Impact of Weather Conditions on Macroscopic Traffic Stream Variables in an Intelligent Transportation System

by

ARCHANA NIGAM
201421001

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
to

Dhirubhai Ambani Institute of Information and Communication Technology

November, 2021

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Archana Nigam

Certificate

This is to certify that the thesis work entitled IMPACT OF WEATHER CONDITIONS ON MACROSCOPIC TRAFFIC STREAM VARIABLES IN AN INTELLIGENT TRANSPORTATION SYSTEM has been carried out by ARCHANA NIGAM for the degree of Doctor of Philosophy at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Sanjay Srivastava
Thesis Supervisor

To my mother and mother in law,
For all their support and encouragement.

Acknowledgments

This was probably the hardest part of the thesis to put in words. First and above all, I am thankful to God, for providing me this opportunity and granting me the capability to fulfill this dream. I would like to offer my sincere thanks to all who supported directly and indirectly to this research.

My sincere gratitude is for my supervisor Prof. Sanjay Srivastava, with whom I have learned immensely and has had a strong influence on my development as a researcher. As he quotes *"Your Ph.D. is not just your work but the transformation of you as a researcher"*. These many years in DAIICT with his support have transformed me into a better person. I am eternally indebted to him for his support, encouragement, and inspiration. I have never seen someone being more selfless, having more of his students' interests in mind, or giving them enough freedom to work at their own pace and still ensuring that we stay focused on the goal.

A very special thanks to Prof. Maniklal Das, Prof. Rahul Muthu, and Prof. V. Sunitha for their valuable advice and feedback on my research. I would also like to convey thanks to Prof. Alka Parikh for teaching me many skills apart from research during my tenure at DA-IICT. I am also thankful to DA-IICT for providing me the opportunity to pursue Ph.D.

A very warm thanks to my senior Dr. Manish Chaturvedi for motivating me during various phases of my thesis and helping me enormously. A special thanks to Nikita Joshi for always being there for me and kept me in good humor throughout.

I am indebted to all my friends and family who were always so helpful in numerous ways. Special thanks to Nupur Di, Vandana Di, Miral Di, Sarita Di, Sujata, Purvi, Nikita Patel, Hardik, Rishikant for their friendship and the warmth they ex-

tended during my doctoral studies. A special thanks to Nidhi (chai partner) who never looked at the clock to support me, her constant support encouraged me to continue my work despite all odds.

I would also like to say a heartfelt thank you to my mother, sisters, nieces, and in-laws for always believing in me and encouraging me to follow my dreams. And finally to Girish, who has been by my side throughout this Ph.D., living every single minute of it. Thanks for your undying love and support. And to dearest son, Aadhwan, you teach me a lot of things, thanks for being such a good baby who supports mummy in every possible way.

Contents

Abstract **xi**

List of Principal Symbols and Acronyms **xv**

List of Tables **xvi**

List of Figures **xix**

1 Introduction **1**

1.1 Motivation 4

1.2 Observations 5

1.3 Thesis Contribution 8

1.4 Organization of the Thesis 10

2 Related Work **12**

2.1 Data-Driven Approaches for Traffic Stream Variables Prediction . . 13

2.1.1 Parametric Approaches 14

2.1.2 Non-parametric Approaches 18

2.2 Weather Impact on Traffic Stream Variables 27

2.2.1 Traffic Prediction During Adverse Weather Conditions . . . 27

2.2.2 Study of Weather Impact on Traffic Stream Variables 30

2.3 Summary 32

3 Problem Description **34**

3.1 Problem Statement for Traffic Stream Variables Prediction 37

3.2 Problem Statement for Synthetic Traffic Data 41

3.3 Assumptions 42

4 System Model and Proposed Approach 43

4.1 System Model 43

4.2 Proposed Model 48

4.2.1 Input Data 48

4.2.2 Synthetic Data Generation Method 55

4.2.3 Impact of Weather Conditions on Traffic Stream Variables . 66

4.2.4 Data Pre-processing 71

5 Learning Models 73

5.1 Hyperparameters Tuning 74

5.1.1 Determine the Optimization Algorithm, Learning Rate, and Mini-Batch Function 75

5.1.2 Determine the Network Structure 79

5.1.3 Determine the Loss Function 79

5.1.4 Determine the Activation Function 80

5.2 Learning Models 82

5.2.1 Convolutional Neural Network (CNN) 82

5.2.2 Sequence Learning Models 86

5.2.3 Hybrid Deep Learning Models 91

6 Experimental Setup and Result Analysis 96

6.1 Deep Learning Model Performance with Inclusion of Rainfall Data 97

6.1.1 Data Description 97

6.1.2 Experimental Setup 98

6.1.3 Result Analysis 100

6.2 Synthetic Data Calibration and Validation 112

6.2.1 Discussion 117

6.3 Deep Learning Model Performance with Synthetic Data 118

6.3.1 Experimental Setup and Result Analysis for Large Road Network 121

6.3.2 Hybrid Models Comparison with Existing Models 131

6.3.3 Model Robustness 133

6.3.4 Learning Models Complexity Analysis 140
6.4 Discussion 143

7 Conclusion and Future Work 146
7.1 Conclusion 146
7.2 Future Work 151

8 List of Publications 153

References 154

Appendix A SUMO Sample Code 166

Appendix B Vanishing Gradient Problem and Solution 168

Appendix C Sample Code of Traffic Stream Variables Prediction Using Deep Learning Models 171
C.1 Sample RNN code 171
C.2 Sample CNN code 172
C.3 Sample LSTM code 172
C.4 Sample LSTM-LSTM Code 173
C.5 Sample CNN-LSTM code 174

Appendix D Experimental Setup and Result Analysis for Single and Small Road Network 176
D.1 Experimental Setup and Result Analysis for Single Road 176
D.1.1 Experimental Setup 176
D.1.2 Result Analysis for Single Road 179
D.2 Experimental Setup and Result Analysis for Small Road Network . 181
D.2.1 Experimental Setup 182
D.2.2 Result Analysis for Small Road Network 184

Appendix E Hyperparameters for the RNN, CNN, and LSTM Model for Large Road Network 191

Appendix F Traffic Stream Variables Prediction in Presence of Snowfall 193

Abstract

Accurate prediction of the macroscopic traffic stream variables such as speed and flow is essential for the traffic operation and management in an Intelligent Transportation System (ITS). Adverse weather conditions like fog, rainfall, and snowfall affect the driver's visibility, vehicle's mobility, and road capacity. Accurate traffic forecasting during inclement weather conditions is a non-linear and complex problem as it involves various hidden features such as time of the day, road characteristics, drainage quality, etc. With recent computational technologies and huge data availability, such a problem is solved using data-driven approaches. Traditional data-driven approaches used shallow architecture which ignores the hidden influencing factor and is proved to have limitations in a high dimensional traffic state. Deep learning models are proven to be more accurate for predicting traffic stream variables than shallow models because they extract the hidden features using the layerwise architecture.

The impact of weather conditions on traffic is dependent on various hidden features. The rainfall effect on traffic is not directly proportional to the distance between the weather stations and the road segment because of terrain feature constraints. The prolonged rainfall weakens the drainage system, affects soil absorption capability, which causes waterlogging. Therefore, to capture the spatial and prolonged impact of weather conditions, we proposed the soft spatial and temporal threshold mechanism. Another concern with weather data is the traffic data has a high spatial and temporal resolution compared to it. Therefore, missing weather data is difficult to ignore, the spatial interpolation techniques such as Theissen polygon, inverse distance weighted method, and linear regression methods are used to fill out the missing weather data.

The deep learning models require a large amount of data for accurate prediction. The ITS infrastructure provides dense and complete traffic data. The installation and maintenance of ITS infrastructures are costly; therefore, the majority of road segments are dependent on cost-effective alternate sources of traffic data. The alternate source of traffic data provides sparse, incomplete, and erroneous information. To overcome the data sparsity issue, we proposed a mechanism to generate fine-grained synthetic traffic data using the SUMO traffic simulator. We studied the impact of rainfall on the traffic stream variables on the arterial, sub-arterial, and collector roads. An empirical model is designed and calibrated for a variety of traffic and weather conditions. The Krauss car-following model in SUMO is upgraded to use the proposed empirical model for computing the vehicle speed. The simulation model is validated by comparing the synthetic data with the ground truth data under various traffic and weather conditions. We find that the empirical model accurately captures the effect of rainfall on the traffic stream variables, and the synthetic data shows a very good match with the ground truth data.

We adopted multiple deep learning models because of their underlying characteristics to extract the spatiotemporal features from the traffic and weather data. Convolutional Neural Network (CNN) model has the characteristics to extract neighboring pixels correlation. The sequence learning models, Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM) learn dependencies in the data based on the past and the current information. We designed the hybrid deep learning models, CNN-LSTM and LSTM-LSTM. The former model extracts the spatiotemporal features and the latter model uses these features as memory. The latter model predicts the traffic stream variables depending upon the memory and the temporal input. The hybrid models are effective in learning the long-term dependency between the traffic and weather data.

We performed various experiments to validate the deep learning models, we use the synthetic traffic data generated by SUMO using the empirical model for different road types (arterial, sub-arterial, and collector) and different road networks (single, small, and large). The results show that the deep learning model

trained with the traffic and rainfall data gives better prediction accuracy than the model trained without rainfall data. The performance of the LSTM-LSTM model is better than the other models in all the scenarios. Considering the large road network, where roads are prone to waterlogging, under long-term dependency LSTM-LSTM outperforms the other deep learning models including RNN, CNN, LSTM, CNN-LSTM, and existing models. For the worst-case scenario, the traffic prediction error of LSTM-LSTM is between 3-15% for 15 to 60-minute future time instances, which is in line with the accuracy needed for ITS applications.

List of Principal Symbols and Acronyms

α	The rainfall parameter
θ	Parameters
β	The soft spatial threshold weight factor
Δ	The threshold distance in km with respect to road segment for taking weather readings
δ	Traffic data aggregation period
$\delta^{'}$	Weather data aggregation period
γ	The soft temporal threshold weight factor
Ł	Loss function
μ	Friction coefficient
η	Learning rate
E	The number of epoch
L	The number of layer
M	The number of weather stations in a geographical area
N	The number of road segments in a geographical area
NPL	The nodes per layer
p	The length of the sequence (lag value)

P^j	The set of road segments other than i (target road segment) in a geographical area
P^k	The set of k weather stations
PH	The prediction horizon
q	The extended sequence length
R	Road type parameter
$R_{arterial}$	Simulated arterial road
$R_{sub-arterial}$	Simulated sub-arterial road
$R_{collector}$	Simulated collector road
S	Sample size
$S1$	Single road with point traffic and weather data
$S2$	Small road network
$S3$	Large road network
S_t	Spatiotemporal data
T_d	Time of the day
t_g	Prediction horizon, t + gδ
u_w	Traffic speed related to weather parameters
$\breve{W}_t^i$	The interpolated weather variable value
$\hat{W}_t$	The weighted average corresponds to soft spatial threshold
$\overline{W_t^k}$	The moving average weather variable value for station k at time t
W_t^k	The observed weather value during t^{th} time interval at k^{th} weather station
W_t	The spatiotemporal weather data

X_t^i	The traffic data during t^{th} time interval at i^{th} road segment.
X_t	The spatiotemporal traffic data
$Y_{t_g}^i$	The predicted traffic value at target road segment i for prediction horizon t_g
Adam	Adaptive Moment Estimation
AMTS	Automatic Traffic Management System
ANN	Artificial Neural Network
ARIMA	AutoRegressive Integrated Moving Average
ATIS	Automatic Travel Information System
BEP	Borivali East to Parel
BPNN	BackPropagation Neural Network
BPTT	BackPropagation Through Time
CNN	Convolutional Neural Network
DBN	Deep Belief Network
kNN	k Nearest Neighbor
LR	Linear regression
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MEP	Mulund East to Parel
MRSAC	Maharashtra Remote Sensing Application Centre
NCDC-NOAA	National Climate Data Center

PeMS	Performance Measurement System
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SAE	Stacked AutoEncoder
SUMO	Simulator for Urban Mobility
SVR	Support Vector Regression
TraCI	Traffic Control Interface

List of Tables

2.1 Parametric Approaches for Traffic Flow Prediction, PH stands for Prediction Horizon. 17

2.2 Non-parametric Approaches for Traffic Flow Prediction, PH stands for Prediction Horizon. 23

2.3 Comparison Between Data-Driven Approaches. 24

2.4 Deep Learning Approaches for Traffic Stream Variables Prediction, PH stands for Prediction Horizon. 26

2.5 Parametric and Non-parametric Approaches for Traffic Stream Variables Prediction During Adverse Weather Condition, PH stands for Prediction Horizon. 29

2.6 Reduction in Traffic Speed for Different Weather Condition as Observed in Literature. 32

4.1 List of Geometric Features of the Roads. 51

4.2 Types of Roads Considered in This Study. 56

4.3 Rainfall Impact on Traffic Movement. 56

4.4 Rainfall Category Based on Rainfall Intensity. 58

4.5 Percentage Decrease in Traffic Speed (km/hr) as per San Diego Data. 58

4.6 Percentage Decrease in Traffic Flow (vehicle-count / 5 min) as per San Diego Data. 59

4.7 Percentage Decrease in Traffic Speed (km/hr) as per Mumbai Data. 59

4.8 Comparison of Decrease in Traffic Speed Due to Rainfall Observed by the Existing Work and This Study. 60

4.9 Comparison of Decrease in Traffic Flow Due to Rainfall Observed by the Existing Work and This Study. 60

4.10 Road Surface Condition and Friction Coefficient. 61

4.11 Least Square Estimation of Speed Based on the Friction Parameter (μ), Rainfall Parameter (α), and Road Type (R), Where $\mu \in [0, 1]$ and $\alpha \in [0, 1]$. 63

6.1 Cases To Evaluate Model Performance. 97

6.2 Target Road Segment and Other Road Segments and Weather Stations id for the Experiment Corresponds to I-5 Road Segment. . . . 98

6.3 Hyperparameters for the LSTM Model for Different Roads. 99

6.4 Hyperparameters for the RNN Model for Different Roads. 100

6.5 MAPE Comparison Between Models Trained with Temporal Input vs Trained with Spatiotemporal Input, I-5 road segments. 110

6.6 MAPE Comparison Between Models Trained with Temporal Input vs Trained with Spatiotemporal Input, SR-75 Road Segments. . . . 111

6.7 MAPE Comparison Between Models Trained with Recent Rainfall Data vs Trained with Sequential Traffic and Rainfall Data, I-5 Road Segments. 112

6.8 Description of Simulated Road Types. 113

6.9 Simulated Vehicle Flow Calibrated from Study Data for Different Types of Roads under Various Traffic and Weather Conditions. . . . 114

6.10 Scenarios to Test Model Performance. 120

6.11 Hyperparameters for the CNN-LSTM Model. 122

6.12 Hyperparameters for the LSTM-LSTM Model. 123

6.13 Traffic Speed Prediction Error of Different Models when Trained with Rainfall Data for Road Type $R_{arterial}$. 124

6.14 Traffic Speed Prediction Error of Different Models when Trained with Rainfall Data for Road Type $R_{sub-arterial}$. 127

6.15 Traffic Speed Prediction Error of Different Models when Trained with Rainfall Data for Road Type $R_{collector}$. 129

6.16 Training Time and Number of Parameters in Models. 143

6.17 Prediction Performance Comparison of LSTM-LSTM model and Existing Models. 144

D.1 Hyperparameters for the RNN Model for Single Road Network. . . 177
D.2 Hyperparameters for the CNN Model for Single Road Network. . . 177
D.3 Hyperparameters for the LSTM Model for Single Road Network. . 177
D.4 Hyperparameters for the CNN-LSTM Model for Single Road Network. 178
D.5 Hyperparameters for the LSTM-LSTM Model for Single Road Network. 178
D.6 Hyperparameters for the RNN Model for Small Road Network. . . 182
D.7 Hyperparameters for the CNN Model for Small Road Network. . . 182
D.8 Hyperparameters for the LSTM Model for Small Road Network. . . 183
D.9 Hyperparameters for the CNN-LSTM Model for Small Road Network. 183
D.10 Hyperparameters for the LSTM-LSTM Model for Small Road Network. 184

E.1 Hyperparameters for the RNN Model for Large Road Network Scenario. 191
E.2 Hyperparameters for the CNN Model for Large Road Network Scenario. 192
E.3 Hyperparameters for the LSTM Model for Large Road Network Scenario. 192

List of Figures

2.1 Classification of Data-driven Approaches. 13

3.1 System Representation. 35

3.2 Different Sources of Input Data. 35

3.3 System Representation of the Input Data. 38

3.4 System Representation of the Learning Model. 40

4.1 System Model. 44

4.2 Selected Road Stretches of San Diego. 50

4.3 Selected Road Stretches of Mumbai. 53

4.4 Monthly Rainfall in San Diego (Year 2019) 54

4.5 Monthly Rainfall in Mumbai (Year 2017). 55

4.6 Correlation Coefficient of Weather Data versus Distance of Weather Stations. 57

5.1 Framework for Traffic Stream Variables Prediction. 74

5.2 Schematic Diagram of CNN Based Traffic Stream Variables Prediction Model. 84

5.3 An Unrolled Recurrent Neural Network. 87

5.4 Schematic Diagram of RNN Based Traffic Stream Variables Prediction Model. 87

5.5 RNN Cell Architecture. 88

5.6 LSTM Cell Architecture. 90

5.7 CNN-LSTM Model for Traffic Variables Prediction. 92

5.8 LSTM-LSTM Model for Traffic Variables Prediction. 94

6.1 Traffic Flow and Speed During Normal Weather Day Corresponding to I-5 Road, Vertical Lines Signifies the Peak Hour. 100

6.2 Rainfall Data from Weather Station in San Diego. 101

6.3 Traffic Speed as a Function of Hourly Precipitation and Time of the Day. 102

6.4 15-minute and 60-minute Traffic Speed Prediction Performance of LSTM With and Without Rainfall Variable. Model Performance is Tested for **Heavy Rainfall** Day. 103

6.5 15-minute and 60-minute Traffic Speed Prediction Performance of LSTM With and Without Rainfall Variable. Model Performance is Tested for **Moderate Rainfall** Day. 103

6.6 15-minute and 60-minute Traffic Speed Prediction Performance of LSTM With and Without Rainfall Variable. Model Performance is Tested for **Light to No Rainfall** Day. 104

6.7 Traffic Speed Prediction by RNN and LSTM Model Trained With and Without Rain Input and Comparing MAPE Corresponding to I-5, I-5 Downtown, and SR-75 Roads. 105

6.8 Traffic Flow Prediction by RNN and LSTM Model Trained With and Without Rain Input and Comparing MAPE Corresponding to I-5, I-5 Downtown, and SR-75 Roads. 107

6.9 Traffic Flow and Speed Prediction Error Comparison of SAE, BPNN, DBNN, RNN and LSTM for Different Prediction Horizon for Traffic Data Along I-5. Error Bar Shows the Standard Deviation. 108

6.10 Percentage Decrease in Traffic Speed Due to Rainfall on $R_{arterial}$-type Road, Error Bar Reflects the Standard Deviation. 115

6.11 Percentage Decrease in Traffic Speed Due to Rainfall on $R_{sub-arterial}$-type Road, Error Bar Reflects the Standard Deviation. 116

6.12 Percentage Decrease in Traffic Speed Due to Rainfall on $R_{collector}$-type Road, Error Bar Reflects the Standard Deviation. 117

6.13 Small Road Network. 119

6.14 Large Road Network. 119

6.15 Traffic Speed Prediction Performance of Models without and with Rain Data for Simulated $R_{arterial}$ Road. Error Bar Shows the Standard Deviation. 124

6.16 Traffic Flow Prediction Performance of Models without and with Rain Data for Simulated $R_{arterial}$ Road. Error Bar Shows the Standard Deviation. 124

6.17 Traffic Speed Prediction Performance of Models without and with Rain Data for Simulated $R_{sub-arterial}$ Road. Error Bar Shows the Standard Deviation. 127

6.18 Traffic Flow Prediction Performance of Models without and with Rain Data for Simulated $R_{sub-arterial}$ Road. Error Bar Shows the Standard Deviation. 128

6.19 Traffic Speed Prediction Performance of Models without and with Rain Data for Simulated $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 129

6.20 Traffic Flow Prediction Performance of Models without and with Rain Data for Simulated $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 130

6.21 Traffic Flow MAPE Comparison of Hybrid and Existing Models. Error Bar Shows the Standard Deviation. 132

6.22 Traffic Speed MAPE Comparison for Hard and Soft Temporal Threshold for the 60-minute Prediction Horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 134

6.23 Traffic Speed MAPE Comparison for Different Value of Weighing factor (γ), for the 30-minute Prediction Horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 134

6.24 Traffic Speed Prediction Performance of the Models for Soft Spatial Threshold. Error bar Shows the Standard Deviation. 136

6.25 Traffic Speed Prediction Performance of the Models with Complete Data and with Spatial Interpolation Method on Missing Data. Error Bar Shows the Standard Deviation. 137

6.26 Traffic Speed Prediction Performance of the Models with Different Interpolation Techniques for the 30-minute Prediction Horizon on $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 137

6.27 Traffic Speed Prediction Performance of the Models Considering Additional Data (one day average rainfall value). Error Bar Shows the Standard Deviation. 138

6.28 Traffic Flow Prediction Performance of the Models Considering Additional Data (one day average rainfall value). Error Bar Shows the Standard Deviation. 139

6.29 Traffic Flow Prediction Performance of the Models Considering Additional Data (two day average rainfall value). Error Bar Shows the Standard Deviation. 139

6.30 Traffic Speed Prediction Performance of the Models for Different Lag Values, 30-minute prediction horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 141

6.31 Traffic Speed Prediction Performance of the Models for Different Epoch Values, 30-minute prediction horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation. 141

6.32 The Number of Epoch Required for the Particular Lag Value. 142

6.33 Training Time Required for the Each Lag Value. 142

D.1 MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for $R_{arterial}$ Road Type. Error Bar Shows the Standard Deviation. 179

D.2 MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for $R_{sub-arterial}$ Road Type. Error Bar Shows the Standard Deviation. 180

D.3 MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for $R_{collector}$ Road Type. Error Bar Shows the Standard Deviation. 180

D.4 MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{arterial}$ Road Type. Error Bar Shows the Standard Deviation. 185

D.5 MAPE (top) and MAE (bottom) of Traffic Flow Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{arterial}$ Road Type. Error Bar Shows the Standard Deviation. 186

D.6 MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{sub-arterial}$ Road Type. Error Bar Shows the Standard Deviation. 187

D.7 MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{collector}$ Road Type. Error Bar Shows the Standard Deviation. 187

D.8 MAPE (top) and MAE (bottom) of Traffic Flow Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{sub-arterial}$ Road Type. Error Bar Shows the Standard Deviation. 189

D.9 MAPE (top) and MAE (bottom) of Traffic Flow Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{collector}$ Road Type. Error Bar Shows the Standard Deviation. 189

F.1 Traffic Flow Prediction Performance of the Model Trained with and without Snowfall Data Corresponding to TH10 in Minnesota. . . . 193

CHAPTER 1

Introduction

The Intelligent Transportation System (ITS) technologies compute the information about the current and the future traffic conditions [1]. ITS applies Information and Communication Technology (ICT) for data collection and processing, advanced information retrieval techniques, and advanced computation technologies for the operation and management of the transportation network. There are a number of ITS applications such as Advanced Traffic Management Systems (ATMS) [2], Advanced Traveller Information Systems (ATIS) [3], etc. The traveler information applications such as ATIS inform the drivers about the expected travel time, expected congestion, and suggest the shortest and fastest route. The traffic management applications such as ATMS ensure optimal efficiency of the transportation network and solve the problem of traffic congestion. Short-term traffic prediction is defined as the process of estimating the traffic conditions in the short-term future given the historical and the current traffic information [4]. The accurate short-term traffic prediction helps traffic management authorities to develop more sophisticated strategies to anticipate and mitigate network-related problems such as traffic congestion. Similarly, an individual traveler gets to benefit from the accurate short-term traffic prediction, by using this predictive information to plan their journey by choosing the most efficient transport option (route, time of the day) to avoid traffic congestion and optimize travel time.

In this thesis, the short-term refers to a prediction horizon of up to one hour which is the optimal time for individual navigation and global urban traffic planning, and is in agreement with literature studies [5, 6, 7]. The short-term traffic prediction focuses on estimating the value of traffic stream variables such as traffic flow and speed on the road segment. The road segment is the portion of a road with uniform characteristics such that the traffic stream variables remain constant on it. Traffic flow is the number of vehicles passing the road segment per unit time. Traffic speed is the average of all vehicle's speeds on the road segment at a particular time.

The traffic stream variables are measured from sensors installed on the road segment such as inductive loops, radar, or the devices mounted on traffic signals such as the camera. These devices are installed and maintained by designated services and come under ITS infrastructure. Other cost-effective alternate sources of traffic data are taxi GPS, cellular data, etc. Prediction of traffic stream variables is a challenging problem. These variables are affected by several factors like time of the day, road condition, non-recurrent events such as accidents, road construction, weather conditions, social gatherings, etc. Therefore traffic stream characteristic is non-linear and complex. To predict the traffic stream variables two types of approaches are used: Model-driven and Data-driven. The model-driven approach simulates the traffic system and explores the vehicle's performance and behavior in the road network.

The data-driven approaches learn the features from examples (data), and these methods need a considerable amount of data to discover patterns. With intelligent computational technologies and massive data availability, traffic stream variables prediction is addressed frequently using data-driven approaches. The data-driven approaches are categorized as: parametric and non-parametric. Parametric methods summarize the data with a fixed set of parameters independent of the amount of training data, assuming that the collected data follows a similar distribution. These approaches assume that the data is stationary, i.e., mean and variance remain unchanged. The time-series based methods such as AutoRegressive Integrated Moving Average (ARIMA) [8, 9, 10], and the Kalman filter [11, 12] based methods come under the parametric category for short-term traffic stream variables prediction. Due to the traffic stream's non-linear and complex nature, linear methods do not provide an accurate prediction. The non-parametric approaches use a flexible number of parameters, which increases as the model starts learning from the data. The collected data need not follow a similar distribution. Machine learning, artificial neural network, and deep learning methods fall under this category. There are methods under machine learning and artificial neural network category like k Nearest Neighbor (kNN) [13], Bayesian network [14], Support Vector Regression (SVR) [15] and, Artificial Neural Network (ANN) [16] for traffic stream variables prediction, and their results show that non-parametric models provide more accurate forecast than the parametric models.

Traffic stream variables depend upon various hidden features. Both parametric and non-parametric (machine learning and artificial neural network) approaches ignore the hidden influencing factors because of the shallow architectures. For complex learning problems, deep architecture proves to have an advantage over shallow architecture [16]. The multiple linear and non-linear units stacked in layerwise fashion provide the ability to learn features at a different level of abstraction. The deep learning algorithm uses the multi-layer architecture to extract the inherent features in data from a lower level to a higher level [16]. The deep learning method has been used to solve the traffic stream variables prediction problem. The traffic variable's future value depends on the current and historical traffic data. Therefore, the traffic stream variables prediction is considered as a sequence learning problem, where the future outcome is based on the past sequence. The deep learning model called a Recurrent Neural Network (RNN) [17], and it's variant, Long Short Term Memory (LSTM) [18] is designed for the sequence learning problem. The RNN uses memory cells to capture the temporal information from the previous sequence and uses this information along with the current input for future prediction. The LSTM uses a gating mechanism to extract longer temporal dependency. Zhao *et al.* [19] used LSTM for the short-term traffic stream variables prediction. The results show that the LSTM model's accuracy is higher than the shallow machine learning model and non-recurrent neural networks.

The traffic condition on the adjacent road will affect the traffic condition on the target road in the near future. Therefore, traffic stream variables have a spatiotemporal correlation. The Convolutional Neural Network (CNN) has shown a powerful ability to extract the neighboring pixel correlation. Zhang *et al.* [20] used the CNN model for traffic speed prediction considering the spatiotemporal dependency. Their results show the improvement over machine learning and parametric approaches.

1.1 Motivation

Adverse weather conditions like fog reduces visibility, rainfall affects the vehicle's mobility, and snowfall affects the road capacity. During inclement weather conditions, some trips will be postponed or canceled. The traffic stream variables get affected due to the rainfall because of poor visibility, waterlogging, and increased aquaplaning. After hefty snowfall, the thick sheet of snow decreases the road capacity, affecting the traffic stream variables. In developing countries like India, where the cities are densely populated, roads are narrow, and have weak drainage systems, the impact of prolonged heavy rainfall (greater than 100 mm in 6 hours) is significant on the traffic conditions. The commute time between two points increases because of road closure due to accidents, tree falls, or massive waterlogging. Ibrahim *et al.* [21] observed that rainfall intensity values 0-0.01 in/hr, 0.01-0.25 in/hr and $>$ 0.25 in/hr causes speed reduction of 2-4%, 5-8% and 9-14%, respectively. M. Agarwal *et al.* [22] observed that light to heavy rainfall causes 2-6% speed reduction and 2-14% capacity reduction on the freeway roads system. Kyte *et al.* [23] observed the reduction in operating speed due to different weather condition. The wet surface causes 9.5 km/hr reduction, snowfall causes 16.4 km/hr reduction and wind $>$24 km/hr causes 11.7 km/hr reduction in operating speed. Therefore, traffic stream variables prediction during adverse weather conditions needs attention. Deep learning models are complex and can extract the hidden features from the traffic data. We would like to examine, whether the deep learning model traffic prediction accuracy improves with the inclusion of weather data.

The deep learning models require dense and complete traffic data for accurate prediction. The ITS infrastructure consists of traffic sensors (e.g. loop detectors, traffic cameras, etc.) that provide dense and complete traffic data. The ITS requires infrastructure for communicating raw data and aggregated traffic information, computation infrastructure for processing raw data from individual traffic sensors to generate the aggregated traffic information. The ITS Report of U.S. Department of Transportation [24] describes the deployment and maintenance cost of the ITS infrastructure for various projects. According to the report, Caltrans Performance Measurement System (PeMS) collects traffic data in real-time from nearly 40,000 individual detectors spanning the freeway system across all the major metropolitan areas of California. PeMS is also an Archived Data User Service (ADUS) that provides over ten years of data for historical analysis. The system costs about $3.5 million a year considering the deployment and maintenance cost of hardware and software (in 2013). Due to the high cost of sensor deployment and maintenance the alternative source of traffic data such as GPS, cellular data are explored. But these cost-effective data provide sparse, erroneous, and incomplete information. To provide dense traffic data to the deep learning models for accurate prediction, fine-grained synthetic data is needed which approximates the measured data. The synthetic traffic data is given to the deep learning model to predict the traffic stream variables during adverse weather conditions. Various stakeholders can use the results of such models. The urban development authorities can identify the road segments and drainage systems that need improvement. The traffic engineers can use this data to compare various prediction models to mitigate congestion and provide travel time estimation. The general users can better plan their trips, such as postpone trips or change routes to avoid waterlogged sites.

1.2 Observations

In literature [25, 26], the traffic stream variables prediction is performed during inclement weather conditions. The weather data is interpolated from the nearest station to the target road segment. The temporal traffic and weather sequence are considered in both studies. The past and current traffic and weather data corresponds to the target road segment is used for traffic forecasting. A. Koesdwiady *et al.* [27] predicts the traffic stream variables during adverse weather condition. The spatiotemporal traffic and weather data corresponding to adjacent road segments are considered in this study. As per their assumption, rainfall data is complementary, therefore with the past and recent traffic data only the most recent rainfall data is needed for the traffic prediction.

There are few gaps in the literature studies. The traffic disruption during rainfall is not directly related to the rainfall value because of the terrain feature constraint. The two road segments are distant apart but due to terrain features, waterlogging at one road segment affects the other road segment in the near future. Therefore, data related to waterlogging, water flow path, soil absorption capability, drainage capacity, and other hidden parameters should be taken into account. Drainage capacity is a function of time if maintenance is not proper. If the incoming water rate is more than the drainage capacity, waterlogging will happen. Soil and surface act as a sink, and due to prolonged rainfall, total outlet capacity reduces. Baghel A., in a technical report, studied the causes of urban floods in Mumbai and Chennai city of India [28]. As per the report, the urban flood is not because of road design but because of the weak drainage system. These parameters are not static that can be used directly from the city soil and drainage dataset to predict the traffic stream variables. These parameters are a function of history in terms of soil absorption capability and drainage maintenance. For many cities and roads, these datasets are not available. To accurately predict the traffic stream variables, consideration of these features is important. Similarly, there is a need for more extended historical information about weather data than traffic data to capture the prolonged weather effect. The impact of weather conditions such as rainfall is varied as per road types. The broad roads with excellent drainage systems are less prone to waterlogging compared to the narrow roads with the poor drainage system. The roads passing through the densely populated areas are more affected due to the rainfall compared to the freeway roads. All these factors make the traffic stream variables prediction during adverse weather conditions a challenging problem.

The future traffic condition on the target road depends on its previous traffic conditions and the road's hidden characteristics, such as point of interest, peak hour time, etc. Therefore, the traffic condition on the target road is affected by the surrounding traffic and weather conditions and its own traffic condition. For example, the target road gets affected due to the water flow from the adjacent roads and if the target road is narrow and prone to waterlogging then the situation gets worse. To predict the traffic stream variables on the target road segment, there is a need to extract the spatiotemporal and temporal dependency. Therefore, a deep learning model that extracts the hidden features considering spatiotemporal and temporal dependencies is the favorable choice for traffic forecasting during adverse weather conditions.

In literature, the impact of rainfall on traffic stream variables is mainly studied for the expressway roads of developed countries where the road infrastructure and drainage systems are excellent. The analyses are performed on the city-specific traffic and weather data obtained from the traffic sensors and weather stations. In developing countries, roads are relatively narrow and surrounded by dense infrastructure. Further, the drainage capacity is weak and impermeable. The heavy rainfall under such conditions causes waterlogging. As per the Uber movement data [29], heavy rain ($\geq$ 100 mm in 6 hours) in Mumbai, India, reduces the traffic speed by 60% and increases the travel time by 75%. Aiding to the problem, limited traffic monitoring infrastructure is available in the majority of the location due to the high maintenance and deployment cost. The majority of the road segments depend on the alternate source of traffic data for the traffic-related information. The traffic data generated by alternative sources are sparse and erroneous. Also, the traffic data from ITS infrastructure in the majority of the locations are private and not available as open source. Therefore, comparing various predictive models on a similar data-set is not possible. Hence, it is necessary to generate synthetic data that provides an abstraction of real-world traffic conditions and solves the above limitations.

1.3 Thesis Contribution

Traffic stream variables prediction during adverse weather conditions is a non-linear and complex problem that depends upon various hidden features. The suitable deep learning model should extract these hidden features from the traffic and the weather data. We used multiple deep learning models for traffic forecasting during adverse weather conditions. CNN is the suitable model, known for finding the neighboring pixel correlation [30]. In the transportation domain, the traffic condition on the target road is significantly affected by the traffic and the weather condition in the surrounding areas. Also, the past and the current traffic and weather condition affects the future traffic condition on the road segment. The recurrent neural network such as RNN and LSTM is suitable for extracting the spatiotemporal dependence between the target and surrounding road's traffic and weather data for past and current time instances. The traffic condition on the target road has spatiotemporal dependence with the traffic and weather data in surrounding areas and temporal dependence on traffic data of its own. We design two hybrid models, CNN-LSTM and LSTM-LSTM. The former model extracts the spatiotemporal correlation, whereas, the latter model uses this spatiotemporal feature as memory, and based on this memory and temporal data the model predicts the future traffic condition.

Deep learning models are known for extracting the hidden influential features from the input data. These models are complex and require a large amount of data for extracting hidden features. Considering the complexity of the deep learning model, our objective is to examine whether the traffic prediction accuracy of the deep learning models improves with the inclusion of weather data during adverse weather conditions.

To overcome the traffic data sparsity issue, we proposed a mechanism to generate fine-grained synthetic traffic data. We study the impact of rainfall on traffic stream variables on different types of roads such as freeways, city roads, etc. We generate the synthetic traffic data for different rainfall intensities and a variety of roads using a realistic simulation setup. We use the Simulator for Urban MObility

(SUMO) [31] for simulations, which is a widely-used open source traffic simulator. The car-following model is used to derive vehicle speed, such that the distance between the following and leading vehicle should be maintained. A major limitation of SUMO is that its car-following models do not use the weather parameters for computing vehicle speed and other mobility-related parameters. Hence, additional work is required to incorporate the weather parameters in the SUMO road networks that influence the traffic stream variables. To deal with the SUMO's shortcoming, we design an empirical model that quantifies the effect of rainfall on different types of roads under diverse weather conditions. These quantified values are associated with the road segment (called edge in SUMO) and made available to the car-following model. The car-following model is adapted to use these additional parameters while computing vehicle speed and other mobility-related parameters.

The weather data have various hidden dependencies. It has spatial and temporal significance. The weather conditions on the distant road may affect the traffic on the target road at a future time instance. The impact of prolonged rainfall is significant on traffic stream variables. In an ideal situation, we can have data corresponding to each feature such as hydrology map, drainage data, soil data that would be useful for extracting the correlation. Also if there is no computation limitation, weather data from all the stations in the city for extended timestamps are passed to the model. But the data-set is not available for the majority of the location and passing data from all stations and for extended timestamps increases the number of parameters. We proposed the spatial and temporal soft threshold mechanism that captures the weather condition's spatial and prolonged impact. To consider the impact of weather conditions on other road segments, we need to take into account the weather data from other weather stations while predicting the traffic stream variables at the target road segment. We apply the soft spatial threshold to limit the number of weather stations to consider for a target road segment. For a particular geographical location, the weather data corresponds to all weather stations up to threshold distance is taken into account. For all the remaining weather stations in that geographical location, a distance-based weighted average is used to incorporate their weather readings. The prolonged impact of weather conditions is captured by defining a soft temporal threshold. The soft temporal threshold captures the impact of prolonged rainfall and at the same time limits the length of the weather data sequence. The weather sequence up to a particular threshold value and for the extended sequence the weighted moving average of the past weather reading is considered.

The traffic data is collected from the sensors located on the road segment, weather data is interpolated to the road segment based on the distance (nearest weather station). The collection frequency of the traffic and weather data is different. The traffic data has high spatial and temporal resolution compared to the weather data. Therefore missing weather data can't be ignored. We used spatial interpolation techniques such as the Thiessen polygon method, inverse distance weighted method, and linear regression methods to fill out the missing values.

To examine whether deep learning model traffic prediction accuracy improves with the inclusion of weather data during adverse weather conditions, we perform the experiment to forecast the traffic on the road systems of San Diego.

For synthetic data validation, a reference scenario is created and calibrated using the real data, to verify the implemented empirical model's effectiveness. The simulator's synthetic data is compared with the real data to validate the proposed empirical model. We generate the synthetic data for different types of roads (freeway, city roads) and road networks (small and large) for different rainfall intensities. We examine the deep learning models (RNN, CNN, LSTM, CNN-LSTM, LSTM-LSTM) performance on generated synthetic data. The models are compared based on their prediction accuracy and complexities. The models are compared with the models proposed in the literature study to examine the best deep learning model for traffic prediction under inclined weather conditions.

1.4 Organization of the Thesis

Chapter 2 surveys proposals for the impact of weather conditions on the traffic stream variables. Traffic stream variables prediction methods are discussed and compared. Detail review of the traffic stream variables prediction during adverse weather conditions is discussed. The chapter is concluded with remarks on the limitation of the related work and contributions of the thesis. Chapter 3 defines the problem related to traffic forecasting and synthetic data generation along with the assumptions. Chapter 4 describes the concerns related to the input traffic and weather data and the detailed mechanism to solve those concerns. Chapter 5 describes the adopted deep learning models and algorithm with the hyperparameters requires for optimal learning. The proposed hybrid models are discussed in detail. Chapter 6 studies the performance of deep learning models under various experimental setups. We examine whether the weather variable improves the prediction accuracy of the deep learning model. The generated synthetic data is compared with the ground truth data. The proposed hybrid model's performance on synthetic data is examined for various scenarios. The model complexity in terms of space and time is discussed in detail. Finally, the conclusion and future work are written in chapter 7.

Chapter 2

Related Work

Adverse weather conditions such as rainfall, snowfall, fog, etc., can significantly affect the travel demand, driving behavior, and traffic flow characteristics. Advances in sensor technologies make it possible to collect real-time traffic data under various traffic conditions, including adverse weather conditions. Automatic Travel Information System (ATIS) applications such as route guidance system and Automatic Traffic Management System (AMTS) applications such as traffic light control system at junction require accurate traffic stream variables prediction of the road segment for their efficient working. The purpose of the ATMS/ATIS is to improve the overall traffic system performance, e.g., congestion control, reducing emissions, noise, and travel times. The recent advancements in computation technologies and traffic data wealth have rendered promising opportunities to the data-driven forecasting approaches. In data-driven forecasting approaches, models are developed using data from the real system, and this data is used to fit the mathematical structures. We discuss the related work in data-driven approaches for traffic stream variables prediction in section 2.1 and the impact of weather conditions on traffic stream variables in section 2.2.

2.1 Data-Driven Approaches for Traffic Stream Variables Prediction

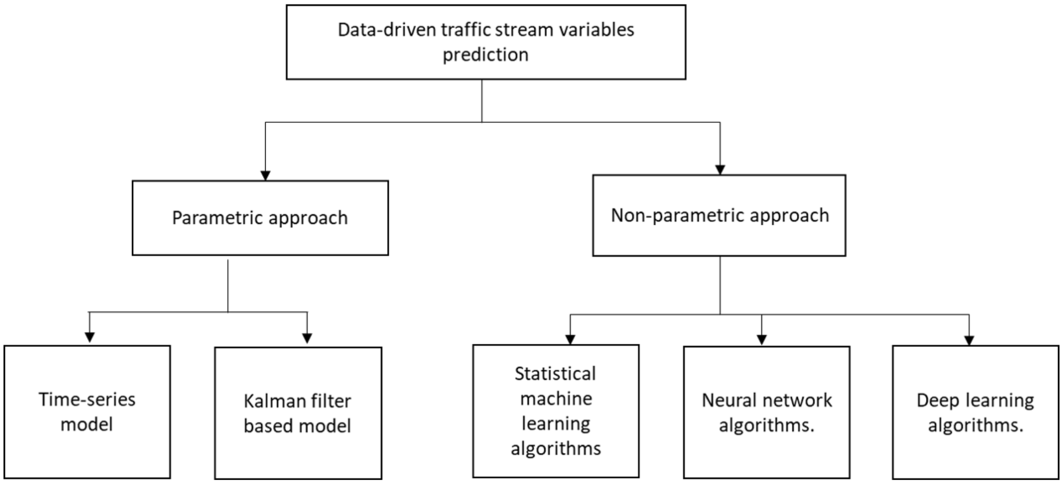

Data-driven traffic stream variables prediction
Parametric approach
Non-parametric approach
Time-series model
Kalman filter based model
Statistical machine learning algorithms
Neural network algorithms.
Deep learning algorithms.

Figure 2.1: Classification of Data-driven Approaches.

Figure 2.1 shows the classification of the data-driven approaches. The data-driven approach is classified into two types parametric and non-parametric.

Parametric methods summarize the data with a fixed set of parameters (θ). Given the parameters, the traffic stream variable's future prediction is independent of the observed data, assuming θ captures every information about the data. So the complexity of the model is bounded even if the amount of data is unbounded [32]. The parametric models use significantly less training data than the other data-driven approaches because the data's complexity is bounded.

These approaches assume that data is stationary, i.e., the mean and the variance remain unchanged. These approaches are classified into two categories: the time-series model and the Kalman filter-based model. The time-series model utilizes the historical and real-time traffic data to predict the future traffic stream variables. The Kalman Filter is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements [33].

In non-parametric approaches, the number of the parameters is flexible and not fixed in advance. The non-parametric techniques are further classified into three categories: statistical machine learning, neural network, and deep learning.

Statistical machine learning learns the patterns in the data and applies the learning for the unseen data. The neural network is an architecture consisting of the input layer, the hidden layer, and the output layer, with connections between each layer's nodes. Information is transformed from the input layer to the output layer through linear and nonlinear transformations. Deep learning is a neural network composed of multiple hidden layers. This multi-layer architecture helps to extract inherent features in data from the lowest level to the highest level. Therefore, a huge amount of structure in the data can be discovered. In traffic stream variables prediction, the data-driven model predicts the output at some future time instance. The length of time into the future for which predictions are to be made is called a prediction horizon. In data-driven approaches, the model's accuracy is reported in terms of the prediction error. The prediction error is defined as the difference between the actual outcome and the predicted outcome.

2.1.1 Parametric Approaches

Parametric models are based on a finite set of known parameters θ about the modelled population [32]. Given the parameters, future predictions y are independent of the observed data, D:

$$P(y|\theta) = P(y|\theta, D)$$

Depending on the parametric model used, the knowledge of the parameters can be utilized in these models, which could help understand the different behaviors of the traffic stream variables.

The parametric techniques include the time-series and the Kalman filter-based techniques for traffic stream variables prediction.

The time-series model utilizes the historical and the real-time traffic data to predict the traffic stream variables at the next time instance. The basic time-series model assumes that future value depends upon the past observations with added random noise [34]. In AutoRegressive Moving Average (ARMA) model, AutoRegressive (AR) technique predicts the future value of the variable using a linear combination of the past value of the variable, while Moving Average (MA) technique uses the past prediction error in a regression model. Since most of the time-series have non-stationary characteristics, the ARMA model is generalized to handle non-stationarity by applying differentiation (computing the difference between consecutive operations). This generalization of the ARMA model is called AutoRegressive Integrated Moving Average (ARIMA) . M. Hamid *et al.* [8] uses the ARIMA model for traffic flow prediction, applied the Box-Jenkins method for model identification. This method selects the appropriate model (whether to choose AR or MA component), based on the ARIMA model's stationarity and parameters. The traffic data is collected across the urban arterials in Amman, Jordan. The traffic data collection frequency is 1 minute. The traffic flow prediction error is 10.6 vehicle-count/minute on an absolute scale for a 1-minute prediction horizon. A heuristic approach with the ARIMA model is introduced to optimize the ARIMA traffic flow model [35]. In a Box-Jenkins method, the coefficient vector defines the number of past values used by the model to predict the next value. For example, AR (3) model uses the last three past values to predict the next value. T. Thomas *et al.* [35] used subset ARIMA, where coefficient vector defines the subset of past values to be used for the next value prediction. For example, the AR (1, 3) model uses only the last and third last values of the time-series for prediction. As per their observation, compared to the full ARIMA model, the subset ARIMA model gave more stable and accurate results. The traffic data is collected across the cross-section of the Netherlands. The traffic data collection frequency is 5-minute. Their results show a 5% traffic flow prediction error for a 30-minute prediction horizon.

In time-series data, the presence of variations that repeats every certain period is called seasonality, which is dealt with using the SARIMA (Seasonal ARIMA) model for traffic flow prediction [9]. The parameters of the SARIMA models are estimated using the least square estimation method. The experiment is performed on the traffic data collected from the Virginia Department of Transportation. It is observed that for less than 5% cases, traffic flow prediction error is greater than 20%, and the average traffic flow prediction error is less than 7%.

Overall, SARIMA gives a 2% error improvement over the ARIMA model for a 15-minute prediction horizon. The linear regression, historical average, ARIMA, and SARIMA are compared by Chung and Rosalion [36], the author concludes that during normal traffic conditions SARIMA outperforms the other algorithms. Still, these methods do not respond well to external system changes such as road accidents.

The Kalman filtering method is a traffic forecasting parametric model and can cast the regression problem in state space by minimizing the variance to obtain the optimal solution [33]. Kalman filter-based traffic volume prediction is introduced by Iwao and Yorgos [11]. The time-series method ignores the dependency of the adjacent road segment's traffic conditions on the target road segment. As per the author, the prediction parameters are improved using the most recent prediction error, and better volume prediction on a road segment is achieved by taking into account the data from the adjacent road segments. The experiment is performed on the freeways of Nagoya city, and it is observed that the average traffic flow prediction error is less than 9% for the 15-minute prediction horizon. In the AR model, the simplifying linearity hypothesis reduces the prediction accuracy in traffic stream prediction because traffic stream characteristics are non-linear and complex. To solve the linearity assumption of linear time-series methods, N. Barimani *et al.* [12] proposed the extended Kalman filter, which solves the non-linear regression in time-series problems by transforming the data into a high dimensional reproducing kernel Hilbert space. For the 15-minute prediction horizon, the average traffic flow prediction error of 3.76% is observed on Minnesota's freeways road segment. Table 2.1 summarizes the works on parametric approaches for traffic stream variables prediction. The summary shows the studied data, the method used for traffic flow prediction, and the analysis results.

