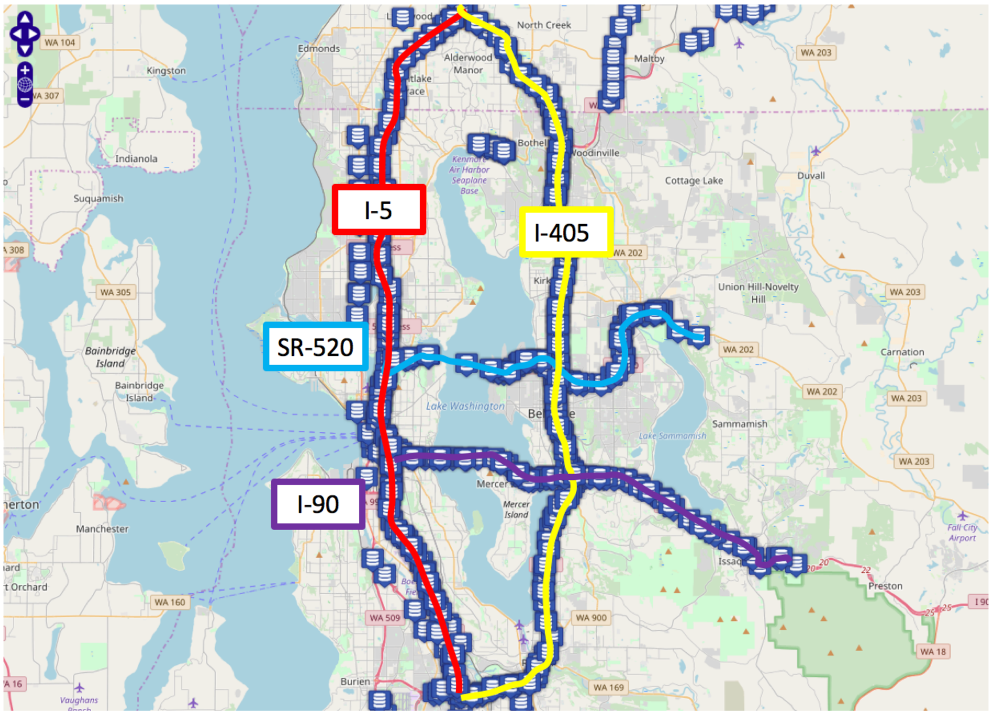


Figure 8.3: The Freeways Included In The Non-stationary TPS Dataset[41]

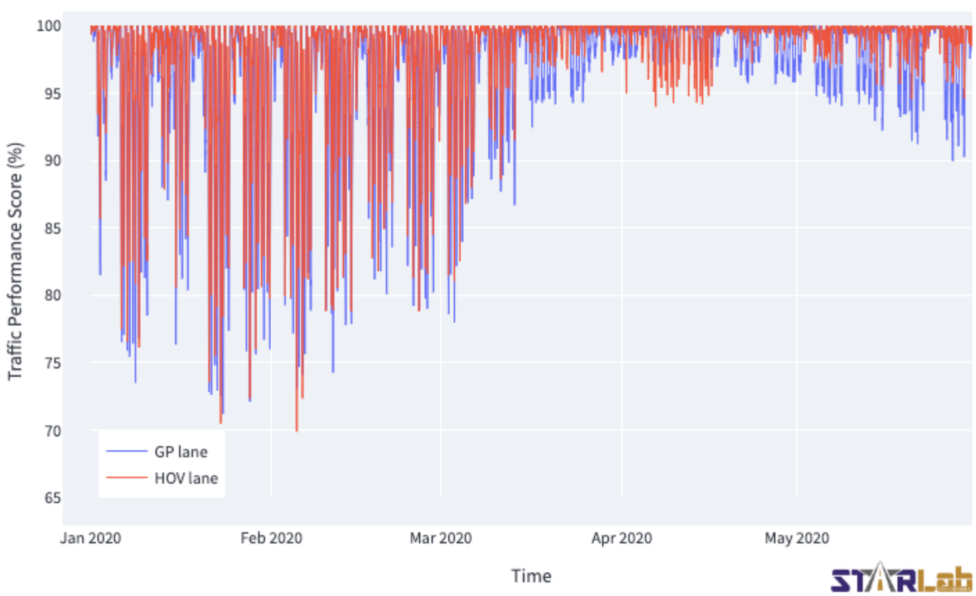


Figure 8.4: Network-wide Non-stationary Traffic Pattern In Training Dataset

Table 8.1: Testing Dataset Properties

Index	Testing Data		Forecasting Horizon	
	From	To	From	To
1	2020-06-01 21:15	2020-06-02 06:00	2020-06-02 06:15	2020-06-02 09:00
2	2020-06-02 22:15	2020-06-03 07:00	2020-06-03 07:15	2020-06-03 10:00
3	2020-06-03 23:15	2020-06-04 08:00	2020-06-04 08:15	2020-06-04 11:00
4	2020-06-05 00:15	2020-06-05 09:00	2020-06-05 09:15	2020-06-05 12:00
5	2020-06-06 01:15	2020-06-06 10:00	2020-06-06 10:15	2020-06-06 13:00
6	2020-06-07 02:15	2020-06-07 11:00	2020-06-07 11:15	2020-06-07 14:00
7	2020-06-08 03:15	2020-06-08 12:00	2020-06-08 12:15	2020-06-08 15:00
8	2020-06-09 04:15	2020-06-09 13:00	2020-06-09 13:15	2020-06-09 16:00
9	2020-06-10 05:15	2020-06-10 14:00	2020-06-10 14:15	2020-06-10 17:00
10	2020-06-11 06:15	2020-06-11 15:00	2020-06-11 15:15	2020-06-11 18:00
11	2020-06-12 07:15	2020-06-12 16:00	2020-06-12 16:15	2020-06-12 19:00
12	2020-06-13 08:15	2020-06-13 17:00	2020-06-13 17:15	2020-06-13 20:00
13	2020-06-14 09:15	2020-06-14 18:00	2020-06-14 18:15	2020-06-14 21:00
14	2020-06-15 10:15	2020-06-15 19:00	2020-06-15 19:15	2020-06-15 22:00
15	2020-06-16 11:15	2020-06-16 20:00	2020-06-16 20:15	2020-06-16 23:00

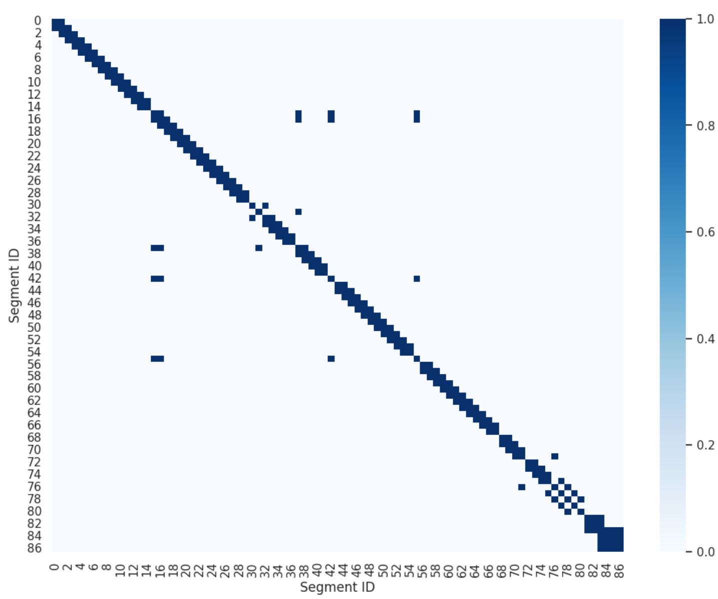


Figure 8.5: Adjacency Matrix Visualization

ent segments in the traffic network, allowing participants to treat the network as a graph if desired.

8.4 Tutorial & Benchmark

This platform provides a comprehensive tutorial (Figure 8.6) for participants in the Traffic Forecasting Challenge. This tutorial covers various topics, including data introduction, data preprocessing, accessing and reading the dataset, model implementation, and model training and validation (Figure 8.7). The tutorial is written in PyTorch style, providing participants with a clear and intuitive framework for developing their models.

In addition to the tutorial, the platform also provides a robust benchmark system for evaluating the performance of participant models. Upon submission of their results, the backend server will evaluate the models using a set of established evaluation metrics. The results will be ranked on a leaderboard as shown in Figure 8.8, giving participants a clear understanding of their model's performance and allowing them to compare their results with others.

8.5 Chapter Conclusion

The TRBAI Open Data Challenge Platform, a new platform introduced in collaboration with the Transportation Research Board's Artificial Intelligence and Advanced Computing Committee, offers a valuable resource for transportation agencies and researchers looking to tackle complex transportation problems. The platform presents challenges to generating AI-driven solutions, provides benchmarked datasets and models, and offers online evaluation systems to rank the results. The platform encourages the development of customized models, enables easy comparisons of results, and saves valuable time for researchers.

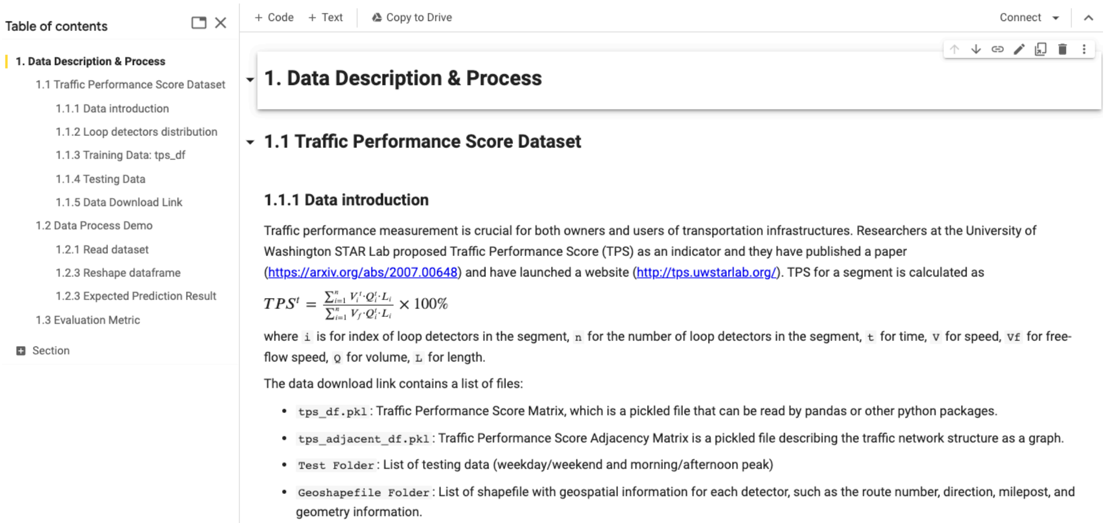
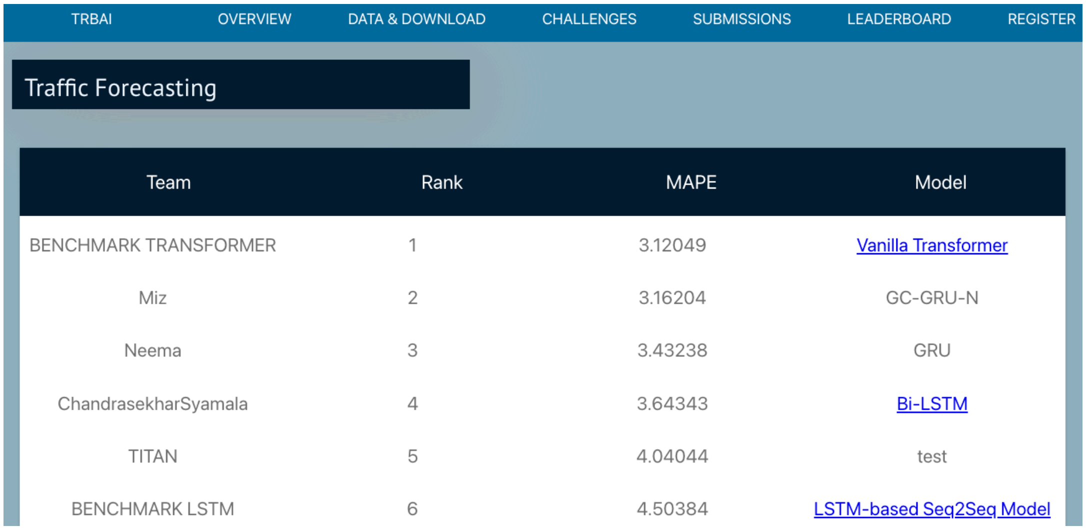


Figure 8.6: Tutorials For Traffic Forecasting Challenge

```python
# Model Implementation
class lstm_encoder(nn.Module):
    ''' Encodes time-series sequence '''

    def __init__(self, input_size, hidden_size, num_layers = 2):

        super(lstm_encoder, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = num_layers

        self.lstm = nn.LSTM(input_size=input_size, hidden_size=hidden_size,
                            num_layers=num_layers, batch_first=True)

    def forward(self, x_input):

        use_gpu = torch.cuda.is_available()
        if use_gpu:
            Hidden_State = Variable(torch.zeros(self.num_layers, x_input.size(0), self.hidden_size).cuda())
            Cell_State = Variable(torch.zeros(self.num_layers, x_input.size(0), self.hidden_size).cuda())
        else:
            Hidden_State = Variable(torch.zeros(self.num_layers, x_input.size(0), self.hidden_size))
            Cell_State = Variable(torch.zeros(self.num_layers, x_input.size(0), self.hidden_size))

        lstm_out, self.hidden = self.lstm(x_input, (Hidden_State, Cell_State))
        return lstm_out, self.hidden

class lstm_decoder(nn.Module):

    def __init__(self, input_size, hidden_size, num_layers = 2):

        super(lstm_decoder, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = num_layers

        self.lstm = nn.LSTM(input_size=input_size, hidden_size=hidden_size,
                            num_layers=num_layers, batch_first=True)

    def forward(self, x_input, encoder_hidden_states):
        lstm_out, self.hidden = self.lstm(x_input.unsqueeze(1), encoder_hidden_states)
        return lstm_out.squeeze(1), self.hidden
```

Figure 8.7: Benchmark Model Implementation Github Link



TRBAI OVERVIEW DATA & DOWNLOAD CHALLENGES SUBMISSIONS LEADERBOARD REGISTER

Traffic Forecasting

Team	Rank	MAPE	Model
BENCHMARK TRANSFORMER	1	3.12049	Vanilla Transformer
Miz	2	3.16204	GC-GRU-N
Neema	3	3.43238	GRU
ChandrasekharSyamala	4	3.64343	Bi-LSTM
TITAN	5	4.04044	test
BENCHMARK LSTM	6	4.50384	LSTM-based Seq2Seq Model

Figure 8.8: Traffic Forecasting Challenge Leaderboard

In conclusion, this platform represents a significant contribution to the field of ITS and has the potential to drive substantial advancements in transportation research and practice. By offering benchmarked datasets, models, and tutorials, the platform provides a valuable resource for transportation data scientists and engineers to evaluate state-of-the-art AI models and develop innovative solutions to real-world transportation problems.

Part V

Final Remarks

9

Final Remarks

9.1 Summary and Contributions

Around 55 percent of the global population resides in urban areas nowadays, and the United Nations predicts that this figure will rise to 68 percent by 2050[144]. As the population grows in metro regions, urban mobility challenges such as traffic congestion and increased demand for public transportation significantly impact traditional urban mobility. There has been an increasing focus on strengthening transportation infrastructure to address these complex challenges, with Intelligent Transportation Systems (ITS) emerging as a solution to improve urban transit operations from an information standpoint. The rapid advancement of traffic sensing and communication technologies has significantly increased the volume and diversity of transportation data available through ITS. This data proliferation has the potential to advance research and applications related to urban transportation and smart cities, such as traffic control, autonomous driving, and smart city infrastructure. Nonetheless, several obstacles exist in understanding non-stationary traffic data and applying relevant solutions to real-world situations. As a result, this dissertation comprehensively examines these challenges through three primary perspectives: (1) data, (2) model, and (3) application. The impact of unanticipated events like the COVID-19 pandemic on traffic patterns, the hurdles of existing traffic forecasting approaches, and the limitations of practical online prediction workflows are discussed.