Table 2.1: Parametric Approaches for Traffic Flow Prediction, PH stands for Prediction Horizon.

Research Work	Data Collection	Collection Frequency	Method	PH	Accuracy
Hamed *et al.* [8]	Urban arterials in Amman, Jordan's capital	1-min	ARIMA	1-min	Average absolute error (veh/min) 10.6
Thomas *et al.* [35]	5 km cross-section of the Netherlands	5-min	ARIMA with heuristic	30-min	5% error is reported
BILLY *et al.* [9]	Virginia Department of Transportation's	5-min	Seasonal ARIMA	15-min	$\leq$5% cases with error $\geq$ 20% and average traffic flow prediction error is $\leq$ 7%. 2% error improvement over ARIMA
Iwao and Yorgos [11]	Nagoya City	5-min	Kalman Filter	15-min	the average prediction error is $\leq$ 9%
Nasim *et al.* [12]	Minnesota	5-min	Extended Kalman Filter	15-min	For 15-min prediction horizon average error is 3.76%.

In a traffic network, it is common that many variables are highly correlated. The linear model assumes that the explanatory variables are statistically independent. Besides, roads are inter-connected, and traffic stream characteristic is nonlinear; hence it is difficult for linear models to capture the stochastic characteristics. These techniques only consider the temporal variation of traffic stream variables. It is necessary to predict traffic stream variables from a network perspective because the transportation networks are complex and highly correlated.

2.1.2 Non-parametric Approaches

A non-parametric method is a mathematical inference method that does not rely on assumptions that the data are drawn from a given probability distribution. Non-parametric models assume that the data distribution cannot be defined based on a finite set of parameters, but they can be defined by assuming infinite-dimensional parameters. The number of parameters is determined from the data. Therefore, these models require more data compared to the parametric models. The amount of information that parameter θ can capture about the data can grow as the amount of data grows. The major advantage of the non-parametric model over the parametric model is that these models can handle non-linear and complex traffic stream characteristics and can also extract the spatial and temporal dependency between the traffic stream variables. The non-parametric models are robust in dealing with situations like missing variables and outliers in the input data. The drawback of a non-parametric model is that the model requires a huge amount of data, and the model training and learning is a computationally intense task. The non-parametric models are a black box; each independent variable's cause and effect are not known. It is not possible to determine the mutual interrelation between the variables. The non-parametric models are classified into the statistical machine learning model, the neural network model, and the deep learning model.

The k Nearest Neighbor (kNN) is a non-parametric method used for classification and regression. It searches for an entry in the training data that are similar to the test data. These found entries are called the nearest neighbors of the test data [37]. The distance functions (Euclidean, Manhattan, Minkowski) are used to find the similarity measure [38]. Davis and Nihan [13] used the kNN method for the short-term freeway traffic forecasting. Their result analysis shows that the kNN performance is comparable with but not better than the parametric approaches. The traffic flow root mean square error for the 1-minute prediction horizon of the kNN model is 13.8 vehicle-count per minute, whereas the ARIMA model is 13.0 vehicle-count per minute. Chang *et al.* [39] used kNN to consider the spatiotemporal dependency for traffic forecasting. The model's input parameters include the k value for the nearest neighbor and the d value for the sequence length (past data to consider). This model performed effectively under shockwaves. Shockwaves are described as transitioning from one traffic state to another, free flow to stopped or stopped back to flowing traffic. Their results show the traffic flow prediction error improvement of 9.3% over ARIMA for 15 minutes prediction horizon. The accuracy of this model depends on the optimal value of parameters k and d. Computation cost can be high if the algorithm needs to compute the distance between the current and all previous traffic states for every prediction. P. Cia *et al.* [40] proposed the improved kNN model for extracting the spatiotemporal correlation for traffic forecasting, the neighbors are selected according to the Gaussian Weighted Euclidean Distance. The results show the traffic flow prediction error improvement of 3% over kNN and 7% over the historical average for 5-minute prediction horizon.

Another approach in statistical machine learning is the Bayesian network for traffic forecasting. A Bayesian network is a directed graphical model that represents the conditional independence between the set of random variables [41]. It can handle the non-linearity and non-stationarity in the traffic data. It can also extract the spatiotemporal dependencies for traffic forecasting. Information from the other road segments is useful in predicting the traffic conditions at the target road segment in the transportation network. It is difficult to directly describe the correlation of traffic conditions on other road segments to the traffic conditions on the target road segment because various variables are involved. These variables need to be determined to access this relationship. In the transportation domain, to make the problem more tractable, it is assumed that the traffic condition on the target road segment is dependent only on its immediate neighboring road seg-

ments [14].
S. Sun *et al.* [14] modeled the transportation network as a Bayesian network. The authors used the Principle Component Analysis (PCA) to find the highly correlated road segment, which selects only a subset of neighboring road segments as nodes of a Bayesian network. PCA reduces the dimensionality of a data-set consisting of many interrelated variables (road segments in this case) while retaining as much as possible of the variation present in the data-set. The joint probability between cause node (data used for forecasting) and effect node (data to be forecast) is described using the Gaussian Mixture Model (GMM). The parameters are estimated via the Competitive Expectation-Maximization (CEM) algorithm. GMM is a probabilistic model that assumes that all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. CEM is an iterative statistical algorithm that calculates the unknown parameters of Gaussian distributions using maximum likelihood estimation. The traffic flow prediction root mean square error improvement is 12.6 vehicle-count per 15 minutes over the AutoRegressive (AR) method. The above method limits the scalability of the Bayesian prediction model which degrades the performance. In the transportation network, due to the various external system changes, the traffic condition's dependency between other and target road segments is not direct. Also, the implementation of the Bayesian network requires domain knowledge, and the outcome depends upon the choice of the root node (prior), which is not directly learned from the data.

Support Vector Regression (SVR) is a statistical machine learning approach in which the best fit line to the data is referred to as hyperplane. The data points on either side of the hyperplane that is closest to the hyperplane are called support vectors which are used to plot the boundary line [42]. To perform this task, SVR used a set of mathematical functions that are defined as the kernel. A Kernel projects the non-linear data onto a higher dimension space making it easier to classify the data where it could be linearly divided by a plane. SVR has been proved to outperform the other regression methods in high dimensional data [43]. SVR is used by J. Hu *et al.* [44] for traffic flow prediction. The model uses a Par-

ticle Swarm Optimization (PSO) algorithm to optimize SVR parameters to predict the traffic flow. The mean absolute error and percentage error improvement of SVR in traffic flow prediction (vehicle-count/5-minute) with PSO is 6.34 and 8.27%, respectively, compared to multiple linear regression. Gaurav *et al.* [45] uses Bayesian SVR (BSVR) for traffic speed prediction. The SVR does not provide any information about the uncertainty in prediction. Bayesian methods are used to select the suitable hyperparameters and estimate variance (error bar) associated with each prediction. The Bayesian SVR shows the traffic flow error improvement of 6.07% over SVR for a 15-minute prediction horizon. Jeong *et al.* [46] used online learning weighted SVR for traffic flow prediction. As per the author, other methods assign equal weights to each time-series data point regardless of their relative order in the data set. Therefore, the author used the SVR with weighted learning methods, and their results show that the average error improvement over SVR is 2.8%. The major limitation of SVR is that it requires domain knowledge; for example, the kernel needs to be set correctly to achieve the optimal result [47].
Feature engineering is a process where domain knowledge is used to hand-craft the features. These hand-crafted features reduce the complexity of the data and make the pattern more visible to the learning algorithm. The above-discussed algorithm used hand-crafted features. The performance of the above-discussed algorithm depends on how accurately the features are identified and extracted. Also, kNN, SVR, and Bayesian consider simplifying assumptions (limited scalability in terms of time and space) that need to be addressed.

Artificial Neural Network (ANN) are biologically inspired computational networks [48]. ANN architecture has three layers, input layer, hidden layer, and output layer. A node or neuron in a layer is connected to all or subsets of nodes in the subsequent layer. Each connection of the neural network is associated with a weight that finds the input data's correlation. Each node has an activation function that defines the output of the nodes. The activation function is used to introduce non-linearity in the modeling capabilities of the network. Kranti K. *et al.* [49] used ANN for short-term traffic flow prediction on the non-urban highway for heterogeneous traffic conditions. This study's main objective is to investigate whether an artificial neural network can be used for short-term prediction of traffic flow in case of heterogeneous traffic conditions, specifically for Indian road scenarios. The model incorporates vehicle category, traffic volume, speed, density, time, and day of the week as input variables. The results show that the ANN model was able to predict the vehicle count accurately even if vehicle category and their corresponding speeds were considered distinct variables. The root mean square error of the ANN model is 0.85 for a 5-minute prediction ahead. A. Csikós *et al.* [50] used ANN for traffic speed prediction, where the model is trained using real and synthetic data generated from Visim [51] because of traffic data scarcity on an urban road. As part of data pre-processing, the feature selection method proposed by P. Devijver and K. Josef [52] is used, which greatly reduces the number of parameters. The designed model gives a 17.35% traffic speed prediction error.

Table 2.2 summarizes the works on non-parametric (machine learning and ANN) approaches for traffic stream variables prediction. The summary shows the studied data, the non-parametric method used for traffic flow prediction, and the used method's accuracy. Compared with table 2.1, we conclude that in most researches, the prediction error is improved with the machine learning and artificial neural network methods due to non-linear and complex characteristics of the transportation network.

Table 2.3 compares the data-driven approach (parametric and non-parametric) with their key points.

The non-parametric (machine learning and artificial neural network) methods require prior knowledge of the specific domain for feature extraction, and their architecture is shallow. The traffic stream variables In general, literature [5, 25, 26, 54] show that the deep learning models have good predictive power and robustness as compared to the parametric and non-parametric (machine learning and ANN) models. Deep learning models extract features with less prior knowledge and handle a large amount of data and the high dimension of features. A deep learning architecture discovers a massive amount of structure in the data because of multi-layer architecture.

Table 2.2: Non-parametric Approaches for Traffic Flow Prediction, PH stands for Prediction Horizon.

Research Work	Data Collection	Collection frequency	Method	PH	Accuracy
Nihan and David [13]	I5, Washington State	1-min	KNN	1-min	RMS error of kNN is 13.8, RMS of ARIMA is 13.0.
Chang *et al.* [39]	Suwon tollgate, Korea	15-min	KNN-NPR	15-min	average 9.3% error improvement over ARIMA
P. Cia *et al.* [40]	Freeways, Liuliqiao District, Beijing	5-min	kNN with Gaussian Weighted Euclidean Distance	5-min	error improvement of 3% over kNN and 7% over historical average
S. Sun *et al.* [53]	UTC/SCOOT system in Traffic Management Bureau of Beijing	15-min	Bayesian Network	15-min	RMSE improvement of 12.6 compared to AR model.
Hu *et al.* [44]	Ring Road of Beijing	15-min	SVR with PSO	15-min	MAPE improvement of 8.27% compared to linear regression.
Gaurav *et al.* [45]	Pan Island Expressway (PIE) in Singapore	5-min	Bayesian SVR	15-min	MAPE 6.07%
K. Kumar *et al.* [49]	NH-58,Muzaffarnagar, India	5-min	ANN	5-min	MAE is .6281
A. Csikós *et al.* [50]	District 6, Budapest	5-min	ANN	5-min	17.35% prediction error

Nagare and Bhatia [54] use the BackPropagation Neural Network (BPNN) for traffic flow prediction. The BPNN consists of the input layer, one or more hidden layers, and the output layer. During the training phase, the input is fed to the hidden layer whose output is the feature vector. The feature vector is a vector that contains information describing data's important characteristics. It is then combined with the model's weights or parameters to predict the final output. The

Table 2.3: Comparison Between Data-Driven Approaches.

Data driven Approach	Methods	Key Points
Parametric	ARIMA, SARIMA, KARIMA, Kalman Filter, Extended Kalman Filter	Suitable for low order model – Based on linear model assumption – Guaranteed to converge – Number of parameters are fixed
Non parametric (Machine learning and ANN)	kNN, SVR, ANN, Bayesian Network	– Parameters are data dependent – Require large amount of data – Computationally expensive – Guaranteed to converge

predicted value is then compared with the measured value, and the difference (the error) is propagated backward through the network to update the parameters. Their results show that the average error improvement of BPNN is up to 2.09% as compared with SVR and up to 11.54% as compared with ARIMA for a 15-minute prediction horizon. Huang *et al.* [55] used a Deep Belief Network (DBN) model for traffic flow prediction. It is a stack of Restricted Boltzmann Machine (RBMs). The RBM has only two layers - the visible layer and the hidden layer. In RBMs, the task is to learn a probability distribution over the input data, which closely approximates the input distribution. Their results show that the model achieves close to 5% improvement over the SVR and kNN methods for traffic flow prediction considering a 15-minute prediction horizon.

Autoencoder is the artificial neural network that learns to compress and encode data such that the compact representation of data is close to the original input [56]. Autoencoder is used for dimensionality reduction. In the case of the traffic data, this method extracts the most influential feature from the data. The hidden layer of the autoencoder gives a compact representation of the data. Yisheng *et al.* [5] used Stacked AutoEncoder (SAE) to learn spatiotemporal traffic state features for short-term traffic flow prediction. In the SAE, multiple autoencoders are stacked such that the output from the hidden layer is fed as an input to other autoencoder layers. The average traffic flow prediction accuracy improvement of SAE is up to

4.8% as compared with BPNN, over 15.34% compared with the SVR.

Convolutional Neural Network (CNN) is a successful model in computer vision, image processing, and natural language processing tasks [57]. CNN is ideal to obtain the local spatial relationships in the input data. Local features can be extracted based on the small size weight matrix ($3 \times 3, 5 \times 5, 7 \times 7$). All the data points share the same weight matrix (filter); therefore, the number of parameters is not huge as compared to other models [30]. Also, it performs the parallel operation, speeds up the processing as compared to the other learning models. W. Zhang *et al.* [20] performs traffic speed prediction on the road considering the traffic state on the adjacent road. The two-dimensional space-time matrix (image) is passed to the CNN to extract a spatiotemporal feature vector containing abstract information about adjacent roads' traffic states. CNN is preferable over other deep learning models because parameters are shareable making it computationally less expensive.

Traffic state forecasting is the sequential problem in which the current state's output depends on the past traffic state. The recurrent learning models, Recurrent Neural Network (RNN) and Long Short Term Memory (LSTM), are designed for the sequence learning task in deep learning approaches. The BPNN, DBN, CNN, and SAE consider each input independently. The output doesn't depend upon previous observations. RNN suffers from a vanishing gradient problem and doesn't hold onto long temporal dependencies. LSTM solves these shortcomings. It uses the gating mechanism (input, forget, and output gate) and holds long and short temporal dependencies [18]. Z. Zhao *et al.* [19] used both LSTM and RNN for traffic flow prediction. For the 15-minute prediction horizon, traffic flow accuracy improvement of RNN is 5.92% as compared with the SVR, and 1.06% as compared with the SAE. The author compared LSTM with the nonrecurrent neural network, and the results show that the average prediction accuracy improvement of LSTM is 10.45% as compared to SAE and 18.3% as compared to SVR.

Table 2.4 summarizes the deep learning approaches for traffic stream variables prediction. The summary shows the studied data, data collection frequency, deep learning method used for traffic stream variables prediction, and the used method's accuracy. Compared with the table 2.2, the error improvement of LSTM is significant, approximately 15% over shallow non-parametric models, and for all the other deep learning models, the error improvement is more than 2%. We conclude that the deep learning models have better prediction power than the parametric and shallow non-parametric approaches, as observed from the literature study.

Table 2.4: Deep Learning Approaches for Traffic Stream Variables Prediction, PH stands for Prediction Horizon.

Research Work	Data Collection	Collection Frequency	Method	PH	Accuracy
Huang *et al.* [55]	PeMS, California	15-min	DBN	15-min	≈5% accuracy improvement over SVR and Bayesian
Yisheng *et al.* [5]	PeMS, California	5-min	SAE	Multi step	4.8% accuracy improvement over BPNN and 15.34% over SVR.
W. Zhang *et al.* [20]	I-5 Seattle	5-min	CNN	5-min	2.8% error improvement over SVR and 4.3% over SARIMA
Z. Zhao *et al.* [19]	Beijing	5-min	RNN and LSTM	Multi step	15-min prediction accuracy improvement of RNN is 5.92% over SVR, 1.06% over SAE. Average prediction accuracy improvement of LSTM is 10.45% over SAE and 18.3% over SVR.

The data-driven approaches discussed above consider temporal or spatiotemporal traffic data for traffic forecasting. The traffic stream characteristics are nonlinear, dynamic, and complex. The deep learning model represented traffic features without prior knowledge and proved to have a better prediction accuracy of traffic stream variables than linear and nonlinear shallow learning models [5, 25, 26, 54]. The models that extract the spatiotemporal dependencies are more accurate in traffic forecasting as compared to the model that only extracts temporal dependencies [58]. Similarly, the model that is dependent on past traffic information while predicting future traffic stream variables is more accurate as compared to the model that is independent of past traffic information [19]. Therefore, the learning model that considers past and current traffic information from adjacent and target road segments is the most appropriate for traffic forecasting.

2.2 Weather Impact on Traffic Stream Variables

The adverse weather conditions such as rainfall, snowfall, fog, etc., affect the traffic stream variables such as traffic speed and flow on the road segment. The heavy rainfall and fog affect the driver's visibility, waterlogging due to prolonged rainfall and heavy snowfall affects the road capacity, which affects the traffic speed and flow on the road segment. In this section, we discuss the literature studies that consider the weather variables for traffic stream variables prediction (section 2.2.1) and the studies that consider the impact of weather on traffic conditions in terms of traffic demand and operation (section 2.2.2).

2.2.1 Traffic Prediction During Adverse Weather Conditions

The majority of the traffic prediction domain research focuses on addressing traffic prediction under normal traffic conditions. Abnormal traffic conditions, such as non-recurrent traffic congestion caused by planned events such as road works or unplanned events such as adverse weather conditions and accidents cannot be neglected. Short-term traffic prediction is more important during abnormal conditions because of the uncertainty of how traffic conditions evolve.

There is very few research that incorporates the weather data for short-term traffic variables prediction. Zhao *et al.* [59] used the parametric approach to examine the impact of snowy and icy conditions on traffic speed data from Buffalo, New York. The author used the linear regression model to estimate the average freeway speed as a weather condition function.

W. Qiao *et al.* [25] used the non-parametric k Nearest Neighbor (kNN) algorithm to predict travel time on a path (origin to destination). As per the author, a traffic sensor on a source road segment is paired with its nearest rainfall station based on the GPS coordinates. The Euclidean distance-based correlation is used to find the correlated k neighbors (road segments and weather stations). The rainfall variable is taken as a point value considering the traffic at a road segment. Their results show that kNN gives an average error improvement of 18% over the ARIMA model for future 30-minute travel time prediction.

Y. Jia *et al.* [26] used the DBN and LSTM deep learning model for traffic flow forecasting on a road segment considering hourly rainfall data. Rainfall data is used as point data in their research. They take into account the temporal dependency of traffic and weather data. Their results show that for 30-minute traffic flow prediction, prediction accuracy improvement of LSTM is 2% over DBN, 4% over BPNN, and 9% over ARIMA. Both [25, 26] consider rainfall data as point data. Rainfall value from the nearest station is interpolated to the target road segment. The correlation between traffic and rainfall is not a direct measurement of the distance between the road and weather stations because of terrain feature constraints. During prolonged heavy rainfall, water flow from one road segment to another, waterlogging blocks the majority of the road segment, which affects the traffic variables of other road segments. Therefore, the nearest station's rainfall data will not capture the impact of rainfall on other roads. There is a need to capture the spatiotemporal dependency between the traffic and weather data of target and adjacent road segments under such a scenario. These studies extract the temporal dependency of traffic and weather data to predict the future traffic conditions ignoring the spatial dependency.

A. Koesdwiady *et al.* [27] used spatiotemporal input data and the DBN model is used to predict the traffic flow during adverse weather conditions. They take into account the spatiotemporal dependency of adjacent roads' weather and traffic data while predicting the traffic flow on the target road. As per the author, with k past traffic data sequence, the most recent rainfall data value is sufficient because rainfall is complementary data. The performance of the model in terms of mean absolute error is 0.002 and 0.0087 as compared to the ANN and least square method for 5-minutes prediction horizon, respectively. The author assumes that the rainfall data is complementary while predicting traffic flow on a road segment. The prolonged impact of weather conditions is significant on traffic stream variables. Waterlogging due to saturated soil absorption capability, weakened drainage system greatly affects the traffic stream variables on the target and adjacent road segments. The extended weather sequence is needed to capture the prolonged impact of weather variables. Table 2.5 summarizes the parametric and non-parametric approaches for traffic stream variables prediction during adverse weather conditions. The summary shows the studied data, the data-driven method used for traffic stream variables prediction during adverse weather conditions, and the used method's accuracy. The shallow non-parametric approaches show error improvement over parametric approaches as observed from the studies. The deep learning model shows better traffic prediction accuracy than the parametric and the shallow non-parametric model during adverse weather conditions.

Table 2.5: Parametric and Non-parametric Approaches for Traffic Stream Variables Prediction During Adverse Weather Condition, PH stands for Prediction Horizon.

Research Work	Data Collection	Collection Frequency	Method	PH	Accuracy
Y. Zhao *et al.* [59]	Buffalo	1-hour	Linear regression (**Parametric**)	1-hour	Model achieve R^2 score 56.1%
W. Qiao *et al.* [25]	I-95, Maryland	5-min	kNN (**Non-parametric** (Machine learning))	Multi step	error improvement of 18% over the ARIMA model
Y. Jia *et al.* [26]	Ring road, Beijing, China	2-min traffic and 1-hour of rainfall data	DBN (**Non-parametric** (Deep learning))	10-min and 30-min	For 30-min traffic flow prediction, error improvement of DBN 2% over BPNN, and 9% over ARIMA
A. Koesdwiady *et al.* [27]	San Francisco, California	15-min	DBN (**Non-parametric** (Deep learning))	15-min	MAE improvement of .002 and .0087 over ANN and least square method, respectively.

2.2.2 Study of Weather Impact on Traffic Stream Variables

The majority of the work considering traffic prediction during adverse weather conditions is performed on dense traffic data obtained from sensor devices installed on a road segment [13, 25, 26, 27]. The deployment of ITS infrastructure is costly. Therefore, in most locations, limited ITS infrastructure is available for traffic monitoring, and the available traffic data are very sparse. Deep learning algorithms require a large amount of data to predict the traffic stream variables during adverse weather conditions. With limited ITS infrastructure and sparse traffic data, it is difficult to predict the traffic stream variables accurately. Therefore, there is a need to generate realistic fine-grained traffic data during adverse weather conditions. To generate fine-grained synthetic data, we should first observe the impact of rain on traffic stream variables. We further used these observations to generate traffic data during rainfall for a variety of roads.

Ibrahim *et al.* [21] observed that the rainfall intensity values 0-0.01 in/hr, 0.01-.25 in/hr and >0.25 in/hr causes speed reduction of 2-4%, 5-8% and 9-14% and capacity reduction of 2-4%, 5-10% and 11-18%, respectively. This study's traffic data were from the freeway traffic management system of the Queen Elizabeth Way in Mississauga, Ontario. The regression analysis is used to select the proper models for representing the flow-occupancy and speed-flow relationship. The multiple regression analysis techniques were used to test for the significant differences in traffic operations between different weather conditions. As per their observation, adverse weather conditions reduce the slope of flow-occupancy function and cause a downward shift in the speed-flow function. M. Kyte *et al.* [23] used the regression analysis to study the effect of weather on free-flow speed on an interstate freeway in Idaho. As per their observation, wet surface causes 9.5 km/hr reduction, snowfall causes 16.4 km/hr reduction, and wind $>$ 24 km/hr causes 11.7 km/hr reduction in operating speed, respectively. M. Agarwal *et al.* [22] classifies the rain and the snow events by their intensity levels and identifies how precipitation intensity changes the impact of speed and capacity of roadways. The previous research on weather impact is obtained from studies outside the United States. This study used the traffic data and the rainfall data collected from the

Minnesota traffic management center and the national climate data center. For this study, the author used the statistical analysis using the Bonferroni method. This allows many comparison statements to be made (or confidence intervals to be constructed) while still assuring an overall confidence coefficient is maintained [60]. As per their observation, heavy rainfall causes 14% reduction, and heavy snowfall causes 22% reduction and fog with visibility < 0.25 mile causes 11% reduction in traffic density, respectively. As per their observation, the light to heavy rainfall causes 1-7% speed reduction on freeway roads. Using the same data and the additional data from cities in the United States of America, Rakha *et al.* [61] concluded that the light and heavy rainfalls cause a speed reduction of 2-3.6% and 6-9%, respectively on the freeways. J. Weng *et al.* [62] observe that snowfall affect the road capacity by 33%, the traffic speed by 10-40% and traffic flow by 12-50%. This study is performed on the expressway in Beijing, China. The heavy snowfall (thickness $\geq$ 30cm) affects the traffic flow and the speed on a subsequent day. Due to slippery ice, the road condition is affected, which decreases the traffic speed by 30% and traffic flow by 40%. Hammad *et al.* [63] studied the impact of fog on traffic stream variables. During fog, a traffic speed reduction of 6-18% is observed on the expressway. There is no road capacity reduction observed. For the peak and the off-peak hours, traffic speed is affected, but there is no significant decrease in traffic flow. For night time, the impact of fog on traffic speed and flow is significant, as drivers avoid unnecessary trips and the vehicle slows down to avoid any accident. The worst effect of rainfall on traffic variables is captured in literature until heavy rainfall conditions [21, 22, 23]. The urban flood or massive waterlogging, in reality, worsens the situation. The impact of rainfall on traffic stream variables is studied in literature considering the developed country's freeway or expressway, where road conditions and drainage systems are exceptional. In developing countries, heavy rainfall causes waterlogging, affecting the traffic condition on the road segments. The study considering the impact of prolonged rainfall on different types of roads is needed to observe the impact of waterlogging or urban flood on traffic stream variables. Table 2.6 summarizes the works on impact of rainfall on traffic speed.

Table 2.6: Reduction in Traffic Speed for Different Weather Condition as Observed in Literature.

Research	Rainfall	Snowfall	Wind
Ibrahim *et al.* [21]	2-14%		
kyte *et al.* [23]	9.5 km/hr	16.4 km/hr	11.7 km/hr
M. Agarwal *et al.* [22]	1-7%	3-10%	2-9%
Rakha *et al.* [61]	2-9%	5-16%	2-10%
Hu *et al.* [64]	2.6-18.4%		
J. Weng *et al.* [62]		10-40%	

2.3 Summary

The traffic stream variables prediction is a non-linear and complex problem. The machine learning and neural network approaches ignore the hidden influencing factor because of shallow architectures. The deep learning model extracts the influential hidden features from the input data and provides better prediction accuracy. Whether the deep learning model can improve traffic stream variables prediction accuracy with additional input, like rainfall variable, is an important question that needs to be answered.

There are limited studies that forecast traffic during adverse weather conditions. The majority of the work focuses on temporal dependency. The traffic on the target road gets affected due to the traffic and weather conditions on the adjacent road segments. Therefore, spatiotemporal dependencies should be taken into account while predicting the traffic stream variables.

To capture the impact of waterlogging or water flow path during heavy rainfall, there is a need to capture the impact of rainfall on other road segments while predicting the traffic stream variables. Similarly, the prolonged impact of rainfall is significant on traffic stream variables which can't be neglected.

The impact of rainfall on traffic operation is captured until heavy rainfall conditions on developed countries' freeway road segments. The prolonged heavy rain causes massive waterlogging in developing countries due to weak drainage systems, narrow roads, and encroachments. Therefore, a study capturing the impact of prolonged heavy rainfall on traffic operation on different roads is needed considering developing countries' scenario.

The majority of the work considering the traffic prediction during adverse weather conditions is performed on the dense traffic data obtained from the ITS infrastructure. The deployment and maintenance of ITS infrastructure are costly, and therefore only a few of the roads worldwide have dense traffic data. The alternate cost-effective traffic data sources such as taxi GPS and cellular data are sparse and incomplete. To overcome the limitation, realistic fine-grained synthetic data is needed that can accurately describe the effect of rainfall on traffic stream variables. The dense traffic data is private and not available for comparison of various learning methods. The fine-grained synthetic data can also be useful for training and testing in ITS and comparing multiple methods on a similar data-set.

CHAPTER 3

Problem Description

Accurate traffic forecasting is important for the applications of the ITS such as ATIS and AMTS. In ATIS, accurate traffic information informs about traffic congestion which can be used to reduce the uncertainty of the future traffic state, improves resource utilization, provides expected delays, alternate routes, travel time estimates, etc. Short-term traffic stream variable's forecasting during adverse weather conditions such as rainfall, snow, fog, etc., is needed for both travelers and traffic management applications. Our objective is to predict the traffic stream variables on the road segment at some future time instance during adverse weather conditions. The traffic stream variables of our interest are traffic speed and flow. The traffic speed is the average of individual vehicle speed on the road segment for a particular time. The traffic flow is the number of vehicles passing the road segment in a given time.

Figure 3.1 shows the representation of the system i.e., set of inputs, output, and function that maps the inputs to the output. For a particular geographical area, we have a large road network that consists of road segments linked with each other. Each road segment has traffic data corresponds to it, which is generated periodically by various sources. The traffic data contains the traffic stream variables values such as average traffic speed, traffic flow, traffic density, etc. We also have multiple weather stations located in the same geographical area. Each weather station provides weather reading periodically. The weather data contains weather variables such as temperature, precipitation, snow, visibility, wind speed, etc. Weather stations also provide the weather forecast value, which gives an insight into the future weather conditions. Figure 3.2 shows the different sources of

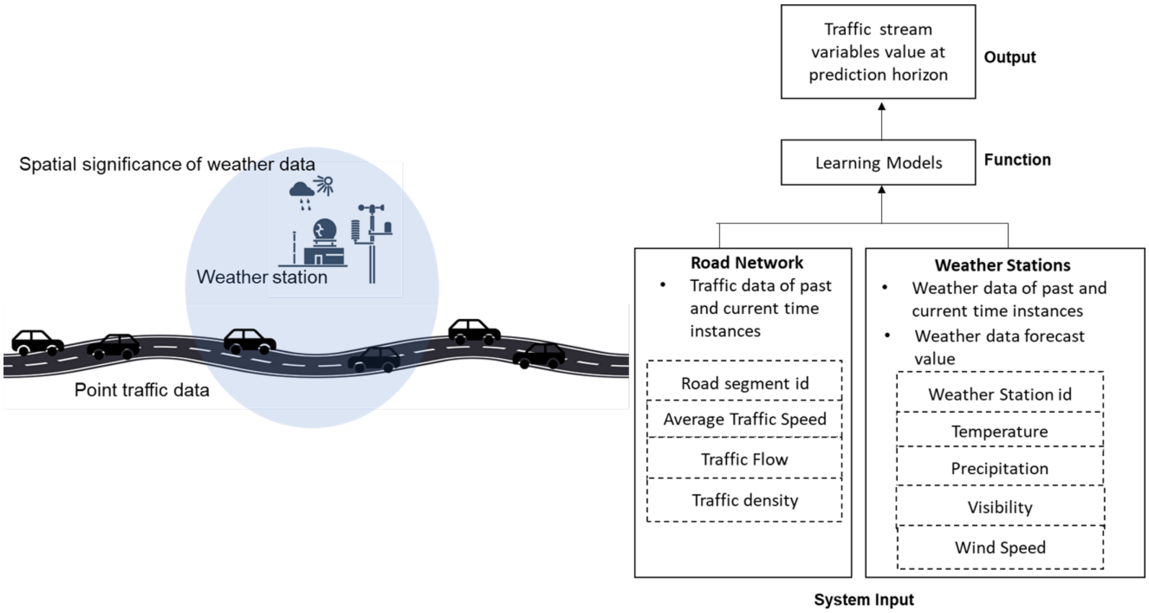


Figure 3.1: System Representation.

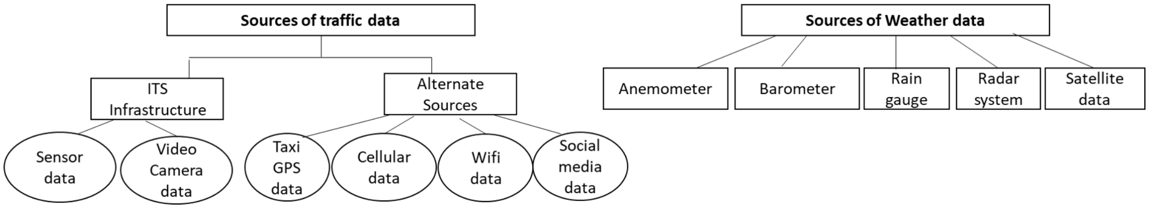


Figure 3.2: Different Sources of Input Data.

the input traffic and weather data. The traffic data is collected either from the ITS infrastructure or from an alternate source. ITS infrastructure provides dense and complete traffic information but they require high deployment cost [24]. Traffic sensors data and video camera data come under this category. The alternative source of traffic data is cost-effective. The Wifi, GPS, cellular data, social media data are the cost-effective but sparse and erroneous source of traffic information.

The weather variables are measured from various devices such as rainfall amount over a particular period is measured by a rain gauge, radar, and satellite. The radar and satellite data provide a volumetric measure of rainfall for an area while a rain gauge provides data from a very specific point. The other devices such as an anemometer are used for measuring wind speed, the barometer is used for measuring atmospheric pressure, etc.

Traffic data is collected from the sensors installed on the road segment and weather data is interpolated from the stations based on the distance between the road segment and station. Therefore, for a road segment, traffic data is more accurate as compared to the weather data. The correlation between the weather station and road segment is not directly related to the distance between them because of the terrain feature constraint. The rainfall at the current time instance will affect the traffic stream variables at some near-future time instance. The water flows or waterlogging at the adjacent road will affect the target road at the future-time instance. The immediate effect of rainfall is visible on traffic speed. Rainfall affects visibility and therefore, the vehicles slow down their speed to avoid crash risk. The impact of rainfall on traffic flow is not immediate. The traffic flow gets affected if, for some duration, the average speed slows down. Road capacity reduces because of a weak drainage system that causes waterlogging due to prolonged rainfall, which affects the traffic speed and flow.

The collection frequency of traffic data and weather data is different. Traffic data is denser as compared to weather data. The weather data doesn't give fine-grained information about the impact of rainfall on traffic stream variables. This makes the traffic prediction during adverse weather conditions a complex problem.

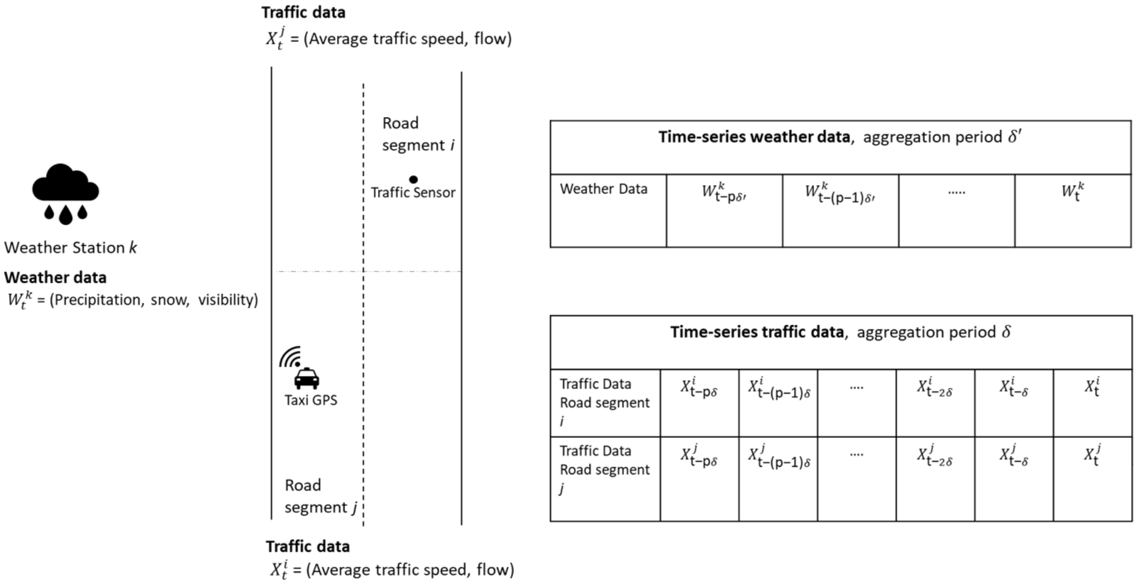
The roads that are not immediate neighbors but due to terrain features, the distant road segment may affect the traffic stream variables of the target road segment during adverse weather conditions. Therefore, weather data related to other road segments should be taken into account to predict the traffic stream variables. We need a mechanism that considers the weather data corresponds to other road segments but also, limits the number of stations to be considered for the target road segment. Similarly, to capture the prolonged impact of rainfall, the model needs a more extended sequence of rainfall data compared to the traffic data. Therefore, we need a mechanism that captures the prolonged impact of rainfall. The traffic and weather sequence along with information related to the weather condition spatial and prolonged significance is passed to the learning model. The learning model is a mathematical function that converts the set of inputs to the output using some learning module. In our case, the output is traffic stream variables value at some future time instance. So our one of the objectives in this research is to predict the accurate traffic stream variables during adverse weather conditions. We need a learning model that extracts the hidden features such as waterlogging, road characteristics, drainage system, water flow path, etc., to make an accurate prediction of traffic stream variables. If we have dense data corresponds to each feature, we will pass these data as input to our learning model. In the absence of these datasets, it is necessary to provide a large amount of fine-grained traffic and weather information such that the learning model extracts the hidden features to forecast traffic during adverse weather conditions.

The fine-grained traffic data collected from ITS infrastructure is not available for a majority of the locations because of the high deployment and maintenance cost. The alternate source of traffic data provides sparse and incomplete information. Therefore, a mechanism to generate the fine-grained traffic data is needed. The traffic condition on a road segment depends upon various parameters. For example, broad roads with excellent drainage systems are less prone to waterlogging as compared to narrow roads with a poor drainage system. These parameters should be considered while generating realistic fine-grained data during adverse weather conditions. Our other objective in this research is to generate fine-grained realistic traffic data considering weather-related parameters.

Section 3.1 gives the definition and mathematical representation of the traffic prediction system along with objective and performance metrics. Section 3.2 gives the definition and mathematical representation of the synthetic data generation system. Section 3.3 list out the assumptions related to the traffic prediction problem.

3.1 Problem Statement for Traffic Stream Variables Prediction

Input traffic data and weather data are generated periodically. An aggregation period is the aggregation of all data points for a single resource over a specified period, which defines the granularity of the data. The traffic and weather data may or may not have the same aggregation period. Time-series data is a set of data collected sequentially at the aggregation period. For a particular geographical area, time-series traffic data from the road segments and time-series weather data from the weather stations are provided as inputs. This time-series traffic and weather data are spatiotemporal data because it provides the traffic and weather reading from past and current time instances along the geographical area. The length of the sequence (past information) is called lag. This spatiotemporal data up to a particular lag value is used to predict the future traffic stream variables. The prediction horizon is how far ahead the model predicts the future traffic stream variables. Our goal is to predict the traffic stream variables on the road segment at the prediction horizon. Figure 3.3 shows the system representation of the input data in our model and the definition of the input data is provided below.


Traffic data
X_t^j = (Average traffic speed, flow)
Road segment *i*
Traffic Sensor
Weather Station *k*
Weather data
W_t^k = (Precipitation, snow, visibility)
Taxi GPS
Road segment *j*
Traffic data
X_t^i = (Average traffic speed, flow)

Time-series weather data, aggregation period δ'				
Weather Data	$W_{t-p\delta'}^k$	$W_{t-(p-1)\delta'}^k$	.....	W_t^k

Time-series traffic data, aggregation period δ						
Traffic Data Road segment *i*	$X_{t-p\delta}^i$	$X_{t-(p-1)\delta}^i$	....	$X_{t-2\delta}^i$	$X_{t-\delta}^i$	X_t^i
Traffic Data Road segment *j*	$X_{t-p\delta}^j$	$X_{t-(p-1)\delta}^j$	....	$X_{t-2\delta}^j$	$X_{t-\delta}^j$	X_t^j

Figure 3.3: System Representation of the Input Data.

Definition 1: **Traffic data** A geographical area has N number of road segments. X_t^i is a traffic data during t^{th} time interval at i^{th} road segment. The traffic data contains information about the traffic flow and average traffic speed.

$$X_t^i = (\text{Average traffic speed, traffic flow })$$

P^j is the set of road segments other than i (target road segment) and p is the lag value.

$$X_t = (X^i_t, X^i_{t-\delta}, \ldots, X^i_{t-p\delta}), \{(X^j_t, X^j_{t-\delta}, \ldots, X^j_{t-p\delta}),\ X^j \in P^j\}$$

X_t is the spatiotemporal traffic data from all N road segments in the geographical area for past and current timestamps. δ is the aggregation period.

Definition 2: **Weather Data** A geographical area has M number of weather stations. These weather stations provide weather data. A Weather data is a sequence of weather readings for past and current timestamps. W^k_t denotes the observed weather value during t^{th} time interval at k^{th} weather station. The weather reading contains information about precipitation, snow, and visibility.

$$W^k_t = (\text{precipitation}, \text{snow}, \text{visibility})$$

P^k is the set of k weather stations out of M and p is the length of the sequence or lag value. $f_{spatial}$ captures the weather impact from $M - k$ stations at time t. $\hat{W}_t$ stores the spatial significance of weather variable.

$$\hat{W}_t = f_{spatial}(\{W^{M-k}_t\}),$$

The prolonged impact of weather variable is captured using $f_{temporal}$, $\overline{W^k_t}$ stores the temporal significance of weather variables during t^{th} time interval at k^{th} weather station. q is the extended sequence length such that $q >> p$.

$$\overline{W^k_t} = f_{temporal}(W^k_{t-q\delta'}, \ldots, W^k_{t-(p+1)\delta'}),\ \ q >> p$$

$$W_t = \{(\{W^k_t, W^k_{t-\delta'}, \ldots, W^k_{t-p\delta'}\}, \hat{W}_t, \overline{W^k_t}),\ W^k \in P^k\}$$

W_t is the spatiotemporal weather data from k weather stations for past and current timestamps alongwith spatial and temporal information. $\delta^{'}$ is the aggregation period and $\delta^{'} \geq \delta$.

Our goal is to predict the traffic stream variables at target road segment i for prediction horizon t_g such that $t_g = t + g\delta$. $Y^i_{t_g}$ is the predicted value and f is the function that maps the input (X_t and W_t) to the predicted output.

$$Y_{t_g}^i = f(X_t, W_t) \tag{3.1}$$

For each predicted $Y_{t_g}^i$ value at target road segment i for prediction horizon t_g, there is true $X_{t_g}^i$ value. Our objective is to predict $Y_{t_g}^i$ which is in the agreement of the $X_{t_g}^i$, by defining an appropriate loss function Ł(θ), where θ are the parameters to be learned. Here, T is the prediction horizon and S is the total number of samples.

$$min\ Ł(\theta) = \frac{1}{S}\sum_{i=1}^{S}\sum_{g=1}^{T}|Y_{t_g}^i - X_{t_g}^i| \tag{3.2}$$

Figure 3.4 shows the system representation of the prediction model. The prediction or learning model learns the parameters that help the model to map the given inputs into the desired output. The objective of the learning model is to minimize the difference between the measured and predicted value of the traffic stream variables. The model keeps on learning till it gets converges.

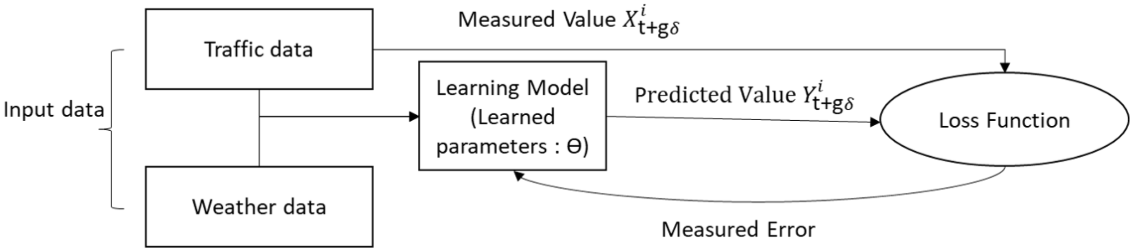

Input data
Traffic data
Weather data
Measured Value $X_{t+g\delta}^i$
Learning Model (Learned parameters : ϴ)
Predicted Value $Y_{t+g\delta}^i$
Loss Function
Measured Error

Figure 3.4: System Representation of the Learning Model.

In this research, three loss functions evaluate model effectiveness: the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), and the Mean Absolute Percentage Error (MAPE). These metrics find the mean error between the actual and predicted value of the traffic stream variables for sample data. They are defined as,

$$\text{MAE} = \frac{1}{S}\sum_{i=1}^{S}\sum_{g=1}^{T}|Y_{t_g}^i - X_{t_g}^i|$$

$$\text{RMSE} = \sqrt{\left[\frac{1}{S}\sum_{i=1}^{S}\sum_{g=1}^{T}(Y_{t_g}^{i} - X_{t_g}^{i})^2\right]}$$

$$\text{MAPE} = \frac{1}{S}\sum_{i=1}^{S}\sum_{g=1}^{S}\frac{|Y_{t_g}^{i} - X_{t_g}^{i}|}{X_{t_g}^{i}}$$

3.2 Problem Statement for Synthetic Traffic Data

One of our objectives in this research is to solve the traffic data sparsity issue. The learning models require a large amount of data for accurate traffic prediction. In the absence of dense traffic data, the fine-grained realistic synthetic traffic data during adverse weather conditions will be used as a traffic data input in our learning model. The quality of the prediction depends upon how fine-grained and realistic is our synthetic data. The traffic data during rainfall depends upon various parameters such as road characteristics, time of the day, rainfall average, etc. Our goal is to generate synthetic traffic data using a realistic simulation setup that considers these parameters.

Road type R is a parameter that captures the road characteristics. It contains information related to the road such as road segment id, number of lanes, elevation, and route.

$$R = (\text{road segment id}, \text{number of lanes}, \text{elevation}, \text{route})$$

Time of the day T_d is categorised into three category. Peak hour, off-peak hour and night time.

Rainfall average α is a parameter that captures the impact of prolonged rainfall.

Friction parameter μ captures the impact of rainfall on vehicle mobility.

u_r is the traffic speed in the real traffic data-set during adverse weather conditions. Our objective is to derive vehicle speed u_w which is a function of all the above parameters, such that

$$u_w = f_{synthetic}(R, \alpha, \mu, T_d)$$

$$u_w \approx u_r$$

3.3 Assumptions

The assumptions in this research related to the traffic stream variables prediction and synthetic traffic data generation during adverse weather conditions are as follows:

- Traffic data provides accurate traffic stream variables information.
- Traffic data obtained from ITS infrastructure is dense and complete.
- Traffic data obtained from alternate sources are sparse and incomplete.
- Weather stations provides accurate weather information.

Chapter 4

System Model and Proposed Approach

In this chapter, the system model discusses the limitations related to the traffic stream variables prediction during adverse weather conditions and the solutions offered to overcome those limitations in section 4.1. The proposed approach discusses the detailed mechanism to solve the related concerns for accurate traffic forecasting in section 4.2.

4.1 System Model

Traffic stream variable prediction during adverse weather conditions is a non-linear and complex problem. Traffic forecasting involves various hidden factors such as time of the day, road characteristics, and non-recurrent events such as weather conditions, road construction, etc. To predict the traffic stream variables, traffic and weather data are required. The learning model predicts the future value of the traffic stream variables based on the input traffic and weather data. It requires dense and complete traffic and weather data for each road segment to predict the traffic stream variables. The ITS infrastructure provides dense traffic data, but deployment and maintenance are costly. Hence limited roads have dense and complete traffic data. The alternate traffic data source, such as taxi GPS, cellular data, etc., provides sparse and incomplete traffic data.

Also, the traffic stream variables during adverse weather conditions depend upon various hidden features such as water flow path, drainage system, soil absorption capability, etc. The hydrology map gives information on the water flow path. The drainage data gives the real-time status of the drainage capacity. The soil data gives the real-time soil absorption capability. If we have fine-grained data corresponds to these hidden features, we will provide the data directly to our model. If there are no computational limitations, then all of the past and current information corresponds to these data-sets for a particular geographical location is provided to the learning model to predict the traffic stream variables accurately. In reality, the data related to the other features (drainage, soil, etc.) are not available in most locations. Also, these features are a function of history in terms of drainage maintenance and soil absorption capability. These data are not static that can be used directly from the soil and drainage data-set of the geographical location. On the other hand, every system has computational limitations. Therefore complete past information is not available to the learning model. Apart from this, to accurately predict the traffic stream variables, the learning model should consider the non-linear, dynamic, and complex transportation network characteristics.

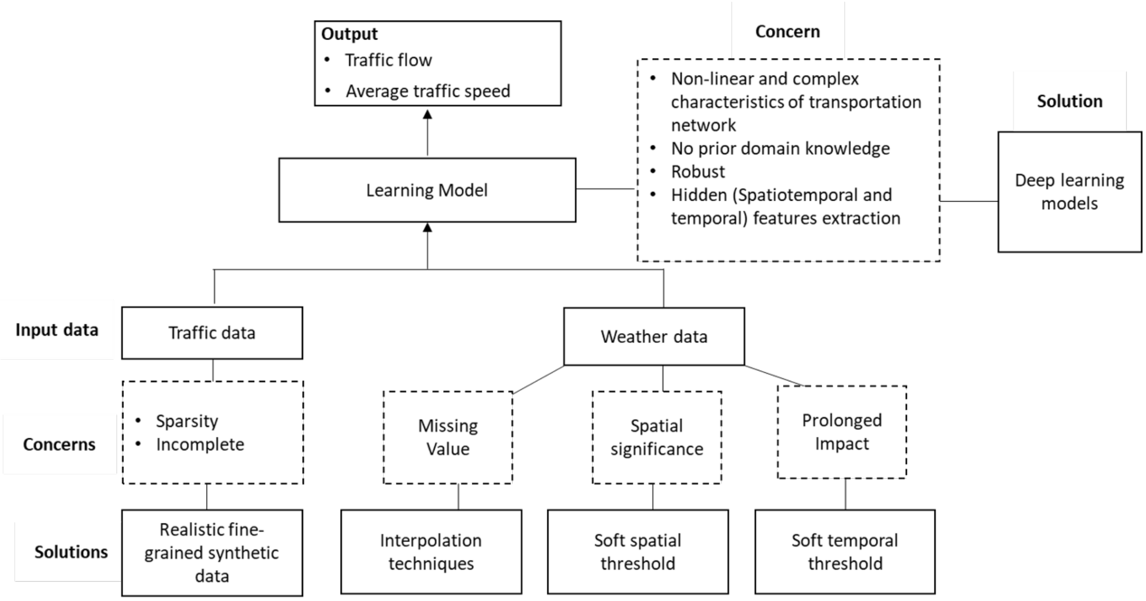


Figure 4.1: System Model.

Figure 4.1 shows the system model. Our system model has traffic data (X_t) and weather data (W_t) as input. The input data is passed to the learning model to predict the future traffic stream variables, such that $Y^i_{t_g} = f(X_t, W_t)$. The main challenge with traffic data is the sparsity problem, it cannot be monitored on some roads or of certain periods. To accurately predict the traffic stream variables on a road segment, there is a need to overcome the traffic data sparsity issue. Therefore, synthetic traffic data is needed to solve this issue. A fine-grained realistic synthetic data will provide a good approximation of the actual traffic data to predict the traffic stream variables. To generate realistic synthetic data during rainfall, we study the impact of rainfall with different intensities on various roads. We simulate the roads prone to waterlogging during prolonged rainfall and those that are not, using Simulator for Urban Mobility (SUMO) [31]. In SUMO, the vehicle speed is derived from the car-following model such that the minimum distance between the following and the leading vehicle on the same lane is maintained. In a current SUMO version, the implemented car-following model does not use the weather-related parameters for computing mobility-related parameters. Therefore, an empirical model is needed that quantifies rainfall's impact on the different types of roads. Based on the road type and rainfall intensity, the quantified values are associated with the road segment, and these values are passed to the car-following model. This model needs to be adapted to use the weather-related parameters while computing the vehicle speed.

The traffic data is more granular compared to the weather data. Therefore with the weather data, a missing value is a significant concern. The interpolation of the weather data is needed to fill out the missing values. We studied and compared various interpolation techniques such as the Theissen polygon, inverse distance weighted method, and linear regression method.

With weather data, another concern is to capture the spatial significance and the prolonged impact of the weather conditions. Prolonged heavy rainfall weakens the drainage system and the soil absorption capability, which results in waterlogging and water flows from one road segment to another. The extended rainfall sequence is needed to capture the prolonged impact of rainfall. To limit the weather data sequence length, we propose the soft temporal threshold mechanism. The model's input should consider the past weather sequence up to the threshold value as a sequence length. Model's input also includes moving weighted average of rainfall value to capture the extended or prolonged weather variable's impact. Similarly, to capture the effect of waterlogging and water flow due to rainfall on the target road segment's traffic stream variables, the input should also consider the rainfall value (intensity) on the other road segments. We propose the soft spatial threshold mechanism to limit the number of weather stations to consider for the target road segment. The model's input should consider the weather data from all the stations in the geographical area that are threshold distance apart. The input should also include the weighted average of weather readings for all the other stations where distance is more than the threshold.

The learning model takes the input traffic and weather data for short-term traffic stream variables prediction on the target road segment. The model should extract the hidden features from the input data. It should consider the non-linear and complex transportation network characteristics during adverse weather conditions. The model should depend upon the learned features instead of hand-crafted features for optimal performance because hand-crafted features require prior domain knowledge. Traffic conditions on the road segment during adverse weather conditions depend upon various hidden features. Therefore, deep learning models are an optimal choice because their multi-level architecture extract hidden influential features in data from a lower level to a higher level. Deep learning models are robust and don't require hand-crafted features for optimal performance. The traffic condition on the target road during adverse weather conditions has a spatiotemporal dependency on the other road segments. Also, the evolution of future traffic conditions is always related to traffic conditions in current and past time instances. Therefore deep learning models that extract the spatiotemporal correlation and consider sequence learning are favorable choices as the learning model.

The Convolutional Neural Network (CNN) model is known for extracting local connections between the pixels from the one-dimensional, and two-dimensional data [30]. In our problem, the traffic condition at the target road segment is dependent on the traffic and weather conditions on the other road segments. Therefore, CNN is a suitable model to extract the spatiotemporal features between the target and other road segments. The spatiotemporal traffic and weather data for past and current time instances are provided as the input to the CNN for regression

prediction.
In a time-series prediction or sequence learning problem, the future outcome is dependent on the current and past observations, i.e., the output from the current timestamp is provided as the input to the next timestamp and so on. The deep learning model designed for the sequence learning problem is the Recurrent Neural Network (RNN) and it's variant Long Short Term Memory (LSTM). The RNN uses memory cells to capture the temporal information from the previous sequence and uses this information along with the current input for future prediction [65]. RNN suffers from the vanishing gradient problem and doesn't hold long temporal dependencies. LSTM solves this problem with the help of a gating mechanism and, therefore, holds long and short temporal dependencies [18]. The spatiotemporal traffic and weather data are used as the input to the recurrent learning models for traffic stream variables prediction.

The traffic conditions on the target road segment are affected by the surrounding area's traffic and weather conditions and traffic conditions of its own. The traffic data sequence of the target road segment contains hidden features like road condition, peak hour traffic behavior, and point of interest influence. All these features affect the traffic condition on the target road irrespective of the traffic condition on the other road segments. Therefore, the traffic stream variables prediction during adverse weather conditions on the target road has the spatiotemporal dependency on other road's traffic and weather conditions and temporal dependency on its own traffic characteristics. LSTM model learns the effective temporal features to predict the traffic stream variables on the target road segment based on the past and the current traffic data sequence. The hybrid models, CNN-LSTM and LSTM-LSTM are designed to extract the spatiotemporal and temporal dependency. There are two submodels in our hybrid model, one which extracts spatiotemporal features, and the other uses the spatiotemporal features and temporal input to predict the traffic stream variables. In the LSTM model, the hidden layer is a memory block, where the cell memorizes the temporal features and passes this information to the next cell state. In LSTM memory cells, the current cell state depends upon the previous cell state and the current input [66].

This cell state is initialized randomly during model training. In our model, we initialize the memory or the cell state of LSTM with the spatiotemporal features extracted by the CNN or LSTM model. Our latter LSTM model learns the hidden features, which depend on the spatiotemporal features of the adjacent road's traffic and weather conditions and its temporal features. These deep learning models predict the traffic stream variables with the objective that the difference between the predicted and actual value should be minimum. Accurate prediction of traffic stream variables is an integral component in the ITS applications.

4.2 Proposed Model

This section discusses the solutions to overcome the limitations related to the input traffic and weather data.

Fine-grained realistic synthetic traffic data is required to overcome the traffic data sparsity issue. The impact of rainfall on the traffic stream variables differs for different road types. The broad and freeway road segments are the least affected, whereas narrow and street roads are the most affected. To generate the synthetic traffic data considering the impact of rainfall on different roads, there is a need to study the impact of rainfall on different road segments. Section 4.2.1 discusses the input traffic and weather data and the impact of rainfall on various roads. Section 4.2.2 discusses the synthetic data generation method. The prolonged impact of weather variables is significant on the traffic stream variables. To accurately predict the traffic stream variables, weather data's spatial and temporal significance needs to be taken into consideration. Section 4.2.3 discusses the soft spatial and temporal mechanisms and weather data interpolation techniques. Section 4.2.4 shows the spatiotemporal representation of the input data.

4.2.1 Input Data

In our model, traffic and weather data are used as an input to predict the traffic stream variables at the prediction horizon. There are two types of traffic data sources, ITS infrastructure (primary) and alternate source (secondary). The traffic data is collected from the road segment. There are different types of roads in the transportation system. We study various roads to observe the impact of inclined weather conditions on the traffic stream variables. The roads are classified into four categories, namely, arterial, sub-arterial, collector, and local, based on their function and capacity. The arterial roads are the high-capacity urban roads. They are the major connecting roads across the country designed for high-speed uninterrupted travel, for example, expressways or freeways. The sub-arterial roads connect the traffic from collector roads to freeways by carrying large traffic volumes, such as state highways. The collector roads connect traffic from the local roads to the sub-arterial road. These roads are mostly passing through the city area and occupy significant city traffic. Local roads are mainly the neighborhood street system. These roads mainly carry the local traffic. Section 4.2.1.1 discusses the traffic data from the ITS infrastructure. Section 4.2.1.2 discusses the traffic data from the alternate source. Section 4.2.1.3 discusses the weather data studied in this research.

4.2.1.1 Traffic Data from ITS Infrastructure: Traffic Sensors

Traffic data corresponds to the ITS infrastructure is collected from the Caltrans Performance Measurement System (PeMS) [67]. Caltrans (California Department of Transportation) requires a freeway performance measurement system that extracts information from the real-time and historical traffic data. PeMS provides traffic information that helps various stakeholders such as traffic engineers, planners, freeway users, researchers, and traveler information service providers. A loop detector is used for measuring traffic flow and occupancy. Two single-loop detectors are placed nearby forms a double-loop detector, which directly measures traffic speed. These loop detectors are wire loops embedded in each lane of the roadway at regular intervals on the network; every half-miles away. A road segment in Caltrans traffic data is a freeway segment that contains a loop detector(s) [68]. PeMS aggregates data at the 5-minute interval and provides value for average speed and vehicle count per 5-minute. There are 12 districts in California, data from San Diego (district 11), along I-5, SR-163, and SR-75, are used in this research. Figure 4.2 shows the location of these roads on a map.

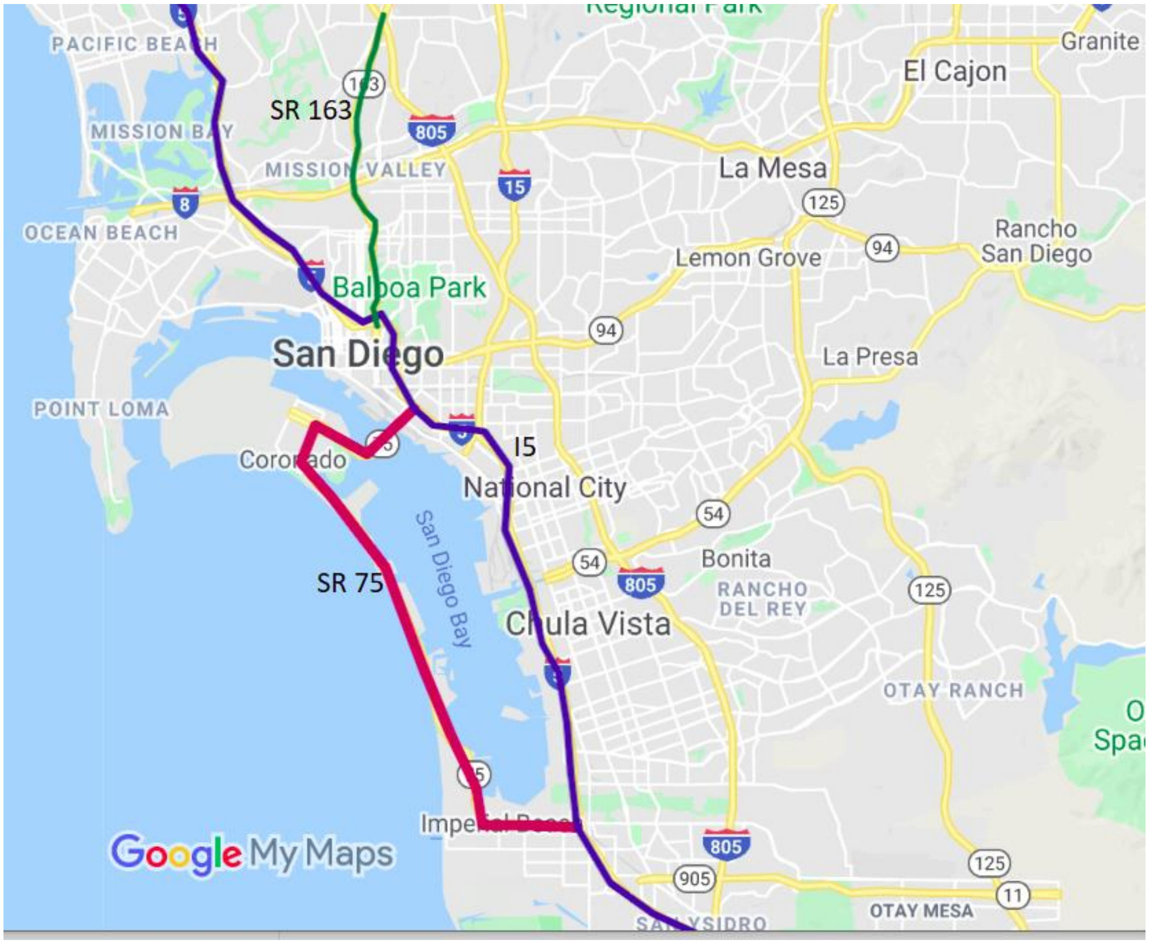


Figure 4.2: Selected Road Stretches of San Diego.

I-5 is a 168 km long arterial road in the San Diego, 124 loop detectors are located on the North headway. It is a 12 lane road with six lanes on each headway (North, South). SR-163 is a 17.8 km long road, and eleven loop detectors are located on the North headway. It is an eight-lane road, four lanes on each headway. SR-75 is a 21 km long road, and eighteen loop detectors are located on the North headway. It is a five-lane road with three lanes on the North headway and two lanes on the South headway. Input features in traffic data are "timestamp", "latitude", "longitude", "road segment", "average speed", "average flow", and "average occupancy" aggregated from all lanes on the single headway, "average speed", "flow", and "occupancy data" individually across all the lanes. The list of geometric features of the roads is shown in table 4.1.

Table 4.1: List of Geometric Features of the Roads.

Road	Lanes	Length	Number of Loop Detector	Data Attribute
I-5 (Arterial)	12, six lanes on each headway (North, South)	168km	124	Timestamp, latitude, longitude, road segment, average speed, average flow, and average occupancy aggregated from all lanes on single headway, average speed, flow, and occupancy data individually across all lanes.
SR-163 (Sub-arterial)	8, four lanes on each headway	17.8km	11	
SR-75 (Sub-arterial)	5, three lanes on North headway and two lanes on South headway	21km	18	

We selected the three different roads based on their characteristics. For the I-5, we use the traffic data corresponds to the road passing through the countryside (Del Mar to Pacific Beach); the elevation is 70-119 m. The part of I-5 passing through the Downtown area between the El Camino Real to the Naval Base, we called it I-5D for this research, the elevation is 41-72 m. We selected the Downtown area of San Diego because the roads passing through the densely populated areas are much affected by the rainfall as compared to the countryside area. I-5 is the broad road compared to the SR-163 and SR-75. The SR-163 passing through the routes Cabrillo to Miramar and SR-75 passing through the routes Coronado to Imperial Beach is considered in this study. The elevation of SR-163 is between 28-53 m and SR-75 is between 4-13 m. Peak hour is the time of the day when most people commute such that traffic flow is at its highest. The peak hour duration on these roads is from 6:00 AM to 10:00 AM and from 3:00 PM to 6:00 PM.

4.2.1.2 Traffic Data from Alternate Source: Taxi GPS

The PeMS requires a system cost of about $3.5 million a year considering deployment and maintenance cost of hardware and software (in 2013) [24]. ITS infrastructure deployment and maintenance are costly. Therefore, the majority of the roads worldwide have limited ITS infrastructure. To manage traffic on roads with limited ITS infrastructure, stakeholders use alternate traffic data sources such as cellular data, taxi GPS, etc. The traffic data from Mumbai corresponds to alternate sources such as taxi GPS are obtained from the Uber Movement website [29]. Uber is a widely used cab service across multiple cities in India. The Uber movement website provides the aggregate traffic data that consist of the road segment, average speed, and travel time computed using the anonymized Uber trips. The data is aggregated daily for a road segment. We use the traffic data of Borivali East to Parel (BEP) and Mulund East to Parel (MEP) roads. Figure 4.3 shows the chosen roads on the map of Mumbai.

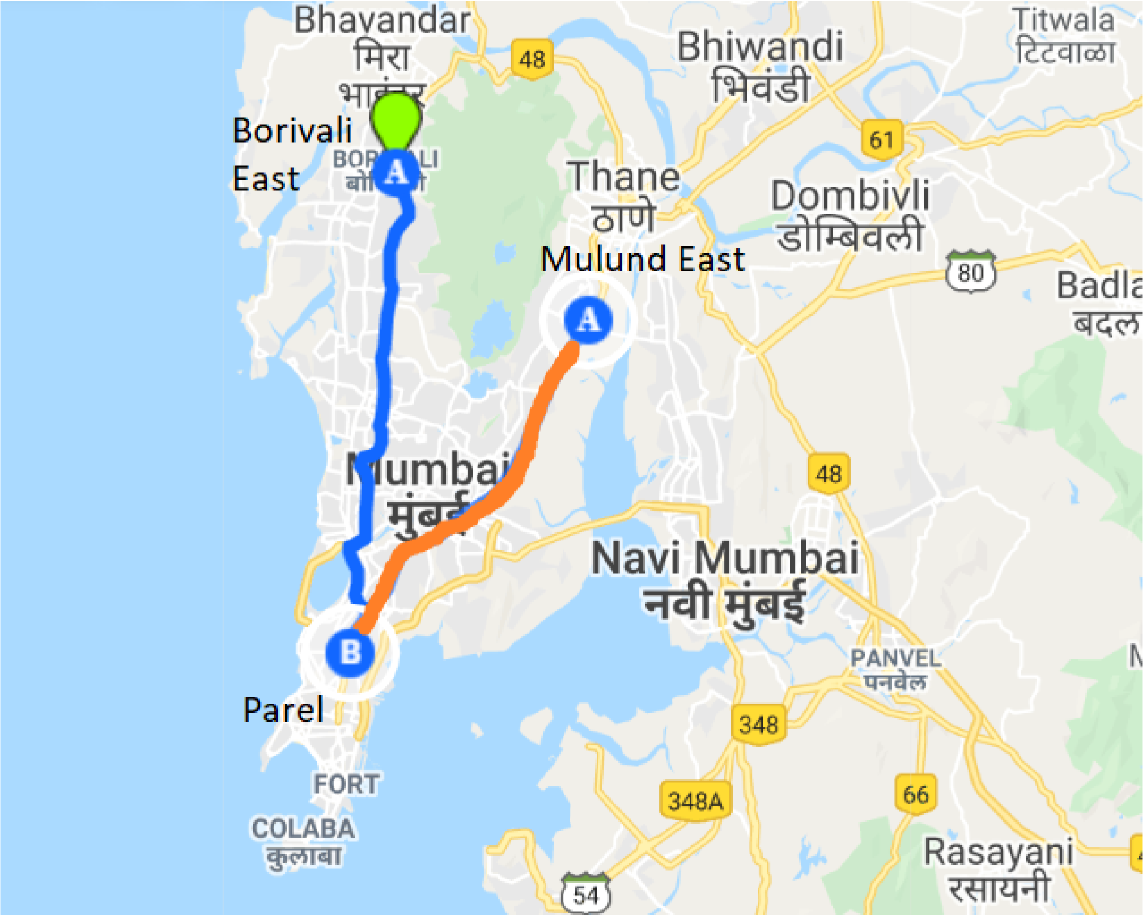


Figure 4.3: Selected Road Stretches of Mumbai.

The BEP road is a 29.1 km long collector road with six lanes (three lanes in each headway). The elevation of the BEP road is between 12-15.8 m. The MEP is a 24.5 km long collector road with four lanes (two lanes in each headway). The elevation of the MEP road is 10-13.4 m. BEP and MEP both pass through the densely populated area. The peak hour duration on these roads is from 7:00 AM to 11:00 AM and from 4:00 PM to 7:00 PM on weekdays.

4.2.1.3 Weather Data

To study the impact of rainfall on traffic conditions of different road types, we use the traffic and the rainfall data of San Diego (district 11), California, and Mumbai, India. The San Diego traffic data is described in section 4.2.1.1. San Diego's rainfall data is available from the National Climate Data Center (NCDC-NOAA)

[69]. The NCDC-NOAA aggregates the weather data every fifteen minutes and records timestamp, latitude, longitude, station id, and rainfall value. There are twenty-two weather stations in San Diego that provide rainfall information. The monthly rainfall data of San Diego is shown in figure 4.4. This study uses the traffic and weather data of duration 21 January 2019 to 16 February 2019. The annual rainfall in the year 2019 was 403.7mm.

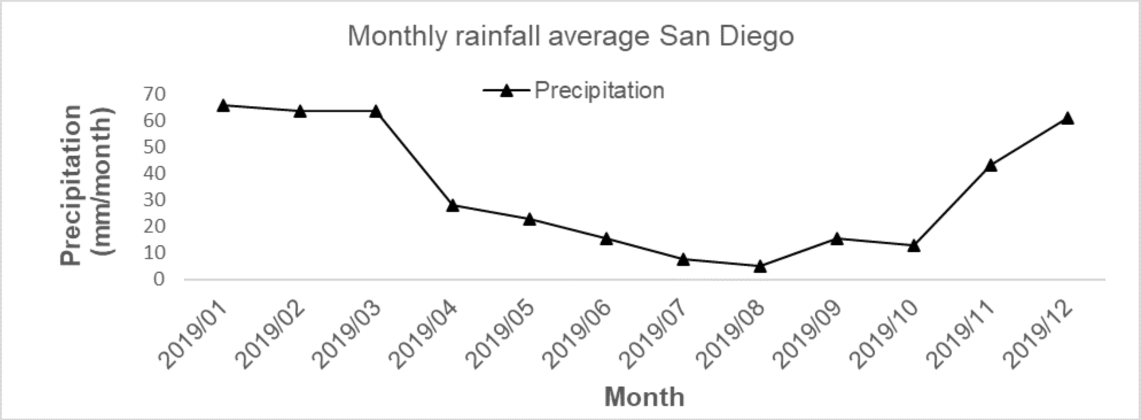


Figure 4.4: Monthly Rainfall in San Diego (Year 2019)

The Mumbai traffic data is described in section 4.2.1.2. The rainfall data of Mumbai are provided by the Maharashtra Remote Sensing Application Centre (MRSAC) [70]. There are 60 automatic rain gauges at 58 locations in Mumbai, and periodically update the data to MRSAC every fifteen minutes. The data consists of a timestamp, station id, latitude, longitude, and rainfall value. The MRSAC aggregates the rainfall data daily, weekly, and monthly. We use traffic and weather data from January 2016 to December 2017. The annual rainfall in these years was 2500 - 2600 mm. Figure 4.5 shows the monthly rainfall data of Mumbai in the year 2017.

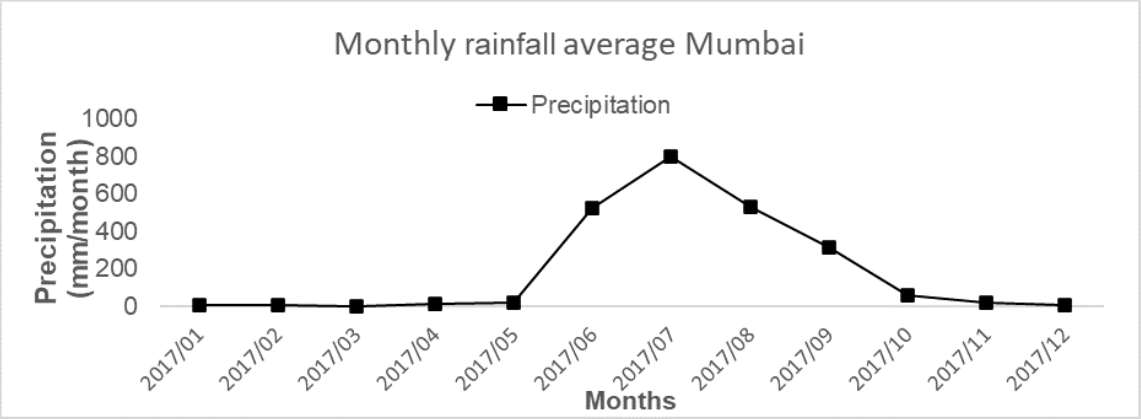


Figure 4.5: Monthly Rainfall in Mumbai (Year 2017).

4.2.2 Synthetic Data Generation Method

One of our objectives is to generate synthetic data that provides an abstraction of a real-world traffic condition during adverse weather conditions to overcome the traffic data sparsity issue. The impact of rainfall on different roads is studied. We proposed an empirical model to generate synthetic data considering the impact of rainfall on traffic stream variables. In section 4.2.2.1 we describe and analyze the traffic and the weather data for different road types under diverse traffic and rainfall conditions; the section 4.2.2.2 presents the empirical model based on the traffic and weather data; section 4.2.2.3 elaborates the simulation setup.

4.2.2.1 Analysis of Traffic and Weather Data

The impact of rainfall is different for various roads. Roads that are broad and have excellent drainage systems are less affected than the roads that are narrow and have a poor drainage system. A brief description of the roads considered in this study is outlined in table 4.2.

Table 4.2: Types of Roads Considered in This Study.

Road Type	Description	Road Stretch
Arterial Road	A high capacity limited access road, provides uninterrupted high speed travel	I-5
Sub-arterial Road	A medium capacity urban road, connects collector road(s) to expressway, less traffic than arterial road	SR-163, SR-75
Collector Road	A low capacity road, connects traffic from local street(s) to sub-arterial and arterial roads	Borivali East to Parel (BEP) and Mulund East to Parel (MEP)

Table 4.3 describes the impact of rainfall on traffic movement in various cases. The traffic movement on the arterial roads is affected by C1 - C3, whereas that on collector roads is affected by all C1 - C4.

Table 4.3: Rainfall Impact on Traffic Movement.

Cases	Description	Affected Roads
C1	Drivers reduce their speed to avoid collision due to aquaplaning, small patches of standing water on road, and slippery road	I-5, I-5D, SR-163, SR-75, BEP, MEP
C2	Drivers reduce their speed to avoid collision due to reduced visibility	I-5, I-5D, SR-163, SR-75, BEP, MEP
C3	Drivers avoid travel at night under heavy rainfall to avoid crash risk, reducing the flow and speed on road	I-5, I-5D, SR-163, SR-75, BEP, MEP
C4	Waterlogging on road (due to heavy rain and water-flow from flooded neighbor roads) affects traffic flow and speed	BEP, MEP

As rainfall is a local phenomenon [71], we associate every road segment with one or more neighboring weather stations. This association is determined by computing the correlation between a road segment's traffic data and the weather data of a weather station. The correlation coefficient is calculated for every pair of road segments and weather stations.

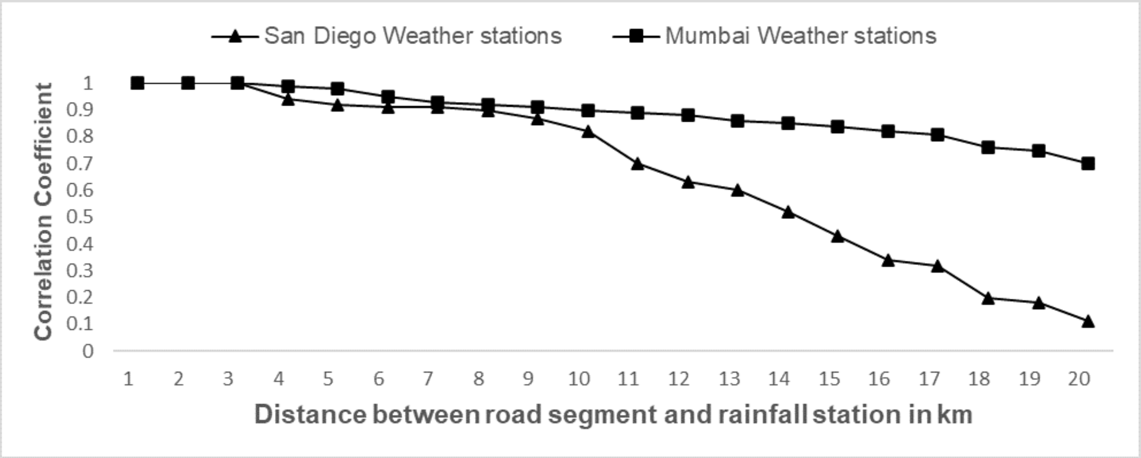


Figure 4.6: Correlation Coefficient of Weather Data versus Distance of Weather Stations.

Figure 4.6 shows the effect of distance between the road segment and weather station on the correlation coefficient. It is observed that the correlation between the traffic and the weather data weakens as the distance increases. Hence, we consider the weather stations within proximity (a distance of three-five kilometers) from a road segment. The distance of three-five kilometers is in line with the past studies conducted by M. Agarwal *et al.* [22] and Tsapakis *et al.* [71]. For this distance, the correlation coefficient between the traffic data and the weather data is within the range of 0.8 - 1.0 (figure 4.6). Table 4.4 shows the category of rainfall based on the rainfall intensity, these values, and category (low-high) is similar with literature studies [21, 22]. We are interested in the prolonged impact of rainfall. Therefore, we introduce the category "extended" for prolonged heavy rainfall.

Table 4.4: Rainfall Category Based on Rainfall Intensity.

Rainfall category	Rainfall intensity
Light	$\leq$ 0.5 mm / hr
Moderate	$\leq$ 5 mm / hr
Heavy	$\geq$ 7 mm / hr
Extended	$\geq$ 40 mm in two hours

Based on the traffic and weather data of San Diego, the table 4.5 and 4.6 shows the percentage decrease in traffic speed (km/hr) and flow (vehicle-count / 5 min), respectively, due to rainfall. The aggregated traffic speed and flow information during the normal day is used as a base condition to compute the percentage decrease in the traffic stream variables during rainfall. It is observed that the light and moderate rainfall (less than 5 mm/hr) shows very less impact on the traffic speed and flow on the expressway, e.g., on I-5, decrease in speed and flow is less than 5% and 4%, respectively. The impact is relatively high on the sub-arterial roads; e.g., on SR-75, the decrease in speed and flow is greater than 8% and 7.5%, respectively. The heavy rainfall (greater than 7 mm/hr) significantly affects the traffic speed and flow on all the roads, and the decrease in speed and flow is 13 - 21.5% and 13.5 - 27.5%, respectively.

Table 4.5: Percentage Decrease in Traffic Speed (km/hr) as per San Diego Data.

Rainfall	I-5	I-5D	SR-163	SR-75
light($\leq$0.5mm/hr)	1.2	2.39	2.02	2.66
moderate($\leq$5 mm/hr)	4.84	6.43	7.74	8.17
heavy($\geq$ 7mm/hr)	13.02	14.31	16.12	17.57
$\geq$ 10mm/hr	14.09	15.12	17.92	19.10
$\geq$ 40mm in 2hr	15.77	16.03	19.59	20.87
$\geq$ 60mm in 4hr	16.03	17.45	20.24	21.46

Table 4.6: Percentage Decrease in Traffic Flow (vehicle-count / 5 min) as per San Diego Data.

Rainfall	I-5	I-5D	SR-163	SR-75
light($\leq$ 0.5mm/hr)	1.32	1.65	1.83	1.94
moderate($\leq$ 5mm/hr)	3.60	4.20	6.96	7.65
heavy($\geq$ 7mm/hr)	13.88	14.66	16.51	16.64
$\geq$ 10mm/hr	14.24	17.78	17.87	19.82
$\geq$ 40mm in 2hr	17.80	19.74	21.38	23.26
$\geq$ 60mm in 4hr	19.44	21.8	24.35	27.63

Based on the traffic and weather data of Mumbai, table 4.7 shows the percentage decrease in the traffic speed (km/hr) on Borivali East to Parel (BEP) and Mulund East to Parel (MEP) roads due to rainfall. Both BEP and MEP are the collector roads. The impact of prolonged heavy rainfall (greater than 7 mm/hr) on these roads is significant, and the decrease in speed is 17-63%.

Table 4.7: Percentage Decrease in Traffic Speed (km/hr) as per Mumbai Data.

Rainfall	BEP	MEP
light($\leq$ 0.5mm/hr)	2.63	2.5
moderate($\leq$ 5mm/hr)	10.36	7.15
heavy($\geq$ 7mm/hr)	20.21	17.13
$\geq$ 10mm/hr	26.87	21.24
$\geq$ 40mm in 2hr	33.38	29.26
$\geq$ 60mm in 4hr	38.34	32.52
$\geq$ 100mm in 6hr	63.23	53.3

We compare our observations about the impact of rainfall on traffic data with those reported in the literature. Table 4.8 and 4.9 shows the comparison between the existing work and our observations for different roads. The effect of prolonged rainfall on the traffic stream variables is not reported in the literature. Hence, we exclude our readings related to the prolonged rainfall during the comparison. On the arterial road, we observe that the traffic speed and flow decrease by 1.2 - 13%

and 1.3 - 13.88%, respectively, for the light to heavy rainfall. These numbers match those reported by Ibrahim *et al.* [21] and M. Agarwal *et al.* [22]. They observed a decrease of 2 - 14% and 1.6-16.5% in the traffic speed and flow, respectively, in their studies. On the sub-arterial road, we observe the reduction of 2.02 - 16% and 1.9 - 16.4% in the traffic speed and flow, respectively, for the light to heavy rainfall. These numbers are similar to those reported by Wang *et al.* [72]. They observed a decrease of 1.72 - 16% and 2 - 16.5 % in speed and flow, respectively. To the best of our knowledge, no study in the literature has analyzed the effect of rainfall on the collector roads. On these roads (MEP and BEP), we observe the traffic speed reduction of 2-20% for the light to heavy rainfall.

Table 4.8: Comparison of Decrease in Traffic Speed Due to Rainfall Observed by the Existing Work and This Study.

Existing Work	Percentage Decrease in Traffic Speed		
	Arterial road	Sub-Arterial road	Collector road
Ibrahim *et al.* [21]	2 - 14%		
M. Agarwal *et al.* [22]	2 - 16.5%		
Wang *et al.* [72]		1.72 - 16%	
This Study	1.2 - 13%	2.02 - 16%	2 - 20%

Table 4.9: Comparison of Decrease in Traffic Flow Due to Rainfall Observed by the Existing Work and This Study.

Existing Work	Percentage Decrease in Traffic Flow	
	Arterial road	Sub-Arterial road
Angel *et al.* [73]	1.6 - 14.2%	
Wang *et al.* [72]		2 - 16.5%
This Study	1.32 - 13.88%	1.94 - 16.4%

4.2.2.2 Empirical Model Design

The rainfall significantly affects the traffic stream variables, as observed in section 4.2.2.1. The traffic speed and flow are affected by various parameters during rain, such as change in driving behavior to avoid crash risk, visibility, road capacity reduction, waterlogging, aquaplaning, etc. To calculate the traffic speed during rainfall, there is a need to design a model that captures most of these parameters. Weber *et al.* [74] introduced the friction coefficient (μ) to represent the effect of road surface conditions on the traffic speed. The friction coefficient is varied to capture the impact of rain and snowfall on the vehicle's speed. Table 4.10 shows how the friction coefficient changes with the road surface conditions.

Table 4.10: Road Surface Condition and Friction Coefficient.

Road Surface Condition	Friction Coefficient
Dry	$\mu \approx 1.0 - 1.2$
Moist	$\mu \approx 0.8 - 1.0$
Wet	$\mu \approx 0.5 - 0.8$
Snow	$\mu \approx 0.3 - 0.6$
Ice	$\mu \approx .1 - .4$

We generalize the friction parameter μ to represent the wet surface's impact on the mobility parameters. It is observed that the water flows from one road segment to another during prolonged rain and affects the traffic stream variables on the target road segment. Similarly, the traffic stream variables are affected by waterlogging caused by prolonged heavy rainfall. We introduce the rainfall parameter α to represent this phenomenon. The α is defined as the weighted average of rainfall intensities values, and it is normalized to have a value between 0 and 1. In the case of light rainfall, the value of friction parameter μ is high, and the rainfall parameter α is low, whereas for the prolonged heavy rainfall, the value of μ is low and α is high. It is observed that the impact of prolonged rainfall varies for different types of roads. The expressways where no water accumulates are the least affected. In contrast, the narrow roads, passing through the densely populated area, and having a weak drainage system, are the most affected. Hence, we introduce a road-type parameter R to consider different types of road. The value of R is low for the roads that are least affected by the rainfall, whereas the value of R is the highest for the roads that are the most affected by rainfall and waterlogging. For an expressway with an excellent drainage system where waterlogging does not occur, the value of R is very low. For a sub-arterial road that is less prone to waterlogging, the value of R is moderate. For the collector roads, the value of

R is the highest as they are the most vulnerable to waterlogging.

The traffic speed u_w is defined as the linear equation of these three parameters (μ, α, and R), and the least-square method is used to find their coefficient. In the matrix notation, a system of linear equations is written using the coefficient matrix A, the parameter vector x, and the result vector b. The size of matrix A is $s \times k$, and that of vectors x and b is $k \times 1$ and $s \times 1$, respectively. In this case, $k = 3$ as we are working with three parameters μ, α and R and s is the sample size.

$$A_{s\times k}x_{k\times 1} = b_{s\times 1}$$

The objective of the least-square estimation is to minimize the difference between the estimated speed and the actual speed (defined as the norm), as shown in equation (4.1), where i is the row, and j is the column.

$$\min_{x\in\mathbb{R}^s} \|b - Ax\| = \sqrt{\sum_{i=1}^{s}\left(b_i - \sum_{j=1}^{k} a_{ij}x_j\right)^2} \tag{4.1}$$

Table 4.11: Least Square Estimation of Speed Based on the Friction Parameter (μ), Rainfall Parameter (α), and Road Type (R), Where $\mu \in [0,1]$ and $\alpha \in [0,1]$.

Road	Traffic Condition	Traffic Speed Considering Weather Condition (u_w)
Arterial	Peak hour	$u_w = -10.815 * \alpha + 16.319 * \mu + 525.042 * R, \quad R \in [0 - 0.1]$
Arterial	Off-peak hour	$u_w = -5.744 * \alpha + 20.054 * \mu + 504.079 * R, \quad R \in [0 - 0.1]$
Arterial	Night time	$u_w = -17.565 * \alpha + 9.609 * \mu + 587.656 * R, \quad R \in [0.1 - 0.2]$
Sub-arterial	Peak hour	$u_w = -15.527 * \alpha + 21.085 * \mu + 90.095 * R, \quad R \in [0.5 - 0.6]$
Sub-arterial	Off-peak hour	$u_w = -20.619 * \alpha + 21.804 * \mu + 87.834 * R, \quad R \in [0.5 - 0.6]$
Sub-arterial	Night time	$u_w = -29.703 * \alpha + 18.38 * \mu + 94.66 * R, \quad R \in [0.6 - 0.7]$
Collector	Peak hour	$u_w = -39.571 * \alpha + 9.218 * \mu + 56.613 * R, \quad R \in [0.8 - 1]$
Collector	Off-peak hour	$u_w = -33.608 * \alpha + 14.832 * \mu + 52.158 * R, \quad R \in [0.8 - 1]$
Collector	Night time	$u_w = -39.278 * \alpha + 5.812 * \mu + 58.224 * R, \quad R \in [0.8 - 1]$

Table 4.11 shows the linear equations for estimating traffic speed using the weather parameters μ, α, and R for various types of roads (arterial, sub-arterial, collector) and traffic conditions (peak hour, off-peak hour, and night time).

We represent the road traffic condition as a ternary variable that takes the values 0 - peak hours, 1 - off-peak hours, and 2 - night hours. The few road network parameters such as maximum speed on edge, road type of an edge remain fixed over time. Whereas the traffic condition, rainfall parameter α, friction parameter

μ varies with time.

4.2.2.3 Simulation Setup

A simulation model provides an abstraction of the real system and allows the user to generate and analyze the results that correspond to an interesting real-world problem. The results obtained using the simulation study are useful only if the simulation model is close to a real-world scenario. Simulator for Urban MObility (SUMO) [31] is an open-source, highly portable, microscopic, and continuous road traffic simulator designed to handle large-scale road networks. SUMO includes many applications that provide the user detailed microscopic traffic simulation model. The vehicle parameters (vehicle type, acceleration, length, etc.), road network parameters (number of lanes, length, maximum speed, etc.), vehicle route and flow per hour, traffic light, etc., are easily configurable. A major limitation of SUMO is that it doesn't include weather parameters (rain, wind, snow, fog, etc.) that influence the traffic stream variables in the real-world scenario.

In SUMO, each vehicle is modeled explicitly, has its own route, and moves individually. A car-following model describes the interaction of a vehicle with surrounding vehicles in the road network. It determines the speed of a vehicle in relation to the vehicle ahead of it. It illustrates that the following vehicle is always trying to keep a safe distance from the leading vehicle. The following vehicle always adapts to the deceleration behavior of the leading vehicle [75]. Weber *et al.* [74] modified the Krauss car-following [75] model to access the friction parameter to derive vehicle's speed. In the Krauss car-following model, the safe speed of a vehicle v_{safe} is computed as follows:

$$v_{safe} = v_l(t) + \frac{g(t) - v_l(t)(t_r)}{\frac{v_l(t)+v_f(t)}{2b} + t_r} \tag{4.2}$$

where $v_l(t)$ and $v_f(t)$ represent the speed of leading and following vehicle speed, respectively, at time t. The $g(t)$ is the gap between leading and the following vehicle at time t, t_r is the driver's reaction time, and b is the maximum deceleration of the following vehicle. Sometimes v_{safe} is greater than the speed limit of the road or the maximum speed that the vehicle can reach due to its acceleration power. Therefore, the desired speed v_{des} of the vehicle is computed as the minimum of these speeds as follows:

$$v_{des} = min\{v_{max}, v + a_t, v_{safe}\} \tag{4.3}$$

where v_{max} is the maximum vehicle speed which is the minimum of the speed limit on the road segment and the vehicle's maximum speed attribute, v is current vehicle speed, a_t is the vehicle acceleration at time t. At the next time step, the vehicle moves at speed v_{des} to avoid collision with the front vehicle. This process is repeated at every time step to update the vehicle's movement speed.

We modify the Krauss car-following model [75] to add weather-related parameters and adapt the vehicle movement. This modified model uses the friction parameter (μ), rainfall parameter (α), and road type (R) for a particular time instance to derive vehicle speed. The vehicle speed u_w considering the weather condition is computed using linear equations described in table 4.11. We use u_w to compute the permissible acceleration of a vehicle at time t (a_t) based on the difference between u_w and its current speed. The value of a_t is passed in equation (4.3) to calculate the desired speed v_{des}.

In SUMO version 1.4.0, it is possible to define one or more generic parameters for the edges and the lanes in a road network. These parameters can be adapted online while the simulation is running either through Traffic Control Interface (TraCI) [76] or via an additional file where the temporal values of these parameters are specified. We use the TraCI to associate the weather parameters (μ, α, and R) with every lane to derive vehicle speed. The parameters can also be introduced as additional attributes of the lane elements of the road network. An XML file contains a specific value for all the attributes which can be set as the key-value pair Friction =$'' < DOUBLE > ''$, Rainfall =$'' < DOUBLE > ''$, and Road =$'' < DOUBLE > ''$ within respective tags. To ensure backward compatibility, the default value of friction parameter (μ) is set to 1.0, the value of rainfall parameter (α) is set to 0.0, and road type parameter (R) is set to 0.0. To directly manipulate the network files, these parameters can also be set in the NETEDIT file. Appendix A discusses the sample code for the TraCI and modified Krauss car-following model.

4.2.3 Impact of Weather Conditions on Traffic Stream Variables

The traffic stream variables prediction on the target road segment is affected by the traffic and the weather conditions on the adjacent road segments. For example, during prolonged rainfall water flows from one road segment to other road segments and affects its traffic stream variables. Also, the correlation between the traffic and the weather data is not directly dependent on the distance between them due to terrain feature constraints. Similarly, prolonged heavy rainfall or snowfall significantly affects the traffic stream variables. We discuss the weather data's spatial and temporal significance on the traffic stream variables in section 4.2.3.1 and 4.2.3.2, respectively. We discuss the weather data spatial interpolation techniques in section 4.2.3.3.

4.2.3.1 Soft Spatial Threshold

Traffic stream variables prediction in the presence of adverse weather conditions is a challenging problem. Changes in the driving patterns to avoid collision risk, poor visibility, waterlogging cause road capacity reduction, unforeseen events due to adverse weather conditions, such as trees fall, which causes road closure, and slippery roads cause an accident. All these factors affect the traffic stream variables on the road segment. The traffic and weather conditions on the target road segment have the spatiotemporal dependency on the other road segment's traffic conditions. For example, prolonged rainfall causes massive waterlogging on road segments where the drainage system is weak and the surroundings are densely populated. In an urban area, waterlogging or flash flood is caused due to inadequate city planning, weak drainage system, over-construction, encroachment, etc.; this will cause the water flow from one road segment to another. The roads that are not adjacent to each other, but due to the terrain feature, the water flows from the distant road segments cause a flash flood or waterlogging at the target road segment. Therefore, weather data from the weather stations near the target road segment will not capture the impact of rainfall on other road segments. If we have data corresponds to these features such as water flow path (hydrology map), waterlogging sites, drainage maintenance, soil data, etc. We will use the data to estimate the impact of rainfall on the road segment's traffic stream variables. For the majority of the location, these datasets are not available. Therefore for an accurate prediction of traffic stream variables on the target road segment during rainfall, the effect of rain on the other road segments needs to be captured. In the literature, the impact of rainfall on the traffic stream variables on a road segment is captured using the interpolation method [25, 26]. The rainfall value from the nearest station is used as point data for the road segment. Multiple interpolation techniques were proposed in the literature [77, 78], where the weight depends upon the distance between the road segment and the weather stations. The impact of rainfall on traffic stream variables depends upon various dynamic parameters. Therefore, distance-based interpolation techniques to provide point weather data on the road segment will not capture the dynamic parameters.