The research objectives of this dissertation concentrate on understanding traffic patterns under interventions, devising robust prediction frameworks, and establishing representation learning methodologies. Additionally, the study seeks to create online learning strategies that can efficiently manage fluctuating traffic conditions. By achieving these goals, this dissertation has the potential to not only inspire the development of resilient algorithms that can adapt effectively to ever-changing traffic conditions but also to contribute valuable datasets and interactive platforms for future investigations.

In **Chapter 3**, the study introduces a cutting-edge multivariate LSTM-based network to forecast traffic parameters under external interference happened, such as those experienced during the COVID-19 pandemic. The proposed MDLSTM model integrates spatial and temporal features to boost its robustness and adapt to overcome changing traffic conditions. The MDLSTM model presents an innovative dual LSTM structure that accommodates external interventions, seamlessly fusing spatial and temporal features as input. Furthermore, an attention-based learning component is developed to balance the contributions of short- and long-term learned representations. Experiments on real-world datasets spanning the COVID-19 outbreak demonstrate the effectiveness of the MDLSTM model. Alongside improved forecasting accuracy, the MDLSTM model also exhibits enhanced computational efficiency, allowing for faster retraining and precise reflection of current traffic situations during unexpected occurrences. This unique advantage positions the MDLSTM model as an ideal solution for handling random events and adapting to fluctuating traffic conditions. The study contributes significantly to traffic forecasting by proposing an integrated dual LSTM-based model capable of accommodating interference. Future research could investigate extending the prediction horizon and enhancing the model's capabilities, which serves as the inspiration for **Chapter 4**.

Building on the success of the MDLSTM model in **Chapter 3** and expanding the prediction horizon from short-term forecasting to a more extended version, we proposed the Traffic-Twitter Transformer in Chapter **Chapter 4**. This innovative model integrates social media features to provide a flexible and comprehensive framework for predicting physical-aware, long-term traffic conditions, which is crucial for better management of future roadway capacity and accommodation of social and human impacts.

The Traffic-Twitter Transformer leverages a novel Natural Language Processing (NLP)-joined social-aware structure, incorporating both traffic and Twitter data intensity. A comprehensive correlation study was employed to evaluate the significance of the correlation between traffic and Twitter datasets. The Traffic-Twitter Transformer integrates natural language representations into traffic data for enhanced long-term traffic prediction. Experimental results demonstrated that the proposed model outperformed baseline models across all evaluation metrics. The Traffic-Twitter Transformer effectively fuses both spatial and temporal features in an end-to-end architecture, improving model robustness. It also employs a time encoder to replace the positional encoder, retaining the time dependency of data with solid temporal characteristics. Overall, the Traffic-Twitter Transformer is a valuable design for network-wide traffic prediction and management, with promising potential for future advancements in traffic forecasting. Future research directions include diving deeper into representation learning techniques, including incremental learning and self-supervised learning, to extract more meaningful and generalizable insights from network-wide traffic datasets.

In **Chapters 5** and **Chapter 6**, we explore the potential of incorporating representation learning techniques into traffic forecasting tasks, aiming to extract more meaningful and generalizable insights from network-wide traffic datasets. **Chapters 5** specifically addresses the challenges posed by learning representative traffic patterns given the constantly evolving circumstances brought on by the COVID-19 pandemic. To tackle these challenges, we propose an incremental learning-based framework for non-stationary data clustering and forecasting within transportation scenarios. This dual-module architecture consists of a Temporal Neighborhood Clustering module and an Incremental Learning module. The Temporal Neighborhood Clustering module dynamically identifies the optimal boundary for clustering statistically similar neighbors, while the Incremental Learning module employs the online-EWC approach to learn new tasks and prevent catastrophic forgetting. Experiment results indicate the framework's reliable prediction performance regarding robustness and accuracy. The key contributions of this research include introducing an incremental learning-based model for regression tasks, developing a dynamic splitting-point detector, comparing the proposed framework with well-known strategies, and utilizing a real-world shifting dataset for evaluation.

Compared to **Chapters 5**, **Chapter 6** focuses more on proposing a unified framework that integrates contrastive learning, an area of representation learning, to deal with incomplete and complex traffic data and improve robustness. The novel unified framework introduced in this chapter combines traffic representation learning and multi-contrastive learning to address these challenges. The framework indicates improved performance in spatial-temporal traffic forecasting by considering multi-scale contextual information, designing multiple perspectives of contrastive learning, and introducing generalized definitions for spatial-temporal positive/negative pairs. Experimental results reveal that the unified framework enhances the accuracy of various base models across all evaluation metrics. Key contributions of this study include the development of the framework that considers multi-scale contextual information, the design of multiple perspectives of contrastive learning, and the proposal of a generalized definition for spatial-temporal positive/negative pairs.

The summaries presented earlier provide a comprehensive understanding of the **research contributions** from Chapter 3 to Chapter 6. In contrast, Chapter 7 and Chapter 8 emphasize **practical contributions** that aim to bridge the gap between academic research and real-world applications.

Chapter 7 presents urban and freeway Traffic Performance Score (TPS) that measures network-wide traffic performance, incorporating multiple parameters for both types of traffic networks. Furthermore, a Seq2Seq-based model with an online learning strategy is introduced for network-wide traffic prediction, outperforming existing methods. This work also presents the implementation of the TPS measurement and its related online prediction functions on a publicly accessible platform, demonstrating its practical application for transportation management.

Chapter 8 introduces a pioneering platform for open data challenges, collaborating with the Transportation Research Board Artificial Intelligence and Advanced Computing Committee. The platform offers benchmarked datasets, models, and comprehensive tutorials, facilitating the development of innovative solutions for complex transportation problems, such as non-stationary traffic forecasting and pavement distress detection. The online evaluation systems rank the modeling results, enabling researchers to compare and select the most effective models for further study or implementation. Overall, **Chapter 7** and **Chapter 8** offer practical solutions that improve transportation management, representing significant contributions with potential benefits for agencies, researchers, and practitioners.

In summary, this dissertation on network-wide traffic feature learning and forecasting under non-stationary circumstances highlights advanced deep-learning models' crucial role in improving traffic forecasting's accuracy and adaptability under fluctuating conditions. The proposed techniques show encouraging solutions for overcoming traditional modeling challenges and have practical applications in real-time traffic scenarios. Our findings emphasize the importance of considering multi-scale contextual information and representation learning techniques for effective traffic forecasting. Further investigation into advanced Deep Neural Network (DNN) models raises the potential for developing more intelligent and robust urban transportation systems capable of effectively managing non-stationary traffic conditions. Our works contribute to leveraging the power of deep learning to address the complex challenges facing transportation management in today's rapidly evolving urban environments.

9.2 Future Work

Several novel pipelines and algorithms have been presented in this dissertation with promising results in addressing non-stationary conditions in traffic forecasting. Therefore, they provide a solid foundation for future work. To further extend this dissertation, a number of areas can be pursued. Firstly, given the rapid evolution of NLP techniques and their applications (such as ChatGPT [i]), it is more likely to extract meaningful semantics from social media datasets. These datasets may contain information that can potentially impact or reflect traffic conditions, thus serving as auxiliary features that can enhance the accuracy of traffic forecasting. Secondly, an alternative approach to address non-stationary circumstances could be to classify current mobility patterns and assign an appropriate forecasting model that can effectively deal with the existing conditions. Such an approach may enhance the framework's adaptability in real-world scenarios, particularly when dealing with fluctuating traffic patterns.

Thirdly, exploring the potential of training the contrastive learning component separately from the forecasting branch to learn the traffic network's representation may enhance the model's generalizability by identifying common patterns in the road network. Lastly, it is worth exploring the integration of *Generative Artificial Intelligence (AI)*[59] into the transportation field, which can create new content or data based on existing patterns and trends. This category can generate more informative and complex representations for traffic net-

[i] OpenAI ChatGPT: http://chat.openai.com/

works and potentially enhance the model's robustness and generalizability to overcome non-stationary traffic conditions. As our current work focuses on *Predictive AI*[181], investigating the potential of *Generative AI* can open new avenues for developing more intelligent transportation systems capable of handling the challenges posed by non-stationary traffic scenarios. Future research in the abovementioned areas can significantly benefit both transportation management and urban mobility.

References

[1] Abadi, A., Rajabioun, T., & Ioannou, P. A. (2014). Traffic flow prediction for road transportation networks with limited traffic data. *IEEE transactions on intelligent transportation systems*, 16(2), 653–662.

[2] Abdullah, M., Ali, N., Shah, S. A. H., Javid, M. A., & Campisi, T. (2021). Service quality assessment of app-based demand-responsive public transit services in lahore, pakistan. *Applied Sciences*, 11(4), 1911.

[3] Abnar, S., Dehghani, M., & Zuidema, W. (2020). Transferring inductive biases through knowledge distillation. *arXiv preprint arXiv:2006.00555*.

[4] Alajali, W., Zhou, W., Wen, S., & Wang, Y. (2018). Intersection traffic prediction using decision tree models. *Symmetry*, 10(9), 386.

[5] Alecsandru, C. & Ishak, S. (2004). Hybrid model-based and memory-based traffic prediction system. *Transportation Research Record*, 1879(1), 59–70.

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

[7] Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., & Tuytelaars, T. (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 139–154).

[8] Aljundi, R., Lin, M., Goujaud, B., & Bengio, Y. (2019). Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32.

[9] Amodei, D., Hernandez, D., Sastry, G., Clark, J., Brockman, G., & Sutskever, I. (2018). Ai and compute. *Retrieved from* https://openai.com/blog/ai-and-compute/.

[10] Asmelash, L. (2020). Black lives matter protests have not led to a spike in coronavirus cases, research says. *CNN, June*, 24.

[11] Bai, L., Yao, L., Li, C., Wang, X., & Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33, 17804–17815.

[12] Beaudoin, J., Farzin, Y. H., & Lawell, C.-Y. C. L. (2015). Public transit investment and sustainable transportation: A review of studies of transit's impact on traffic congestion and air quality. *Research in Transportation Economics*, 52, 15–22.

[13] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157–166.

[14] Benjamin, J. (1986). A time-series forecast of average daily traffic volume. *Transportation Research Part A: General*, 20(1), 51–60.

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

[16] Boquet, G., Morell, A., Serrano, J., & Vicario, J. L. (2020). A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection. *Transportation Research Part C: Emerging Technologies*, 115, 102622.

[17] Boto-Giralda, D., Díaz-Pernas, F. J., González-Ortega, D., Díez-Higuera, J. F., Antón-Rodríguez, M., Martínez-Zarzuela, M., & Torre-Díez, I. (2010). Wavelet-based denoising for traffic volume time series forecasting with self-organizing neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 25(7), 530–545.

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

[19] Byon, Y.-J., Ha, J. S., Cho, C.-S., Kim, T.-Y., & Yeun, C. Y. (2017). Real-time transportation mode identification using artificial neural networks enhanced with mode availability layers: A case study in dubai. *Applied Sciences*, 7(9), 923.

[20] Cai, L., Janowicz, K., Mai, G., Yan, B., & Zhu, R. (2020). Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3), 736–755.

[21] Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., & Sun, J. (2016). A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. *Transportation Research Part C: Emerging Technologies*, 62, 21–34.

[22] Campos, L. (2015). Tweets as sources in the history of contemporary science-social media trends in medical history. *Medical History*, 59(1), 148–150.

[23] Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., & Han, L. D. (2009). Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert systems with applications*, 36(3), 6164–6173.

[24] CBS News (2020). Black lives matter protesters stop traffic on fdr drive. *Retrieved from* https://www.cbsnews.com/newyork/news/protest-fdr-drive-black-lives-matter/.

[25] Center for Sustainable Systems at University of Michigan (2022). U.s. cities factsheet. *Retrieved from* https://css.umich.edu/sites/default/files/2022-09/U.S.%20Cities_CSS09-06.pdf.

[26] Cetin, M. & Comert, G. (2006). Short-term traffic flow prediction with regime switching models. *Transportation Research Record*, 1965(1), 23–31.

[27] Chan, K. Y., Dillon, T. S., Singh, J., & Chang, E. (2011). Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg–marquardt algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 13(2), 644–654.

[28] Chemouil, P., Garnier, B., & Issy-les Moulineaux, F. (1985). An adaptive short-term traffic forecasting procedure using kalman filtering. In *Proc. Eleventh Intl. Teletraffic Conf.* (pp. 516–522).

[29] Chen, C., Hu, J., Meng, Q., & Zhang, Y. (2011). Short-time traffic flow prediction with arima-garch model. In *2011 IEEE Intelligent Vehicles Symposium (IV)* (pp. 607–612).: IEEE.

[30] Chen, H. & Grant-Muller, S. (2001). Use of sequential learning for short-term traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 9(5), 319–336.

[31] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607).: PMLR.

[32] Cheng, Z., Pang, M.-S., & Pavlou, P. A. (2020). Mitigating traffic congestion: The role of intelligent transportation systems. *Information Systems Research*, 31(3), 653–674.

[33] Chikkakrishna, N. K., Hardik, C., Deepika, K., & Sparsha, N. (2019). Short-term traffic prediction using sarima and fbprophet. In *2019 IEEE 16th India Council International Conference (INDICON)* (pp. 1–4).: IEEE.

[34] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[35] Cools, M., Moons, E., & Wets, G. (2009). Investigating the variability in daily traffic counts through use of arimax and sarimax models: assessing the effect of holidays on two site locations. *Transportation research record*, 2136(1), 57–66.

[36] Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2019). Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 21(11), 4883–4894.

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

[38] Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2018). Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. *arXiv preprint arXiv:1801.02143*.

[39] Cui, Z., Ke, R., Pu, Z., & Wang, Y. (2020b). Stacked bidirectional and unidirectional lstm recurrent neural network for forecasting network-wide traffic state with missing values. *Transportation Research Part C: Emerging Technologies*, 118, 102674.

[40] Cui, Z., Tsai, M.-J., Zhu, M., Yang, H., Liu, C., & Wang, Y. (2022). Traffic performance score 2.0: Measure urban mobility and online predict near-term traffic like weather forecast. *Transportation Research Board 101st Annual Meeting*.

[41] Cui, Z., Zhu, M., Wang, S., Wang, P., Zhou, Y., Cao, Q., Kopca, C., & Wang, Y. (2020c). Traffic performance score for measuring the impact of covid-19 on urban mobility. *arXiv preprint arXiv:2007.00648*.

[42] Dalgkitsis, A., Louta, M., & Karetsos, G. T. (2018). Traffic forecasting in cellular networks using the lstm rnn. In *Proceedings of the 22nd Pan-Hellenic Conference on Informatics* (pp. 28–33).

[43] De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., & Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7), 3366–3385.

[44] De Luca, S. & Di Pace, R. (2015). Modelling users' behaviour in inter-urban carsharing program: A stated preference approach. *Transportation research part A: policy and practice*, 71, 59–76.

[45] Deloitte (2017). Fueling the future of mobility. *Electronic resource - Deloitte,* https://www2.deloitte.com/content/dam/Deloitte/cn/Documents/finance/deloitte-cn-fueling-the-future-of-mobility-en-200101.pdf.

[46] Djukic, T., Flötteröd, G., Van Lint, H., & Hoogendoorn, S. (2012). Efficient real time od matrix estimation based on principal component analysis. In *2012 15th International IEEE Conference on Intelligent Transportation Systems* (pp. 115–121).: IEEE.

[47] Doshi, K. & Yilmaz, Y. (2020). Continual learning for anomaly detection in surveillance videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 254–255).