There are two ways to solve this problem; either use the weather data from all the stations in the city or define a threshold distance for the weather station selection. If there is no computational limitation, we can use data from all the stations in the city, but using this will increase the number of parameters. Similarly, there is no fixed threshold value to decide the relevant number of weather stations for the target road segment because various dynamic parameters are involved. Therefore, instead of defining a hard threshold on the number of weather stations relevant to the target road segment, we propose a soft threshold mechanism. The number of stations is selected based on the threshold distance value, and for all other stations (falling out of the threshold distance), we are using the weighted average of the weather variable's value. The weight is dependent on the distance between the target road segment and the weather station. The threshold value is set as a hyperparameter, the parameters that are not learned but need to be fixed before training the learning model such that the model gives a minimum error for a particular value.

There are total M weather stations in a geographical location. For a target road segment i where the traffic stream variables need to be predicted, we take into account k weather stations (that are under threshold distance Δ). For other $M-k$ stations, we apply the weighted average. The equation (4.4) shows the weighted average corresponds to the soft spatial threshold.

$$\hat{W}_t = \sum_{j \in \{M-k\}} \beta_j W_t^j \tag{4.4}$$

$$\beta_j = \frac{\frac{1}{dist_{i,j}}}{\sum \frac{1}{dist_{i,j}}} \qquad \text{such that} \quad \sum \beta_j = 1$$

Where $\hat{W}_t$ is the weighted average from the $M-k$ weather stations at time t, W_t^j is a weather reading from j^{th} station at time t, β_j is the weight factor, and $dist_{i,j}$ is the distance between target road segment i and j^{th} weather station.

4.2.3.2 Soft Temporal Threshold

The prolonged effect of rainfall has a significant impact on the traffic stream variables of a road segment. For example, due to prolonged heavy rainfall, the soil absorption and drainage capacity become weak; therefore, even the moderate continuous rainfall will significantly affect the traffic stream variables. After prolonged rainfall, roads that are prone to waterlogging, take time to clear waterlogging. Suppose data corresponding to different features is available like waterlogging sites, hydrology maps, drainage systems, soil, etc. These datasets are used to extract the features to predict the traffic stream variables during adverse weather conditions. For the majority of the location, these datasets are not available. Therefore it is needed to provide the extended past sequence of weather data such that the model captures the prolonged rainfall impact on traffic stream variables.

The temporal sequence of the traffic and the weather data is used for the traffic stream variables prediction during adverse weather conditions [25, 26, 27]. We propose a soft temporal threshold mechanism to capture the prolonged impact of rainfall on the traffic stream variables. To capture the prolonged impact, with the weather data sequence we also pass the weighted moving average of the weather variable value. The weighted moving average captures the prolonged effect of the weather variable. If past values have higher significance, then more weight is assigned to them. Otherwise, current values have a higher weight. The weighting factor and sequence length are the hyperparameters, need to be tuned before model training.

A weight factor γ captures the impact of relevant factors such as soil absorption capability, drainage system quality, etc., that affects the traffic stream variables. γ is in the range [0,1]. Closer the γ to 1, more weight is given to the past rainfall value, whereas a smaller value of γ gives more weight to the current rainfall value.

$$\overline{W_t^k} = \gamma \overline{W_{t-1}^k} + (1-\gamma) W_t^k \ ; \quad \text{such that } \ 0 \leq \gamma \leq 1 \tag{4.5}$$

For station *i*, $\overline{W_t^k}$ is current rainfall value average at time *t*, $\overline{W_{t-1}^k}$ is average of past rainfall value, W_t^k is current rainfall value.

4.2.3.3 Spatial Interpolation of Weather Data

The traffic data is collected from the sensors located on the road segment, whereas the weather data on the road segment is interpolated from the nearest weather station. The aggregation period is different for both the data. Traffic data is more granular and dense compared to the weather data. Therefore, for the road segment, traffic data is more accurate compared to the weather data. If a weather station has missing weather reading for a particular period, then with limited weather data, it is difficult to ignore it. Therefore spatial interpolation technique is needed for the weather data.

Spatial interpolation is carried out by estimating the missing value based on weights of measured values. The general formula for spatial interpolation is described as:

$$W_t^i = \sum_{j=1}^{n} \Lambda_j W_t^j$$

W_t^i is the interpolated weather variable's value at the point *i*, W_t^j is the measured weather variable's value at the point *j*, *n* is the total number of measured points, and Λ is the weight contributing to the interpolation. The interpolation technique used in the literature is discussed below.

Theissen polygon The Thiessen polygon method assumes that the estimated values can take the closest station's measured values. The Thiessen polygon method is also known as the nearest neighbor method [78]. This method is best suitable for the regions where weather stations are dense. If weather stations are sparse, then due to terrain the estimated value doesn't give the correct result based on the nearest neighbor method.

Inverse distance weighted method In an inverse distance weighted method, the estimated value is replaced with the weighted measured value from the surrounding stations [77]. The opposite of the distance defines the weights. Weights are normalized such that sum of all weights equals one. The weight decreases as distance increases, the farthest stations have the lowest weight, and the nearest ones have the highest weights.

$$W_t^i = \frac{\sum_j \frac{W_t^j}{dist_{i,j}}}{\sum_j \frac{1}{dist_{i,j}}} \tag{4.6}$$

W_t^i is the estimated weather value at time t at station i and W_t^j measured weather value at time t at station j. The distance between target station i and measured station j is given by $dist_{i,j}$.

Linear regression method Linear regression method is used to find the best fit line for the data given dependent output value (predicted value) and independent input value (predictor value) [79]. In linear regression-based spatial interpolation, the best fit line is used to estimate the variable's unknown value given the other variable's measured value.

$$\breve{W}_t^i = \theta_0 + \sum_{j=1}^{M} \theta_j W_t^j \tag{4.7}$$

For each predicted $\breve{W}_t^i$ value, there is true W_t^i value. Our objective is to predict $\breve{W}_t^i$ which is in the agreement of the W_t^i, by defining an appropriate loss function Ł(θ), where θ are the parameters to be learned. Our objective is to minimize the loss between the measured value and predicted value, S is the total number of samples.

$$min\ \text{Ł}(\theta) = \frac{1}{S}\sum_{i=1}^{S}|\breve{W}_t^i - W_t^i| \tag{4.8}$$

The missing weather data is interpolated by one of the above discussed spatial interpolation techniques. The weather data from all the weather stations up to the threshold distance and the weighted distance-based average weather value capture the weather condition's spatial significance on traffic stream variables. Similarly, past weather sequences up to a particular lag value and moving weighted rainfall average value are used to capture the prolonged impact of weather conditions on the traffic stream variables.

4.2.4 Data Pre-processing

In the previous sections, we discuss the concerns related to the input traffic and weather data and the approaches used to solve them. The deep learning models learn the hidden parameters given a large amount of data. The fine-grained traffic and weather data are required as input in the deep learning model to predict the accurate traffic stream variables. In the absence of dense traffic data, we use the empirical model to generate the fine-grained synthetic data. The traffic data is granular for a road segment compared to the weather data. Weather data has a long-term impact on traffic stream variables. The soft spatial and temporal threshold will capture the spatial and prolonged impact of weather variables on a road segment. The fine-grained synthetic traffic data and weather information with spatial and temporal significance captured using soft threshold is used as an input in our model. In transportation network, traffic condition at future time instance depends upon the past and the recent traffic conditions. The traffic condition on a road segment also depends upon the traffic and the weather conditions on the other road segments. Therefore, deep learning models require the spatiotemporal input to extract this spatiotemporal dependency in the traffic and weather data. The input traffic data from n road segments for t (current) to $t - p\delta$ (past) time and input weather data from k weather stations for t (current) to

$t - p\delta'$ (past) time, makes the spatiotemporal data S_t . Here p defines the sequence length. The spatiotemporal input S_t is shown below.

$$S_t = \begin{bmatrix} X_t^1 & X_t^2 & \dots & X_t^n & \{W_t^k\} & \hat{W}_t & \{\overline{W_t^k}\} & \\ X_{t-\delta}^1 & X_{t-\delta}^2 & \dots & X_{t-\delta}^n & \{W_{t-\delta'}^k\} & \hat{W_{t-\delta'}} & \{\overline{W_{t-\delta'}^k}\} & \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ X_{t-p\delta}^1 & X_{t-p\delta}^2 & \dots & X_{t-p\delta}^n & \{W_{t-p\delta'}^k\} & \hat{W_{t-p\delta'}} & \{\overline{W_{t-p\delta'}^k}\} & \end{bmatrix}$$

Here, n and k are representing the spatial length, and p represents the temporal length and $\delta' \geq \delta$. The weather data from k station for a particular time instance is represented as, $\{W_t^k\} = \{W_t^1, W_t^2, \dots, W_t^k\}$. The weather readings from rest of the stations (other than k) is captured using $\hat{W}_t$ (soft spatial threshold). The prolonged impact of rainfall from k stations at time t is captured using weighted moving average (soft temporal threshold), represented as, $\{\overline{W_t^k}\} = \{\overline{W_t^1}, \overline{W_t^2}, \dots, \overline{W_t^k}\}$. This spatiotemporal input is passed to the learning model to extract the spatiotemporal dependency to predict the traffic stream variables accurately.

CHAPTER 5

Learning Models

The deep learning models are appropriate for traffic forecasting due to the transportation network's non-linear and complex characteristics. The model should be robust enough to deal with incomplete data and outlier. The appropriate learning model for the traffic stream variables prediction should consider the spatiotemporal dependency given the input traffic and the weather sequence for the adjacent and target road segments. We adopt the RNN, LSTM, and CNN deep learning models, that have been used to extract the spatiotemporal dependency from the input data. The traffic data sequence of the target road segment contains hidden features like road condition, peak hour traffic behavior, and point of interest influence. All these features affect the traffic condition on the target road. Therefore, the traffic stream variables prediction during adverse weather conditions on the target road segment has a spatiotemporal dependency on adjacent road segments traffic and weather conditions and temporal dependency on traffic characteristic of itself. We design the hybrid model that considers the spatiotemporal and temporal dependency between traffic and weather data while predicting the future traffic stream variables. The framework of the traffic stream variables prediction is shown in figure 5.1. Our framework has three main blocks: data preprocessing, learning model, and prediction output. The data preprocessing is discussed in the previous chapter (section 4.2.4). In the learning method block, for each selected deep learning model, we have a set of hyperparameters that can affect the results of the model's learning and the final regression prediction. The output block in the figure is the predicted output of the model on the test dataset. Section 5.1 discusses the hyperparameters required for efficient learning of the model. Section

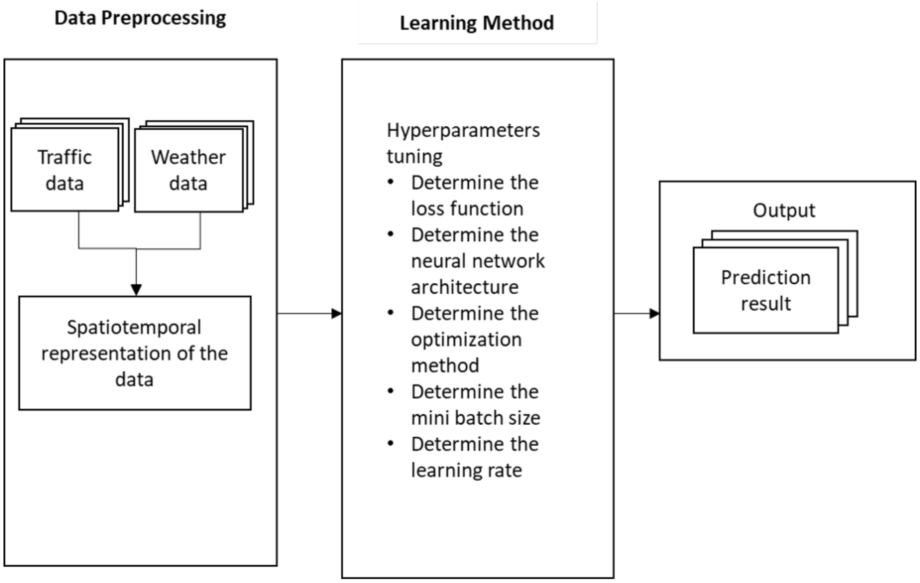

Data Preprocessing
Learning Method
Traffic data
Weather data
Spatiotemporal representation of the data
Hyperparameters tuning
• Determine the loss function
• Determine the neural network architecture
• Determine the optimization method
• Determine the mini batch size
• Determine the learning rate
Output
Prediction result

Figure 5.1: Framework for Traffic Stream Variables Prediction.

5.2.1 discusses the Convolutional Neural Network in detail, followed by sequence learning model in section 5.2.2. Section 5.2.3 discusses the hybrid CNN-LSTM model and LSTM-LSTM model.

5.1 Hyperparameters Tuning

The deep learning model is the mathematical model with a number of parameters that need to be learned from the data. By training the model with existing data, we are able to fit the model parameters. However, another kind of parameter, known as hyperparameters, cannot be directly learned from the data while training. Instead, they are usually fixed before the actual training process begins. In most cases, hyperparameters tuning is performed either by using a search-based approach or by trial and error for the given problem.

We use the random search method for hyperparameters tuning [80]. In this method, the deep learning model is evaluated for a range of hyperparameter values. The different values for each hyperparameter are provided to the random search approach. Out of the values mentioned, this approach randomly makes combinations of its own and tries to fit the dataset, test the accuracy, and return the set of hyperparameters for which model accuracy is optimal.

Some examples of model hyperparameters include:

- Determine the optimization algorithm
- Determine the learning rate of the model
- Determine the mini-batch function
- Determine the neural network structure
- Determine the loss function
- Determine the activation function

5.1.1 Determine the Optimization Algorithm, Learning Rate, and Mini-Batch Function

We use the BackPropagation and BackPropagation Through Time (BPTT) methods for model training. Section 5.1.1.1 discusses the BackPropagation and BackPropagation Through Time (BPTT) methods and section 5.1.1.2 gives insight about the optimization algorithm and also discusses the learning rate and mini-batch function.

5.1.1.1 BackPropagation and BackPropagation Through Time (BPTT) Methods

In the deep neural network, the backpropagation training algorithm is used to update the weights [81]. The main aim of this algorithm is to update the deep neural network's weights such that the difference between the actual value and the predicted value is minimized. Algorithm 1 describes the working of the backpropagation algorithm.

Algorithm 1 Backpropagation

Step 1: The training input is given to the network. It passes through the whole network to produce an output.
Step 2: The predicted outcome is compared with the actual outcome, and the error is calculated.
Step 3: The derivatives of the error are calculated with respect to the network weights.
Step 4: Weights are updated based on the optimization algorithm to minimize the error.
Step 5: Repeat the steps, till convergence.

In a recurrent neural network, at time $t = 0$ an input x_0 is feed into the network, and output y_0 is produced by the network. At time $t = 1$, an input x_1 is feed into the network and output y_1 is produce by the network. The output y_1 is calculated based on the input x_1 and the cell state from the previous timestamp. To calculate the error, instead of having a single output, here we have multiple outputs, one at each timestamp. Therefore, backpropagation is not applicable here. The modified backpropagation called BPTT is proposed for the recurrent neural network [82]. Algorithm 2 describes the working of the BPTT algorithm. For each time step, we are running the usual backpropagation. Therefore, BPTT is computationally expensive for more extended dependency.

Algorithm 2 Backpropagation Through Time

Step 1: The time-series input is given to the network. It passes through the whole network to produce an output.

Step 2: Unroll the network, and the error is calculated by comparing the actual and the predicted outcomes. Accumulate the errors across each timestamp.

Step 3: Roll the network, and the weights are updated to minimize the error.

Step 4: Repeat the steps, till convergence.

5.1.1.2 Optimization Algorithm

In the previous section, we have discussed the backpropagation training algorithm. The training algorithm's main aim is to calculate the error derivatives with respect to the network weights. The weights of a neural network cannot be calculated using an analytical method. Instead, the weights must be updated via an empirical optimization procedure called stochastic gradient descent.

In deep learning, training algorithms are optimization algorithms used to find the coefficient that minimizes the loss function (the difference between the actual and the predicted value). A gradient descent optimization algorithm is used during backpropagation where the error vector (a derivative of the loss function) is computed backward, starting from the final layer. Depending upon the activation function, the algorithm identifies how much change is required by taking the partial derivative of the function with respect to the weights or parameters. The learning rate η is a hyperparameter that controls the change in the model corresponds to the estimated error each time the model parameters are updated. The value for which model gives the minimum error is considered as the learning rate. This value is subtracted from the previous output value to get the updated value. This process repeats till convergence. There are three variants of gradient descent, depending on the amount of data we use to compute the objective function's gradient. Depending upon the amount of data, these variants are a trade-off between the parameter update's accuracy and the time it takes to perform an update. Determining the batch size is another hyperparameter that needs to be tuned before model training.

Batch gradient descent: It computes the gradient of the loss function with respect to the parameters θ for the entire training dataset. Here, η is the learning rate, ∇_θ is the partial derivative with respect to θ, and $J(\theta)$ is the loss function [83]. The parameters are updated as shown in the equation (5.1).

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} J(\theta_t) \tag{5.1}$$

Stochastic gradient descent : It computes the gradient of the loss function with respect to the parameter θ for each training example ($x^{(i)}$) [83]. It is faster than the batch gradient descent but due to frequent updates, the steps taken towards the minima are very noisy. It may take longer to achieve convergence to the minima of the loss function due to noisy steps.

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} J(\theta_t : x^{(i)})$$

Mini batch gradient descent : It is a combination of batch and stochastic gradient descent. It computes the gradient of the loss function with respect to the parameter θ for n training example ($x^{(i,i+n)}$) [83]. The n is defined as a hyperparameter and therefore needed to be tune before actual training of the model.

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} J(\theta_t : x^{(i,i+n)})$$

The gradient descent algorithm takes a lot of time to navigate through the region with a gentle slope, resulting in slow learning. Therefore momentum-based gradient descent is proposed [84]. In this approach, momentum is added based on the past observation, resulted in the faster convergence. It does this by adding a fraction κ of the updated vector (m_{t-1}) of the past time step to the currently updated vector.

$$m_t = \kappa m_{t-1} + \eta \nabla_{\theta_t} J(\theta_t)$$

$$\theta_{t+1} = \theta_t - m_t$$

Adam: Adaptive moment estimation (Adam) method computes adaptive learning rates for each parameter [85]. It performs smaller updates (i.e. low learning rates) for parameters associated with frequently occurring features, and larger updates (i.e. high learning rates) for parameters associated with infrequent features. It is well-suited for dealing with sparse data. The update rule is shown in equation (5.2).

$$v_t = \kappa_1 v_{t-1} + (1 - \kappa_1)(\nabla_{\theta_t} J(\theta_t))^2 \tag{5.2}$$

From the update rule, it is clear that the gradient's history is stored in v. Adam also keeps an exponentially decaying average of the past gradients m_t, similar to momentum.

$$m_t = \kappa_2 m_{t-1} + (1 - \kappa_2)(\nabla_{\theta_t} J(\theta_t))$$

As m_t and v_t are initialized to 0's, when the decay rates are small at that time these vectors are biased towards zero. This problem is solved by Zaheer *et al.* [86] by computing bias-corrected.

$$\hat{v}_t = \frac{v_t}{1 - \kappa_1^t}$$

$$\hat{m}_t = \frac{m_t}{1 - \kappa_2^t}$$

These vectors are used to update the parameters, the Adam update rule is defined as shown in equation (5.3), to avoid the denominator from becoming zero, a small value ϵ is added in the denominator. Smaller the v for smaller gradient accumulated and it will lead to a bigger learning rate.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} * (\hat{m}_t) \tag{5.3}$$

We use Adam as an optimization algorithm with all the learning models because the weather variables are sparse compared to the traffic variables. Therefore adaptive learning rate is required such that the frequent update corresponds to the traffic features doesn't overshadow the weather features.

5.1.2 Determine the Network Structure

The network structure is the total number of layers used in the learning model. The number of layers is different for different learning models. In determining the learning model's architecture, there is currently no widely accepted network parameter selection strategy. The network structure is dependent on the complexity of the specific problem. In the experimental setup (Chapter 6) we will discuss the network structure for each learning model used in this research.

5.1.3 Determine the Loss Function

The loss function reflects the error between the model output and the actual data. We use Mean Absolute Percentage Error (MAPE), which is compatible with the input data and was found to reduce the impact of a large amount of noise existing in the data [87]. Here, $\hat{y}_i$ is the predicted value and y_i is the actual value, S is the sample size.

$$Loss = \sum_{i=1}^{S} \frac{|\hat{y}_i - y_i|}{y_i}$$

The model adds L2 regularization to prevent the neural network from overfitting [88]. The overfitting of the model means that the model's prediction accuracy is very high on the training set, but it does not perform well on the validation and the test set. L2 regularization is introduced by adding the regularization term to the loss function. The formulation can be expressed as:

$$Loss = \sum_{i=1}^{S} \left[\frac{|\hat{y}_i - y_i|}{y_i} + \lambda \left\| w \right\|^2 \right]$$

where λ denotes the L2 regularization coefficient, w represents the weight of layers that uses L2 regularization. This method reduces the model's overfitting risk by penalizing the large weighting coefficients.

5.1.4 Determine the Activation Function

In a deep neural network, the weighted sum of the input is transformed into the output and this transformation is defined by the activation function. The activation function decides whether to activate the particular node or not. If information produce by the node is not useful then the node will not be activated else it will be activated. Without activation function, the neurons perform a linear transformation on the input using the weights and the biases as shown below.

$$x = \sum (weight * input) + bias$$

Here, output x is just the linear transformation of the input variable. The activation function is applied on the above equation as shown below.

$$f(x) = Activation(x)$$

Therefore, the non-linearity in the network is introduced by the activation function. Activation functions are differentiable, the first-order derivative can be calculated for a given input value. The neural network is trained using the backpropagation algorithm that requires the derivative of the prediction error to update the weights of the model. Different types of activation functions are discussed below:

Sigmoid Activation Function: This function (σ) is also called the logistic function [89]. It takes any real value as input and produces the output in the range of 0.0 to 1.0. The larger the input value, the transformed value will be closer to 1.0 and for the smaller value, the transformed value will be closer to 0.0. The mathematical expression for sigmoid is shown below.

$$\sigma = \frac{1}{1 + e^{-x}}$$

Where e is a mathematical constant, which is the base of the natural logarithm. The sigmoid function is not zero-centered. So the output of all the neurons will be of the same sign. This makes the gradients saturated and therefore results in slow or no learning. Sigmoid functions are useful for predicting probability-based output, filtered output, and binary classification problems [90]. It is used with the LSTM gates to give the filtered output.

Hyperbolic Tangent (Tanh) Activation Function: The tanh function is similar to the sigmoid function but it is zero centered [91]. This function takes any real value as input and produces the output in the range of -1.0 to 1.0. The tanh function is zero centered and therefore preferred over the sigmoid function, it gives better training performance for the deep neural networks [90, 92]. The mathematical expression for tanh is shown below.

$$\tanh = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$$

The tanh functions have been used in recurrent neural networks for natural language processing and speech recognition tasks [93, 94].

Rectified Linear Unit (ReLU) Activation Function: The mathematical expression for ReLU is shown below.

$$\text{ReLU} = max(0.0, x)$$

The main advantage of using the ReLU function is that it does not activate all the neurons at the same time. This function helps the deep neural network realize sparse activation [95]. The neurons will be deactivated if the output of the linear transformation is less than 0. The ReLU function is computationally efficient compared to the sigmoid and tanh functions because only a certain number of neurons are activated. The ReLU has been used in different architectures of deep learning because of its simplicity and reliability, which includes the convolutional neural network architectures [96, 97, 98].

Other hyperparameters such as the number of epoch and lag values are set using the random search method. The lag value is defined as the length of the past sequence passed as the input to the model. An epoch is when an entire training dataset is passed both forward and backward through the neural network, and the model requires multiple numbers of epoch to converge.

5.2 Learning Models

Traffic stream variables on the target road segment depend upon the other road segments' traffic and weather conditions. Also, the future traffic condition on the road segment depends upon its previous and current traffic conditions. We consider the deep learning models that extract the spatiotemporal and recurrent relationship from the traffic and weather data. The deep learning model should be robust enough to deal with the outlier. Also, the model accuracy should not be dependent on the prior domain knowledge to extract the hidden features.

5.2.1 Convolutional Neural Network (CNN)

In deep learning, the model used for object detection, image classification, and face recognition is Convolutional Neural Network (CNN) [99]. The main focus of CNN is local connectivity. It performs local dimensionality reduction. Each pixel shares the maximum correlation with its local neighbor. The traffic condition on a road segment is highly correlated to neighboring road's traffic and weather conditions. Therefore to extract the spatiotemporal features of the traffic and the weather data, CNN is the suitable choice. The spatial extent of this connectivity is called the receptive field of the node. The CNN derives its name from the convolution operator. The convolution preserves the spatial relationship between pixels by learning features using a small set of input data at a time. A convolution is a linear operation that involves multiplication between a set of weights with the input. The set of weights in the context of CNN is called a filter or kernel [100].

The filter's size is smaller than the input data, and element-wise multiplication is applied between the filter and filter-sized patches of input data to obtain the local connectivity. The resulting value shows the local connectivity between the pixels. The filter is applied to each filter-sized patch of the input data horizontally and vertically. The same filter applied over whole input data thus decreases the number of parameters. If the filter is designed to extract a specific feature in the input data, it will discover a similar feature anywhere in the input data. This is called translation invariance, where our model does not just tell whether the feature is present or not but also tells where it is present. For example, suppose the filter detects the change in the traffic variables due to weather variables at a particular location. In that case, the same filter will detect a similar pattern in other locations also. A dot product between the filter and input data gives a single value or a feature and filter slide over complete input data, resulting in a two-dimensional array called a feature map. The equation (5.4) shows the convolution operation between the input and the filter, where $X(m, n)$ is the convolutional input, $w(i, j)$ is the convolutional filter, b is the bias vector and $y(m, n)$ is the output of the convolution, here m and n are the data point of interest and filter size is $M \times N$ [101].

$$y(m,n) = \sum_{i=1}^{M} \sum_{j=1}^{N} X(m+i, n+j) w(i,j) + b \tag{5.4}$$

The feature map is passed through the nonlinear activation function that performs the nonlinear transformation. This transformed output is sent to the next layer for further computation. In the convolutional layer, multiple filters of different sizes are applied to each filter-sized patch of input data horizontally and vertically. The resulted feature map is passed as the input to the further convolutional layer. The convolutional layer's final output gives the feature vector that contains the hidden abstract information regarding the input. In deep learning, the feature vector is a collection of the measured values of a particular feature. In our case, it is a spatiotemporal feature corresponding to the input traffic and weather data.

The CNN model used in this study is shown in figure 5.2. The spatiotemporal input is passed to the first convolutional layer. In the process of convolution, multiple convolution kernels or filters are applied to the input such that each filter

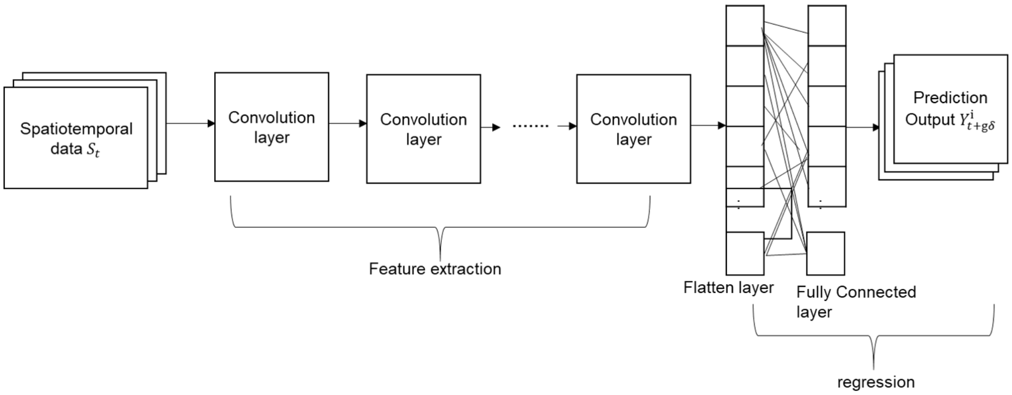


Figure 5.2: Schematic Diagram of CNN Based Traffic Stream Variables Prediction Model.

will extract a particular feature. When one convolution filter w^{conv1} is applied to the input the resultant output vector is y^{conv1} [102].

$$y^{conv1}(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} S_t(m+i,n+j)w^{conv1}(i,j) + b^1$$

Equation (5.5) shows the complete formulation for multiple filters (c_l) for each layer.

$$y_k^{conv}(m,n) = \text{ReLU}(\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{l=1}^{c_l} S_t(m,n,l)w_k^{conv}(i,j,l) + b_k^{conv}) \quad ,k \in [1,2,\dots L] \tag{5.5}$$

Here c_l represents the number of convolution kernel contained in the k^{th} convolution layer. L represents the total number of layers in the CNN architecture [100]. The ReLU is used as the activation function. After feature extraction, we need to predict the output value corresponding to the input sequence; this can be done using a fully connected layer. Before the fully connected layer, the feature vector should be converted into a one-dimensional vector, making it suitable for the fully connected layer processing. The flatten layer is used to convert the feature vector into a one-dimensional vector.

$$y^{flatten} = \text{flatten}([y_1^{conv}, y_2^{conv}, \dots, y_L^{conv}])$$

The flattened one-dimensional vector is passed through the activation function, and the resultant flatten output is given to the fully connected layer for the output prediction. The fully connected layer is similar to that in ANN, where each neuron is connected to all neurons in the last layer. The fully connected layer captures the complex relationships between high-level features. This layer's output is the one-dimensional vector; it is the predicted value of the traffic stream variables. In the below equation, w^{fc} and b^{fc} are the fully connected layer's weight and bias. $Y^{i}_{t_g}$ is the predicted traffic stream variable at i^{th} road segment for prediction horizon t_g.

$$Y^{i}_{t_g} = w^{fc} \, y^{flatten} + b^{fc}$$

Algorithm 3 shows the steps to train the CNN model. The traffic and the weather data are pre-processed. The input consists of traffic and weather sequence up to lag value, and output consists of traffic stream variables value at the prediction horizon. After pre-processing, the spatiotemporal data is split into the training and the test data-set for different prediction horizons. The hyperparameters tuning is performed. For each spatiotemporal input sequence, our CNN model extracts the spatiotemporal feature vector. This spatiotemporal feature vector is given to the flatten layer. The one-dimensional vector is passed to the fully connected layer to predict the output value. The error is calculated based on the predicted output value. The calculated error is backpropagated to the model using the backpropagation method. To update the weight, the optimization algorithm is used, and this process repeats till convergence. After completion of the training, test data is used to evaluate the model performance.

Algorithm 3 Training CNN Model

Step 1: Pre-process the traffic and the weather data. Split the spatiotemporal data into the training and the test data set for 15-minute, 30-minute, 45-minute, and 60-minute prediction horizons.
Step 2: Hyperparameters tuning is performed.
Step 3: Input is passed to the first convolutional layer. The output from the first layer is passed to the next layer and so on. The feature vector from the last convolutional layer is passed to the flatten layer which transforms the input to one-dimensional output. This output is passed to the fully connected layer to predict the output.
Step 4: The predicted output is compared to the true output, and the error is generated. The error is backpropagated to the model with the backpropagation method. The optimization algorithm is used to update the weights; hence the CNN model is trained.
Step 5: Use test data to examine the prediction performance.

5.2.2 Sequence Learning Models

For accurate traffic stream variables prediction, we use the model that considers a recurrent learning approach to learn the relevant spatiotemporal features from the weather and traffic data. In a sequence learning problem, the future outcome is dependent on the current and the past observations, i.e., the output from the current timestamp is provided as input to the next timestamp. The deep learning model designed for sequence learning is Recurrent Neural Network (RNN), and Long Short Term Memory (LSTM). Section 5.2.2.1 discusses the RNN model and section 5.2.2.2 discusses the LSTM model in detail.

5.2.2.1 Recurrent Neural Network (RNN)

The idea behind RNN is to use sequential information. Unlike other neural networks where inputs are independent, RNN takes sequential input into account [100]. Hidden layers in the RNN have the same parameters (weight and bias) throughout the procedure, which works as a working memory to process the information. In the figure 5.3, the RNN received input x_t and produce the output feature vector h_t [65]. A loop allows information to be passed from one step of the network to the next [103].

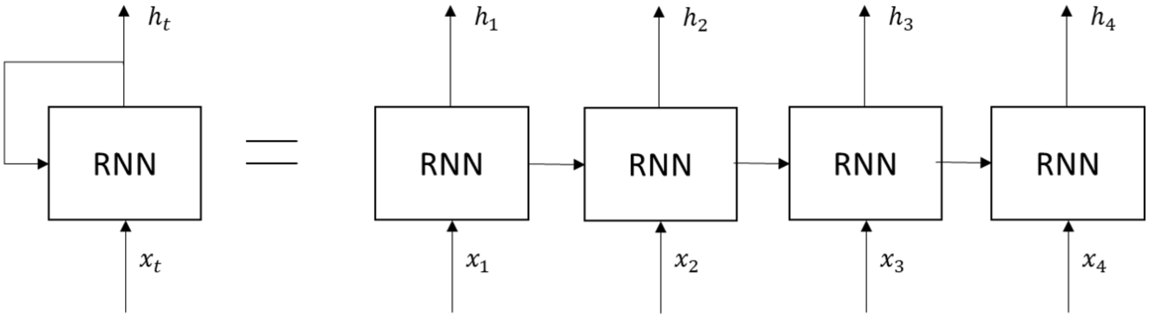


Figure 5.3: An Unrolled Recurrent Neural Network.

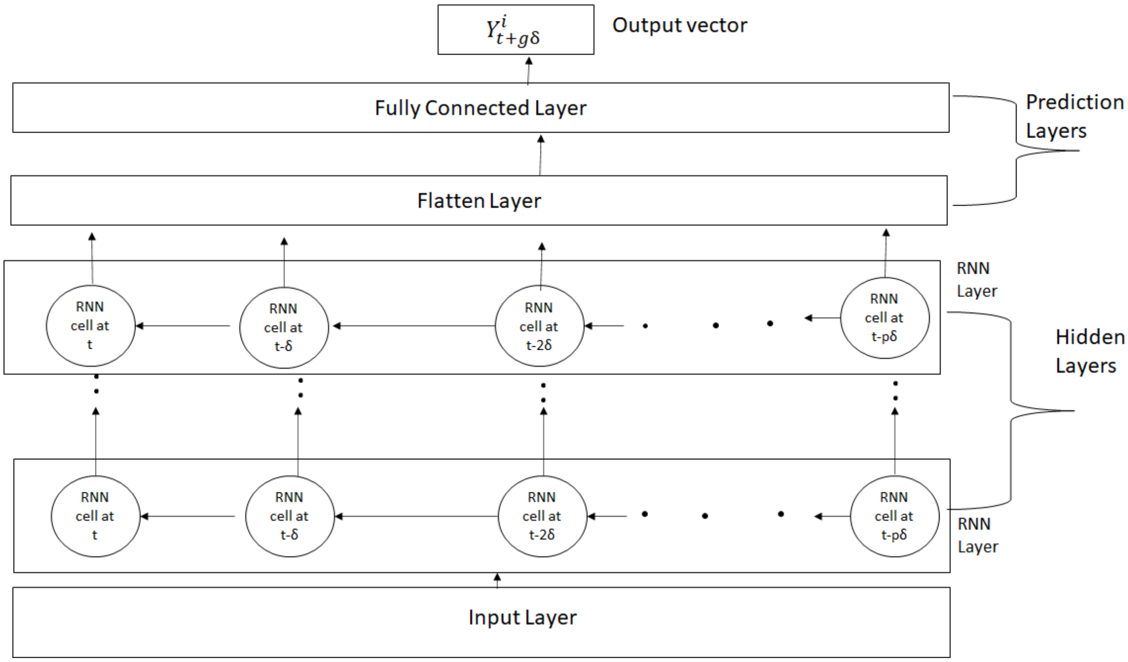


Figure 5.4: Schematic Diagram of RNN Based Traffic Stream Variables Prediction Model.

The design of our RNN based model is shown in figure 5.4. The model consists of the input layer, the hidden layers, and the prediction layer. The input layer contains a spatiotemporal sequence of traffic and weather data. The spatiotemporal sequence is passed to the hidden layer. The hidden layer is a memory block, where the cell memorizes the spatiotemporal features and passes this information to the next cell state. In the RNN model, the hidden layer consists of RNN memory cells. The input sequence is passed through the RNN cell, one at a time. The cell state remembers the past sequence and combines this information with the current input data to provide the output. The multiple hidden layers provide the ability to learn features at different levels of abstraction. For each time instance, the traffic data of *n* road segment and weather data from *k* weather stations is given to the model. Therefore, the cell state of RNN updates this information based on the previous cell state and current input. If there is a spatiotemporal correlation between traffic conditions on road segments and weather conditions, the RNN updates its feature vector. The last hidden layer's output, the feature vector, is given as input to the prediction layer. After feature extraction, we need to predict the output value corresponding to the input sequence. This can be done using a fully connected layer. The output of a fully connected layer is a one-dimensional vector. It is a vector of traffic stream variables (traffic speed and flow) predicted value.

Figure 5.5 shows the details of the RNN cell [103].

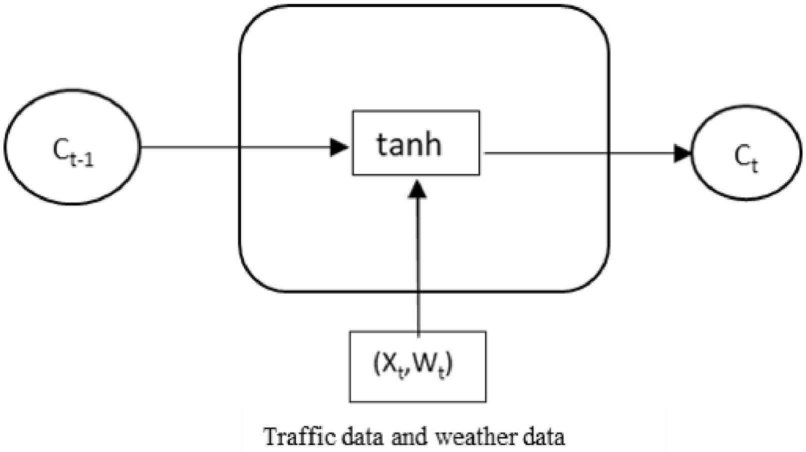


Figure 5.5: RNN Cell Architecture.

Formula for current cell state considering traffic and weather data,

$$C_t = \tanh\left(w_c C_{t-1} + w_x (X_t, W_t) + b_c\right)$$

Where, (X_t, W_t) is the current traffic data and weather data from *n* road segment and *k* weather stations, respectively. C_{t-1} is the previous cell state, w_c is the weight of previous cell state, w_x is weight of current input state, b_c is the bias vector and tanh is non-linear activation function [65].

The output from the last hidden layer is provided as input to the flatten layer and then provide the one-dimensional vector to fully connected layer for the prediction task.

$$Y^i_{t_g} = w_y \, C_t + b_v$$

$Y^i_{t_g}$ is the traffic stream variables for i^{th} road segment at prediction horizon t_g, w_y is the weight and b_v is the bias vector.

5.2.2.2 Long Short Term Memory (LSTM)

The RNN model suffered from the vanishing gradient problem and this problem is solved by the LSTM [18]. LSTM introduces the concept of working memory (short-term memory) as the hidden state and long-term memory as cell states to solve the problem of vanishing gradient due to the long-term dependency [18]. Appendix B discusses the vanishing gradient problem suffered by RNN and the solution offered by the LSTM model. LSTM uses the forget, input, and output gate to update the long and short-term memory [66, 100]. Rainfall has both short-term and long-term effects on the traffic stream variables. The short-term impact of rainfall is significant on traffic speed as vehicles slow down their speed due to poor visibility or to avoid collision risk. The long-term impact of rainfall is substantial on both traffic flow and speed. Our model remains the same, as shown in figure 5.4. The only change is in the hidden layer where LSTM cells are used. Figure 5.6 shows the detail of the LSTM cell. A set of equations used in LSTM are as follows:

$$f_t = \sigma(w_f[h_{t-1}, (X_t, W_t)] + b_f)$$

$$i_t = \sigma(w_i[h_{t-1}, (X_t, W_t)] + b_i)$$

$$o_t = \sigma(w_o[h_{t-1}, (X_t, W_t)] + b_o)$$

$$\widetilde{C}_t = \tanh\left(w_c[h_{t-1}, (X_t, W_t)] + b_c\right)$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \widetilde{C}_t$$

$$h_t = o_t \otimes \tanh\left(C_t\right)$$

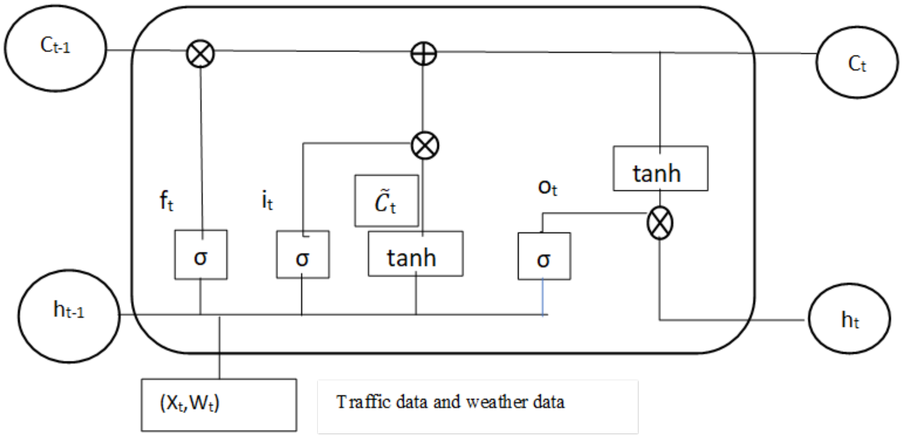


Figure 5.6: LSTM Cell Architecture.

Where (X_t, W_t) is the current traffic and weather data, w_f, w_i, w_o, w_c are the weight of the forget gate, input gate, output gate, and cell state, respectively. b_f, b_i, b_o and b_c is the bias vector of the forget gate, input gate, output gate, and cell state, respectively. f_t, i_t, o_t, C_t, h_t is the output of the forget gate, input gate, output gate, current cell state and current hidden state, respectively. C_{t-1} is the previous cell state and h_{t-1} is the previous hidden state, $\otimes$ is element-wise multiplication, tanh and σ are the activation function. The output from the last hidden layer is given to the fully connected layer after passing through flatten layer and predicted value is calculated based on the linear activation function.

$$Y^i_{t_g} = w_y(h_t) + b_v$$

$Y^i_{t_g}$ is the output at prediction horizon t_g, w_y is weight of current hidden state and b_v is the bias vector.

The algorithm to train the RNN and the LSTM model is described in the Algorithm 4.

Algorithm 4 Training RNN and LSTM Model

```
Step 1: Split the spatiotemporal data into training and test data sets. In training data, for each input sequence, there is a true output value.
Step 2: Perform hyperparameters tuning.
Step 3: Based on the optimal set of hyperparameters, train the RNN and the LSTM model using the training data.
Step 4: For each input, the current state is updated based on the current input and the previous state. The output from the last state is given as input to the fully connected layer. This layer calculates the output value. The predicted output is then compared to the actual output, and error is computed. The error is backpropagated to the model with the BackPropagation Through Time (BPTT) method. The optimization algorithm is used to update the weights. Hence the model is trained.
Step 5: Use the test data to examine the prediction performance.
```

5.2.3 Hybrid Deep Learning Models

The traffic stream variables prediction in the presence of weather conditions on the target road has a spatiotemporal dependency on adjacent roads traffic and weather conditions and temporal dependency on traffic characteristic of itself.

In our model, there are two submodels, one which extracts the spatiotemporal features and the other that uses these spatiotemporal features as memory and extracts features based on memory and temporal input for traffic forecasting.

CNN is the suitable model to extract the spatiotemporal features between the target and adjacent road's traffic and weather data because of its characteristics to extract local connections between pixels from one-dimensional and two-dimensional data. LSTM is a favorable choice to extract the short and long-term spatiotemporal dependency between the weather and the traffic data of the target and other roads. Also, the LSTM model learns the effective temporal features of the traffic and the weather data to predict the traffic stream variables. Therefore, we design two hybrid models, CNN-LSTM and LSTM-LSTM.

In the LSTM model, the hidden layer is the memory block, where the cell memorizes the temporal features and passes this information to the next cell state. In LSTM memory cells, the current cell state depends upon the previous cell state and the current input. This cell state is initialized randomly during model training. The traffic condition on the target road is spatiotemporal dependent upon the past and the current surrounding traffic and weather conditions and temporal dependency on previous and current traffic conditions of its own. Therefore, in our model, we initialize the memory or cell state of LSTM with the spatiotemporal features extracted by the CNN or the LSTM model.

5.2.3.1 CNN-LSTM

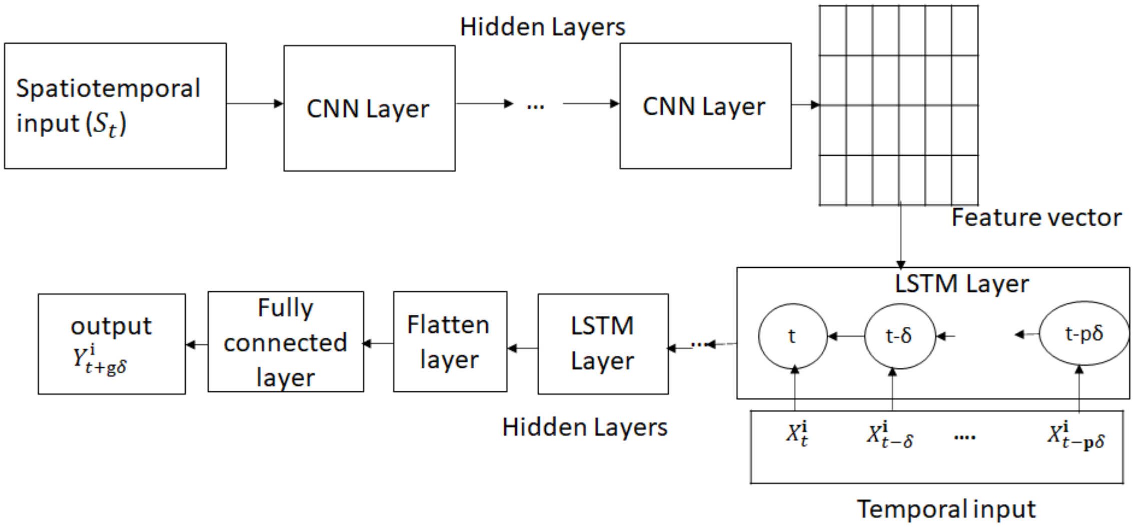


Figure 5.7: CNN-LSTM Model for Traffic Variables Prediction.

The design of our hybrid CNN-LSTM model is shown in figure 5.7. In our model, the spatiotemporal input consists of traffic, and weather data is passed to the convolutional layer. The CNN model considers each input independently; the output doesn't depend upon the previous input. Ideally, CNN can handle a single image at a time and transform it from input pixels into a vector representation. In our model, we need to repeat the same operation across multiple inputs, where inputs are dependent on each other. This mechanism will allow the LSTM model to build up internal state and update weights using the BPTT algorithm across a sequence of the input's internal vector representation. In our case input to the CNN is the spatiotemporal data from n road segments and k weather station from time $t - p\delta$ to t. The CNN model transforms the spatiotemporal input data into a vector representation. We want to apply the CNN model to each spatiotemporal input and pass the output vector to the LSTM model as a single time step. We can achieve this by wrapping the CNN model (one layer or more) in a TimeDistributed layer [104]. The TimeDistributed layer applies the same set of operations to every temporal slice of an input. Therefore, this layer achieves the desired outcome of applying the same layer or layers multiple times. In our case, the TimeDistributed layer helps us in applying the CNN layer multiple times to multiple input time steps. A sequence of the feature vector is provided to the LSTM model to work on. This feature vector, along with the target road segment's temporal data, is passed to the LSTM layer.