[48] Draelos, T. J., Miner, N. E., Lamb, C. C., Cox, J. A., Vineyard, C. M., Carlson, K. D., Severa, W. M., James, C. D., & Aimone, J. B. (2017). Neurogenesis deep learning: Extending deep networks to accommodate new classes. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 526–533).: IEEE.

[49] Erhardt, G. D., Hoque, J., Chen, M., Souleyrette, R., Schmitt, D., Chaudhary, A., Rapolu, S., Kim, K., Weller, S., Sall, E., et al. (2020). *Traffic Forecasting Accuracy Assessment Research*. Number Project 08-110.

[50] Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2020). A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*, (pp. 1–24).

[51] Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2021). A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. *World Wide Web*, 24(4), 1345–1368.

[52] Fan, J. & Yao, Q. (2008). *Nonlinear time series: nonparametric and parametric methods*. Springer Science & Business Media.

[53] Fang, K. (2015). "smart mobility": Is it the time to re-think urban mobility? *Retrieved from The World Bank Group* https://blogs.worldbank.org/transport/smart-mobility-it-time-re-think-urban-mobility.

[54] Fu, R., Zhang, Z., & Li, L. (2016). Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (pp. 324–328).: IEEE.

[55] Gao, S., Rao, J., Kang, Y., Liang, Y., & Kruse, J. (2020). Mapping county-level mobility pattern changes in the united states in response to covid-19. *SIGSpatial Special*, 12(1), 16–26.

[56] Ghosh, B., Basu, B., & O'Mahony, M. (2007). Bayesian time-series model for short-term traffic flow forecasting. *Journal of transportation engineering*, 133(3), 180–189.

[57] Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

[58] Google LLC (2020). See how your community moved differently due to covid-19. *Retrieved from* https://www.google.com/covid19/mobility/.

[59] Google LLC (2023). Google cloud and nvidia bring next-generation ai infrastructure and software for large scale models and generative ai applications to enterprises. *Retrieved from* https://cloud.google.com/blog/products/ai-machine-learning/nvidia-and-google-cloud-bring-generative-ai-to-enterprises.

[60] Gu, Y., Qian, Z. S., & Chen, F. (2016). From twitter to detector: Real-time traffic incident detection using social media data. *Transportation research part C: emerging technologies*, 67, 321–342.

[61] Guo, J., Liu, Y., Yang, Q., Wang, Y., & Fang, S. (2021a). Gps-based citywide traffic congestion forecasting using cnn-rnn and c3d hybrid model. *Transportmetrica A: transport science*, 17(2), 190–211.

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

[63] Guo, S., Lin, Y., Li, S., Chen, Z., & Wan, H. (2019b). Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(10), 3913–3926.

[64] Guo, S., Lin, Y., Wan, H., Li, X., & Cong, G. (2021b). Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Transactions on Knowledge and Data Engineering*.

[65] Halko, N., Martinsson, P.-G., & Tropp, J. A. (2010). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions.

[66] Hall, F. L. (1996). Traffic stream characteristics. *Traffic Flow Theory. US Federal Highway Administration*, 36.

[67] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[68] Hassani, K. & Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. In *International conference on machine learning* (pp. 4116–4126).: PMLR.

[69] Haworth, J. & Cheng, T. (2012). Non-parametric regression for space–time forecasting under missing data. *Computers, Environment and Urban Systems*, 36(6), 538–550.

[70] He, J., Shen, W., Divakaruni, P., Wynter, L., & Lawrence, R. (2013). Improving traffic prediction with tweet semantics. In *Twenty-Third International Joint Conference on Artificial Intelligence*.

[71] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738).

[72] Heisler, S. (2020). Where and when local travel decreased from covid-19 around the world. *Retrieved from* https://blog.mapbox.com/movement-changes-around-the-world-from-covid-19-cc79db7e04c7.

[73] Hin, L. T. W. & Subramaniam, R. (2012). Creating smart cities with intelligent transportation solutions: Experiences from singapore. In *Green and ecological technologies for urban planning: Creating smart cities* (pp. 174–190). IGI Global.

[74] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02), 107–116.

[75] Hochreiter, S. & Schmidhuber, J. (1996). Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, 9.

[76] Hochreiter, S. & Schmidhuber, J. (1997a). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

[77] Hochreiter, S. & Schmidhuber, J. (1997b). Lstm can solve hard long time lag problems. *Advances in neural information processing systems*, (pp. 473–479).

[78] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29, 3451–3460.

[79] Huang, C.-W., Chiang, C.-T., & Li, Q. (2017). A study of deep learning networks on mobile traffic forecasting. In *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)* (pp. 1–6).: IEEE.

[80] Huang, R., Huang, C., Liu, Y., Dai, G., & Kong, W. (2020a). Lsgcn: Long short-term traffic prediction with graph convolutional networks. In *IJCAI* (pp. 2355–2361).

[81] Huang, T., Chakraborty, P., & Sharma, A. (2020b). Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images. *arXiv preprint arXiv:2005.04188*.

[82] Hung, C.-Y., Tu, C.-H., Wu, C.-E., Chen, C.-H., Chan, Y.-M., & Chen, C.-S. (2019). Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32.

[83] Ibrahim, M. F. (2003). Improvements and integration of a public transport system: the case of singapore. *Cities*, 20(3), 205–216.

[84] INRIX (2020). Transportation trends during a global pandemic: Monitoring covid-19's impact on our cities, businesses roadways. *Retrieved from* `https://inrix.com/covid-19-transportation-trends/`.

[85] INRIX (2022). Inrix 2022 globaltraffic scorecard. *Retrieved from* `https://inrix.com/scorecard/`.

[86] Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2020). A survey on contrastive self-supervised learning. *Technologies*, 9(1), 2.

[87] Järv, O., Ahas, R., Saluveer, E., Derudder, B., & Witlox, F. (2012). Mobile phones in a traffic flow: a geographical perspective to evening rush hour traffic analysis using call detail records. *PloS one*, 7(11), e49171.

[88] Jeon, S. & Hong, B. (2016). Monte carlo simulation-based traffic speed forecasting using historical big data. *Future Generation Computer Systems*, 65, 182–195.

[89] Jiang, W. & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, (pp. 117921).

[90] Kamarianakis, Y., Shen, W., & Wynter, L. (2012). Real-time road traffic forecasting using regime-switching space-time models and adaptive lasso. *Applied stochastic models in business and industry*, 28(4), 297–315.

[91] Karlaftis, M. G. & Vlahogianni, E. I. (2011). Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. *Transportation Research Part C: Emerging Technologies*, 19(3), 387–399.

[92] Kingma Diederik, P. & Adam, J. B. (2014). A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[93] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13), 3521–3526.

[94] Koesdwiady, A., Bedawi, S., Ou, C., & Karray, F. (2018). Non-stationary traffic flow prediction using deep learning. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)* (pp. 1–5).: IEEE.

[95] Kopuklu, O., Zheng, J., Xu, H., & Rigoll, G. (2021). Driver anomaly detection: A dataset and contrastive learning approach. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 91–100).

[96] Kuang, L., Yan, X., Tan, X., Li, S., & Yang, X. (2019). Predicting taxi demand based on 3d convolutional neural network and multi-task learning. *Remote Sensing*, 11(11), 1265.

[97] Kumar, K., Parida, M., & Katiyar, V. (2013). Short term traffic flow prediction for a non urban highway using artificial neural network. *Procedia-Social and Behavioral Sciences*, 104, 755–764.

[98] Kurzhanskiy, A. A. & Varaiya, P. (2015). Traffic management: An outlook. *Economics of transportation*, 4(3), 135–146.

[99] Lan, L. W., Sheu, J.-B., & Huang, Y.-S. (2008). Investigation of temporal freeway traffic patterns in reconstructed state spaces. *Transportation Research Part C: Emerging Technologies*, 16(1), 116–136.

[100] Lana, I., Del Ser, J., Velez, M., & Vlahogianni, E. I. (2018). Road traffic forecasting: Recent advances and new challenges. *IEEE Intelligent Transportation Systems Magazine*, 10(2), 93–109.

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

[102] Lartey, B., Homaifar, A., Girma, A., Karimoddini, A., & Opoku, D. (2021). Xgboost: a tree-based approach for traffic volume prediction. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 1280–1286).: IEEE.

[103] Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *Ieee Access*, 8, 193907–193934.

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

[105] Lee, S. & Fambro, D. B. (1999). Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation research record*, 1678(1), 179–188.

[106] Leng, D., Guo, J., Pan, L., Li, J., & Wang, X. (2021). Enhance information propagation for graph neural network by heterogeneous aggregations. *arXiv preprint arXiv:2102.04064*.

[107] Li, C., Bai, L., Liu, W., Yao, L., & Waller, S. T. (2021). Urban mobility analytics: A deep spatial-temporal product neural network for traveler attributes inference. *Transportation Research Part C: Emerging Technologies*, 124, 102921.

[108] Li, L., Su, X., Zhang, Y., Hu, J., & Li, Z. (2014). Traffic prediction, data compression, abnormal data detection and missing data imputation: An integrated study based on the decomposition of traffic time series. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)* (pp. 282–289).: IEEE.

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

[110] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*.

[111] Li, Z.-P., Yu, H., Liu, Y.-C., & Liu, F.-Q. (2008). An improved adaptive exponential smoothing model for short-term travel time forecasting of urban arterial street. *Acta automatica sinica*, 34(11), 1404–1409.

[112] Liao, B., Zhang, J., Wu, C., McIlwraith, D., Chen, T., Yang, S., Guo, Y., & Wu, F. (2018). Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 537–546).

[113] Litman, T. (2013). Transportation and public health. *Annual review of public health*, 34, 217–233.

[114] Liu, S.-y., Li, D.-w., Xi, Y.-g., & Tang, Q.-f. (2015). A short-term traffic flow forecasting method and its applications. *Journal of Shanghai Jiaotong University (Science)*, 20(2), 156–163.

[115] Liu, Y., Li, Z., Pan, S., Gong, C., Zhou, C., & Karypis, G. (2021). Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE transactions on neural networks and learning systems*, 33(6), 2378–2392.

[116] Liu, Y., Zheng, H., Feng, X., & Chen, Z. (2017). Short-term traffic flow prediction with conv-lstm. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)* (pp. 1–6).: IEEE.

[117] Lomonaco, V., Pellegrini, L., Rodriguez, P., Caccia, M., She, Q., Chen, Y., Jodelet, Q., Wang, R., Mai, Z., Vazquez, D., et al. (2022). Cvpr 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions. *Artificial Intelligence*, 303, 103635.

[118] Lopez-Paz, D. & Ranzato, M. (2017). Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

[119] Lu, H., Ge, Z., Song, Y., Jiang, D., Zhou, T., & Qin, J. (2021a). A temporal-aware lstm enhanced by loss-switch mechanism for traffic flow forecasting. *Neurocomputing*, 427, 169–178.

[120] Lu, J., Li, B., Li, H., & Al-Barakani, A. (2021b). Expansion of city scale, traffic modes, traffic congestion, and air pollution. *Cities*, 108, 102974.

[121] Luo, X., Li, D., Yang, Y., & Zhang, S. (2019). Spatiotemporal traffic flow prediction with knn and lstm. *Journal of Advanced Transportation*, 2019.

[122] Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4), 818.

[123] Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54, 187–197.

[124] Ma, X., Zhong, H., Li, Y., Ma, J., Cui, Z., & Wang, Y. (2020). Forecasting transportation network speed using deep capsule networks with nested lstm models. *IEEE Transactions on Intelligent Transportation Systems*, 22(8), 4813–4824.

[125] Madan, R. & Mangipudi, P. S. (2018). Predicting computer network traffic: a time series forecasting approach using dwt, arima and rnn. In *2018 Eleventh International Conference on Contemporary Computing (IC3)* (pp. 1–5).: IEEE.

[126] Mai, T., Ghosh, B., & Wilson, S. (2012). *Multivariate short-term traffic flow forecasting using Bayesian vector autoregressive moving average model*. Technical report.

[127] Maitra, B., Sikdar, P., & Dhingra, S. (1999). Modeling congestion on urban roads and assessing level of service. *Journal of Transportation Engineering*, 125(6), 508–514.

[128] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3), e0194889.

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

[130] Marchant, A. (2020). What can traffic data tell us about the impact of the coronavirus? *Retrieved from* https://www.tomtom.com/newsroom/explainers-and-insights/covid-19-traffic/.

[131] Margiotta, R. A., Lomax, T. J., Hallenbeck, M. E., Dowling, R. G., Skabardonis, A., Turner, S., Systematics, C., et al. (2013). *Analytical procedures for determining the impacts of reliability mitigation strategies*. Technical report, National Research Council (US). Transportation Research Board.

[132] Martin, E. & Shaheen, S. (2011). The impact of carsharing on public transit and non-motorized travel: an exploration of north american carsharing survey data. *Energies*, 4(11), 2094–2114.

[133] Mermillod, M., Bugaiska, A., & Bonin, P. (2013). The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Frontiers in psychology*, 4, 504.

[134] Mescheder, D. (2020). Can traffic data help us find a way out of the lockdown? *Retrieved from* https://www.tomtom.com/newsroom/explainers-and-insights/covid-19-traffic-mobility/.

[135] Metz, D. (2018). Tackling urban traffic congestion: The experience of london, stockholm and singapore. *Case Studies on Transport Policy*, 6(4), 494–498.

[136] Mfenjou, M. L., Ari, A. A. A., Abdou, W., Spies, F., et al. (2018). Methodology and trends for an intelligent transport system in developing countries. *Sustainable Computing: Informatics and Systems*, 19, 96–111.

[137] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[138] Min, X., Hu, J., & Zhang, Z. (2010). Urban traffic network modeling and short-term traffic flow forecasting based on gstarima model. In *13th International IEEE Conference on Intelligent Transportation Systems* (pp. 1535–1540).: IEEE.

[139] Mirshahi, M., Obenberger, J., Fuhs, C. A., Howard, C. E., Krammes, R. A., Kuhn, B. T., Mayhew, R. M., Moore, M. A., Sahebjam, K., Stone, C. J., et al. (2007). *Active traffic management: the next step in congestion management*. Technical report, United States. Federal Highway Administration.

[140] Moorthy, C. & Ratcliffe, B. (1988). Short term traffic forecasting using time series methods. *Transportation planning and technology*, 12(1), 45–56.

[141] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., & Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern recognition*, 45(1), 521–530.

[142] Nadi, A., Sharma, S., van Lint, J., Tavasszy, L., & Snelder, M. (2022). A data-driven traffic modeling for analyzing the impacts of a freight departure time shift policy. *Transportation Research Part A: Policy and Practice*, 161, 130–150.

[143] Nagy, A. M. & Simon, V. (2018). Survey on traffic prediction in smart cities. *Pervasive and Mobile Computing*, 50, 148–163.

[144] Nations, U. (2018). 68% of the world population projected to live in urban areas by 2050. *Retrieved from* https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html.

[145] Nguyen, C. V., Li, Y., Bui, T. D., & Turner, R. E. (2017). Variational continual learning. *arXiv preprint arXiv:1710.10628*.

[146] Nguyen, H., Liu, W., & Chen, F. (2016). Discovering congestion propagation patterns in spatio-temporal traffic data. *IEEE Transactions on Big Data*, 3(2), 169–180.

[147] Nihan, N. L. & Holmesland, K. O. (1980). Use of the box and jenkins time series technique in traffic forecasting. *Transportation*, 9(2), 125–143.

[148] Njord, J., Peters, J., Freitas, M., Warner, B., Allred, K. C., Bertini, R. L., Bryant, R., Callan, R., Knopp, M., Knowlton, L., et al. (2006). *Safety applications of intelligent transportation systems in Europe and Japan.* Technical report, United States. Federal Highway Administration. Office of International Programs.

[149] Oh, S., Byon, Y.-J., Jang, K., & Yeo, H. (2015). Short-term travel-time prediction on highway: a review of the data-driven approach. *Transport Reviews*, 35(1), 4–32.