The temporal sequence as shown below from i^{th} target road segment for t to $t - p\delta$ time is used as the input in our LSTM model.

$$\begin{bmatrix} X_t^i & X_{t-\delta}^i & \dots & X_{t-p\delta}^i \end{bmatrix}$$

The temporal data contains all the information about the hidden characteristics of the target road segment. The feature vector from the CNN will work as an initial cell state of the LSTM network. The traffic stream variable's temporal correlation to the target road segment given spatiotemporal feature vector is identified through the LSTM layer. After feature extraction, we need to predict the output value corresponding to the input sequence; this can be done using the fully connected layer. The fully connected layer gives the predicted value of the traffic stream variables. Algorithm 5 shows the steps to train the CNN-LSTM model. The traffic and the weather data are pre-processed. The input consists of the traffic and the weather sequence up to the lag value, and the output consists of the traffic stream variable value at the prediction horizon. After pre-processing, the spatiotemporal and temporal data is split into the training and the test dataset for different prediction horizons. The hyperparameters are tuned for efficient model performance. For each spatiotemporal input sequence, our CNN model extracts the spatiotemporal feature vector. This spatiotemporal feature vector and the temporal input sequence are given to the LSTM model to predict the output value. The error is calculated based on the predicted output value. The calculated error is backpropagated to the model using the BPTT method. The optimization algorithm is used to update the weights, and this process repeats till convergence. After completion of the training, test data is used to evaluate the model performance.

Algorithm 5 Training CNN-LSTM Model

Step 1: Split the data into spatiotemporal and temporal training and test data set, for each input instance, there is the output value.
Step 2: Perform hyperparameters tuning.
Step 3: Each layer in the CNN model wrapped in a TimeDistributed layer. The CNN model's output feature vector is stored as an initial cell state of the first layer of the LSTM model. The CNN model is trained based on spatiotemporal training data, and the LSTM model is trained based on temporal training data.
Step 4: The predicted output is compared to the true output, and the error is generated. The error is backpropagated to the model with the BPTT method. The optimization algorithm is used to update the weights; hence the CNN-LSTM model is trained.
Step 5: Use test data to examine the prediction performance.

5.2.3.2 LSTM-LSTM

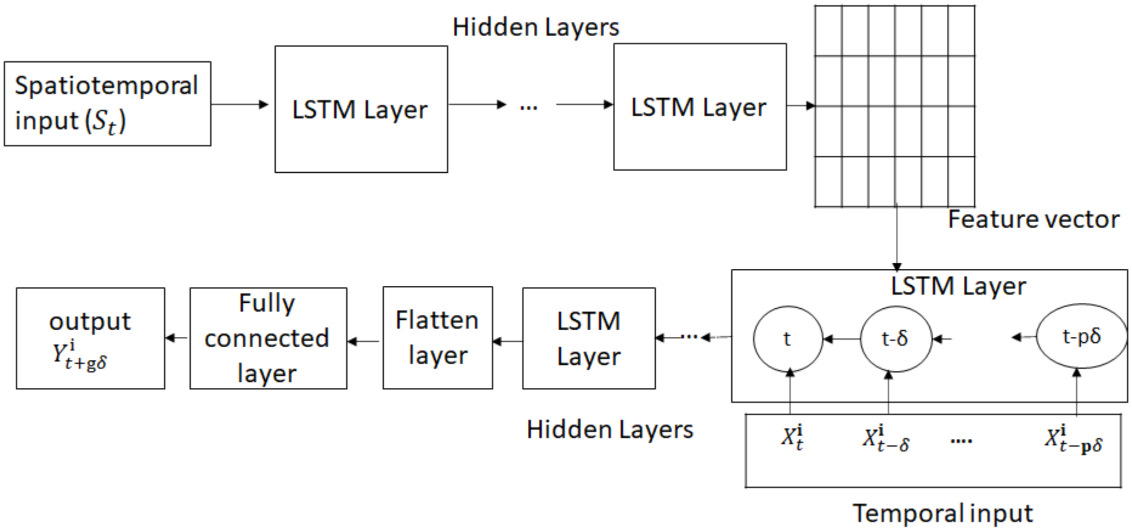


Figure 5.8: LSTM-LSTM Model for Traffic Variables Prediction.

The design of our LSTM-LSTM model is shown in figure 5.8. The spatiotemporal input is passed to the first LSTM model, and the hidden state vector from the last layer is passed to the second LSTM model with the temporal input. The LSTM layer finds the correlation between the feature vectors consisting of a summary of spatiotemporal traffic and weather sequence and temporal traffic sequence at the target road segment. The design of the latter LSTM model is similar to the design of the LSTM model of CNN-LSTM as discussed above.

Algorithm 6 gives the details about training LSTM-LSTM model. The spatiotemporal and temporal data is split into the training and the test dataset for different prediction horizons. For example, to predict the traffic stream variables at the prediction horizon of 15-minute, we pre-process the data. The input consists of the sequence up to lag value and output consisting of traffic stream variables value at 15-minute from the current time instance. The pre-processed data is split into training and test dataset. For each spatiotemporal input sequence, our former LSTM model extracts the spatiotemporal feature vector. This spatiotemporal feature vector and temporal input sequence are given to the latter LSTM model to predict the output value. The difference between the actual and predicted value is used to calculate the prediction error. The calculated error is backpropagated to the model using the BPTT method. The optimization algorithm is used to update the weights, and this process repeats till convergence. After completion of the training, test data is used to evaluate the model performance.

Algorithm 6 Training LSTM-LSTM Model

Step 1: Split the data into spatiotemporal and temporal training and test dataset.
Step 2: Perform hyperparameters tuning.
Step 3: Output feature vector from the former LSTM model is stored as an initial cell state of the first layer of the latter LSTM model. The first LSTM model is trained based on spatiotemporal training data, and the second LSTM model is trained based on the temporal training data and features extracted by the first one.
Step 4: The predicted output is compared to the actual output, and the error is generated. The error is backpropagated to the model with the BPTT method, and the optimization algorithm is used to update the weights. The process repeats until convergence, and hence the LSTM-LSTM model is trained.
Step 5: Use the test data to examine the prediction performance.

Appendix C shows the sample code of all the above discussed deep learning models.

CHAPTER 6

Experimental Setup and Result Analysis

We perform two different kinds of experiments, one is to examine the performance of adopted deep learning models under different scenarios and the other is to examine the quality of synthetic traffic data for different types of roads and rainfall intensities.

To examine the performance of adopted deep learning models, each model is trained with a large dataset depends upon the scenario and for the unseen dataset, the model prediction performance is observed. One of our main objectives is to examine the performance of deep learning models with weather data inclusion. The accuracy of the prediction model is defined as the difference between the actual and the predicted value. In an ideal situation, this difference is zero. Section 6.1 discusses the performance of the recurrent learning models, RNN and LSTM, with and without the weather variables inclusion. The model's prediction accuracy is used to analyze the model's performance. We compare the recurrent learning models with the models discussed in the literature.

To examine the quality of synthetic data, the comparison between ground truth data and synthetic data is necessary to validate the synthetic data generation's empirical model. Section 6.2 discusses the simulation model's calibration for generating traffic data for different rainfall intensities for different road types. We compare the synthetic traffic data with the ground truth traffic data for different rainfall intensities and different road types.

Section 6.3 analyzes the prediction accuracy of the adopted deep learning models for synthetic traffic data. The RNN, CNN, LSTM, CNN-LSTM, and LSTM-LSTM models predict the traffic stream variables using synthetic traffic data. We generate the traffic data for different types of roads such as arterial, sub-arterial, and collector for different road network types, i.e., single, small and large road networks. We discuss the complexity involves with each learning model. We also compare the hybrid models with the existing models.

6.1 Deep Learning Model Performance with Inclusion of Rainfall Data

We examine whether the inclusion of the weather variable improves the deep learning model's prediction accuracy. We use the recurrent learning model to perform this experiment. We exclude the soft spatial and temporal threshold for this experiment and consider only the spatiotemporal traffic and weather data as input. Our main aim is to observe the performance of the learning model with the inclusion of weather variables. We examine RNN and LSTM model performance when trained with traffic data only and trained with traffic and weather data to predict the traffic stream variables during adverse weather conditions. We design two cases for this study as described in Table 6.1.

Table 6.1: Cases To Evaluate Model Performance.

Case	Data Consider for Training and Test
I	Traffic data only
II	Traffic and weather data

6.1.1 Data Description

For this experiment, we use the traffic data from the ITS infrastructure [67]. Traffic and weather data from San Diego (district 11) are used in this experiment. Details of the traffic and weather data are provided in section 4.2.1. Traffic data contains road segment id, timestamp, traffic flow, and average traffic speed along the lanes. The weather data contains station id, timestamp, latitude, longitude, precipitation value. We check the optimal distance to consider the other road segments and weather stations for the target road segment. The distance for which models give minimum error is considered in this study. For the target road segment, the other road segments up to 11 km and weather stations up to 15 km are considered. Table 6.2 shows the example of the target road segment id and corresponding road segments and weather station ids. In our model, if we consider a lag value of 12, then 1-hour traffic data is passed (collection frequency is a 5-minute interval), and 3-hour rainfall data is passed (collection frequency is a 15-minute interval). Therefore we can access a more extended rainfall sequence as compared to the traffic sequence.

Table 6.2: Target Road Segment and Other Road Segments and Weather Stations id for the Experiment Corresponds to I-5 Road Segment.

Target Road Segment id	Weather Stations ids	Other Road Segments ids
5N1108459	Coop:040136, Coop:047740, Coop:040983, Coop:041424, Coop:042239, Coop:042350, Coop:042406	5N1108648, 5N1108450, 5N1108452, 5N1108456, 5N1108458 5N1108461, 5N1108464, 5N1108465, 5N1108467

6.1.2 Experimental Setup

For the model's optimal structure, we need to determine the number of hidden layers, the number of hidden units per layer, the number of the epoch, and the lag value. We use the random search method for hyperparameter tuning [80]. The number of parameters to be estimated is dependent on the optimal structure, and the normal distribution is used to initialize the parameters. The LSTM and RNN model is trained based on Adam optimizer [85]. The learning rate is set to be 0.006, and the batch size is set to 215. To avoid overfitting, the dropout layer and early stopping are used. For the implementation, we used the TensorFlow version 1.0.1 framework, and GPU is used to improve the computational capability to fasten the training module's execution. Python version 2.7 is used for the programming. Table 6.3 and 6.4 show the most useful set of hyperparameters for the LSTM and RNN model, respectively, for different road traffic data. The small change in the hyperparameter's value doesn't reflect the significant change in the model's output, which shows the deep learning model robustness. For both the LSTM and RNN models for different road types, the lag value in the range [12-20] for a 5 to 30 minutes prediction horizon provides no significant difference. For a 45 to 60 minutes prediction horizon, the lag value is in the range [20-25]. For 5 to 30 minutes, the number of layers is in the range [1-2], and 45 to 60 minutes is in the range [2-4]. The number of epoch for 5 to 15 minutes is in the range of [10-18], and 30 to 60 minutes is in the range [20-27] for I-5 and I-5D road. The number of epoch for 5 to 15 minutes is in the range of [15-20] for 30 to 60 minutes is in the range [22-35] for SR-75 road.

In all the hyperparameter table, PH stands for prediction horizon, p defines the sequence length or lag value, L defines the number of layer, NPL defines the nodes per layer, E defines the number of epoch.

Table 6.3: Hyperparameters for the LSTM Model for Different Roads.

Road	PH	p	L	NPL	E
I-5	5	12	1	30	12
	15	18	1	40	15
	30	18	2	[20,40]	18
	45	24	2	[30,60]	22
	60	24	2	[40,80]	25
I-5D	5	15	1	30	15
	15	18	1	50	18
	30	20	2	[30,50]	22
	45	24	2	[40,60]	25
	60	28	3	[30,50,80]	30
SR-75	5	12	1	40	18
	15	18	2	[30,50]	20
	30	18	2	[40,60]	27
	45	24	3	[40,60,80]	30
	60	24	3	[50,80,120]	35

Table 6.4: Hyperparameters for the RNN Model for Different Roads.

Road	PH	p	L	NPL	E
I-5	5	10	1	15	10
	15	12	1	20	12
	30	18	2	[10,30]	15
	45	22	2	[15,40]	20
	60	22	2	[20,40]	20
I-5D	5	10	1	20	12
	15	12	1	30	15
	30	18	2	[30,40]	18
	45	22	2	[30,60]	22
	60	22	3	[40,70,90]	27
SR-75	5	10	1	60	15
	15	12	2	[20,50]	18
	30	18	2	[30,70]	22
	45	22	3	[30,60,80]	25
	60	22	3	[40,70,90]	32

6.1.3 Result Analysis

The traffic data on the I-5 road segment on a normal weather day is shown in figure 6.1. Peak hour shows the maximum flow, which resulted in the lower traffic speed. The rainfall data for a similar geographical location is shown in figure 6.2.

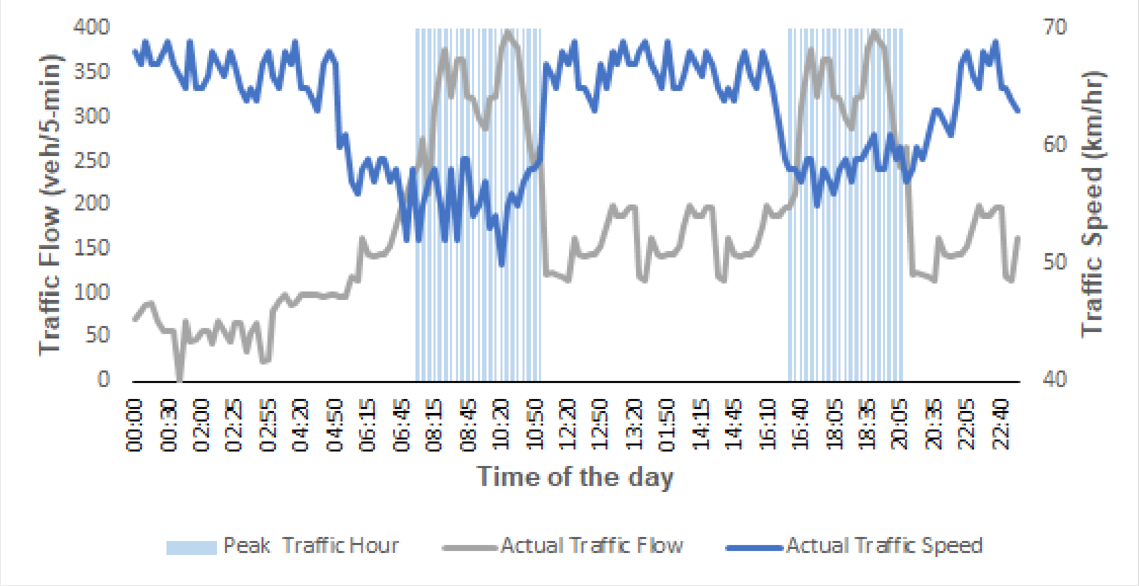


Figure 6.1: Traffic Flow and Speed During Normal Weather Day Corresponding to I-5 Road, Vertical Lines Signifies the Peak Hour.

The rainfall value ≤ 0.5 mm/hr is categorized as light rainfall, ≤ 5 mm/hr is categorized as moderate rainfall and ≥ 7 mm/hr is categorized as heavy rainfall.

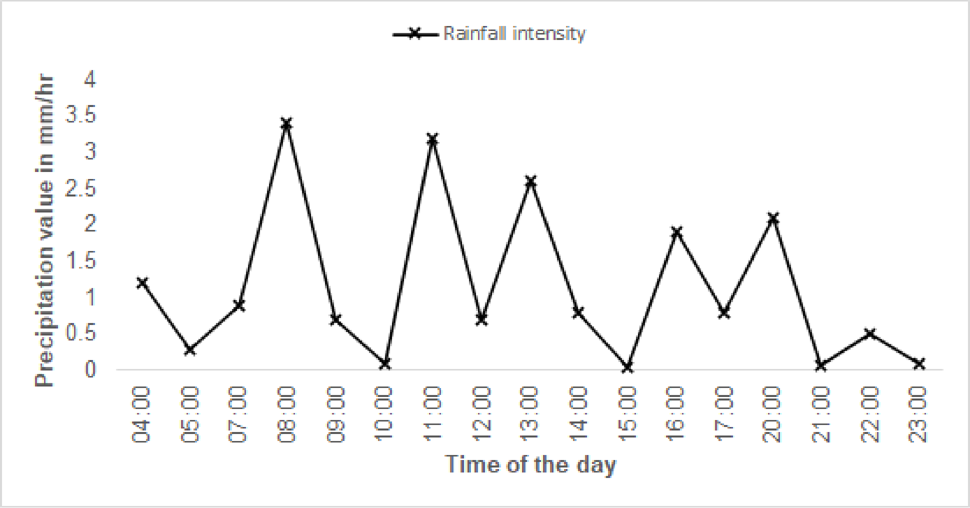


Figure 6.2: Rainfall Data from Weather Station in San Diego.

Figure 6.3 shows the heat-map, the darker color represents the lower traffic speed value and the lighter color represents the higher traffic speed value. The light to moderate rainfall affects the traffic speed during peak hour, and during the off-peak hour, the impact is small. Heavy rainfall significantly affects the traffic speed during the entire day.

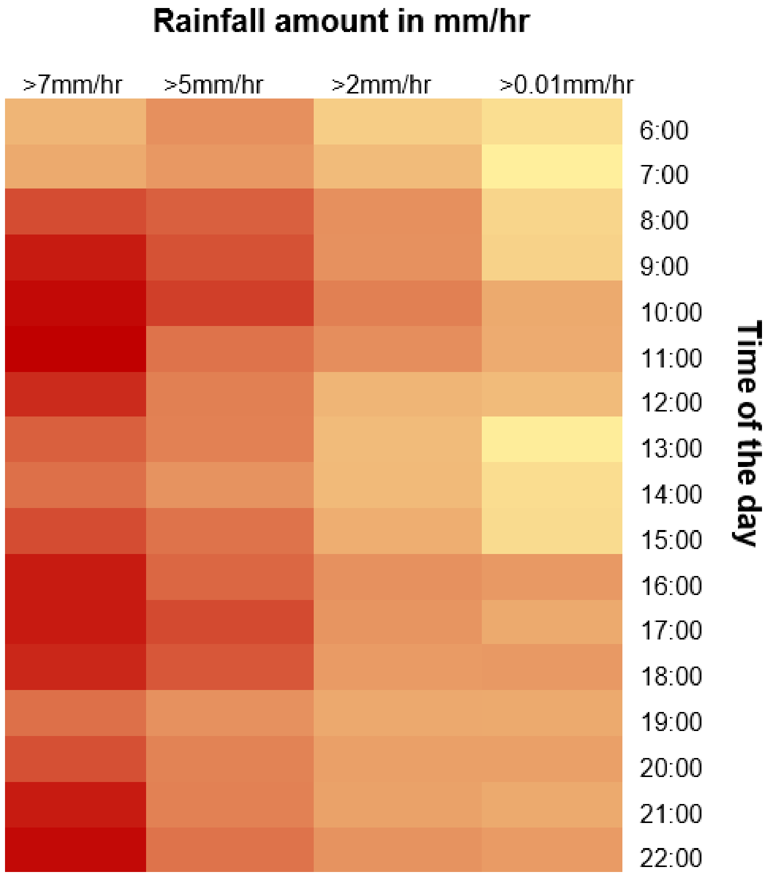


Figure 6.3: Traffic Speed as a Function of Hourly Precipitation and Time of the Day.

We examine the RNN and LSTM model's performance to predict the traffic stream variables during adverse weather conditions when trained with traffic and weather data and when trained only with traffic data. Figure [6.4-6.6] shows the value predicted by the LSTM model trained with and without rainfall variable of I-5 road for the 15-minute and 60-minute prediction horizon, respectively, for different rainfall intensities. We test our model performance for a rainy, moderate, and light to no rainfall day.

On heavy rainfall day, the figure shows that the value predicted by the model trained with rainfall input is close to the measured traffic speed value compared to the model trained with traffic data only. The absolute difference between the measured and the predicted speed for the LSTM model trained with rainfall data and tested on the heavy rainfall day is in the range [0.32-3.4] and [1.99-8.46] km/hr for 15-minute and 60-minute prediction horizon, respectively. For the same test data,

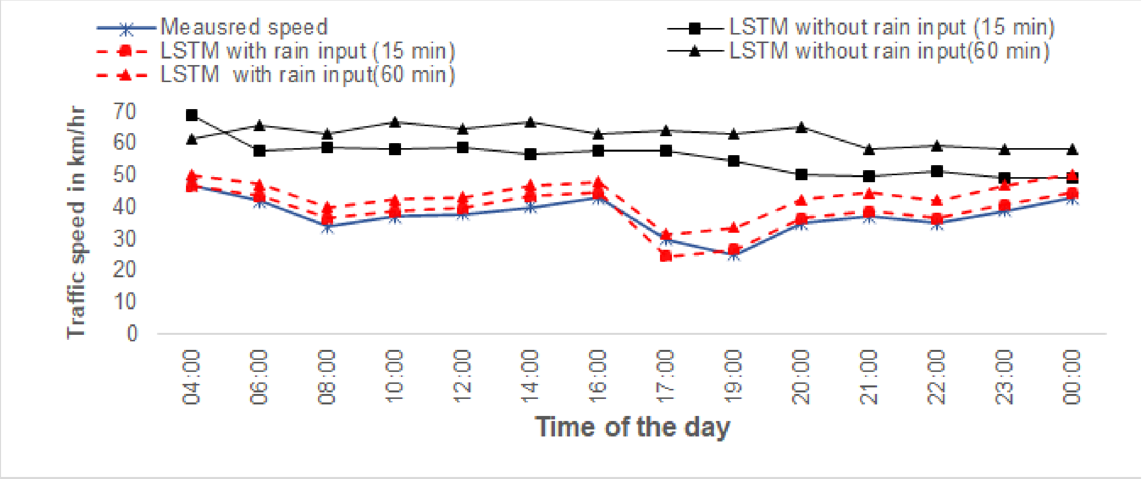


Figure 6.4: 15-minute and 60-minute Traffic Speed Prediction Performance of LSTM With and Without Rainfall Variable. Model Performance is Tested for **Heavy Rainfall** Day.

the absolute difference between the measured and predicted speed for the model trained without rainfall data is in the range [2.4-25.3] and [3.45-34.8] km/hr for 15-minute and 60-minute prediction horizon, respectively. Therefore, we can conclude that a significant difference is observed in the model's performance when trained with and without rainfall data for a heavy rainfall day.

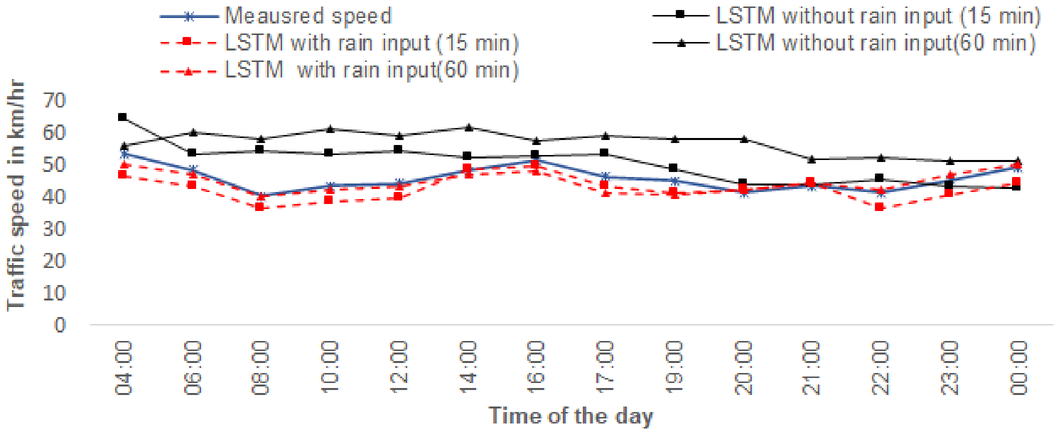


Figure 6.5: 15-minute and 60-minute Traffic Speed Prediction Performance of LSTM With and Without Rainfall Variable. Model Performance is Tested for **Moderate Rainfall** Day.

For a moderate rainfall day, as shown in figure 6.5, the absolute difference between the measured and predicted speed for the LSTM model trained with rainfall data is in the range [0.29-2.8] and [1.7-5.8] km/hr for 15-minute and 60-minute prediction horizon, respectively. The absolute difference between the measured and predicted speed for the model trained without rainfall data is in the range [1.7-11.6] and [2.7-19.6] km/hr for 15-minute and 60-minute prediction horizon, respectively. There is a difference between the model performance when trained with and without rainfall data but not as significant as heavy rainfall days.

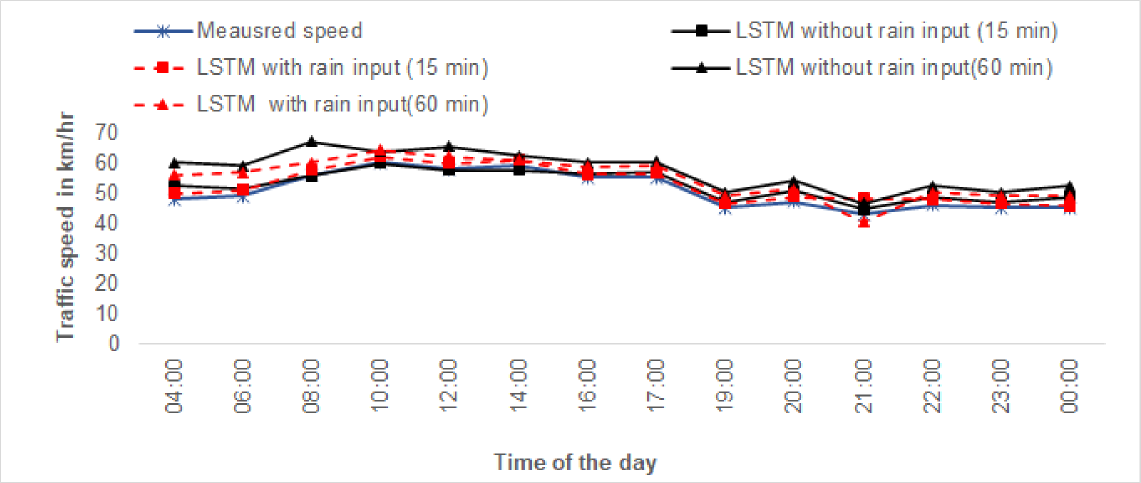


Figure 6.6: 15-minute and 60-minute Traffic Speed Prediction Performance of LSTM With and Without Rainfall Variable. Model Performance is Tested for **Light to No Rainfall** Day.

As shown in figure 6.6, the performance of the model trained with rainfall data and tested on light to no rainfall day is in the range [0.22-1.93] and [1.92-4.3] km/hr for a 15-minute and 60-minute prediction horizon. For the same test data, the model's performance when trained without rainfall data is in the range [0.24-2.2] and [2.3-6.35] km/hr for a 15-minute and 60-minute prediction horizon, respectively. Therefore we can conclude that for light to no rainfall day, there is no significant difference in the performance of the model when trained with and without rainfall data.

To examine the impact of rainfall on different roads, we test the model performance on San Diego's roads. The broader road faces less reduction in road capacity during rainfall as compared to the narrow road. Similarly, low-elevation and narrow roads are more prone to waterlogging as compare to high-elevation and broad roads. The roads passing through the densely populated area are more affected compared to the freeway road segment. Instead of passing the data related to every hidden variable like elevation, road width, road type, etc., that affects the traffic stream variables during rainfall; we use the deep learning model that learns hidden features from the traffic and weather data. Therefore, we test the model performance for various roads (I-5, I-5D, and SR-75). Figure 6.7 shows the traffic speed prediction accuracy of the RNN and LSTM model for different prediction horizons. The performance is evaluated corresponding to the data from road segments of I-5 passing through the countryside area, the Downtown area, and SR-75.

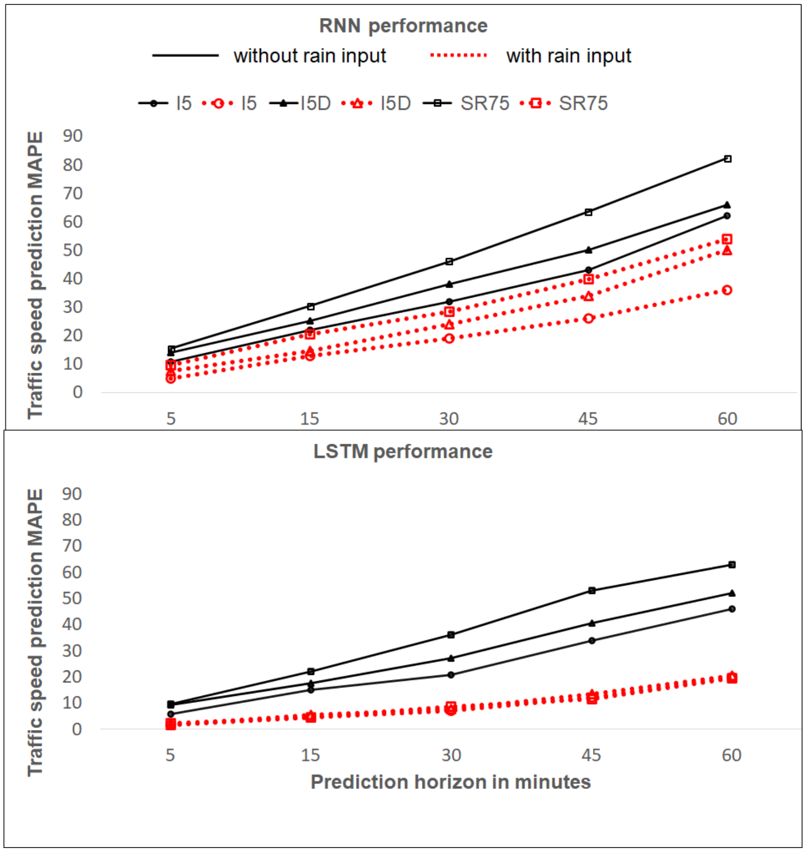


Figure 6.7: Traffic Speed Prediction by RNN and LSTM Model Trained With and Without Rain Input and Comparing MAPE Corresponding to I-5, I-5 Downtown, and SR-75 Roads.

Comparing the prediction performance, less error is observed when trained with rainfall data as compare to trained without rainfall data. The traffic speed prediction error of the LSTM model trained with rainfall variable is the lowest for all road segments corresponding to all prediction horizons. The traffic speed prediction error of both the models (RNN and LSTM) trained without rainfall input is the highest for SR-75. The traffic speed MAPE of the RNN model trained without rainfall data for the SR-75 road segment considering the 30-minute prediction horizon is 46% and of the LSTM model is 36%. The traffic speed MAPE of the RNN model trained without rainfall data for the I-5 road segment considering the 30-minute prediction horizon is 31% and of the LSTM model is 20%. SR-75 is a narrow and low-elevation road compared to I-5. Therefore it gets most affected due to rainfall compared to the I-5. For SR-75 road, the MAPE for the prediction horizon of 30 minutes of RNN model when trained with rain input is 28%, whereas, for the LSTM model trained with rain input, it is 8%. For the I-5 road segment, the MAPE of the RNN and LSTM models when trained with rainfall data for 30-minute prediction horizon is 19% and 7%, respectively.

Figure 6.8 shows the traffic flow prediction accuracy of the RNN and LSTM model for different prediction horizons for all road segments. It is observed that for a shorter prediction horizon ($\leq$ 15 minutes) traffic speed prediction is affected by rainfall events because as rainfall starts, it affects the driver's visibility and mobility. For the same prediction horizon, the model's performance trained with and without rainfall data is almost the same for traffic flow prediction.

The percentage traffic flow error of the LSTM model for SR-75 is 9% and 4% when trained without and with rainfall data, respectively, for a 15-minute prediction horizon.

However, for a longer prediction horizon, the rainfall shows an impact on traffic flow prediction accuracy. The LSTM model trained with rain inputs shows the minimum error for all the prediction horizons. LSTM model's traffic flow MAPE when trained without rainfall data for SR-75 road segment and 30-minute prediction horizon is 22% whereas for RNN model trained without rainfall data is 35%. The percentage error of a model for the same prediction horizon when trained with rainfall data is 10% for LSTM and 21% for RNN. The memory component of LSTM holds long spatial and temporal dependency. Therefore it can find the spatiotemporal relationship between the traffic and rainfall data.

It is observed from both the figures that the error corresponds to I-5 passing through the Downtown area is more than I-5 passing from the countryside. The

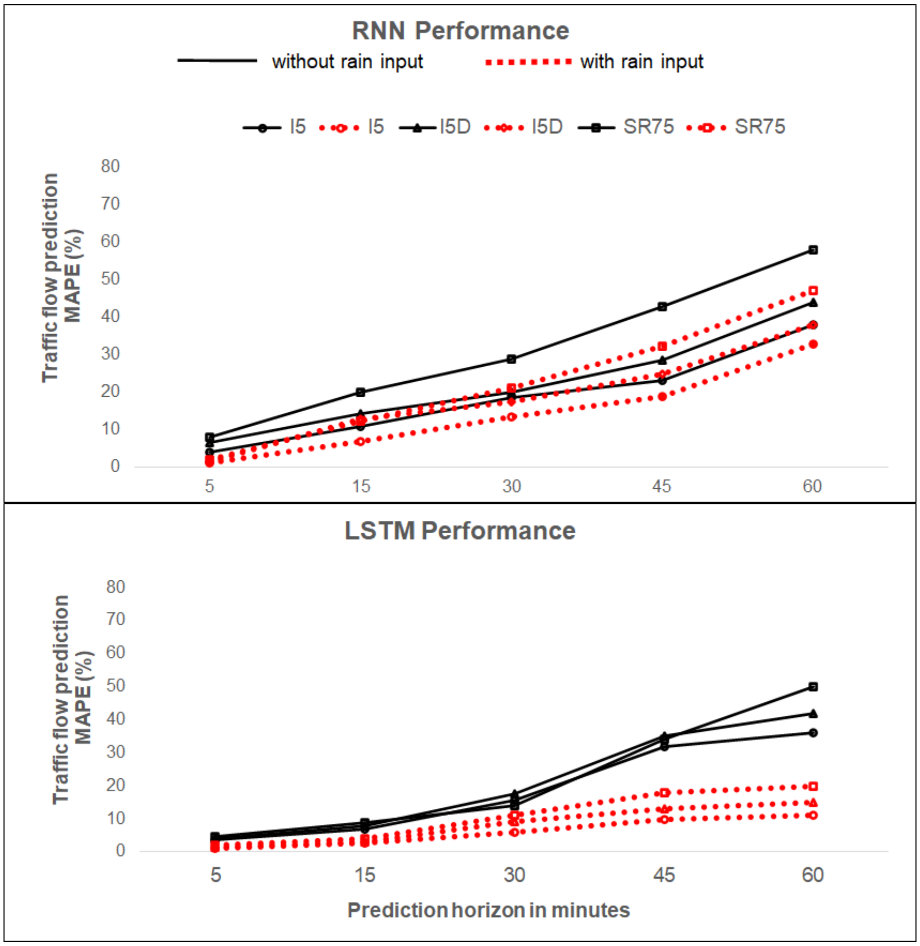


Figure 6.8: Traffic Flow Prediction by RNN and LSTM Model Trained With and Without Rain Input and Comparing MAPE Corresponding to I-5, I-5 Downtown, and SR-75 Roads.

LSTM traffic flow prediction error when trained without rainfall data is 32% and 43% for I-5 and I-5D, respectively, for 60-minute prediction horizon. The prediction error reduces with the inclusion of rainfall data. For I-5 road it is 11% and 15% for I-5D road. Downtown is a densely populated area, and the I-5 has a lower elevation near the Downtown area (41 - 72 m) then the countryside area (70 - 119 m). Therefore, the traffic flow and speed of I-5 passing through the Downtown area are affected more than the countryside road segments of I-5.

We observe that the prediction accuracy decreased with an increase in the prediction horizon. The reason for this phenomenon is that the multi-step prediction needs to use the previous prediction results. Unless the prediction is completely accurate, there must be an error between the predicted value and the observed value. Therefore, the larger is the step ahead of the prediction, the more errors of the previous prediction will be accumulated, leading to greater errors in the end.

6.1.3.1 Comparison with Other Deep Learning Models

We compare the performance of the recurrent learning models with the other deep learning models used in the literature. To compare the model performance, we use the same algorithm as suggested in the literature. The hyperparameters tuning is performed using a random search method such that the adopted model gives minimum prediction error for the particular set of hyperparameters. The deep learning models adopted from literature are Stacked AutoEncoder (SAE) [5], Back Propagation Neural Network (BPNN) [54], and Deep Belief Neural Network (DBNN) [27]. All the models are trained with rainfall and traffic data. The dataset corresponds to I-5 road segments is used for the performance comparison.

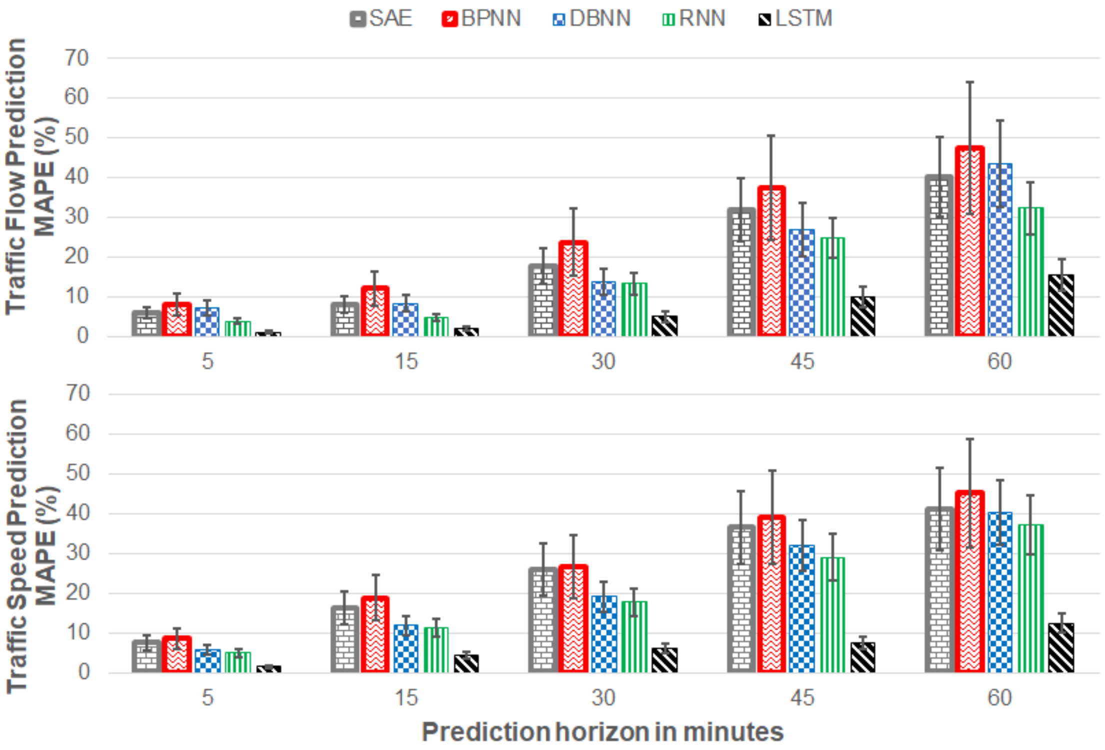


Figure 6.9: Traffic Flow and Speed Prediction Error Comparison of SAE, BPNN, DBNN, RNN and LSTM for Different Prediction Horizon for Traffic Data Along I-5. Error Bar Shows the Standard Deviation.

Figure 6.9 shows that for the prediction horizon of 15 minutes, the performance of SAE, BPNN, DBNN, and RNN has a negligible difference. However, LSTM shows the lowest error value. It indicates that spatial and temporal features improve the prediction accuracy of the model. For the prediction horizon (greater than 30 minutes), the performance of SAE is better than BPNN. The performance of RNN is better than SAE and DBNN, which indicates that spatial and temporal features are dependent on previous information. Therefore sequence learning models are better than non-sequential models. The traffic flow prediction error for LSTM is 15%, RNN is 41%, DBNN is 43%, SAE is 44%, BPNN is 55%, for 60-minute prediction horizon. The LSTM yields the lowest prediction error with a stable trend. The LSTM model holds long spatiotemporal dependency due to its gate mechanism compared to SAE, BPNN, and DBNN where the current input is independent of the past observations. The traffic speed prediction error for LSTM is 12%, RNN is 40%, DBNN is 43%, SAE is 49%, BPNN is 55% for 60-minute prediction horizon. From this result, we conclude that the LSTM model is suitable for holding long spatiotemporal dependency where future prediction is dependent on the past and current data. Therefore the prediction error of LSTM is less as compared to the non-recurrent models. With longer dependency, the RNN model suffers from the vanishing gradient problem, therefore for longer dependency performance of the LSTM is significantly better than the RNN.

6.1.3.2 Spatiotemporal Data vs Temporal Data Performance

Y. Jia *et al.* [26] used Deep Belief Network (DBN) and LSTM to predict the traffic flow on the road segment. The rainfall data from the nearest station is interpolated to the target road segment for traffic prediction. The temporal traffic and weather data are used to predict the traffic flow on a road segment. The experiment is performed on the arterial road segments of Beijing, China. The dataset is private and inaccessible. We trained our RNN and LSTM model with temporal traffic data of the I-5 road segment and rainfall data corresponding to the target road segment, ignoring the spatial dependency. The rainfall data from the nearest weather station is interpolated to the target road segment. We compare the MAPE of the LSTM and the RNN model when trained with spatiotemporal traffic and weather data and with the LSTM and RNN model when trained only with temporal traffic and weather data. Table 6.5 listed out the performance of LSTM and RNN considering spatiotemporal input vs temporal input. The MAPE recorded in [26] for DBN is 13% and 19% for 10-minute and 30-minute prediction horizon, respectively. The MAPE recorded for LSTM is 11% and 17% for 10-minute and 30-minute prediction horizons, respectively. In their result, the performance of the LSTM is better than the DBN considering temporal input. The performance of our LSTM model considering temporal input shows similar performance. The performance of the LSTM model trained with spatiotemporal input is better compared to the LSTM model trained with temporal input. The MAPE of the LSTM model trained with spatiotemporal input is 2% and 6% for 15-minute and 30-minute prediction horizons. The MAPE of the LSTM model trained with temporal input is 8% and 15% for 10-minute and 30-minute prediction horizons, respectively. The MAPE for the DBN and LSTM model is from the result reported in the study [26].

Table 6.5: MAPE Comparison Between Models Trained with Temporal Input vs Trained with Spatiotemporal Input, I-5 road segments.

Model	Input	**10 min**	**30 min**
LSTM	Temporal input	8%	15%
	Spatiotemporal input	2%	6%
RNN	Temporal input	12%	19%
	Spatiotemporal input	7%	12%
DBN [26]	Temporal input	13%	19%
LSTM [26]	Temporal input	11%	17%

Table 6.6 shows the MAPE comparison between the models for the SR-75 road segment. The difference between the MAPE of the two models (trained with spatiotemporal input and temporal input) is more significant for the SR-75 road segment compared to the I-5 road segment. The target road segment gets affected due to the rainfall at the other road segments because SR-75 is narrow and low elevation compared to I-5 road. The performance of the LSTM model trained with spatiotemporal input is better compared to the LSTM model trained with

temporal input. The MAPE of the LSTM model trained with spatiotemporal input is 4% and 10% for 10-minute and 30-minute prediction horizons, respectively. The MAPE of the LSTM model trained with temporal input is 11% and 19% for 10-minute and 30-minute prediction horizons, respectively. Therefore, it is important to consider the spatiotemporal dependency for the accurate prediction of traffic stream variables.

Table 6.6: MAPE Comparison Between Models Trained with Temporal Input vs Trained with Spatiotemporal Input, SR-75 Road Segments.

Model	Input	**10 min**	**30 min**
LSTM	Temporal input	11%	19%
	Spatiotemporal input	4%	10%
RNN	Temporal input	15%	28%
	Spatiotemporal input	11%	20%

6.1.3.3 Single Rainfall Data vs Sequential Rainfall Data Performance

Koesdwiady *et al.*[27] used DBN for traffic flow prediction during rainfall. The rainfall data is considered as complementary data, for traffic data the sequence length p is taken into account but only the most recent value of rainfall is considered for traffic flow prediction. The prediction horizon is 5-minute in this study. The author considers the spatiotemporal rainfall and traffic data. The experiment is performed on the arterial road segments of San Francisco. We trained our RNN and LSTM model with spatiotemporal traffic and rainfall data corresponds to I-5, the recent value of rainfall is considered as input. Table 6.7 shows the performance of LSTM, RNN, and DBN considering recent rainfall data for a 5-minute prediction horizon. The MAPE for the DBN model is from the result reported in the study [27]. The author reported the results for a 5-minute prediction horizon. For such a smaller prediction horizon, the dependency is not very large, the traffic prediction is dependent on recent traffic conditions, therefore the difference between models performance is not significant. For a 30-minute prediction horizon, the difference between the model considering only recent rainfall data and the model considering past and recent rainfall data is significant. The LSTM model performance when trained with the most recent rainfall data is 13% and when trained with sequential rainfall data is 6% for 30-minute prediction horizon. For longer dependency, the sequential rainfall data provides better prediction accuracy compared to the recent rainfall data.

Table 6.7: MAPE Comparison Between Models Trained with Recent Rainfall Data vs Trained with Sequential Traffic and Rainfall Data, I-5 Road Segments.

Model	Input	**5 min**	**30 min**
LSTM	Recent rainfall input	2%	13%
	Timeseries rainfall input	1%	6%
RNN	Recent rainfall input	2%	25%
	Timeseries rainfall input	1%	12%
DBN [27]	Recent rainfall input	4%	

We can conclude that the rainfall variable improves the traffic stream variables prediction accuracy of the deep learning model. The narrow and low-elevation roads are more affected by the rainfall than the broad and high-elevation, as traffic prediction error is higher for SR-75 than the I-5 when the model trained without rainfall data. The traffic condition on a road segment depends upon the traffic and weather conditions on adjacent roads, therefore, spatiotemporal input provides better prediction accuracy compared to the temporal input. The sequential rainfall input improves the prediction accuracy of the model compared to single rainfall input because of the prolonged dependency of rainfall on traffic stream variables. The prediction error increases with the prediction horizon due to the error accumulation.

6.2 Synthetic Data Calibration and Validation

We have developed the empirical model to generate the traffic data considering the rainfall impact on a variety of roads as shown in section 4.2.2. We perform a calibration to use the proposed empirical model to generate synthetic traffic data during adverse weather conditions. We generate the traffic data for different roads (arterial, sub-arterial, and collector) during rainfall. The simulation parameters are calibrated using the traffic data described in section 4.2.1.

Table 6.8: Description of Simulated Road Types.

Name	Description	Relevant Study data
$R_{arterial}$	An expressway road consists of 12 lanes (6 lanes in each headway); Speed limit is 70 km/hr; No waterlogging occurs on this road	Arterial road I-5
$R_{sub-arterial}$	A sub-arterial road consists of 8 lanes (4 lanes in each headway); Speed limit is 65 km/hr; passes through densely populated area; less prone to waterlogging	Sub-arterial road SR-75 and SR-163
$R_{collector}$	A collector road consisting of 4 lanes (2 lanes in each headway); Speed limit is 60 km/hr; passes through densely populated area; highly prone to waterlogging	Collector road BEP and MEP

Table 6.8 describes the simulated road types. We simulate the expressway ($R_{arterial}$), sub-arterial road ($R_{sub-arterial}$), and collector roads ($R_{collector}$), which corresponds to I-5, SR-75 and SR-163, and BEP and MEP roads, respectively, in the study data.

Table 6.9: Simulated Vehicle Flow Calibrated from Study Data for Different Types of Roads under Various Traffic and Weather Conditions.