[150] Ojeda, L. L., Kibangou, A. Y., & De Wit, C. C. (2013). Adaptive kalman filtering for multi-step ahead traffic flow prediction. In *2013 American Control Conference* (pp. 4724–4729).: IEEE.

[151] Okutani, I. & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1), 1–11.

[152] Papageorgiou, M., Ben-Akiva, M., Bottom, J., Bovy, P. H., Hoogendoorn, S. P., Hounsell, N. B., Kotsialos, A., & McDonald, M. (2007). Its and traffic management. *Handbooks in operations research and management science*, 14, 715–774.

[153] Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310–1318).: PMLR.

[154] Paul, A., Chilamkurti, N., Daniel, A., & Rho, S. (2017). Big data collision analysis framework. In *Intelligent Vehicular Networks and Communications* (pp. 177–184).: Elsevier.

[155] Pavlyuk, D. (2017). Short-term traffic forecasting using multivariate autoregressive models. *Procedia Engineering*, 178, 57–66.

[156] Petalas, Y. G., Ammari, A., Georgakis, P., & Nwagboso, C. (2016). A big data architecture for traffic forecasting using multi-source information. In *International Workshop of Algorithmic Aspects of Cloud Computing* (pp. 65–83).: Springer.

[157] Pfülb, B., Gepperth, A., Abdullah, S., & Kilian, A. (2018). Catastrophic forgetting: still a problem for dnns. In *International conference on artificial neural networks* (pp. 487–497).: Springer.

[158] Prasad, K. S. N. & Ramakrishna, S. (2014). An efficient traffic forecasting system based on spatial data and decision trees. *Int. Arab J. Inf. Technol.*, 11(2), 186–194.

[159] Price, L. (2020). Impact of covid-19 on small businesses – where is it worst? *Retrieved from* https://smallbiztrends.com/2020/04/impact-of-coronavirus-on-small-businesses.html.

[160] Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., & Cui, Y. (2021). Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6964–6974).

[161] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10), 1872–1897.

[162] Ramakrishnan, N. & Soni, T. (2018). Network traffic prediction using recurrent neural networks. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 187–193).: IEEE.

[163] Reza, S., Ferreira, M. C., Machado, J., & Tavares, J. M. R. (2022). A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. *Expert Systems with Applications*, 202, 117275.

[164] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

[165] Ryu, S., Kim, D., & Kim, J. (2020). Weather-aware long-range traffic forecast using multi-module deep neural network. *Applied Sciences*, 10(6), 1938.

[166] Sankaran, J. K., Gore, A., & Coldwell, B. (2005). The impact of road traffic congestion on supply chains: insights from auckland, new zealand. *International Journal of Logistics: Research and Applications*, 8(2), 159–180.

[167] Schwarz, J., Czarnecki, W., Luketina, J., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., & Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning* (pp. 4528–4537).: PMLR.

[168] Shekhar, S. & Williams, B. M. (2007). Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record*, 2024(1), 116–125.

[169] Shi, Z. & Fang, Y. (2020). Temporal relationship between outbound traffic from wuhan and the 2019 coronavirus disease (covid-19) incidence in china. *MedRxiv*.

[170] Shin, H., Lee, J. K., Kim, J., & Kim, J. (2017). Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

[171] Singh, B. & Gupta, A. (2015). Recent trends in intelligent transportation systems: a review. *Journal of Transport Literature*, 9, 30–34.

[172] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics*, 14(11), 4724–4734.

[173] Smith, B. L. & Demetsky, M. J. (1997). Traffic flow forecasting: comparison of modelling approaches. *Journal of Transportation Engineering, 123(4):261–266*.

[174] Smith, B. L., Williams, B. M., & Oswald, R. K. (2002). Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4), 303–321.

[175] Stathopoulos, A. & Karlaftis, M. G. (2003). A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C: Emerging Technologies*, 11(2), 121–135.

[176] Sugiyama, M. & Kawanabe, M. (2012). *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

[177] Sun, H., Liu, H. X., Xiao, H., He, R. R., & Ran, B. (2003). Use of local linear regression model for short-term traffic forecasting. *Transportation Research Record*, 1836(1), 143–150.

[178] Sun, S., Zhang, C., & Yu, G. (2006). A bayesian network approach to traffic flow forecasting. *IEEE Transactions on intelligent transportation systems*, 7(1), 124–132.

[179] Surnin, O., Ivaschenko, A., Sitnikov, P., Suprun, A., Stolbova, A., & Golovnin, O. (2019). Urban public transport digital planning based on an intelligent transportation system. In *2019 25th Conference of Open Innovations Association (FRUCT)* (pp. 292–298).: IEEE.

[180] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

[181] SwissCognitive (2023). Predictive analytics vs. generative ai choosing the right technology for your marketing business. *Retrieved from* https://swisscognitive.ch/2023/03/30/predictive-analytics-vs-generative-ai-choosing-the-right-technology-for-your-marketing-bu

[182] Szeto, W., Ghosh, B., Basu, B., & O'Mahony, M. (2009). Multivariate traffic forecasting technique using cell transmission model and sarima model. *Journal of Transportation Engineering*, 135(9), 658–667.

[183] Terama, E., Peltomaa, J., Rolim, C., & Baptista, P. (2018). The contribution of car sharing to the sustainable mobility transition. *Transfers*, 8(2), 113–121.

[184] Tian, Y. & Pan, L. (2015). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)* (pp. 153–158).: IEEE.

[185] Tonekaboni, S., Eytan, D., & Goldenberg, A. (2021). Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*.

[186] Tsai, M.-J., Chen, H.-Y., Cui, Z., & Wang, Y. (2021). Multivariate long and short term lstm-based network for traffic forecasting under interference: Experiments during covid-19. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)* (pp. 2169–2174).: IEEE.

[187] Tsai, M.-J., Cui, Z., Liu, C., Yang, H., & Wang, Y. (2022a). An incremental learning-based framework for non-stationary traffic representations clustering and forecasting. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 3237–3242).: IEEE.

[188] Tsai, M.-J., Cui, Z., Wang, Y., et al. (2022b). Traffic-twitter transformer: A nature language processing-joined framework for network-wide traffic forecasting. *arXiv preprint arXiv:2206.11078*.

[189] Tsirigotis, L., Vlahogianni, E. I., & Karlaftis, M. G. (2012). Does information on weather affect the performance of short-term traffic forecasting models? *International journal of intelligent transportation systems research*, 10(1), 1–10.

[190] Turner, S. M., Margiotta, R., Lomax, T., et al. (2004). *Lessons learned: monitoring highway congestion and reliability using archived traffic detector data*. Technical report, United States. Federal Highway Administration. Office of Operations.

[191] U. S. Department of Transportation Federal Highway Administration (2021). *Retrieved from* https://www.fhwa.dot.gov/policyinformation/travel_monitoring/tvt.cfm.

[192] UW STAR Lab (2020). Traffic performance score website. *Retrieved from* http://tps.uwstarlab.org/.

[193] Van Arem, B., Kirby, H. R., Van Der Vlist, M. J., & Whittaker, J. C. (1997). Recent advances and applications in the field of short-term traffic forecasting. *International journal of forecasting*, 13(1), 1–12.

[194] Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining kohonen maps with arima time series models to forecast traffic flow. *Transportation Research Part C: Emerging Technologies*, 4(5), 307–318.

[195] Van Lint, J., Hoogendoorn, S., & van Zuylen, H. J. (2002). Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks. *Transportation Research Record*, 1811(1), 30–39.

[196] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[197] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[198] Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2018). Deep graph infomax. *arXiv preprint arXiv:1809.10341*.

[199] Vinayakumar, R., Soman, K., & Poornachandran, P. (2017). Applying deep learning approaches for network traffic prediction. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 2353–2358).: IEEE.

[200] Vlahogianni, E. I., Golias, J. C., & Karlaftis, M. G. (2004). Short-term traffic forecasting: Overview of objectives and methods. *Transport reviews*, 24(5), 533–557.

[201] Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43, 3–19.

[202] Wang, J. & Shi, Q. (2013). Short-term traffic speed forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory. *Transportation Research Part C: Emerging Technologies*, 27, 219–232.