Road Type and Traffic Scenario	Traffic Flow (x100 v/hr) Varying Rain Fall (mm per hour)				
	No rain 00 mm/hr	Low $\leq$ 0.5 mm/hr	Moderate $\leq$ 5 mm/hr	Heavy $\geq$ 7 mm/hr	Extended $\geq$ 40mm in 2hr
$R_{arterial}$ (peak)	45-62	42-62	35-58	27-53	18-46
$R_{arterial}$(off-peak)	30-56	27-53	25-46	19-40	15-34
$R_{arterial}$ (night)	21-38	19-33	15-25	12-21	10-18
$R_{sub-arterial}$ (peak)	40-55	38-53	33-48	25-39	12-30
$R_{sub-arterial}$(off-peak)	32-43	28-40	25-35	18-27	10-22
$R_{sub-arterial}$ (night)	20-24	16-22	12-19	10-15	06-09
$R_{collector}$ (peak)	28-38	24-35	20-32	15-20	08-12
$R_{collector}$(off-peak)	18-22	15-18	10-15	08-10	04-05
$R_{collector}$ (night)	05-09	04-07	03-06	01-03	01-1.5

As noted in section 4.2.2.1, rainfall has a varying effect on traffic flow and speed on different types of roads. We calibrate the traffic flow on the $R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$ under various traffic and weather conditions to observe the effect on traffic speed. It is expected that the traffic speed in the simulation environment is similar to one in the study data under the relevant traffic and weather condition. Table 6.9 describes the simulated vehicle flows on $R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$ under different traffic and weather conditions. As described earlier, for BEP and MEP (corresponding to $R_{collector}$ in simulation), the vehicle flow data is not available. Hence, the vehicle flow on $R_{collector}$ is simulated using the typical peak and off-peak hour traffic conditions reported on the Indian collector roads [105]. The traffic flow is reduced by 10 - 15% for the low to moderate rainfall, 25 - 30% for the heavy rainfall, and 50 - 60% for the prolonged heavy rainfall, based on the approximate knowledge of ground truth data.

Every simulation scenario is executed seven times by changing the random speed, and the aggregate traffic speed is recorded. The percentage decrease in traffic speed along the three types of roads ($R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$) under different traffic and rainfall conditions is shown in figure 6.10, 6.11, and 6.12. The variations in percentage decrease in the traffic speed are denoted by error bars in the graph. The height of an error bar equals the standard-deviation in percentage decrease in traffic speed.

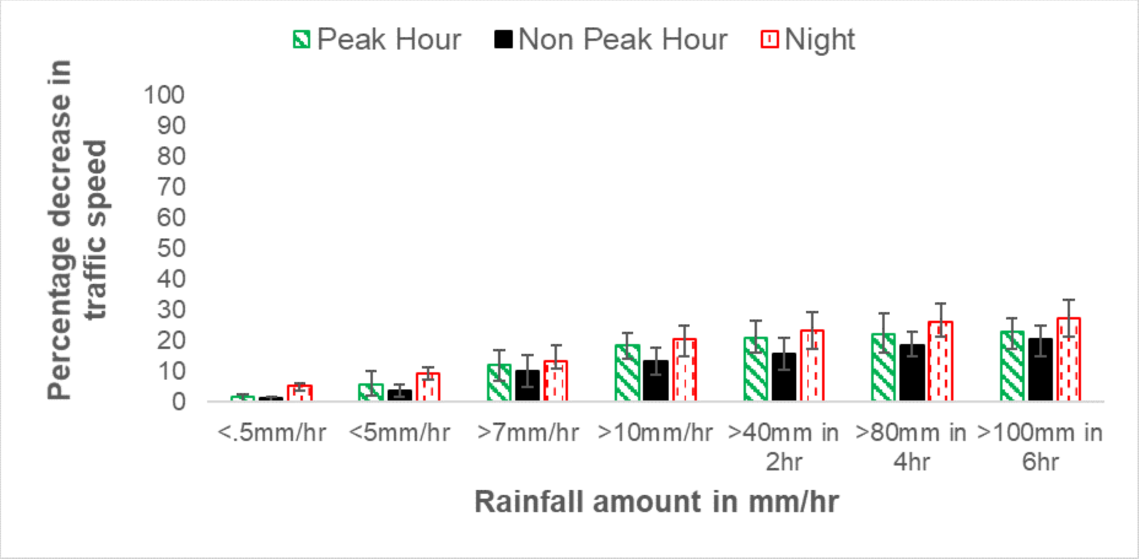


Figure 6.10: Percentage Decrease in Traffic Speed Due to Rainfall on $R_{arterial}$-type Road, Error Bar Reflects the Standard Deviation.

As shown in figure 6.10, the traffic speed decreases by 1.3 - 14% on the $R_{arterial}$ road under the light ($\leq$ 0.5 mm/hr), moderate ($\geq$ 5 mm/hr) and heavy ($\geq$ 7 mm/hr) rainfall condition. This decrease in speed observed in the simulations matches with the study data (table 4.8), where the decrease of 1.2 - 13 % in traffic speed on arterial roads is reported. As noted earlier, the effect of prolonged rainfall on traffic stream variables is not reported in the literature. Under this weather condition, the decrease of 14 - 27% in traffic speed is observed in simulations.

On the $R_{sub-arterial}$ roads, the traffic speed decreases by 1 - 21% under the light,

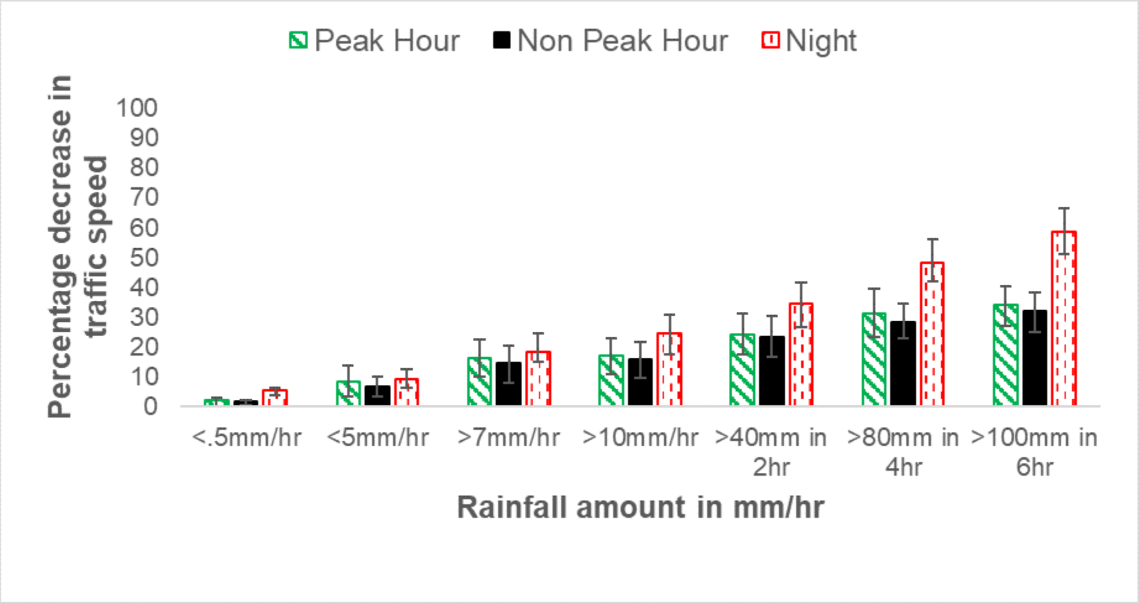


Figure 6.11: Percentage Decrease in Traffic Speed Due to Rainfall on $R_{sub-arterial}$-type Road, Error Bar Reflects the Standard Deviation.

moderate, and heavy rainfall conditions as shown in figure 6.11. This decrease in speed observed in the simulations matches with the study data (table 4.8), where the decrease of 1.2 - 16 % in traffic speed on sub-arterial roads is reported. Under the prolonged rainfall condition, a decrease of 18 - 59% in traffic speed is observed in simulations.

Figure 6.12 shows the speed decrease of 2 - 22% on $R_{collector}$-type roads under the light, moderate, and heavy rainfall conditions. This decrease in speed observed in the simulations matches with the study data (table 4.8) of the collector roads - BEP and MEP in Mumbai - where the decrease of 2 - 20% in traffic speed is reported. No literature study has analyzed the effect of rainfall on the collector roads, as noted earlier. The simulations report a decrease of 20 - 65% in traffic speed on collector roads under the prolonged rainfall condition.

From the figures 6.10 - 6.12 and table 4.8, it is clear that the simulation model replicates the effect of rainfall on traffic parameters as reported in the literature and study data (section 4.2.2.1).

The simulation results also provide additional insight into the effect of rainfall on traffic parameters during peak hours, off-peak hours, and night time. The effect of rainfall during peak-hours is more severe on all types of roads. On the

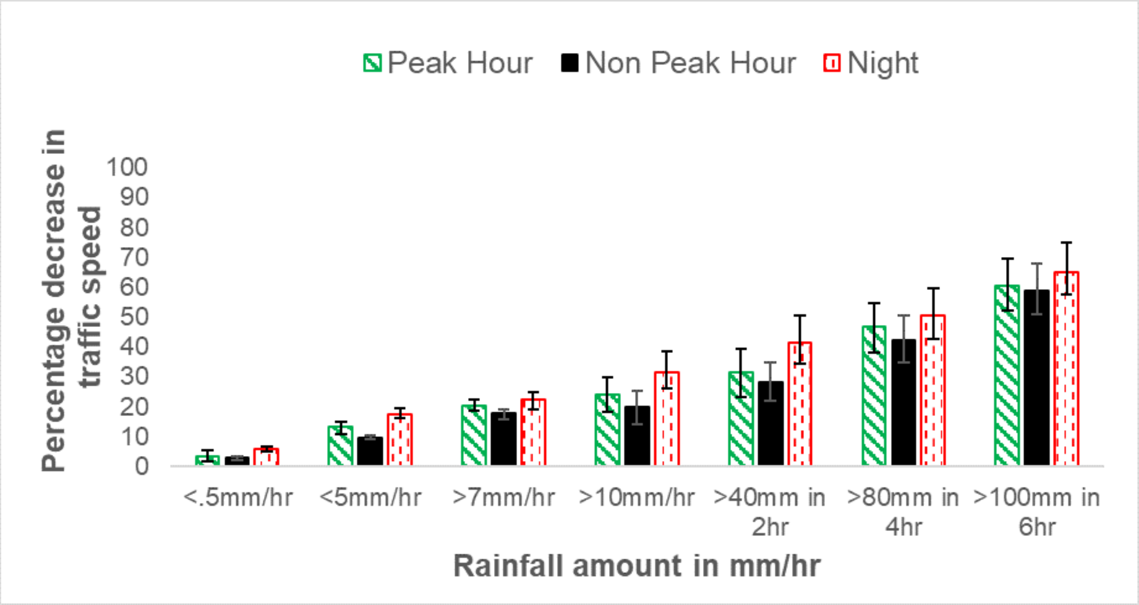


Figure 6.12: Percentage Decrease in Traffic Speed Due to Rainfall on $R_{collector}$-type Road, Error Bar Reflects the Standard Deviation.

$R_{arterial}$ roads, during peak hours, the traffic speed decreases by 1.82 - 22.98% due to rainfall, whereas during off-peak hours, the rainfall causes the decrease of 1.2 - 20% in traffic speed. On the $R_{sub-arterial}$ roads, the traffic speed decreases by 2 - 34% due to rainfall during peak hours. During off-peak hours, the rainfall causes a decrease of 2 - 32% in traffic speed. Similarly, on the $R_{collector}$ roads, during peak hours, the traffic speed decreases by 2 - 60% due to rainfall, whereas during off-peak hours, the rainfall causes the decrease of 2 - 58% in traffic speed.

A significant decrease in traffic speed is observed during night time due to rainfall. The low (≤ 0.5 mm/hr) to very heavy prolonged rainfall (≥ 100 mm in 6hrs) causes the percentage decrease in traffic speed during night time 2 - 27%, 5 - 58%, and 5 - 65% on the arterial, sub-arterial, and collector roads, respectively. Vehicle flow also decreases significantly during severe rainfall at night as drivers try to avoid night journeys due to the risk of major crashes or accidents.

6.2.1 Discussion

The following observations are made from the study data and the generated synthetic data:

On the $R_{arterial}$ road, we observe that the traffic speed decrease by 1.2 - 13.8% for the low to heavy rainfall. These numbers match with those reported by Ibrahim *et al.* [21] and M. Agarwal *et al.* [22]. On the $R_{sub-arterial}$ road, we observe the reduction of 2.1 - 16.56% in the traffic speed for the low to heavy rainfall. These numbers are similar to those reported by Wang *et al.* [72]. They observed a decrease of 1.72 - 16% in traffic speed. To the best of our knowledge, no study in the literature has analyzed the effect of rainfall on the collector roads. On the $R_{collector}$ road, we observe the traffic speed reduction of 2.58 - 20.43% for the low to heavy rainfall. These number matches with the Mumbai's MEP and BEP road data, the decrease of 2 - 20% in traffic speed is observed.

This opens the door for generating realistic synthetic data for a variety of traffic and weather conditions. The availability of realistic synthetic data enables further studies based on the data-driven models.

The rain causes a decrease in the traffic flow and speed, and this reduction is associated with rainfall intensity - the light rainfall has little impact on the traffic stream variables. The impact of rainfall on traffic stream variables is different on different types of roads. On the expressway ($R_{arterial}$), the rainfall impact is less than the collector road ($R_{collector}$). The effect of rainfall during peak-hours is more severe. The decrease in speed during peak hours is more than that in off-peak hours. A significant decrease in traffic flow and speed is observed during nighttime.

6.3 Deep Learning Model Performance with Synthetic Data

In the previous section, we generated synthetic data corresponding to different rainfall intensities for three roads (arterial, sub-arterial, and collector). To validate learning model performance, we generate traffic data using a similar empirical model and simulation study, for the worst-case rainfall scenario. We consider the case when both target and other roads receive heavy rainfall for a prolonged time greater than 100 mm in 6 hours. Table 6.8 describes the three simulated road types ($R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$). To validate learning model performance considering small and large size road networks, synthetic data is generated for three road networks as shown in table 6.10, for each road type. The small road network size is 5 × 7 km, and the size of the large road network is 20 × 28 km. Here rainfall station symbol is for the representation purpose.

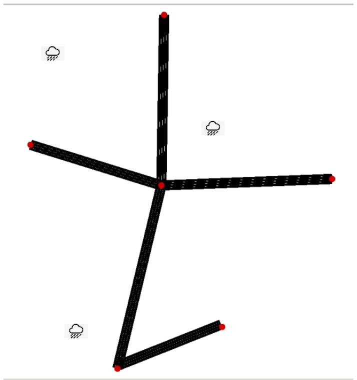


Figure 6.13: Small Road Network.

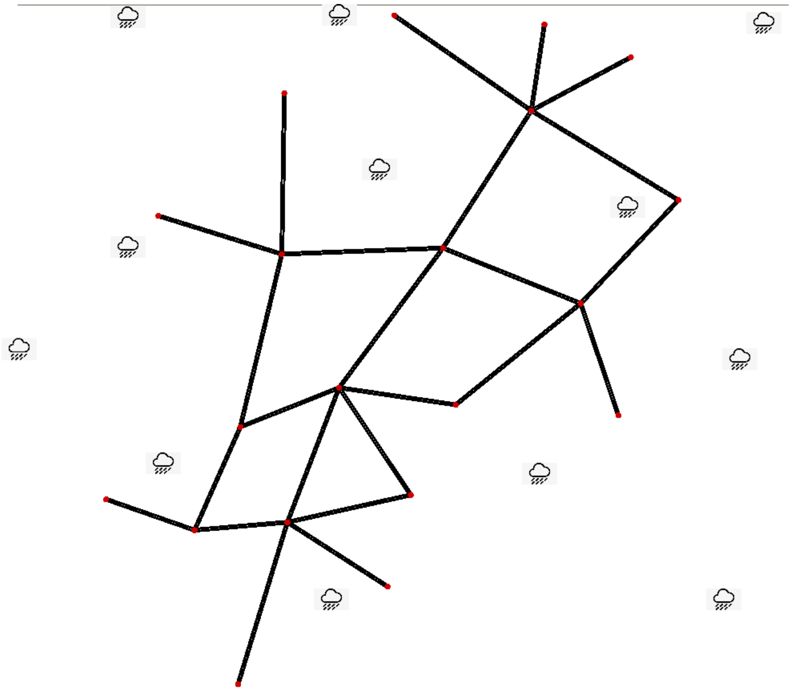


Figure 6.14: Large Road Network.

Table 6.10: Scenarios to Test Model Performance.

Scenarios	Description
S1	Single road with point traffic and weather data
S2	Small road network, figure 6.13
S3	Large road network, figure 6.14

In all three scenarios, the following is applicable. We use the random search method for hyperparameter tuning [80]. The models are trained based on Adam optimizer [85]. The learning rate is set to be 0.005, and the batch size is set to 256. To avoid overfitting, L2 regularisation is used. The normal distribution is used to initialize the set of parameters. For the implementation, we used the TensorFlow version 1.0.1 framework, and GPU is used to improve the computational capability and fasten the training module's execution. For the programming, Python version 2.7 is used.

Appendix D discusses the experimental setup and result analysis for the single road (S1) and the small road network (S2).
For the single road, there is no dependency on other road segments. The traffic and weather data are available as point data. When trained with rain data, the model's traffic speed prediction improves compared to the model's traffic speed prediction when trained without rain data. For the single road, hybrid and non-hybrid model's performance shows no significant difference as there is no dependency from other road segments are involved.

Considering the small road network, for road type $R_{sub-arterial}$ and $R_{collector}$, the target road traffic variables get affected due to rainfall at adjacent roads, as water flows from one road segment to another. Similarly, waterlogging makes the situation worse. Therefore for road type $R_{sub-arterial}$ and $R_{collector}$ the performance of LSTM-LSTM and CNN-LSTM shows significant improvement over LSTM, CNN, and RNN models. For the prediction horizon ($>$ 30 minutes) the traffic prediction error improvement of the hybrid model is 10% over the non-hybrid models.

6.3.1 Experimental Setup and Result Analysis for Large Road Network

We generate the traffic data for a large road network as shown in figure 6.14 considering each road segment receives an average rainfall of 100 mm in 6 hours (scenario S3). The traffic data is generated separately for arterial ($R_{arterial}$), sub-arterial ($R_{sub-arterial}$) and collector road ($R_{collector}$). The traffic and weather condition of other road segment affects the traffic stream variables of the target road segment. The dependency in this scenario is large compared to the other two scenarios, as the impact of rainfall on other road segments (distant one) is significant on the target road segment at some later time instance.

6.3.1.1 Experimental Setup

Table 6.11 and 6.12 shows the most useful set of hyperparameters for the CNN-LSTM and LSTM-LSTM model, respectively. The hyphen's left side's value indicates the hyperparameter for the first sub-model (CNN/LSTM), and the right side value shows the hyperparameter for the second sub-model (LSTM). The hyperparameters for the other models (RNN, CNN, and LSTM) is shown in Appendix E. For a large road network of road type $R_{arterial}$, the lag value of the former model in the hybrid model for 15 to 30 minutes prediction horizon is in the range [15-22], and for the later LSTM model, the range is [10-15]. The number of the epoch of hybrid models is in the range [15-20] for 15 to 30 minutes prediction horizon and [27-40] for 45 to 60 minutes prediction horizon. The spatial threshold value Δ is in the range [3-5] km. The value of weight in weighted moving average (γ) is in the range [0.5-0.7]. The number of layers is in the range [2-3].

For $R_{sub-arterial}$ road, the lag value of the former model in the hybrid model is in the range [18-24] and for the later LSTM model is in the range [12-18]. For the hybrid models, the epoch value for the 15 to 30 minutes prediction horizon is in the range [18-22] and for the 45 to 60 minutes prediction horizon is in the range [30-45]. The spatial threshold value Δ is in the range [3-7] km.

For $R_{collector}$ road, for the CNN-LSTM model, the lag value of the former model is in the range [20-30] and for the later LSTM model is in the range [12-18] for 15 to 45 minutes prediction horizon and is in the range [18-23] for 60 minutes prediction horizon. For the LSTM-LSTM model, the lag value of the former model is in the range [20-30] and for the later LSTM model is in the range [17-20] for 15 to 30 minutes prediction horizon and is in the range [22-26] for 45 to 60 minutes prediction horizon. For the CNN-LSTM model, the epoch value is in the range [18-25] for 15 to 30 minutes prediction horizon and is in the range [40-50] for 45 to 60 minutes prediction horizon. For the LSTM-LSTM model, the epoch value is in the range [20-28] for 15 to 30 minutes prediction horizon and is in the range [40-55] for 45 to 60 minutes prediction horizon. The spatial threshold value Δ is in the range [3-7] km.

The reason for providing this range is because for a new data-set considering similar road types for prolonged rainfall conditions, the prediction of the model using the hyperparameters within the range will not affect much the performance of the models for the particular prediction horizon.

Table 6.11: Hyperparameters for the CNN-LSTM Model.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	18-12	1-1	[16,3×3-12]	15	.6	3
	30	18-12	2-1	[(32,3×3),(16,3×3)-20]	18	.7	5
	45	22-15	3-2	[(32,3×3),(16,3×3)-(30,10)]	27	.7	5
	60	22-15	3-2	[(32,3×3),(32,3×3),(16,3×3)]-[52,20]	35	.7	7
$R_{sub-arterial}$	15	18-12	1-1	[16,3×3-18]	18	.6	5
	30	20-12	2-1	[(32,3×3),(16,3×3)-24]	20	.7	5
	45	22-15	3-2	[(32,3×3),(16,3×3)-(30,15)]	30	.7	7
	60	22-18	3-3	[(48,3×3),(32,3×3),(16,3×3)]-[50,30,20]	40	.7	7
$R_{collector}$	15	20-12	1-1	[16,3×3-22]	18	.6	5
	30	24-15	2-1	[(32,3×3),(16,3×3)-28]	20	.7	7
	45	24-18	3-2	[(32,3×3),(16,3×3)-(40,20)]	40	.7	7
	60	28-22	3-3	[(64,3×3),(32,3×3),(16,3×3)-(60,40,20)]	50	.7	7

6.3.1.2 Result Analysis for Large Road Network

We examine the performance of models for the large road network (scenario 3) as shown in figure 6.14. We test the model performance for the worst-case scenario

Table 6.12: Hyperparameters for the LSTM-LSTM Model.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	18-12	2-1	[20,40]-[20]	16	.5	5
	30	18-12	2-2	[30,70]-[20,10]	19	.7	5
	45	22-12	3-2	[20,40,60]-[30,10]	35	.7	7
	60	22-15	3-2	[30,50,80]-[40,20]	40	.7	7
$R_{sub-arterial}$	15	20-12	2-1	[20,50]-[25]	18	.5	5
	30	20-12	2-2	[30,80]-[30,10]	20	.7	7
	45	22-15	3-2	[30,60,90]-[40,20]	37	.7	7
	60	24-15	3-3	[50,80,100]-[50,30,10]	45	.7	7
$R_{collector}$	15	18-18	2-1	[30,50]-[28]	20	.5	7
	30	22-20	2-2	[40,80]-[30,20]	26	.7	7
	45	27-24	3-2	[30,60,90]-[50,20]	45	.7	7
	60	30-24	3-3	[50,80,110]-[60,40,20]	55	.7	7

where both the target and other roads receive prolonged heavy rainfall ($\geq$ 100 mm in 6 hours).

Figure 6.15 shows the performance of models for traffic speed prediction for road type $R_{arterial}$ when trained without and with rainfall data. From the result, it is observed that the inclusion of rainfall variables decreases the error of the model. For the 60-minute prediction horizon, the traffic speed prediction error for LSTM-LSTM is 42%, and CNN-LSTM is 47% when the model is trained without rainfall variable. When both the models are trained with rainfall variable, then traffic speed prediction error for LSTM-LSTM is 9%, and CNN-LSTM is 11%. The CNN-LSTM and LSTM-LSTM model's performance is better in extracting spatiotemporal and temporal dependency than the model that considers only spatiotemporal dependency. When trained with the rainfall variable, RNN, CNN, and LSTM model's performance is 43%, 40%, and 29%, respectively, for a 60-minute prediction horizon. For 15 minutes prediction horizon, the traffic speed prediction error of RNN is 13%, CNN is 13%, LSTM is 10%. LSTM-LSTM is 1% and CNN-LSTM is 3%. With the higher value of the prediction horizon, the model's prediction error increases due to error accumulation. Therefore, a significant difference is observed between the performance of hybrid and non-hybrid models. The model's traffic speed prediction performance when trained with rainfall variables in MAE and RMSE is discussed in Table 6.13. The MAE and RMSE are in agreement with MAPE. LSTM-LSTM model performance is better than other models.

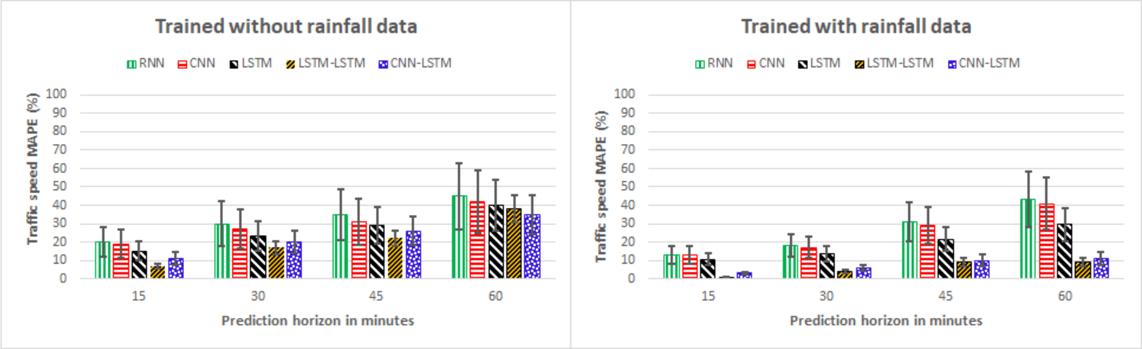


Figure 6.15: Traffic Speed Prediction Performance of Models without and with Rain Data for Simulated $R_{arterial}$ Road. Error Bar Shows the Standard Deviation.

Table 6.13: Traffic Speed Prediction Error of Different Models when Trained with Rainfall Data for Road Type $R_{arterial}$.

Model	Error	15 min	30 min	45 min	60 min
RNN	MAE	8.7	14.3	19.4	23.4
	RMSE	13.8	21.8	29.7	36.5
CNN	MAE	4.8	6.6	9.8	13.2
	RMSE	11.8	15.5	24.5	32.1
LSTM	MAE	3.76	5.4	7.4	11.8
	RMSE	9.6	12.6	22.8	30.7
CNN-LSTM	MAE	2.1	3.1	4.7	5.7
	RMSE	7.5	10.7	18.5	25.4
LSTM-LSTM	MAE	1.2	1.7	2.5	4.2
	RMSE	5.4	8.3	14.8	21.9

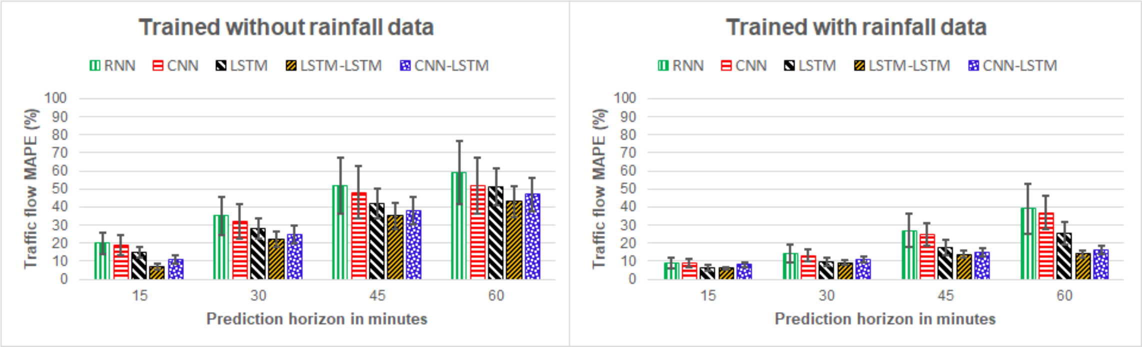


Figure 6.16: Traffic Flow Prediction Performance of Models without and with Rain Data for Simulated $R_{arterial}$ Road. Error Bar Shows the Standard Deviation.

Figure 6.16 shows the traffic flow prediction performance of the models for road type $R_{arterial}$. The figure shows that the traffic flow prediction error of models trained without rainfall variables is higher than the models trained with rainfall variables. The traffic flow prediction error (60-minute prediction horizon), when trained without rainfall variables is 51% and 55% for LSTM-LSTM and CNN-LSTM models, respectively. The traffic flow prediction error for a similar prediction horizon decreases when trained with rainfall variables. The error is 14% and 15% for LSTM-LSTM and CNN-LSTM models, respectively. When trained without rainfall variable, the model's error is more for traffic flow than traffic speed. For the road type $R_{arterial}$, we assume no waterlogging; therefore, the speed reduction is due to reduced visibility or avoiding crash risk during heavy rainfall. But considering the traffic flow, the impact of rainfall on other road segments affects the traffic flow at the target road segment. For 60 minutes prediction horizon, the traffic flow prediction error of the non-hybrid models, RNN is 39%, CNN is 36%, and LSTM is 25%. The hybrid model's traffic flow prediction performance is better than the non-hybrid model for traffic flow prediction in extracting longer dependencies. But there is no significant difference between the performance of hybrid models observed for large road networks consisting of arterial roads.

Figure 6.17 shows the performance of models for traffic speed prediction for road type $R_{sub-arterial}$ when trained without and with rainfall data. The result shows that the inclusion of rainfall variables decreases the prediction error of the models. For the 60-minute prediction horizon, the traffic speed prediction error for LSTM-LSTM is 48%, and CNN-LSTM is 59% when the model is trained without rainfall variable. When both the models are trained with rainfall variable, then traffic speed prediction error for LSTM-LSTM is 10%, and CNN-LSTM is 22%. For road type $R_{sub-arterial}$, the model's prediction error when trained without rainfall variable is higher than road type $R_{arterial}$. The $R_{arterial}$ road is exceptional, and traffic speed reduction due to rainfall is less than the $R_{sub-arterial}$ road. The prolonged heavy rainfall affects $R_{sub-arterial}$ roads more compared to $R_{arterial}$ road. Therefore, the error in traffic speed prediction is more in $R_{sub-arterial}$ compared to $R_{arterial}$. With the inclusion of the rainfall variable, for 15 minutes prediction horizon, the traffic speed prediction error of RNN is 15%, CNN is 14%, LSTM is 12%, LSTM-LSTM is 2%, and CNN-LSTM is 7%. For 60 minutes prediction horizon, the traffic speed prediction error of RNN is 45%, CNN is 42%, LSTM is 31%, LSTM-LSTM is 10%, and CNN-LSTM is 22%. The CNN-LSTM and LSTM-LSTM model's performance is better in extracting spatiotemporal and temporal dependency than the model that considers only spatiotemporal dependency. The LSTM-LSTM model shows better prediction accuracy compared to the CNN-LSTM model. The RNN and CNN model doesn't hold such long-term dependencies. RNN suffers from vanishing gradient problem, and the CNN model performs local dimensionality reduction, which is limited by their localized receptive field.

The model needs to handle large memory proportional to the sequence length to handle longer dependencies, making it difficult to scale to a large problem. Therefore for learning long-term dependency, the LSTM model shows significant performance. The LSTM model is designed initially to improve gradient flow in a recurrent network such that the network can learn long-term dependencies, which is achieved with proper use of the gating mechanism. In the LSTM-LSTM model, the first LSTM model learns the spatiotemporal dependency between weather and traffic data. It learns both short and long-term dependencies. The second LSTM uses the feature vector generated from the first LSTM and traffic data from the target road segment for traffic forecasting. The model's traffic speed prediction performance when trained with rainfall variables in MAE and RMSE is discussed in Table 6.14, which reflects the same findings as discussed above, the performance of the LSTM-LSTM model is better than other models.

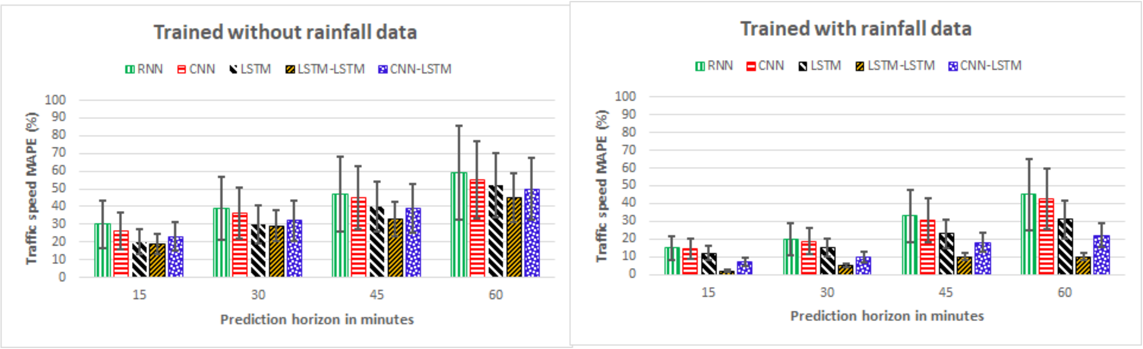


Figure 6.17: Traffic Speed Prediction Performance of Models without and with Rain Data for Simulated $R_{sub-arterial}$ Road. Error Bar Shows the Standard Deviation.

Table 6.14: Traffic Speed Prediction Error of Different Models when Trained with Rainfall Data for Road Type $R_{sub-arterial}$.

Model	Error	15 min	30 min	45 min	60 min
RNN	MAE	17.56	20.42	26.45	32.22
	RMSE	25.4	31.2	39.7	49.7
CNN	MAE	10.42	14.34	17.45	22.24
	RMSE	15.7	24.5	35.4	43.3
LSTM	MAE	7.34	12.42	15.35	20.24
	RMSE	11.2	20.4	30.2	39.6
CNN-LSTM	MAE	4.5	7.2	10.4	15.2
	RMSE	9.4	16.3	26.5	35.4
LSTM-LSTM	MAE	2.2	4.25	5.2	7.8
	RMSE	7.8	15.4	23.2	31.23

Figure 6.18 shows the traffic flow prediction performance of the models for road type $R_{sub-arterial}$. The traffic flow prediction error for 60-minute prediction horizon when trained without rainfall variables is 59% and 64% for LSTM-LSTM and CNN-LSTM models, respectively. The traffic flow prediction error of the model for a similar prediction horizon when trained with rainfall variables is 14% and 26% for LSTM-LSTM and CNN-LSTM models, respectively. For a similar prediction horizon and with the inclusion of rainfall variable, the traffic flow prediction error of RNN is 47%, CNN is 44%, LSTM is 33%. The impact of rainfall

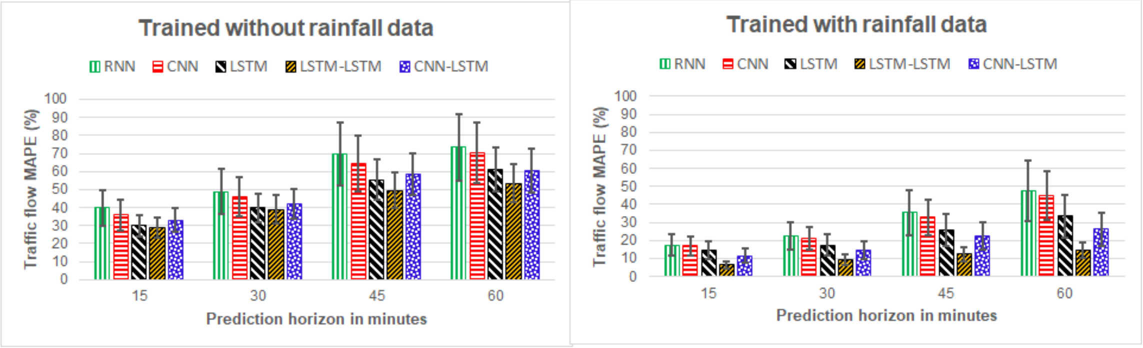


Figure 6.18: Traffic Flow Prediction Performance of Models without and with Rain Data for Simulated $R_{sub-arterial}$ Road. Error Bar Shows the Standard Deviation.

is more on traffic flow than traffic speed because traffic speed comes to the saturation point after prolonged heavy rainfall. People avoid traveling during prolonged heavy rainfall, especially in off-peak hours, which drastically affects traffic flow. Therefore, the performance of the LSTM-LSTM model and other models show a significant difference.

Figure 6.19 shows the performance of models for traffic speed prediction for $R_{collector}$ type road (highly prone to waterlogging) when trained without and with rainfall data. For the LSTM-LSTM model, when trained without rainfall data, the prediction error (60-minute prediction horizon) for road type $R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$ is 42%, 48%, and 59%, respectively. The impact of prolonged heavy rainfall is most significant on $R_{collector}$ type road as these are prone to waterlogging and narrow compared to $R_{arterial}$ and $R_{sub-arterial}$ roads. The prediction error decrease with the inclusion of the rainfall variable. There is a significant difference in the performance of LSTM-LSTM and CNN-LSTM. With the inclusion of the rainfall variable, for 15 minutes prediction horizon, the traffic speed prediction error of RNN is 21%, CNN is 19%, and LSTM is 17%, and LSTM-LSTM is 3%, and CNN-LSTM is 16%. For 60 minutes prediction horizon, the traffic speed prediction error of RNN is 51%, CNN is 47%, LSTM is 36%, LSTM-LSTM is 11%, and CNN-LSTM is 31%. There is a considerable drop in CNN and CNN-LSTM model performance for a large-scale problem because there is a more extended spatiotemporal dependency. For a large network, the LSTM-LSTM model's performance is better than the other models considering the impact of other roads ′ traffic and weather conditions on target road traffic conditions. The LSTM-LSTM model learns the

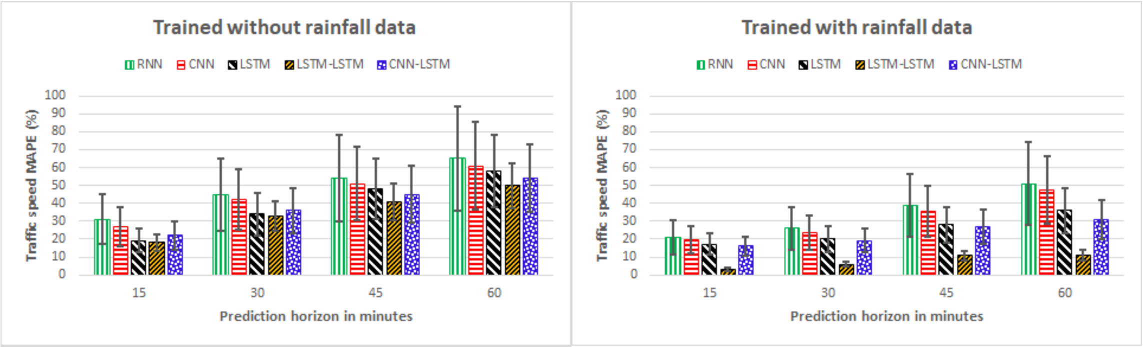


Figure 6.19: Traffic Speed Prediction Performance of Models without and with Rain Data for Simulated $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

longer spatiotemporal and temporal dependency more accurately as compared to other models. The model's traffic speed prediction performance when trained with rainfall variables in MAE and RMSE is discussed in Table 6.15, which shows that the performance of the LSTM-LSTM model is better than the other models.

Table 6.15: Traffic Speed Prediction Error of Different Models when Trained with Rainfall Data for Road Type $R_{collector}$.

Model	Error	15 min	30 min	45 min	60 min
RNN	MAE	21.22	26.45	32.4	37.7
	RMSE	32.3	40.23	52.1	64.2
CNN	MAE	10.2	15.3	22.2	29.7
	RMSE	18.3	31.2	44.5	58.7
LSTM	MAE	7.3	12.2	18.7	24.5
	RMSE	15.6	27.5	36.6	55.3
CNN-LSTM	MAE	6.09	9.99	17.5	21.5
	RMSE	14.4	29.4	35.2	49.4
LSTM-LSTM	MAE	2.2	4.5	6.1	8.3
	RMSE	10.2	20.3	29.3	36.4

Figure 6.20 shows the traffic flow prediction performance of the models for road type $R_{collector}$. The traffic flow prediction error for the 60-minute prediction horizon when trained without rainfall variables is 59% and 62% for LSTM-LSTM and CNN-LSTM models, respectively. The traffic flow prediction error when

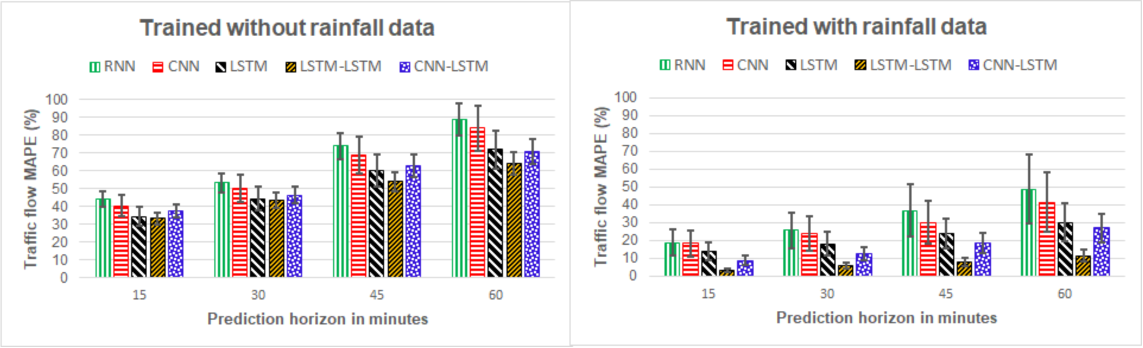


Figure 6.20: Traffic Flow Prediction Performance of Models without and with Rain Data for Simulated $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

trained with rainfall variables is 11% and 28% for LSTM-LSTM and CNN-LSTM models, respectively. For a similar prediction horizon with the inclusion of rainfall variable, the traffic flow prediction error of RNN is 48%, CNN is 41%, and LSTM is 30%. The impact of traffic flow is significant on $R_{collector}$ type road. Due to waterlogging, road capacity reduces. In addition to that, the traffic flow drastically reduces because the driver avoids waterlogging sites.

One of the major observations from the results is that the model's prediction error increased with the prediction horizon because of error accumulation. There is a longer dependency for the large road network scenario in terms of the road network and the prolonged rainfall impact. With the non-hybrid models, the error accumulated will be more. Therefore, there is a large difference between the prediction accuracy of the hybrid and non-hybrid models. The LSTM-LSTM model's performance is better than the other non-hybrid and hybrid CNN-LSTM models for the longer dependency. The impact of rainfall is significant on all road types. The prediction error of the models reduces with the inclusion of the rainfall data for all road types. But for the collector road types, the error of the models when trained without rainfall data is higher than the arterial and sub-arterial road types. We can conclude that the performance of the LSTM-LSTM model is best in all three scenarios (from the best to the worst case).

6.3.2 Hybrid Models Comparison with Existing Models

There is two existing work that performs the traffic flow prediction during adverse weather conditions. We compare the prediction performance of our developed hybrid models with the existing models.

We developed the DBN model as proposed by Y. Jia *et al.* [26] and called it DBN (1). The author also implemented the LSTM model, but we have already compared the performance of the hybrid model with the LSTM model. Therefore, in this experiment, we are comparing the hybrid model only with DBN (1). To predict the traffic flow on the target road segment, temporal traffic, and rainfall data corresponds to the target road segment are used in this study. Rainfall data from the nearest station is interpolated to the target road segment. For the comparison, we train the DBN(1) with temporal traffic and weather data. Another study was done by A. Koesdwiady *et al.* [27], used the DBN model to predict the traffic flow on the road segment. The prediction horizon of 5 minutes is considered in their study. We developed the DBN model as proposed in their work and called it DBN (2). The author estimates the traffic flow on the target road segment, spatiotemporal traffic, and rainfall data is used. As per their assumption, for traffic flow prediction sequence of traffic readings from past and current is needed, but rainfall data is complementary. Therefore, only the most recent rainfall value is provided as input to the model. For the comparison, we train the DBN (2) with spatiotemporal traffic and weather data. The traffic data up to optimal lag value is provided for the model training, but rainfall reading from the current time instance is provided.

As per previous results, hybrid deep learning model's performance is better than the non-hybrid models. Specifically for the worst-case scenario, i.e., large road network and prolonged rainfall. The LSTM-LSTM model's performance surpasses the other hybrid and non-hybrid models in extracting the long-term dependency of traffic and weather data. There is a need for a common dataset to compare our hybrid model's performance with the existing models for traffic flow prediction during rainfall. The traffic data used in the literature are either private or the complete dataset information is not shared. Therefore, to compare the

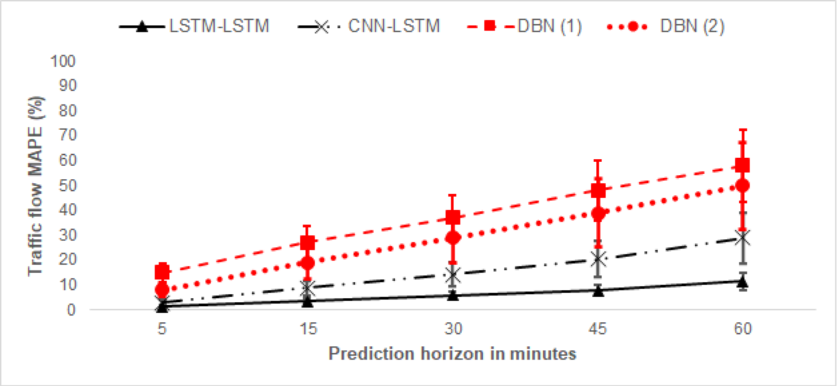


Figure 6.21: Traffic Flow MAPE Comparison of Hybrid and Existing Models. Error Bar Shows the Standard Deviation.

model on a similar dataset, we used our synthetic data. We compare the model's performance for large road network scenarios, and road type is $R_{collector}$. The traffic data is generated considering the average prolonged rainfall case (100 mm in 6 hours). Both the existing work perform traffic flow prediction only, not considering the traffic speed, therefore for the comparison, we also perform traffic flow prediction only. As traffic speed measurement requires a double loop detector, therefore the majority of the roads with ITS infrastructure provide traffic flow measurement only. The fine-grained synthetic data will solve this issue by providing the approximate measurement of the traffic speed and flow.

Figure 6.21 shows the traffic flow prediction performance of the DBN (1), DBN (2), LSTM-LSTM and CNN-LSTM model for different prediction horizons. The result shows that for a smaller prediction horizon (5 minutes) there is a small difference in model's performance. The traffic flow prediction error of LSTM-LSTM is 2%, CNN-LSTM is 4%, DBN (1) is 12% and DBN (2) is 8%. With the increase in the prediction horizon, we can observe the difference between model's performance increase significantly. For 60 minutes prediction horizon, the traffic flow prediction error of LSTM-LSTM is 11%, CNN-LSTM is 28%, DBN (1) is 58% and DBN (2) is 50%. DBN (1) consider only temporal traffic and rainfall input ignores the spatiotemporal dependency. DBN (2) consider only recent rainfall value for traffic flow prediction. Considering the large road network and prolonged rainfall scenario, the traffic stream variables on the target road segment get affected due to traffic and rainfall conditions on the other road segment. We are considering the $R_{collector}$ roads that are prone to waterlogging, which greatly affects the traffic flow on target and other road segments. The traffic and rainfall information from other road segments is important for making an accurate prediction. Similarly, the extended rainfall information is necessary to capture the prolonged impact of rainfall. In DBN, the current input is independent of the previous outcome. Our LSTM-LSTM model learns the longer dependency due to the gating mechanism, which works as a memory to store the previous outcome and use it with current input for prediction. We conclude that the LSTM-LSTM model's performance is better than the existing models.

6.3.3 Model Robustness

A model is considered robust if changing the hyperparameter value by a small amount doesn't affect the model's prediction accuracy significantly. We examine the hybrid models for different threshold values and analyze the changes in prediction accuracy.

6.3.3.1 Soft Temporal Threshold

A more extended past sequence is required to observe the impact of prolonged rainfall, but a more extended sequence increases the model's training time and the number of parameters. The soft temporal threshold considers the weather data sequence up to the lag value and exponential moving average of the past weather data to consider the prolonged impact of weather conditions. In the hard threshold, only the weather data sequence up to the lag value is considered. In figure 6.22, the performance of models trained with soft temporal threshold and hard threshold are compared considering the 60-minute prediction horizon for road type $R_{collector}$. The result shows that for the LSTM-LSTM model, the traffic speed prediction error with a hard threshold is 17%, and with a soft temporal threshold is 11%. For the CNN-LSTM model, the traffic speed prediction error with hard and with soft temporal threshold is 39% and 31%, respectively.

Therefore, including the information related to the extended rainfall sequence will increase the model's accuracy.

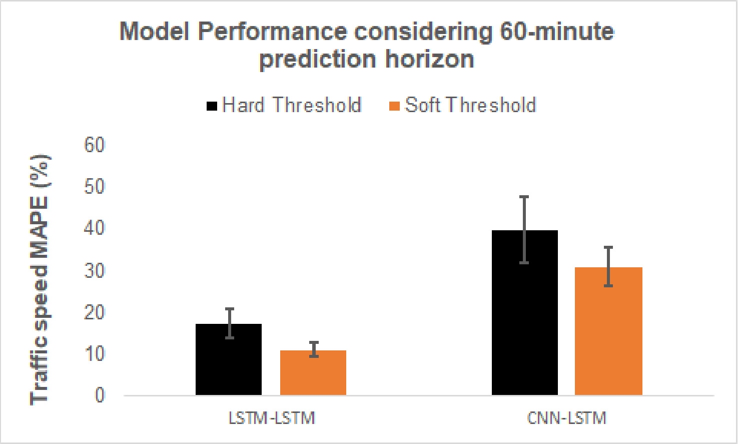


Figure 6.22: Traffic Speed MAPE Comparison for Hard and Soft Temporal Threshold for the 60-minute Prediction Horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

Figure 6.23 shows the model performance for 30-minute prediction horizon. The result shows that the model gives the best performance for weighing factor γ equals to 0.7. The traffic speed prediction error of LSTM-LSTM model when $\gamma = 0$ is 9% and for $\gamma = 0.7$ is 5%. The traffic speed prediction error of CNN-LSTM model when $\gamma = 0$ is 24% and for $\gamma = 0.7$ is 18%. The CNN-LSTM and LSTM-LSTM models are robust, and there is no significant change in prediction output with a small change in the value of γ. The traffic speed prediction error of the LSTM-LSTM model for $\gamma = 0.7$ and $\gamma = 0.9$ is 5.82% and 5.99%, respectively.

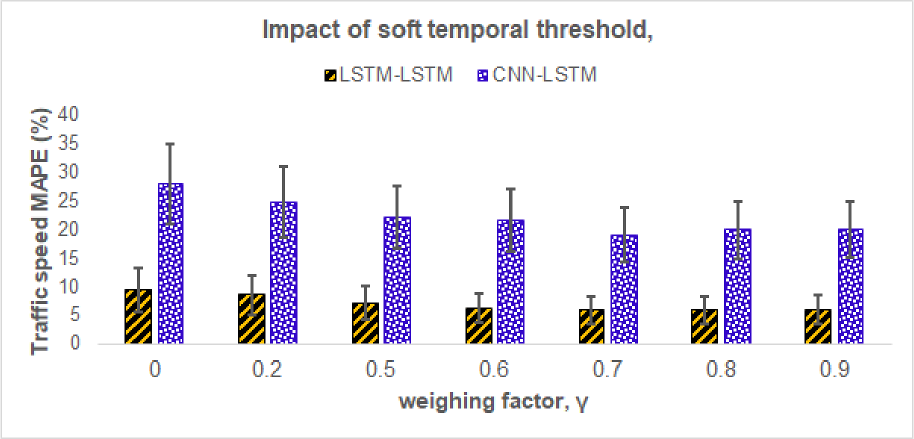


Figure 6.23: Traffic Speed MAPE Comparison for Different Value of Weighing factor (γ), for the 30-minute Prediction Horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

6.3.3.2 Soft Spatial Threshold

Due to the poor drainage system and saturated soil, prolonged rainfall causes waterlogging. The traffic stream variables on the target road get affected due to waterlogging and water flow from the other roads during rainfall. For an accurate prediction of traffic stream variables on the target road segment during rainfall, along with the traffic data from the target road segments, the effect of rain on the other road segments in the geographical location needs to be captured. We proposed the soft spatial threshold mechanism to capture the spatial significance of weather conditions, as discussed in section 4.2.3.1. The rainfall data from stations under the threshold distance are considered, and for all other stations, the weighted average rainfall value is used as the input. In a large road network scenario S3, there are 24 roads. Each road is receiving an average rainfall of 100 mm in 6 hours. The traffic and weather data from other roads are passed to predict the traffic stream variables at the target road. Instead of passing the rainfall value corresponding to each road, we used a soft spatial threshold mechanism. We set the threshold distance as a hyperparameter. The distance up to which rainfall value corresponds to all road segments is passed, and for all the other roads falling outside the threshold value, we pass the weighted average rainfall value.

Figure 6.24 shows the model's performance for different threshold values for the 30-minute prediction horizon. For the threshold value equals to 7 km model gives the minimum error. With the change in the threshold value, no significant difference in performance is observed, which shows the model's robustness. The traffic speed prediction error of LSTM-LSTM model for threshold value 3, 7, and 24 is 9%, 5%, and 7%, respectively. The traffic speed prediction error of CNN-LSTM model for threshold value 3, 7, and 24 is 22%, 18%, and 21%, respectively. The error increase for larger threshold value because of overfitting. The result concludes that the weather data from other road segments improve the model's accuracy while predicting the traffic stream variables at the target road segment.

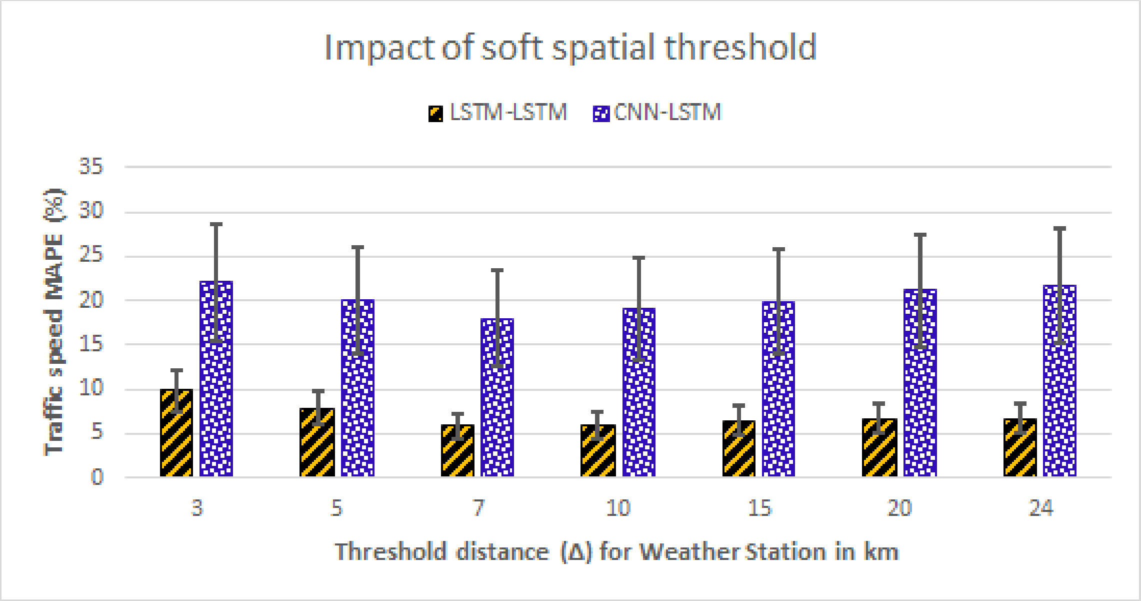


Figure 6.24: Traffic Speed Prediction Performance of the Models for Soft Spatial Threshold. Error bar Shows the Standard Deviation.

6.3.3.3 Spatial interpolation

We test our hybrid model's performance for the linear regression-based spatial interpolation method. To test the performance of the model, we make two cases, case one where we train and test our model with complete rainfall and traffic data and case two where we randomly remove 20% of the rainfall data and interpolate the removed data with linear regression interpolation method as discussed in section 4.2.3.3. Figure 6.25 shows the result comparison with complete data and interpolation using the linear regression method on 20% missing data. It is observed from the result that for 15-minute and 30-minute prediction horizon, the model performance is almost similar to the model's performance with complete data. For 45-minute and 60-minute prediction, the model performance degrades using spatial interpolation because there is an increase in error with the increase in prediction horizon, error from the previous prediction gets accumulated. For interpolation, this error increases further. The LSTM-LSTM traffic speed prediction error with complete data for 15-minute, 30-minute, 45-minute, and 60-minute is 3%, 5%, 8%, and 11%, respectively, for road type $R_{collector}$ and large road network. With spatial interpolation using a linear regression technique, the LSTM-LSTM model output is 3%, 6%, 12%, and 14% for 15-minute, 30-minute, 45-minute, and 60-minute, respectively.

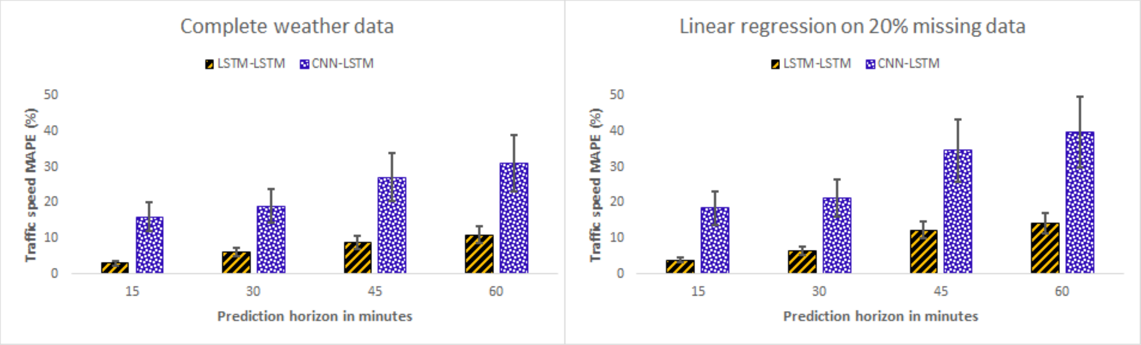


Figure 6.25: Traffic Speed Prediction Performance of the Models with Complete Data and with Spatial Interpolation Method on Missing Data. Error Bar Shows the Standard Deviation.

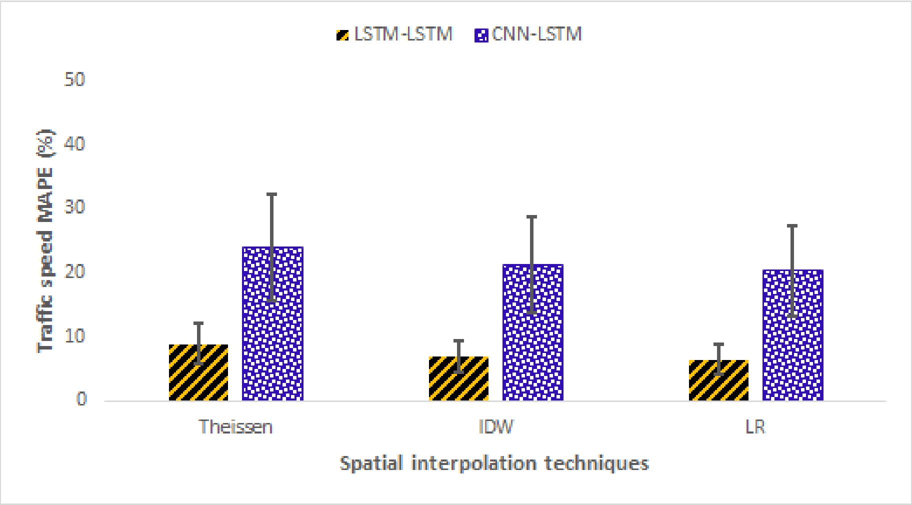


Figure 6.26: Traffic Speed Prediction Performance of the Models with Different Interpolation Techniques for the 30-minute Prediction Horizon on $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

We also examine the performance of different spatial interpolation techniques as discussed in section 4.2.3.3, Theissen polygon method, Inverse Distance Weighted (IDW) method, and Linear Regression (LR) method. Model using linear regression interpolation technique shows the lowest prediction error and highest error with Theissen polygon method. The error difference between linear regression and inverse distance weighted method is not significant. The linear regression method is preferred over inverse distance weighted. In the IDW method, performance is dependent on the radius (area up to which station is considered for the interpolation). In the linear regression method, the model learns this dependency from the given input data. Therefore, no external tuning is required. Figure 6.26 shows the deep learning model's performance with different spatial interpolation techniques considering the 30-minute prediction horizon for road type $R_{collector}$. The traffic speed prediction error of the LSTM-LSTM model using Theissen, IDW, and linear regression interpolation method is 8%, 6.9%, and 6.5%, respectively. The traffic speed prediction error of the CNN-LSTM using Theissen, IDW, and linear regression interpolation method is 24.1%, 21.3%, and 20.3%, respectively.

6.3.3.4 Additional Variables

We examine the performance of the model providing additional data sources as input. The average rainfall value from the past days will give more insight into the soil absorption capability and drainage system. We provide the additional variable as the average rainfall value for a day along with the input data. We perform this experiment for road type $R_{collector}$ and large road network. The model's performance is shown in figure 6.27, where we compare the traffic speed prediction error for the cases when additional data is used versus not used. The figure shows that for the 60-minute prediction horizon, the traffic speed prediction error of LSTM-LSTM is 9% and 11% for the with additional data (one-day average rainfall) and without additional data cases, respectively. There is no significant difference observed in the performance of the model.

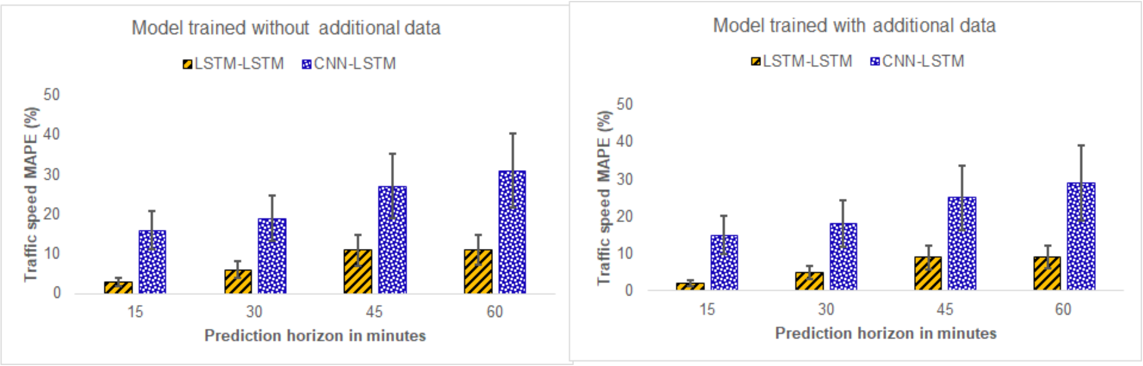


Figure 6.27: Traffic Speed Prediction Performance of the Models Considering Additional Data (one day average rainfall value). Error Bar Shows the Standard Deviation.

Figure 6.28 shows the traffic flow prediction error, considering the 60-minute prediction horizon, the performance of LSTM-LSTM is 12% and 14% for with the additional data and without additional data cases, respectively.

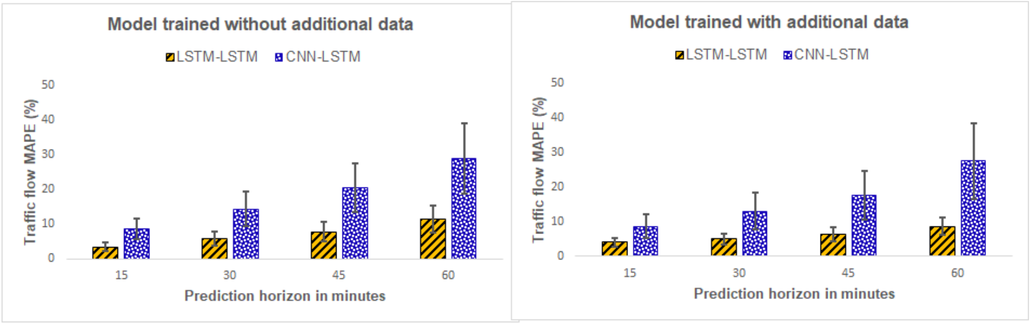


Figure 6.28: Traffic Flow Prediction Performance of the Models Considering Additional Data (one day average rainfall value). Error Bar Shows the Standard Deviation.

We also test model performance when the average rainfall value for the past two days is passed for road type $R_{collector}$. For the 60-minute prediction horizon, the traffic flow prediction error of the LSTM-LSTM model is 11% and 14% for the additional data and without additional data cases, respectively, as shown in figure 6.29. The results are similar to the previous scenario (rainfall average of one day). We conclude that considering the heavy rainfall scenario, the traffic flow comes to the saturation point (minimal value) after prolonged heavy rainfall. Therefore, not much of the difference is observed in model performance. We conclude that the LSTM-LSTM model is robust, increasing the training data will not significantly improve the model's performance.

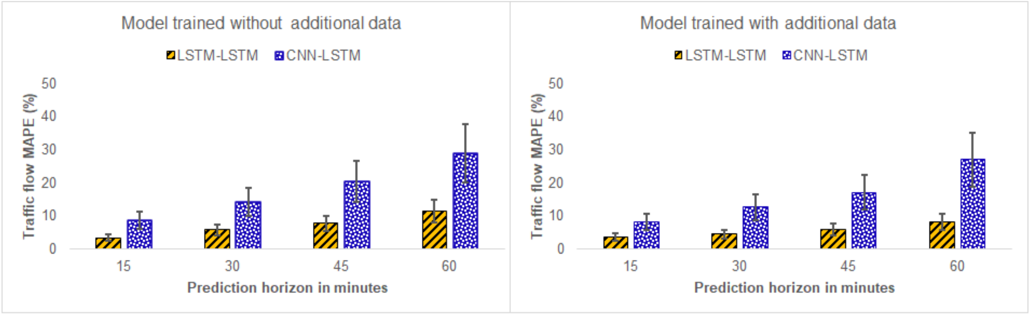


Figure 6.29: Traffic Flow Prediction Performance of the Models Considering Additional Data (two day average rainfall value). Error Bar Shows the Standard Deviation.

6.3.4 Learning Models Complexity Analysis

We evaluate the time and number of parameters needed to train the models. The main factors that influence the model's training time are the lag value and number of the epoch. The lag value is the sequence length passed as an input to the model. For model learning, the entire training dataset is passed both forward and backward through the model defined as the epoch, and the model requires multiple epochs to converge. These are the two hyperparameters that are required to be set before the model training. The large value of lag requires more training time per epoch. The number of epoch until convergence is data-dependent. Figure 6.30 shows the performance of models for different lag value for $R_{collector}$ road type (large road network) considering 30-minute prediction horizon. It is observed that LSTM and LSTM-LSTM models require longer lag values for making an accurate prediction as compared to the RNN, CNN, and CNN-LSTM models. The traffic speed prediction error of the LSTM-LSTM model is the lowest (6%) for the lag value 27. The same reason is applicable for the better performance of the LSTM-LSTM model as it looks out for a longer past range to make a future prediction. The CNN-LSTM model shows the traffic speed prediction error of 19% for the lag value 18. We observe that there is not much difference in the model's performance for the small change in these hyperparameters value. For the RNN model, the lag value 18 shows the error of 26%, CNN shows the error of 23% for the lag value 18, and LSTM shows the error of 20% for the lag value 22.

Figure 6.31 shows the model's performance for the different number of epoch considering the 30-minute prediction horizon for $R_{collector}$ road type. It is observed that the LSTM and LSTM-LSTM model require larger epochs to converge than CNN and CNN-LSTM models. The traffic speed prediction error of LSTM-LSTM is 6% for the number of epoch equals 35. The traffic speed prediction error of CNN-LSTM is 19% for the number of epoch equals 25. For the RNN, CNN, and LSTM model, epoch value is 18, 25, and 35, respectively.

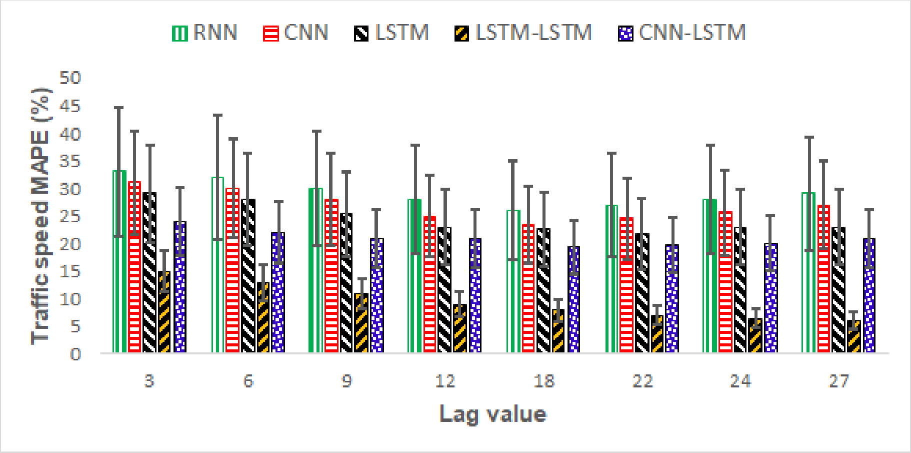


Figure 6.30: Traffic Speed Prediction Performance of the Models for Different Lag Values, 30-minute prediction horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

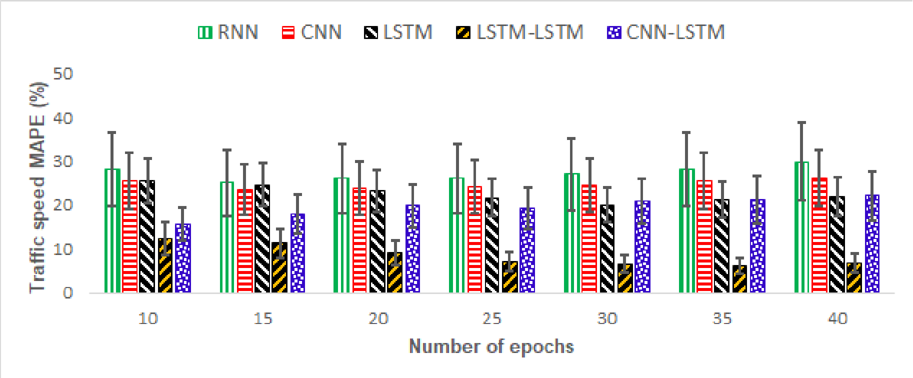


Figure 6.31: Traffic Speed Prediction Performance of the Models for Different Epoch Values, 30-minute prediction horizon for $R_{collector}$ Road. Error Bar Shows the Standard Deviation.

Figure 6.32 and 6.33 shows the relation between the lag value, the number of epoch, and the model's training time considering the 30-minute prediction horizon for $R_{collector}$ road type. The CNN model shows the optimal value (lowest prediction error) at a lower lag value than the LSTM model; therefore, the epoch's number is less. Similarly, the training time is less than the LSTM model. The training time required to train the RNN model for a lag value of 18 is 46 minutes. The training time required to train the CNN model for a lag value of 18 is 42 minutes. The training time required to train the LSTM model for a lag value of 27 is 84 minutes. The recurrent learning model uses the BPTT algorithm, which runs backpropagation for each lag value. Therefore it increases the time complexity of the recurrent learning model. The LSTM-LSTM and CNN-LSTM are the slowest models to train. The training time required by the LSTM-LSTM model is 159.7 minutes, and for the CNN-LSTM model, it is 108.7 minutes. This forms a trade-off between training time and prediction accuracy. LSTM-LSTM being the most accurate model is the slowest model to train.

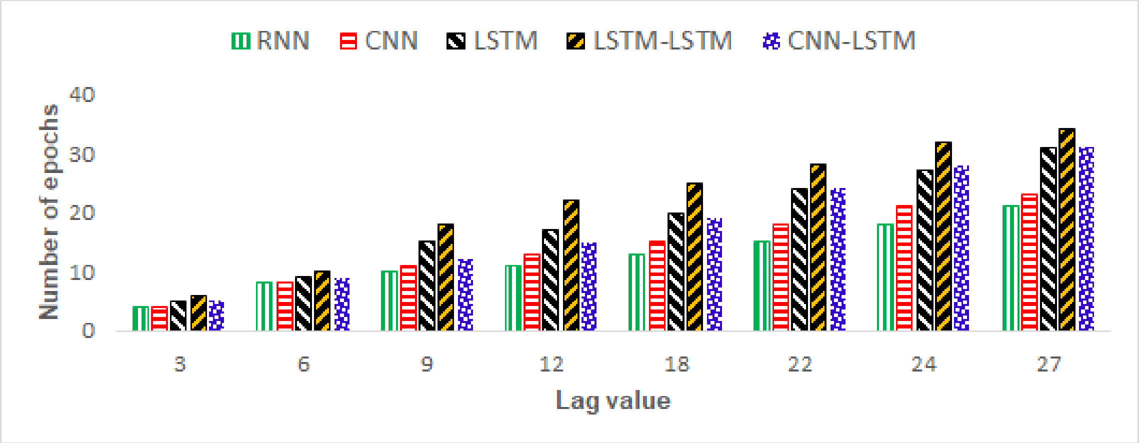


Figure 6.32: The Number of Epoch Required for the Particular Lag Value.

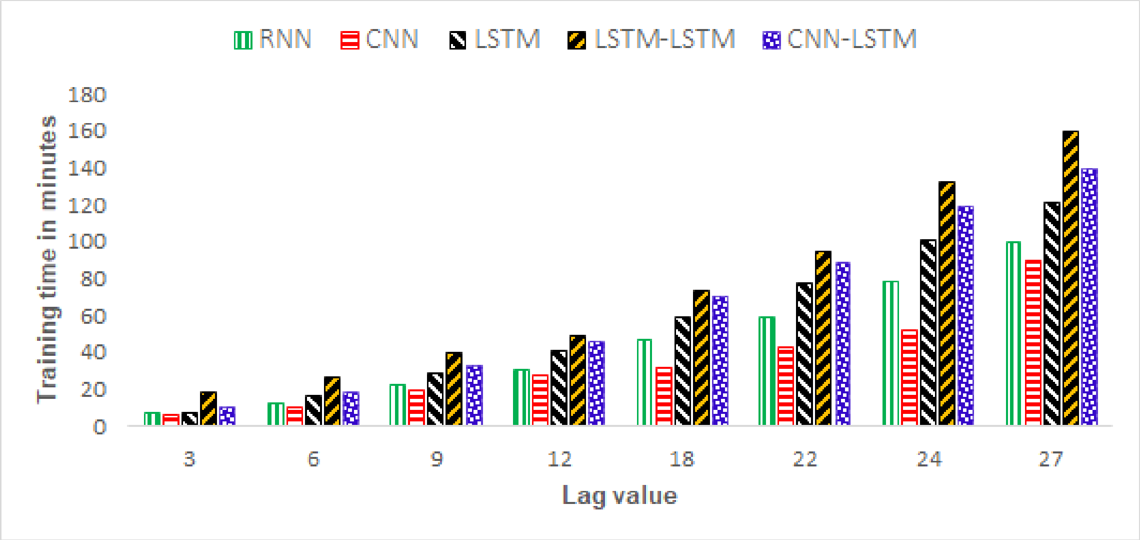


Figure 6.33: Training Time Required for the Each Lag Value.

The number of parameters in a model is the sum of parameters of all layers. The parameters on each layer depend upon the input feature vector and output feature vector. The number of parameters is $\mathcal{O}(k \times n^2)$, where k is the constant which is different for the RNN, LSTM, and CNN.

In the recurrent learning model for each layer, there are two matrices W and U and one bias vector b. The m is input vector size and n is output vector size. The size of W is $m \times n$ which shows the relationship between input and present cell state, and the size of U is $n \times n$ which shows the relation between previous cell state and current cell state. The size of bias vector b is n. The LSTM uses four gates, and therefore for the LSTM value of k is four and for the RNN is one.

For CNN, the constant value depends upon the size of the filter. If size of filter is $k \times k$, the number of parameters are $\mathcal{O}(k \times k \times n^2)$.

Table 6.16 shows the training time and the number of parameters required by the models to predict traffic speed and flow for a 30-minute prediction horizon for $R_{collector}$ road type.

Table 6.16: Training Time and Number of Parameters in Models.

	RNN	CNN	LSTM	LSTM-LSTM	CNN-LSTM
Training time in minute	46	42	84.6	159.7	108.7
Number of parameters in million	0.8	1.7	1.4	2.2	2.8

We also discuss the impact of other environmental variables such as snowfall on traffic stream variables prediction in Appendix F.

6.4 Discussion

The results show that with the inclusion of rainfall data prediction accuracy of the model improves. The result reinforces the fact that the LSTM-LSTM model learns the long-term spatiotemporal dependency between rain and traffic data for a large network and longer temporal dependency between traffic data. The LSTM-LSTM model shows the trade-off between complexities and prediction accuracy. It is the best model in terms of prediction accuracy but at the same time requires longer training time and a large number of parameters. For the large road network of

Table 6.17: Prediction Performance Comparison of LSTM-LSTM model and Existing Models.

Weather Condition	Prediction Horizon in Minutes	Prediction Accuracy
Normal Weather [5, 20, 54]	5 to 15 minutes	92-96%
	30 to 60 minutes	78-82%
Rainy Weather [26, 27]	5 minutes	96%
	10 to 30 minutes	81-87%
LSTM-LSTM model	5 to 15 minutes	97-98%
	30 to 60 minutes	92-85%

road type $R_{collector}$ and prolonged heavy rainfall scenario, the LSTM-LSTM model surpasses all the other models in learning the spatial and temporal features from the rain and traffic data. For the prediction horizons less than 30 minutes, the prediction error of the LSTM-LSTM model is $\leq$ 10%. For the prediction horizons greater than 30 minutes the prediction error is $\leq$ 15%. Compared with the existing research in traffic stream variables prediction during rainfall for shorter prediction horizon ($\leq$ 15-minute) the average error improvement of LSTM-LSTM over existing work is 10% and for larger prediction horizon ($\geq$ 45-minute) it is 30%. The results also show that the LSTM-LSTM model is robust, with small changes in threshold values there is no significant change in model performance.

In the majority of the literature, [5, 20, 54] the prediction accuracy $\geq$ 90% is observed for 15 to 20-minute prediction horizon. The prediction accuracy $\geq$ 80% is observed for a 45 to 60-minute prediction horizon. Our experimental results show the prediction accuracy of the LSTM-LSTM model for the worst case is $\geq$ 97% to 85% for 15-minute to 60-minute prediction horizon. The LSTM-LSTM model performs best because other models update their current state and can't hold more prolonged dependency. With the use of a proper gating mechanism, the LSTM-LSTM model holds a more prolonged dependency.

As per the U.S. Department of Transportation [106], the traffic flow prediction error up to 10-18% reduces the severe congestion (unstable traffic regime) by 5-9%. The ATIS developed by M. Chaturvedi and S. Srivastava [107], where vehicles get real-time traffic information and based on this information the travel time decision are made. With an ATIS penetration rate of 40-50%, for the traffic speed estimation error of 10-15%, the trip duration reduces more than 40%. R. Anil *et* *al.* [108] developed the ATIS for emergency vehicles (ambulance). As per their observation, the traffic speed estimation error of 3-10% provides accurate travel time estimation to the emergency vehicles, based on the analysis of ground truth and synthetic data. A Bisht *et al.* [109] develop a local adaptive light system for ATMS application. As per their observation, the traffic flow error up to 10-15% is acceptable without degrading the network performance. Our LSTM-LSTM model gives the error between 2-15% for a 5 to 60 minutes prediction horizon. Therefore, the ITS application can plug the LSTM-LSTM model for traffic prediction during adverse weather conditions. The lower prediction error achieved by the LSTM-LSTM model makes it suitable for ITS applications used by normal and emergency vehicle users. The experimental results are shown for the rainfall data, a similar model can be used with other environmental data.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

Adverse weather conditions such as rainfall, snowfall, and fog significantly affect the traffic stream variables of the road segment. The short-term traffic stream variables prediction during adverse weather conditions is needed for effective ITS applications such as route guidance and traffic management-related applications. We adopted multiple deep learning models because of their underlying architecture to extract the spatiotemporal features from the input data. The CNN model is known for finding the correlation between the neighboring pixels. The recurrent learning models, RNN and LSTM, specifically designed for sequence learning problems, are used to extract the spatiotemporal features from the historical and current traffic and weather data. The traffic condition on the road segment is affected by traffic and weather conditions of other connected roads and the previous traffic conditions of its own. We designed the hybrid deep learning models CNN-LSTM and LSTM-LSTM, where the former model extracts the spatiotemporal features and the latter model uses these features as memory. Based on memory and temporal input, the traffic prediction is performed. During prolonged rainfall, the rainfall impact on distant roads affects the traffic condition on the target road at a much later time instance. The hybrid models are specifically designed to extract the longer spatiotemporal dependencies.

Traffic stream variables prediction during adverse weather conditions should consider the spatial and the temporal significance of the weather conditions. The impact of weather conditions on other road segments is important to capture the waterlogging and water flow paths. Similarly, the extended weather sequence would capture the impact of prolonged weather conditions.

We proposed the soft spatial and temporal threshold mechanism to capture the spatial and prolonged impact of weather conditions. The soft spatial threshold mechanism considers the time-series weather data from all the stations that are threshold distance apart and the weather reading from the other weather stations that fall out of the threshold range through distance-based weighted average. The soft temporal threshold mechanism considers the moving weighted average of the extended weather readings.

Weather data is less granular than the traffic data; therefore, for accurate traffic forecasting, missing weather data can't be ignored. We used the spatial interpolation method such as the Theissen polygon, inverse distance weighted method, and linear regression method to fill out the missing weather readings. The linear regression method is preferred over other methods because it predicts the interpolated values from the given data without any external tuning.

The deep learning models require a large amount of data to extract the hidden features. The majority of road segments depend upon the alternate traffic data source because of the high maintenance and deployment cost of the ITS infrastructure. The alternate source of traffic data gives sparse and incomplete traffic information. Therefore, to overcome data sparsity, we proposed the mechanism to generate realistic synthetic traffic data. We studied the impact of rainfall on the traffic stream variables of different roads such as arterial, sub-arterial, and collector. The traffic and the rainfall data of different types of roads are analyzed to design the empirical model. The empirical model computes traffic speed using the friction, rainfall, and road-type parameters. These parameters are estimated using the ground truth data under diverse traffic and weather conditions.

To generate the synthetic data under a realistic simulation setting, we used the SUMO traffic simulator. SUMO car-following models don't consider the weather parameters for estimating the traffic speed of the vehicle. To overcome this limitation, the Krauss car-following model in SUMO is upgraded to use the friction, rainfall, and road-type parameters to compute the vehicle speed during rainfall.

We performed various experiments to observe the prediction performance of the deep learning models. We performed the experiment to test whether the traffic prediction accuracy of the deep learning model improves with the inclusion of weather data or not. We used the RNN and LSTM models for this experiment. The traffic and the rainfall data of San Diego are used to test the model performance. Results show that traffic speed and flow prediction of the model trained with traffic and rainfall data is more accurate than the model trained without rainfall data. The average prediction improvement of LSTM and RNN trained with rainfall data is 10% compared to the LSTM and RNN when trained without rainfall data for a shorter prediction horizon ($\leq$ 30 minutes). For a higher prediction horizon ($>$ 30 minutes and $\leq$ 1 hour), the prediction error improvement of models when trained with rainfall data is approximately 30% compared to models when trained without rainfall data. We observe that due to error accumulation, as the prediction horizon increases, prediction error also increases.

We experiment with different types of roads, I-5, I-5D, and SR-75 road segments. We observe that when trained without rainfall data, the prediction error are high for narrow and low elevation roads (SR-75) as well as for the road passing through the densely populated area (I-5D) as compared to the broad, high elevation road passing through the countryside area (I-5). Therefore apart from rainfall, other factors like drainage system, soil absorption, waterlogging, road width, densely populated area, etc., also play an important role in traffic stream variables prediction. Our deep learning model learns these additional features from the input traffic and rainfall data.

We compare the prediction performance of recurrent learning models with the other deep learning models (SAE, BPNN, DBN) during adverse weather conditions. For a prediction horizon greater than 15 minutes, LSTM shows approximately 15% error improvement over other models. The LSTM model holds long spatiotemporal dependency due to its gating mechanism, which works as a memory component.

We compare the LSTM and RNN model's performance with the existing models to predict the traffic stream variables during adverse weather conditions. The existing model is trained on traffic data which is either private or complete data information is not shared. Compared with the reported results, LSTM considers spatiotemporal traffic and weather input, show an average error improvement of 2-9%.

We perform the experiment to test the quality of the synthetic traffic data. The synthetic data generated using simulations are compared with the ground truth data for all roads (arterial, sub-arterial, and collector road) under various traffic and rainfall conditions to validate the empirical model and simulation setup. We calibrate the real traffic flow data during rainfall to generate synthetic data using the proposed empirical model. On the arterial road, traffic speed decrease by 1 - 13%, for the low to heavy rainfall. On the sub-arterial road, the reduction of 2 - 16% in the traffic speed for the low to heavy rainfall is observed. These numbers are similar to the literature studies. On the collector road, we observe the traffic speed reduction of 2 - 20% for the low to heavy rainfall. The availability of realistic synthetic data enables further studies based on data-driven models. Most of the traffic data from ITS infrastructure is not open for free access. Therefore, using fine-grained synthetic data, the comparison study between different learning models will be possible.

To validate the deep learning models, we use the synthetic traffic data generated by SUMO using an empirical model for various road types (arterial, sub-arterial, and collector) and different road network types (single, small and large). The traffic data is generated for a prolonged heavy rainfall scenario (100 mm in 6 hours) for each road network type. We test the performance of RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM. We observe that for all the road segments and every road network, the inclusion of rainfall variable improves the models' traffic prediction accuracy. The prolonged rainfall significantly affects the traffic stream variables, both traffic speed and flow.

For large road networks, the performance of LSTM-LSTM is better than the other models for road type sub-arterial and collector. For the sub-arterial road segment, for a larger prediction horizon ($\geq$ 45 minutes), the average error improvement of LSTM-LSTM is 25% compared to non-hybrid models and 10% over

CNN-LSTM. For collector road, the average error improvement of LSTM-LSTM is 20% compared to CNN-LSTM. RNN suffers from vanishing gradient problem, and the CNN model performance is limited by their localized receptive field. For prolonged rainfall and large road network scenario, the LSTM-LSTM model shows significant improvement in handling longer dependencies. In the LSTM-LSTM model, the former LSTM model learns the spatiotemporal dependency between weather and traffic data. It extracts both short and long-term dependencies. The latter LSTM model uses the spatiotemporal features extracted by the former LSTM model as memory components. It learns weather and traffic dependency on the target road segment based on spatiotemporal features and traffic data from the target road segment.

One of the main advantages of realistic synthetic data is that we can compare different deep learning models' performance on a similar data set. We compare the existing models with our hybrid models for large road networks and collector road segments. In literature, either the temporal traffic and weather data is used for traffic prediction, or recent rainfall value is used with spatiotemporal traffic data. For smaller prediction horizons, the average error improvement of LSTM-LSTM is 15%. For a larger prediction horizon, the average error improvement is 35% compared to the existing models.

We observe that the performance of deep learning models improves with the inclusion of spatial and prolonged information of weather variables. Also, the spatial interpolation technique such as linear regression provides a good approximation of the measured value for smaller prediction horizons, but for larger prediction horizons due to error accumulation, the difference between the predicted value and measured value increases.

We perform various experiments to validate the robustness of the model. We change the threshold values and examine the performance of the hybrid models. We observe that no significant difference in model performance is observed with a small change in the threshold values.

To learn about the model's complexities, we compare the model's training time and the number of parameters. The training time is dependent on the lag value and the number of the epoch. LSTM-LSTM model holds the longer dependency and, therefore, required a larger lag value compared to other models. It uses the BPTT algorithm, which runs backpropagation for each lag value. Therefore, the training time of the LSTM-LSTM model is higher compared to the other models. Similarly, the number of parameters learned is higher for the hybrid models compared to the non-hybrid models. This forms a trade-off; the hybrid models are accurate but require more training time and parameters compared to the non-hybrid models.

The performance of LSTM-LSTM shows significant improvement over existing models, specifically during the worst-case scenario (large road network and prolonged heavy rainfall scenario). The literature suggests that ATIS and AMTS applications work accurately for traffic speed and flow error of 10 to 15%. Our LSTM-LSTM model shows an error of less than 15% for a 60-minute prediction horizon under the worst-case scenario. Therefore, this model can be easily plugin for traffic stream variables prediction during adverse weather conditions.

We have focused on rainfall conditions but a similar method can be used to predict the traffic stream variables in the presence of other environmental variables also.

7.2 Future Work

The empirical model is developed to generate realistic synthetic traffic data during adverse weather conditions. As future work, a more generic empirical model can be built and evaluated. The empirical model should consider other weather-related parameters such as visibility, sleet, etc. The availability of such a standard data-set may enable the evaluation of various learning algorithms on common ground and open up doors for research in traffic variables prediction using a cost-effective source of traffic data. Development of the data set is part of future research.

It is assumed that the simulation model considers homogeneous vehicles (car), and all vehicles follow lane discipline. As future work, we aim to generate traffic data for heterogeneous vehicles environment where vehicles may follow lane less discipline.

We studied alternate traffic data sources such as taxi GPS in the absence of traffic data from the ITS infrastructure. Social media are increasingly being used as an information source. The user shares information related to waterlogging, traffic jams, snowfall, etc. As future work, social media information with other traffic data sources such as taxi GPS and cellular data gives more insight into the actual traffic during adverse weather conditions.

We examined the impact of weather conditions on traffic stream variables prediction. As future work, we aim to examine the impact of other non-recurrent events such as road accidents, social events, etc., on traffic stream variables prediction.

The shortcoming of the multistep prediction model is with an increase in prediction horizon, the error increases due to error accumulation. The model learns based on some loss function. Therefore, there is a need for proper loss function for multistep output. As future work, we aim to use the multistep loss function such that the problem of an increase in prediction error due to error accumulation can be resolved.

As future work, we extend our designed models for other environmental data such as snowfall, fog, etc. Depending upon the weather condition at a particular geographical location, the prediction result will be helpful for various stakeholders worldwide.

CHAPTER 8

List of Publications

- **Journal**

  1. Nigam, Archana, and Sanjay Srivastava. "Weather impact on macroscopic traffic stream variables prediction using recurrent learning approach." Journal of Intelligent Transportation Systems (2021): 1-17. Taylor & Francis.

     This publication is based on the work discussed in section 5.2.2 and 6.1.

- **Conferences**

  1. Archana Nigam and Sanjay Srivastava, "Macroscopic Traffic Stream Variable Prediction with Weather Impact using Recurrent Learning Approach." 2020 IEEE-HYDCON, pp. 1–6, Sept 2020.

     This publication is based on the work discussed in section 5.2.2.

  2. Archana Nigam and Sanjay Srivastava, "Macroscopic Traffic Stream Variables Prediction with Weather Impact Using Hybrid CNN-LSTM model." Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking (ICDCN), pp. 1–6, Jan 2021.

     This publication is based on the work discussed in section 4.2.3.2 and 5.2.3.1.

  3. Archana Nigam, Manish Chaturvedi, and Sanjay Srivastava, "Impact of Rainfall and Waterlogging on Traffic Stream Variables in Developing Countries." 2021 International Conference on COMmunication Systems NETworkS (COMSNETS), pp. 728–735, Jan 2021.

     This publication is based on the work discussed in section 4.2.2 and 6.2.

References

[1] L. Figueiredo, I. Jesus, J. T. Machado, J. R. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*. IEEE, 2001, pp. 1206–1211.

[2] J. L. Kay, "Advanced traffic management systems—an element of intelligent vehicle-highway systems," *SAE transactions*, pp. 1240–1251, 1990.

[3] G. W. Euler, "Intelligent vehicle/highway systems: Definitions and applications," *ITE journal*, vol. 60, no. 11, pp. 17–22, 1990.

[4] E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transport reviews*, vol. 24, no. 5, pp. 533–557, 2004.

[5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[6] J. Barros, M. Araujo, and R. J. Rossetti, "Short-term real-time traffic prediction methods: A survey," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2015, pp. 132–139.

[7] X. Fu, W. Luo, C. Xu, and X. Zhao, "Short-term traffic speed prediction method for urban road sections based on wavelet transform and gated recurrent unit," *Mathematical Problems in Engineering*, vol. 2020, 2020.

[8] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.

[9] B. M. Williams, P. K. Durvasula, and D. E. Brown, "Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transportation Research Record*, vol. 1644, no. 1, pp. 132–141, 1998.

[10] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[11] S.-R. Hu, S. M. Madanat, J. V. Krogmeier, and S. Peeta, "Estimation of dynamic assignment matrices and od demands using adaptive kalman filtering," *Journal of Intelligent Transportation Systems*, vol. 6, no. 3, pp. 281–300, 2001.

[12] N. Barimani, A. R. Kian, and B. Moshiri, "Real time adaptive non-linear estimator/predictor design for traffic systems with inadequate detectors," *IET Intelligent Transport Systems*, vol. 8, no. 3, pp. 308–321, 2014.

[13] G. A. Davis and N. L. Nihan, "Nonparametric regression and short-term freeway traffic forecasting," *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.

[14] E. Castillo, J. M. Menéndez, and S. Sánchez-Cambronero, "Predicting traffic flow using bayesian networks," *Transportation Research Part B: Methodological*, vol. 42, no. 5, pp. 482–509, 2008.

[15] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.

[16] G. Zhang, B. E. Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks:: The state of the art,” *International journal of forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[17] S. Kombrink, T. Mikolov, M. Karafiát, and L. Burget, “Recurrent neural network based language modeling in meeting recognition,” in *Twelfth annual conference of the international speech communication association*, 2011.

[18] S. Hochreiter and J. Schmidhuber, “Lstm can solve hard long time lag problems,” in *Advances in neural information processing systems*, 1997, pp. 473–479.

[19] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, “Lstm network: a deep learning approach for short-term traffic forecast,” *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[20] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, “Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning,” *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1688–1711, 2019.

[21] A. T. Ibrahim and F. L. Hall, “Effect of adverse weather conditions on speed-flow-occupancy relationships,” *Proceedings of theTransportation Research Record 1457. Transportation Research Board, Washington, DC.*, pp. 184–191, 1994.

[22] M. Agarwal, T. H. Maze, and R. Souleyrette, “Impacts of weather on urban freeway traffic flow characteristics and facility capacity,” in *Proceedings of the 2005 mid-continent transportation research symposium*, 2005, pp. 18–19.

[23] M. Kyte, Z. Khatib, P. Shannon, and F. Kitchener, “Effect of weather on free-flow speed,” *Transportation Research Record*, vol. 1776, no. 1, pp. 60–68, 2001.

[24] L. Greer, J. L. Fraser, D. Hicks, M. Mercer, K. Thompson *et al.*, “Intelligent transportation systems benefits, costs, and lessons learned: 2018 update report,” United States. Dept. of Transportation. ITS Joint Program Office, Tech. Rep., 2018.

[25] W. Qiao, A. Haghani, C.-F. Shao, and J. Liu, "Freeway path travel time prediction based on heterogeneous traffic data through nonparametric model," *Journal of Intelligent Transportation Systems*, vol. 20, no. 5, pp. 438–448, 2016.

[26] Y. Jia, J. Wu, and M. Xu, "Traffic flow prediction with rainfall impact using a deep learning method," *Journal of advanced transportation*, vol. 2017, 2017.

[27] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508–9517, 2016.

[28] A. Baghel, "Causes of urban floods in india: Study of mumbai in 2006 and chennai in 2015."

[29] "Uber movement india," https://movement.uber.com/, accessed: 2020-12-30.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[31] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility)-an open-source traffic simulation," in *Proceedings of the 4th middle East Symposium on Simulation and Modelling (MESM20002)*, 2002, pp. 183–187.

[32] J. Van Lint and C. Van Hinsbergen, "Short-term traffic and travel time prediction models," *Artificial Intelligence Applications to Critical Transportation Issues*, vol. 22, no. 1, pp. 22–41, 2012.

[33] P. S. Maybeck, "The kalman filter: An introduction to concepts," in *Autonomous robot vehicles*. Springer, 1990, pp. 194–204.

[34] W. W. Wei, "Time series analysis," in *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2*, 2006.

[35] T. Thomas, W. Weijermars, and E. Van Berkum, “Predictions of urban volumes in single time series,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 71–80, 2009.

[36] E. Chung and N. Rosalion, “Short term traffic flow prediction,” in *AUSTRALASIAN TRANSPORT RESEARCH FORUM (ATRF), 24TH, 2001, HOBART, TASMANIA, AUSTRALIA*, 2001.