[203] Wang, X., Ma, Y., Wang, Y., Jin, W., Wang, X., Tang, J., Jia, C., & Yu, J. (2020). Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020* (pp. 1082–1092).

[204] Wang, Y., Papageorgiou, M., & Messmer, A. (2006). Renaissance – a unified macroscopic model-based approach to real-time freeway network traffic surveillance. *Transportation Research Part C: Emerging Technologies*, 14(3), 190–212.

[205] Washington State Department of Transportation (2022). *Retrieved from* https://wsdot.wa.gov/travel/holiday-weekend-travel/labor-day-weekend-travel-times/i-90-travel-charts.

[206] Wei, W., Wu, H., & Ma, H. (2019). An autoencoder and lstm-based traffic flow prediction method. *Sensors*, 19(13), 2946.

[207] Westervelt, M., Huang, E., Schank, J., Borgman, N., Fuhrer, T., Peppard, C., & Narula-Woods, R. (2018). Uprouted: Exploring microtransit in the united states.

[208] Williams, B. M., Durvasula, P. K., & Brown, D. E. (1998). Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record*, 1644(1), 132–141.

[209] Williams, B. M. & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6), 664–672.

[210] Work, D. B. & Bayen, A. M. (2008). Impacts of the mobile internet on transportation cyberphysical systems: traffic monitoring using smartphones. In *National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation, & Rail* (pp. 18–20).

[211] Wu, H., Xu, J., Wang, J., & Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34, 22419–22430.

[212] Wu, X., Huang, C., Zhang, C., & Chawla, N. V. (2020). Hierarchically structured transformer networks for fine-grained spatial event forecasting. In *Proceedings of The Web Conference 2020* (pp. 2320–2330).

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

[214] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.

[215] Xia, D., Wang, B., Li, H., Li, Y., & Zhang, Z. (2016). A distributed spatial–temporal weighted model on mapreduce for short-term traffic flow forecasting. *Neurocomputing*, 179, 246–263.

[216] Xiangxue, W., Lunhui, X., & Kaixun, C. (2019). Data-driven short-term forecasting for urban road network traffic based on data processing and lstm-rnn. *Arabian Journal for Science and Engineering*, 44(4), 3043–3060.

[217] Xiao, H., Xiao, J., Deng, X., & Li, L. (2021). Traffic flow prediction based on traffic and meteorological data fusion in non-stationary environments. In *2021 International Conference on Electronic Information Technology and Smart Agriculture (ICEITSA)* (pp. 154–158).: IEEE.

[218] Xing, W., Yao, J., Liu, Z., Liu, W., Zhang, S., & Wang, L. (2022). Contrastive js: A novel scheme for enhancing the accuracy and robustness of deep models. *IEEE Transactions on Multimedia*.

[219] Xu, Y., Zhong, X., Yepes, A. J. J., & Lau, J. H. (2020). Forget me not: Reducing catastrophic forgetting for domain adaptation in reading comprehension. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).: IEEE.

[220] Yan, H., Ma, X., & Pu, Z. (2021). Learning dynamic and hierarchical traffic spatiotemporal features with transformer. *IEEE Transactions on Intelligent Transportation Systems*.

[221] Yang, B., Sun, S., Li, J., Lin, X., & Tian, Y. (2019). Traffic flow prediction using lstm with feature enhancement. *Neurocomputing*, 332, 320–327.

[222] Yao, W. & Qian, S. (2021). From twitter to traffic predictor: Next-day morning traffic prediction using social media data. *Transportation research part C: emerging technologies*, 124, 102938.

[223] You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., & Shen, Y. (2020). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33, 5812–5823.

[224] Yu, B., Lee, Y., & Sohn, K. (2020). Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (gcn). *Transportation research part C: emerging technologies*, 114, 189–204.

[225] Yu, B., Song, X., Guan, F., Yang, Z., & Yao, B. (2016). k-nearest neighbor model for multiple-time-step prediction of short-term traffic condition. *Journal of Transportation Engineering*, 142(6), 04016018.

[226] Yu, B., Yin, H., & Zhu, Z. (2017a). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.

[227] Yu, H., Jiang, R., He, Z., Zheng, Z., Li, L., Liu, R., & Chen, X. (2021). Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives. *Transportation research part C: emerging technologies*, 127, 103101.

[228] Yu, H., Wu, Z., Wang, S., Wang, Y., & Ma, X. (2017b). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7), 1501.

[229] Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017c). Deep learning: A generic approach for extreme condition traffic forecasting. In *Proceedings of the 2017 SIAM international Conference on Data Mining* (pp. 777–785).: SIAM.

[230] Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., & Xu, B. (2022). Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36 (pp. 8980–8987).

[231] Zeng, D., Xu, J., Gu, J., Liu, L., & Xu, G. (2008). Short term traffic flow prediction using hybrid arima and ann models. In *2008 Workshop on Power Electronics and Intelligent Transportation System* (pp. 621–625).: IEEE.

[232] Zenke, F., Poole, B., & Ganguli, S. (2017). Continual learning through synaptic intelligence. In *International Conference on Machine Learning* (pp. 3987–3995).: PMLR.

[233] Zhang, C., Yang, Z., He, X., & Deng, L. (2020). Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 14(3), 478–493.

[234] Zhang, G. P. (2003). Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50, 159–175.

[235] Zhang, J., Wang, F.-Y., Wang, K., Lin, W.-H., Xu, X., & Chen, C. (2011). Data-driven intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 12(4), 1624–1639.

[236] Zhang, L., Alharbe, N. R., Luo, G., Yao, Z., & Li, Y. (2018a). A hybrid forecasting framework based on support vector regression with a modified genetic algorithm and a random forest for traffic flow prediction. *Tsinghua Science and Technology*, 23(4), 479–492.

[237] Zhang, W., Yu, Y., Qi, Y., Shu, F., & Wang, Y. (2019a). Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning. *Transportmetrica A: Transport Science*, 15(2), 1688–1711.

[238] Zhang, Y. & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, 58, 308–324.

[239] Zhang, Y., Wang, S., Chen, B., & Cao, J. (2019b). Gcgan: Generative adversarial nets with graph cnn for network-scale traffic prediction. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).: IEEE.

[240] Zhang, Y. & Zhang, Y. (2016). A comparative study of three multivariate short-term freeway traffic flow forecasting methods with missing data. *Journal of Intelligent Transportation Systems*, 20(3), 205–218.

[241] Zhang, Z., He, Q., Gao, J., & Ni, M. (2018b). A deep learning approach for detecting traffic accidents from social media data. *Transportation research part C: emerging technologies*, 86, 580–596.

[242] Zhang, Z., Li, M., Lin, X., Wang, Y., & He, F. (2019c). Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation research part C: emerging technologies*, 105, 297–322.

[243] Zhao, W. (2022). Accurate non-stationary short-term traffic flow prediction method. *arXiv preprint arXiv:2205.00517*.

[244] Zhao, Z., Chen, W., Wu, X., Chen, P., & Liu, J. (2017a). Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11, 68–75.

[245] Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017b). Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68–75.

[246] Zheng, C., Fan, X., Wang, C., & Qi, J. (2020a). Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34 (pp. 1234–1241).

[247] Zheng, H., Lin, F., Feng, X., & Chen, Y. (2020b). A hybrid deep learning model with attention-based conv-lstm networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 6910–6920.

[248] Zheng, Z. & Su, D. (2014). Short-term traffic volume forecasting: A k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm. *Transportation Research Part C: Emerging Technologies*, 43, 143–157.

[249] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35 (pp. 11106–11115).

[250] Zhou, T., Jiang, D., Lin, Z., Han, G., Xu, X., & Qin, J. (2019). Hybrid dual kalman filtering model for short-term traffic flow forecasting. *IET Intelligent Transport Systems*, 13(6), 1023–1032.

[251] Zhou, X. & Mahmassani, H. S. (2007). A structural state space model for real-time traffic origin–destination demand estimation and prediction in a day-to-day learning framework. *Transportation Research Part B: Methodological*, 41(8), 823–840.

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