[37] L. E. Peterson, “K-nearest neighbor,” *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[38] J. Walters-Williams and Y. Li, “Comparative study of distance functions for nearest neighbors,” in *Advanced techniques in computing sciences and software engineering*. Springer, 2010, pp. 79–84.

[39] H. Chang, Y. Lee, B. Yoon, and S. Baek, “Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences,” *IET intelligent transport systems*, vol. 6, no. 3, pp. 292–305, 2012.

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

[41] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine learning*, vol. 29, no. 2, pp. 131–163, 1997.

[42] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik *et al.*, “Support vector regression machines,” *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

[43] Y. Wang, Y. Fan, P. Bhatt, and C. Davatzikos, “High-dimensional pattern regression using machine learning: from medical images to continuous clinical variables,” *Neuroimage*, vol. 50, no. 4, pp. 1519–1535, 2010.

[44] J. Hu, P. Gao, Y. Yao, and X. Xie, “Traffic flow forecasting with particle swarm optimization and support vector regression,” in *17th international ieee*

conference on intelligent transportation systems (itsc). IEEE, 2014, pp. 2267–2268.

[45] G. Gopi, J. Dauwels, M. T. Asif, S. Ashwin, N. Mitrovic, U. Rasheed, and P. Jaillet, "Bayesian support vector regression for traffic speed prediction with error bars," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. IEEE, 2013, pp. 136–141.

[46] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.

[47] L. Nanni, S. Ghidoni, and S. Brahnam, "Handcrafted vs. non-handcrafted features for computer vision classification," *Pattern Recognition*, vol. 71, pp. 158–172, 2017.

[48] M. H. Hassoun *et al.*, *Fundamentals of artificial neural networks*. MIT press, 1995.

[49] K. Kumar, M. Parida, and V. K. Katiyar, "Short term traffic flow prediction in heterogeneous condition using artificial neural network," *Transport*, vol. 30, no. 4, pp. 397–405, 2015.

[50] A. Csikós, Z. J. Viharos, K. B. Kis, T. Tettamanti, and I. Varga, "Traffic speed prediction method for urban networks—an ann approach," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2015, pp. 102–108.

[51] N. Saquib, M. S. R. Sakib, and A.-S. K. Pathan, "Visim: A user-friendly simulation tool for manet routing protocols," BRAC University, Technical Report—TR-BU-CSE001-Oct09, Tech. Rep., 2009.

[52] P. A. Devijver and J. Kittler, *Pattern recognition: A statistical approach*. Prentice hall, 1982.

[53] S. Sun, C. Zhang, and Y. Zhang, "Traffic flow forecasting using a spatio-temporal bayesian network predi ctor," in *International conference on artificial neural networks*. Springer, 2005, pp. 273–278.

[54] A. Nagare and S. Bhatia, "Traffic flow control using neural network," *Traffic*, vol. 1, no. 2, pp. 50–52, 2012.

[55] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[56] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[57] B. Cheung, "Convolutional neural networks applied to human face classification," in *2012 11th International Conference on Machine Learning and Applications*, vol. 2. IEEE, 2012, pp. 580–583.

[58] S. Sun, H. Wu, and L. Xiang, "City-wide traffic flow forecasting using a deep convolutional neural network," *Sensors*, vol. 20, no. 2, p. 421, 2020.

[59] Y. Zhao, A. W. Sadek, and D. Fuglewicz, "Modeling the impact of inclement weather on freeway traffic speed at macroscopic and microscopic levels," *Transportation research record*, vol. 2272, no. 1, pp. 173–180, 2012.

[60] R. E. Tarone, "A modified bonferroni method for discrete data," *Biometrics*, pp. 515–522, 1990.

[61] H. Rakha, M. Farzaneh, M. Arafeh, and E. Sterzin, "Inclement weather impacts on freeway traffic stream behavior," *Proceedings of the Transportation Research Record*, vol. 2071, no. 1, pp. 8–18, 2008.

[62] J. Weng, L. Liu, and J. Rong, "Impacts of snowy weather conditions on expressway traffic flow characteristics," *Discrete dynamics in nature and society*, vol. 2013, 2013.

[63] H. M. Hammad, M. Ashraf, F. Abbas, H. F. Bakhat, S. A. Qaisrani, M. Mubeen, S. Fahad, and M. Awais, "Environmental factors affecting the frequency of road traffic accidents: a case study of sub-urban area of pakistan," *Environmental Science and Pollution Research*, vol. 26, no. 12, pp. 11 674–11 685, 2019.

[64] S. Hu, H. Lin, K. Xie, J. Dai, and J. Qui, "Impacts of rain and waterlogging on traffic speed and volume on urban roads," in *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2943–2948.

[65] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[66] S. Hochreiter and J. Schmidhuber, "Lstm can solve hard long time lag problems," *Advances in neural information processing systems*, pp. 473–479, 1997.

[67] "Caltrans, performance measurement system (pems)," http://pems.dot.ca.gov, accessed: 2020-12-30.

[68] C. Chen, "Freeway performance measurement system (pems)," 2003.

[69] "National oceanic and atmospheric administration," https://www.ncdc.noaa.gov/cdo-web/dataset, accessed: 2020-12-30.

[70] "Maharashtra remote sensing application centre," https://mrsac.gov.in/MRSAC, accessed: 2020-12-30.

[71] I. Tsapakis, T. Cheng, and A. Bolbol, "Impact of weather conditions on macroscopic urban travel times," *Proceedings of the Journal of Transport Geography*, vol. 28, pp. 204–211, 2013.

[72] L. Wang, T. Yamamoto, T. Miwa, and T. Morikawa, "An analysis of effects of rainfall on travel speed at signalized surface road network based on probe vehicle data," *Proceedings of the ICTTS, Xian, China*, pp. 2–4, 2006.

[73] M. L. Angel, T. Sando, D. Chimba, and V. Kwigizile, "Effects of rain on traffic operations on florida freeways," *Proceedings of the Transportation Research Record*, vol. 2440, pp. 51–59, 2014.

[74] T. Weber, P. Driesch, and D. Schramm, "Introducing road surface conditions into a microscopic traffic simulation." in *SUMO*, 2019, pp. 172–186.

[75] J. Song, Y. Wu, Z. Xu, and X. Lin, "Research on car-following model based on sumo," pp. 47–55, 2014.

[76] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "Traci: an interface for coupling road traffic and network simulators," *Proceedings of the 11th communications and networking simulation symposium*, pp. 155–163, 2008.

[77] F.-W. Chen and C.-W. Liu, "Estimation of the spatial rainfall distribution using inverse distance weighting (idw) in the middle of taiwan," *Paddy and Water Environment*, vol. 10, no. 3, pp. 209–222, 2012.

[78] I. A. Nalder and R. W. Wein, "Spatial interpolation of climatic normals: test of a new method in the canadian boreal forest," *Agricultural and forest meteorology*, vol. 92, no. 4, pp. 211–225, 1998.

[79] L. Mitas and H. Mitasova, "Spatial interpolation," *Geographical information systems: principles, techniques, management and applications*, vol. 1, no. 2, 1999.

[80] F. J. Solis and R. J.-B. Wets, "Minimization by random search techniques," *Mathematics of operations research*, vol. 6, no. 1, pp. 19–30, 1981.

[81] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[82] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[83] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[84] R. Sutton, "Two problems with back propagation and other steepest descent learning procedures for networks," in *Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986*, 1986, pp. 823–832.

[85] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[86] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar, "Adaptive methods for nonconvex optimization," in *Advances in neural information processing systems*, 2018, pp. 9793–9803.

[87] A. De Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Using the mean absolute percentage error for regression models," in *Proceedings*. Presses universitaires de Louvain, 2015, p. 113.

[88] T. Van Laarhoven, "L2 regularization versus batch and weight normalization," *arXiv preprint arXiv:1706.05350*, 2017.

[89] J. Turian, J. Bergstra, and Y. Bengio, "Quadratic features and deep architectures for chunking," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, 2009, pp. 245–248.

[90] R. M. Neal, "Connectionist learning of belief networks," *Artificial intelligence*, vol. 56, no. 1, pp. 71–113, 1992.

[91] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning. nature 521 (7553), 436-444," *Google Scholar Google Scholar Cross Ref Cross Ref*, 2015.

[92] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[93] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[94] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[95] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[96] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[97] S. S. Md Noor, J. Ren, S. Marshall, and K. Michael, "Hyperspectral image enhancement and mixture deep-learning classification of corneal epithelium injuries," *Sensors*, vol. 17, no. 11, p. 2644, 2017.

[98] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[99] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[100] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[101] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[102] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, "Implementation of training convolutional neural networks," *arXiv preprint arXiv:1506.01195*, 2015.

[103] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[104] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[105] M. Chaturvedi and S. Srivastava, "Multi-modal design of an intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 2017–2027, 2017.

[106] F. Habtemichael, W. A. Perez, R. K. Boyapati *et al.*, "Advanced traveler information systems (atis) 2.0 precursor system," United States. Dept. of Transportation. ITS Joint Program Office, Tech. Rep., 2018.

[107] M. Chaturvedi and S. Srivastava, "Advanced traveler information system using cocomo and ecomo," in *2017 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2017, pp. 1–5.

[108] R. Anil, M. Satyakumar, and J. Jayakumar, "Travel time estimation and routing for emergency vehicles under indian conditions," in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2018, pp. 247–252.

[109] A. Bisht, K. Ravani, M. Chaturvedi, and N. Kumar, "A feasibility study on upgrading the static tlc infrastructure to adaptive tlc," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2563–2568.

[110] MNDOT, "Minnesota department of transportation (mndot)," 2020. [Online]. Available: https://www.dot.state.mn.us/

[111] NWS, "National weather service twin cities," 2020. [Online]. Available: https://www.weather.gov/mpx/

Appendix A SUMO Sample Code

The parameters related to road-type, friction, average rainfall value, and time are passed to the car following model through the Traffic Control Interface (TraCI). The below program shows the sample code of TraCI, the function run_test is written in Python. The value corresponds to the particular parameter is set (using setParameter) which will be further retrieved by the car-following model.

```
def run_test():
    while  traci.simulation.getMinExpectedNumber() >0:
        traci.simulationStep()
        traci.lane.setParameter("Lane1_name", "friction", fric_val)
        traci.lane.setParameter("Lane1_name", "alpha", alpha_val )
        traci.lane.setParameter("Lane1_name", "road", road_val )
    traci.close()
    sys.stdout.flush()
```

Once the parameters are set, these parameters are retrieved by the Krauss car-following model (using getParameter) to derive vehicle speed. The below sample code shows the function to calculate speed based on parameters.

```
        // save old v for optional acceleration computation
        const double oldV = veh->getSpeed();
        /*below statements define the parameters and get the values passed
        through TraCI. If value is not defined than for
        backward compatibility values are provided.*/
         double fric = StringUtils::toDouble(veh->getLane()->
         getParameter("friction", "1.0"));
         double alpha = StringUtils::toDouble(veh->getLane()->
         getParameter("alpha", "0.0"));
         double road = StringUtils::toDouble(veh->getLane()->
         getParameter("road", "0.1"));
         double time = StringUtils::toDouble(veh->getLane()->
         getParameter("time", "0"));
         double factor = -0.3491 * fric * fric + 0.8922 *
         fric + 0.4493; //2nd degree polyfit
        double u_w;
       if(time == 0.000000 && road == 0.100000)
           u_w = (43.3548 + 26.9938 * fric - 0.5541 * alpha ) * .27778;
        else if(time == 0.000000 && road == 0.900000)
          {
           u_w = ( -39.5716 * alpha + 56.613 * road + 9.2185 * fric ) * .27778;
        else if(time == 1.000000 && road == 0.900000)
          {
           u_w = ( -33.608 * alpha + 52.158 * road + 14.832 * fric) * .27778;
          //printf("speed t=1 %f  ",u_w);
          }
        else if(time == 2.000000 && road == 0.900000)
          {
          u_w = (-39.278 * alpha + 58.224 * road + 5.8123 * fric) * .27778;
      }
       double acc=(u_w-oldV)/veh->getActionStepLengthSecs();
       double vMax = MIN3(oldV + ACCEL2SPEED(acc),
        maxNextSpeed(oldV, veh), vStop);
```

Appendix B Vanishing Gradient Problem and Solution

After the RNN outputs the prediction vector $\hat{Y^i_{t_g}}$, the model computes the prediction error E_{t_g} and uses the BPTT algorithm to compute the gradient.

$$\frac{\partial E_{t_g}}{\partial W}$$

For the simplification we called E_{t_g} as E_t and $\hat{Y^i_{t_g}}$ as Y_t. The gradient is used to update the model parameters, and η is the learning rate. The below equation is used to update the parameters.

$$W \leftarrow W - \eta \frac{\partial E_t}{\partial W}$$

and the learning process continues using the gradient descent (GD) algorithm. After t time stamps, gradient is computed as,

$$\frac{\partial E_t}{\partial W} = \frac{\partial E_t}{\partial Y_t} \frac{\partial Y_t}{\partial C_t} \frac{\partial C_t}{\partial C_{t-1}} \cdots \frac{\partial C_2}{\partial C_1} \frac{\partial C_1}{\partial W}$$

$$\frac{\partial E_t}{\partial W} = \frac{\partial E_t}{\partial Y_t} \frac{\partial Y_t}{\partial C_t} (\prod_{k=2}^{t} \frac{\partial C_k}{\partial C_{k-1}}) \frac{\partial C_1}{\partial W} \tag{B.1}$$

Notice that since $W = [w_c, w_x]$, C_k can be written as:

$$C_k = tanh(w_c C_{k-1} + w_x(X_k, W_k))$$

Compute the derivative of C_k and get:

$$\frac{\partial C_k}{\partial C_{k-1}} = tanh'(w_c C_{k-1} + w_x(X_k, W_k)).\frac{d}{dc_{k-1}}[w_c C_{k-1} + w_x(X_k, W_k)]$$

$$= tanh'(w_c C_{k-1} + w_x(X_k, W_k)).w_c \tag{B.2}$$

Plug equation B.2 into B.1

$$\frac{\partial E_t}{\partial W} = \frac{\partial E_t}{\partial Y_t}\frac{\partial Y_t}{\partial C_t}(\prod_{k=2}^{t} tanh'(w_c C_{k-1} + w_x(X_k, W_k)).w_c)\frac{\partial C_1}{\partial W}$$

The last expression tends to vanish when t is large, this is due to the derivative of the activation function *tanh* which is smaller or equal to 1.

We have:

$$\prod_{t=2}^{k} tanh'(w_c C_{t-1} + w_x(X_t, W_t)).w_c \to 0$$

so,

$$\frac{\partial E_k}{\partial w} \to 0$$

So the network's weights update will be:

$$w \leftarrow w - \eta\frac{\partial E_t}{\partial w} \approx w$$

And no significant learning will be done.

This is called the vanishing gradient problem, due to this problem the RNN is not capable of holding long-term dependencies. This shortcoming of RNN is solved by the LSTM model.

The equation corresponds to cell state gate in LSTM is represented as,

$$C_k = C_{k-1} \otimes \sigma(w_f.[h_{k-1}(X_k, W_k)]) \oplus tanh(w_c.[h_{k-1}(X_k, W_k)]) \otimes \sigma(w_c.[h_{k-1}(X_k, W_k)])$$

$$C_t = C_{k-1} \otimes f_k \oplus \tilde{C}_k \otimes i_k$$

$$\frac{\partial C_k}{\partial C_{k-1}} = \sigma(w_f.[h_{k-1}(X_k, W_k)]) + \frac{d}{dc_{k-1}}(tanh(w_c.[h_{k-1}(X_k, W_k)]) \otimes \sigma(w_c.[h_{k-1}(X_k, W_k)])) \tag{B.3}$$

Plug equation B.3 into B.1,

$$\frac{\partial C_k}{\partial C_{k-1}} \approx f_k$$

$$\frac{\partial E_t}{\partial W} = \frac{\partial E_t}{\partial Y_t}\frac{\partial Y_t}{\partial C_t}(\prod_{k=2}^{t} f_k)\frac{\partial C_1}{\partial W}$$

$$f_k \approx 1$$

Hence gradient will not vanish, forget gate output will close to zero for the cases where long term dependencies are not important under such cases gradient will vanish but the vanishing of such gradients is allowed in the model.

Appendix C Sample Code of Traffic Stream Variables Prediction Using Deep Learning Models

The data pre-processing makes the spatiotemporal data. The spatiotemporal data is divided into training and validation or test data. For RNN, CNN, and LSTM Xtrain and Ytrain are input and output training data. XVad and YVad are input and output validation data. For LSTM-LSTM and CNN-LSTM, Xtrain1 and Xtrain2 are the spatiotemporal and temporal input training data, and Ytrain2 is the output training data. XVad1 and XVad2 are the spatiotemporal and temporal input validation data, and YVad2 is the output validation data. Adam is the optimizer used in all the deep learning models with MAPE loss function. ep is the number of epoch required by the model to converge.

C.1 Sample RNN code

The sample code for single-layer RNN is given below. node_rnn is the number of units in the RNN layer, n_input is the size of input which is defined by the length of the sequence, and n_dim is the number of attributes in the data. Similarly, RNN with multiple layers will be created.

```
from keras.models import Sequential
from keras.layers import Dense
from keras.models import Model
from keras.layers import Input
from keras.layers import SimpleRNN
model = Sequential()
model.add(SimpleRNN(node_rnn, input_shape=(n_input,n_dim)))
model.add(Dense(n_dim, activation='linear'))
model.compile(optimizer='adam', loss='mape')
```

```
history = model.fit(Xtrain, Ytrain, validation_data
= (XVad, YVad), epoch=ep, verbose=0)
```

C.2 Sample CNN code

The sample code for single layer 2 dimensional CNN with flatten and dense layer is given below. n_kernel is the number of filters in the CNN layer, s_kernel is the size of the filter, n_input is the size of input which is defined by the length of the sequence, n_dim is the number of attributes in the data. In a similar manner, CNN with multiple layers will be created.

```
from keras.models import Sequential
from keras.layers import Dense, Conv2D, Flatten
input_shape = (n_input,n_dim, 1)
model = Sequential()
model.add(Conv2D(n_kernel, kernel_size=(s_kernel,s_kernel),
input_shape=input_shape))
model.add(Flatten())
model.add(Dense(n_dim))
model.compile(optimizer='adam', loss='mape')
history = model.fit(Xtrain, Ytrain, validation_data
= (XVad, YVad), epoch=ep, verbose=0)
```

C.3 Sample LSTM code

The sample code for single-layer LSTM is given below. node_lstm is the number of units in the LSTM layer, n_input is the size of input which is defined by the length of the sequence, and n_dim is the number of attributes in the data. Similalry, LSTM with multiple layers will be created.

```
from keras.models import Sequential
from keras.layers import Dense
from keras.models import Model
from keras.layers import Input
from keras.layers import LSTM
model = Sequential()
model.add(LSTM(node_lstm, input_shape=(n_input,n_dim)))
model.add(Dense(n_dim, activation='linear'))
model.compile(optimizer='adam', loss='mape')
history = model.fit(Xtrain, Ytrain, validation_data
= (XVad, YVad), epoch=ep, verbose=0)
```

C.4 Sample LSTM-LSTM Code

The former LSTM model learns the feature vector and stores it as cell state to the latter LSTM model. The latter LSTM model learns the features from input data and current cell state. Here, LSTM1_n_input and LSTM2_n_input is the size of the input in the former and latter LSTM model, and LSTM1_n_units and LSTM2_n_units define the number of nodes in a layer. n_out is the number of units in the dense layer of the latter LSTM model. This code uses a single layer for each LSTM model. Depending upon hyperparameters LSTM-LSTM with multiple layers can be defined.

```
from keras.layers import Dense
from keras.models import Model
from keras.layers import Input
from keras.layers import LSTM
from keras.layers import Dense
def define_models(LSTM1_n_input, LSTM2_n_input, LSTM1_n_units, LSTM2_n_units):
        # define training LSTM-LSTM
        LSTM1_inputs = Input(shape=(None, LSTM1_n_input))
        LSTM1 = LSTM(LSTM1_n_units, return_state=True)
        LSTM1_outputs, state_h, state_c = LSTM1(LSTM1_inputs)
        LSTM1_states = [state_h, state_c]
        print(state_h.shape)
        # define training LSTM2
        LSTM2_inputs = Input(shape=( None, LSTM2_n_input))
        LSTM2_lstm = LSTM(LSTM2_n_units, return_sequences=True,
        return_state=True)
        LSTM2_output, _, _ = LSTM2_lstm(LSTM2_inputs,
        initial_state = LSTM1_states)
        LSTM2_outputs = Dense(n_out, activation='linear')(LSTM2_output)
        model = Model([LSTM1_inputs, LSTM2_inputs], LSTM2_outputs)
        return model
model = define_models(LSTM1_n_input, LSTM2_n_input,
LSTM1_n_units, LSTM2_n_units)
model.compile(optimizer='adam', loss='mape')
history = model.fit(([Xtrain1, Xtrain2], Ytrain2),
validation_data=(([XVad1, XVad2], YVad2), epoch=ep, verbose=0)
```

C.5 Sample CNN-LSTM code

The former CNN model wrapped in timedistributed provides the sequential feature vector to the LSTM model. The LSTM model uses this feature vector as a cell state. The LSTM model predicts the output based on cell states and temporal input. CNN_n_input and LSTM_n_input is the size of the input in CNN and LSTM input and n_out is the number of units in the dense layer of CNN. CNN_unit is the number of filters and kernel_size_cnn is the size of the filter. This code uses a single layer of CNN and LSTM model and similarly depends upon hyperparameters CNN-LSTM with multiple layers can be defined.

```
from keras.layers import Input, Reshape
from numpy import *
from keras.models import Sequential
from keras.layers import Dense
from keras.models import Model
from keras.layers import Input
from keras.layers import LSTM
from keras.layers import Conv2D
from keras.layers import TimeDistributed
def define_models(CNN_n_input, LSTM_n_input, n_out, LSTM_unit):
    CNN_inputs = Input(shape=((None, CNN_n_input,1))
    x1 = Conv2D(CNN_unit, kernel_size=(kernel_size_cnn,kernel_size_cnn)
    ,padding='same')(CNN_inputs)
    time_x1 = TimeDistributed()(x1)
    flat= Flatten()(time_x1)
    time_flat = TimeDistributed()(flat)
    outputscnn=Dense(n_out, name='predictions')(time_flat)
    LSTM_inputs = Input(shape=(none,LSTM_n_input))
    x = LSTM(LSTM_unit, return_sequences=True)(LSTM_inputs,
    initial_state=[outputscnn])
    LSTM_outputs = Dense(LSTM_output, name='predictions_op')(x)
    model = Model([CNN_inputs,LSTM_inputs],LSTM_outputs)
    return model
model = define_models(CNN_n_input, LSTM_n_input, n_out, LSTM_unit)
model.compile(optimizer='adam', loss='mape')
history = model.fit(([Xtrain1, Xtrain2], Ytrain2),
validation_data=(([XVad1, XVad2], YVad2), epoch=ep, verbose=0)
```

Appendix D Experimental Setup and Result Analysis for Single and Small Road Network

D.1 Experimental Setup and Result Analysis for Single Road

We generate the traffic data for a single road considering prolonged rainfall (100 mm in 6 hours) (S1). The traffic data is generated separately for arterial ($R_{arterial}$), sub-arterial ($R_{sub-arterial}$) and collector road ($R_{collector}$). We examine the performance of deep learning models for a single road. There is no dependency on other road segments. The traffic and weather data are available as point data.

D.1.1 Experimental Setup

The number of parameters is dependent on optimal structure, which is obtained by performing hyperparameters tuning. Table [D.1 - D.5] shows the most useful set of hyperparameters for the RNN, CNN, CNN-LSTM, LSTM and LSTM-LSTM model, respectively.

Table D.1: Hyperparameters for the RNN Model for Single Road Network.

Road Type	PH	p	L	NPL	E	γ
$R_{arterial}$	15	3	1	[10]	10	0
	30	6	1	[15]	12	0
	45	9	1	[20]	15	0.1
	60	12	1	[30]	18	0.1
$R_{sub-arterial}$	15	5	1	[12]	10	0
	30	7	1	[18]	12	0.1
	45	10	1	[27]	15	0.1
	60	12	1	[35]	20	0.2
$R_{collector}$	15	6	1	[18]	12	0.1
	30	9	1	[27]	15	0.1
	45	12	1	[30]	18	0.2
	60	15	1	[40]	22	0.2

Table D.2: Hyperparameters for the CNN Model for Single Road Network.

Road Type	PH	p	L	NPL	E	γ
$R_{arterial}$	15	3	1	[(16,3×3)]	8	0
	30	6	1	[(16,3×3)]	10	0
	45	9	1	[(32,3×3)]	12	0.1
	60	12	1	[(32,3×3)]	15	0.1
$R_{sub-arterial}$	15	3	1	[(16,3×3)]	10	0
	30	6	1	[(16,3×3)]	15	0.1
	45	9	1	[(32,3×3)]	18	0.1
	60	12	1	[(32,3×3)]	22	0.1
$R_{collector}$	15	5	1	[(16,3×3)]	12	0.1
	30	7	1	[(32,3×3)]	18	0.1
	45	10	1	[(32,3×3)]	20	0.2
	60	13	1	[(64,3×3)]	24	0.2

Table D.3: Hyperparameters for the LSTM Model for Single Road Network.

Road Type	PH	p	L	NPL	E	γ
$R_{arterial}$	15	3	1	[15]	12	0
	30	7	1	[20]	15	0
	45	10	1	[30]	18	0.1
	60	12	1	[40]	20	0.1
$R_{sub-arterial}$	15	5	1	[15]	12	0
	30	7	1	[30]	15	0.1
	45	10	1	[40]	18	0.1
	60	15	1	[50]	20	0.2
$R_{collector}$	15	5	1	[15]	12	0.1
	30	7	1	[30]	15	0.1
	45	10	1	[50]	18	0.2
	60	15	1	[60]	20	0.2

Table D.4: Hyperparameters for the CNN-LSTM Model for Single Road Network.

Road Type	PH	p	L	NPL	E	γ
$R_{arterial}$	15	3-3	1-1	16,3×3-10	12	0
	30	6-5	1-1	[(16,3×3)-12]	15	0
	45	9-6	1-1	[(32,3×3)-15]	18	0.1
	60	12-9	1-1	[(32,3×3)-20]	20	0.1
$R_{sub-arterial}$	15	3-3	1-1	16,3×3-10	15	0
	30	7-5	1-1	[((16,3×3)]-12	18	0.1
	45	10-6	1-1	[(32,3×3)-18]	22	0.1
	60	12-9	31-1	[(32,3×3)-24]	25	0.1
$R_{collector}$	15	3-3	1-1	16,3×3-12	15	0.1
	30	7-5	1-1	[(32,3×3)-18	18	0.1
	45	10-7	1-1	[(32,3×3)-30]	22	0.2
	60	13-9	1-1	[(64,3×3)-40]	25	0.2

Table D.5: Hyperparameters for the LSTM-LSTM Model for Single Road Network.

Road Type	PH	p	L	NPL	E	γ
$R_{arterial}$	15	3-3	1-1	[15-10]	12	0
	30	6-5	1-1	[20-12]	15	0
	45	9-6	1-1	[(30-15]	18	0.1
	60	12-9	1-1	[40-20]	20	0.1
$R_{sub-arterial}$	15	3-3	1-1	[15-10]	15	0
	30	7-5	1-1	[30-12]	18	0.1
	45	10-6	1-1	[(40-18]	22	0.1
	60	12-9	1-1	[50-24]	25	0.1
$R_{collector}$	15	3-3	1-1	[15-12]	15	0.1
	30	7-5	1-1	[30-18]	18	0.1
	45	10-7	1-1	[50-30]	22	0.2
	60	13-9	1-1	[60-40]	25	0.2

D.1.2 Result Analysis for Single Road

We test the model's performance for the point traffic and weather data. We are considering a single road; therefore, there is no spatiotemporal dependency. For scenario S1, we only consider the prolonged heavy rainfall ($\geq$ 100 mm in 6 hr) impact on traffic stream variables. Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) performance metric is used to compare the model's performance.

Figure D.1-D.3 shows the traffic speed prediction performance of RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM models for scenario S1 for $R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$ road type.

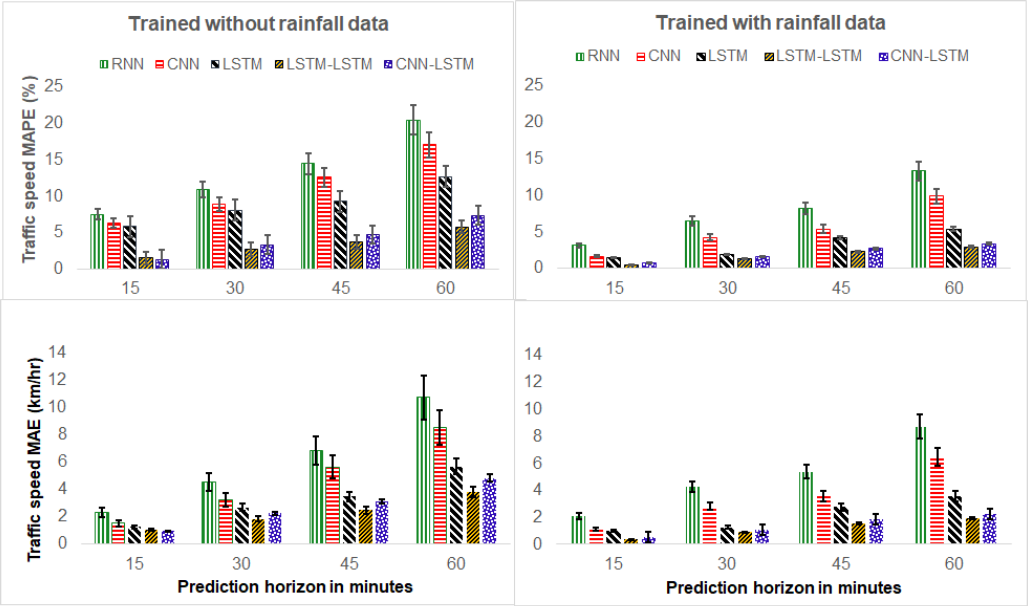


Figure D.1: MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for $R_{arterial}$ Road Type. Error Bar Shows the Standard Deviation.

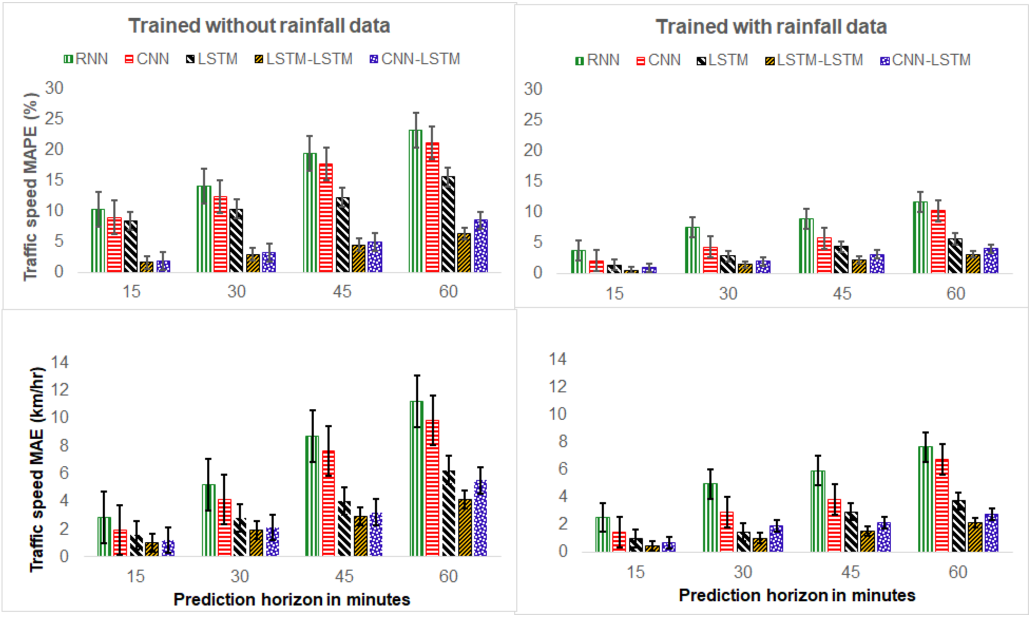


Figure D.2: MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for $R_{sub-arterial}$ Road Type. Error Bar Shows the Standard Deviation.

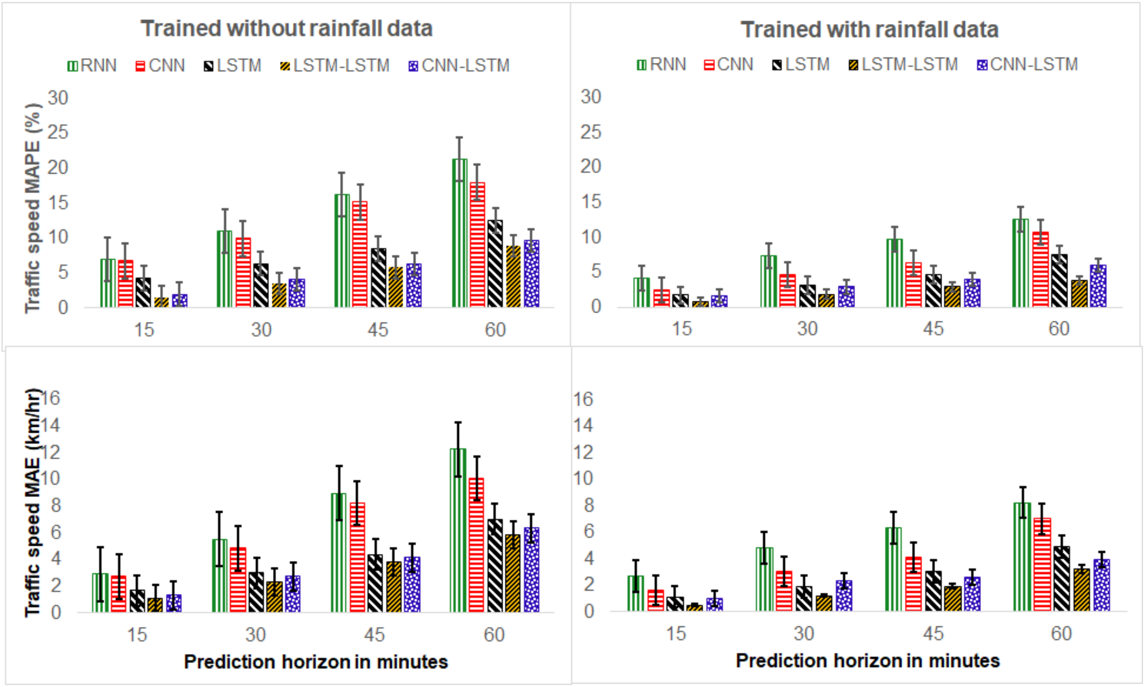


Figure D.3: MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for $R_{collector}$ Road Type. Error Bar Shows the Standard Deviation.

When trained with rain data, the model's traffic speed prediction improves compared to the model's traffic speed prediction when trained without rain data. For the 60-minute prediction horizon, the traffic speed prediction error of the LSTM-LSTM model when trained without rainfall data and when trained with rainfall data is 5% and 2%, respectively, for road type $R_{arterial}$. For similar prediction horizon and $R_{sub-arterial}$ road type, the prediction error of the LSTM-LSTM model when trained without rainfall data and when trained with rainfall data is 6% and 3%, respectively. For the $R_{collector}$ road type, the prediction error when trained only with traffic data and when trained with traffic and weather data is 8% and 3%, respectively. Comparing all the models' performance when trained with rainfall data, no significant difference is observed for each road type. Considering the 30-minute prediction horizon, for $R_{arterial}$ road type, the traffic speed prediction error of RNN is 6%, CNN is 4% LSTM is 1%, LSTM-LSTM is 1%, CNN-LSTM is 1%. For same prediction horizon the prediction error of RNN is 7%, CNN is 4%, LSTM is 2%, LSTM-LSTM is 1%, CNN-LSTM is 2%, for road type $R_{sub-arterial}$. The prediction error of RNN is 7%, CNN is 4%, LSTM is 3%, LSTM-LSTM is 1%, CNN-LSTM is 2%, for $R_{collector}$ road type. This is because it involves a single road, and both traffic and weather are point data, which provides accurate information about the weather impact on the traffic stream variables. Therefore even a non-hybrid model can learn the dependency between the traffic and weather data. With this observation regarding the single road network, we conclude that the traffic stream variables prediction during rainfall improves with rainfall data inclusion. Since there is no dependency on other road segment's traffic and weather data, hybrid and non-hybrid model's performance shows no significant difference.

D.2 Experimental Setup and Result Analysis for Small Road Network

We generate the traffic data for small road networks, considering each road segment receives an average rainfall of 100 mm in 6 hours (scenario S2). The traffic data is generated separately for arterial ($R_{arterial}$), sub-arterial ($R_{sub-arterial}$) and collector road ($R_{collector}$). We examine the performance of deep learning models for the small road network. The traffic and weather conditions of other road segments affect the target road segment's traffic stream variables.

D.2.1 Experimental Setup

Table [D.6-D.10] shows the most useful set of hyperparameters for the RNN, CNN, LSTM, CNN-LSTM, and LSTM-LSTM model, respectively.

Table D.6: Hyperparameters for the RNN Model for Small Road Network.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	6	1	[20]	12	0.1	3
	30	9	1	[40]	15	0.1	3
	45	12	2	[20,40]	16	0.3	3
	60	16	2	[20,50]	18	0.3	3
$R_{sub-arterial}$	15	6	1	[30]	15	0.1	3
	30	9	1	[45]	18	0.2	3
	45	14	2	[30,50]	20	0.3	3
	60	17	2	[20,60]	22	0.3	3
$R_{collector}$	15	10	1	[30]	15	0.2	3
	30	15	1	[50]	18	0.2	3
	45	18	2	[40,50]	20	0.3	4
	60	20	2	[40,60]	22	0.3	4

Table D.7: Hyperparameters for the CNN Model for Small Road Network.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	5	1	$[(16,3\times3)]$	10	0.1	3
	30	8	1	$[(32,3\times3)]$	12	0.1	3
	45	10	1	$[(32,3\times3)]$	15	0.3	3
	60	14	2	$[(32,3\times3),(16,3\times3)]$	20	0.3	3
$R_{sub-arterial}$	15	6	1	$[(16,3\times3)]$	12	0.1	3
	30	9	1	$[(32,3\times3)]$	14	0.2	3
	45	12	2	$[(28,3\times3),(14,3\times3)]$	18	0.3	3
	60	16	2	$[(48,3\times3),(16,3\times3)]$	22	0.3	3
$R_{collector}$	15	7	1	$[(32,3\times3)]$	15	0.2	3
	30	9	1	$[(48,3\times3)]$	18	0.2	3
	45	15	2	$[(32,3\times3),(16,3\times3)]$	20	0.3	4
	60	18	2	$[(64,3\times3),(32,3\times3)]$	22	0.3	4

Table D.8: Hyperparameters for the LSTM Model for Small Road Network.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	6	1	[20]	12	0.1	3
	30	9	1	[30]	15	0.1	3
	45	12	2	[20,50]	16	0.3	3
	60	15	2	[30,60]	18	0.3	3
$R_{sub-arterial}$	15	8	1	[30]	14	0.1	3
	30	10	1	[40]	16	0.2	3
	45	14	2	[20,60]	18	0.3	4
	60	17	2	[30,80]	21	0.3	4
$R_{collector}$	15	10	1	[40]	15	0.2	3
	30	12	1	[50]	18	0.2	4
	45	17	2	[30,50]	20	0.3	5
	60	20	2	[30,70]	24	0.3	5

Table D.9: Hyperparameters for the CNN-LSTM Model for Small Road Network.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	6-3	1-1	[16,3×3-10]	12	0.1	3
	30	8-5	1-1	[(16,3×3)-12]	15	0.1	3
	45	10-7	1-1	[(32,3×3)-(16)]	20	0.3	3
	60	12-7	1-1	[((32,3×3)]-[20]	25	0.3	3
$R_{sub-arterial}$	15	9-3	1-1	[16,3×3-14]	15	0.1	3
	30	12-6	1-1	[(16,3×3)-15]	18	0.2	4
	45	15-6	2-1	[(32,3×3),(16,3×3)-(20)]	25	0.3	4
	60	18-9	2-1	[((32,3×3),(16,3×3)]-[30]	29	0.3	4
$R_{collector}$	15	9-3	1-1	[16,3×3-16]	15	0.2	3
	30	13-6	1-1	[(32,3×3)-20]	19	0.3	4
	45	18-6	2-1	[(28,3×3),(12,3×3)-(30)]	25	0.3	4
	60	20-9	2-1	[(32,3×3),(16,3×3)-(40)]	32	0.3	5

Table D.10: Hyperparameters for the LSTM-LSTM Model for Small Road Network.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	7-3	1-1	[20]-[10]	14	0.1	3
	30	10-5	1-1	[40]-[20]	17	0.1	3
	45	12-5	1-1	[60]-[27]	25	0.3	4
	60	15-7	2-1	[20,40]-[30]	30	0.3	4
$R_{sub-arterial}$	15	9-3	1-1	[30]-[15]	15	0.1	3
	30	12-5	1-1	[40]-[20]	18	0.2	4
	45	15-5	2-1	[20,40]-[30]	27	0.3	4
	60	18-7	2-1	[30,60]-[30]	30	0.3	5
$R_{collector}$	15	9-3	1-1	[30]-[20]	18	0.2	4
	30	13-5	1-1	[50]-[20]	22	0.2	4
	45	17-6	2-1	[20,50]-[30]	29	0.3	5
	60	22-8	2-1	[30,80]-[30]	35	0.3	5

D.2.2 Result Analysis for Small Road Network

Figure D.4-D.7 shows the traffic speed prediction error of RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM models for scenario S2 for road type $R_{arterial}$, $R_{sub-arterial}$, and $R_{collector}$.

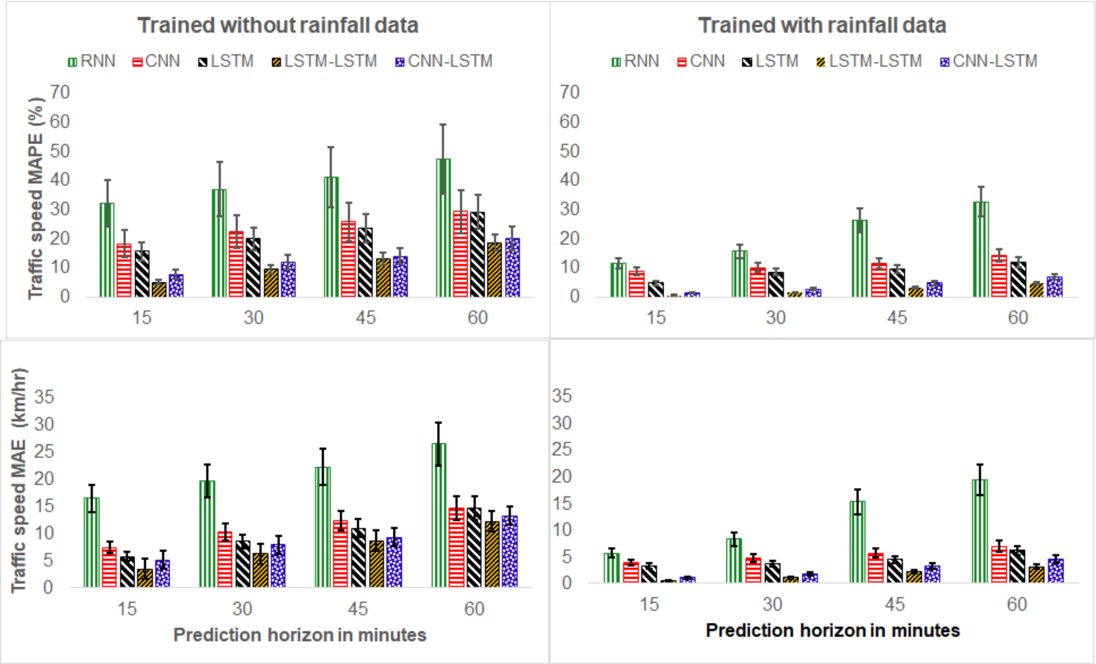


Figure D.4: MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{arterial}$ Road Type. Error Bar Shows the Standard Deviation.

In the scenario S2 for all road types, the models' traffic speed prediction performance when trained with rain data is more accurate than the models' traffic speed prediction performance when trained without rain data. For $R_{arterial}$ road type, the prediction error of the LSTM-LSTM model when trained without rainfall data is 18% and 4% when trained with rainfall data for a 60-minute prediction horizon.

Figure D.4 shows the traffic speed prediction performance of the models for the small road network of arterial road type, the model's traffic speed prediction error when trained with rainfall data is 31%, 14%, 11%, 4%, and 6% for RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM, respectively, for 60-minute prediction horizon. Figure D.5 shows the performance of the models for traffic flow prediction for arterial road type. The RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM models' traffic flow prediction error when trained with rainfall data is 42%, 20%, 19%, 5%, and 8%, respectively. Considering the case when model trained with rainfall data, for $R_{arterial}$ type of road, there is no waterlogging and water flow from the adjacent neighbors. For traffic flow prediction on $R_{arterial}$ type road, if there is heavy rainfall on adjacent roads, traffic flow on the target road is affected. For traffic flow prediction, LSTM-LSTM and CNN-LSTM model performance show significant improvement over LSTM, CNN, and RNN model for the $R_{arterial}$ road type. As dependency increases, the hybrid model performance is much better than the non-hybrid model performance.

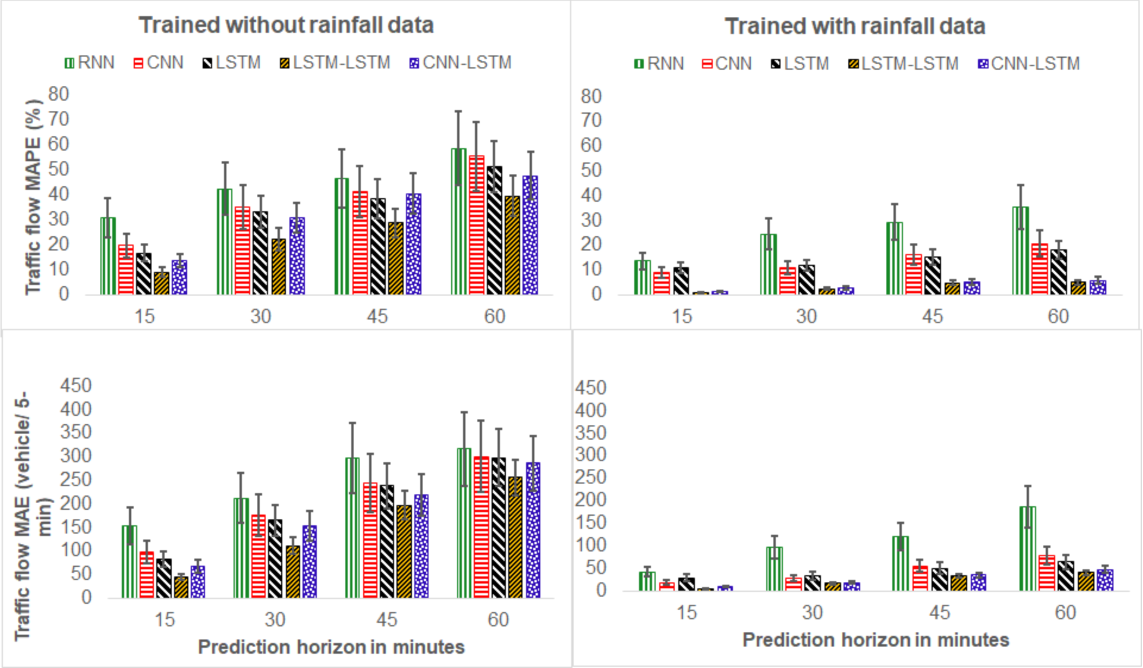


Figure D.5: MAPE (top) and MAE (bottom) of Traffic Flow Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{arterial}$ Road Type. Error Bar Shows the Standard Deviation.

For road types $R_{sub-arterial}$ and $R_{collector}$, a significant difference between the model performance when trained with and without rainfall data can be observed as shown in figure D.6 and D.7.

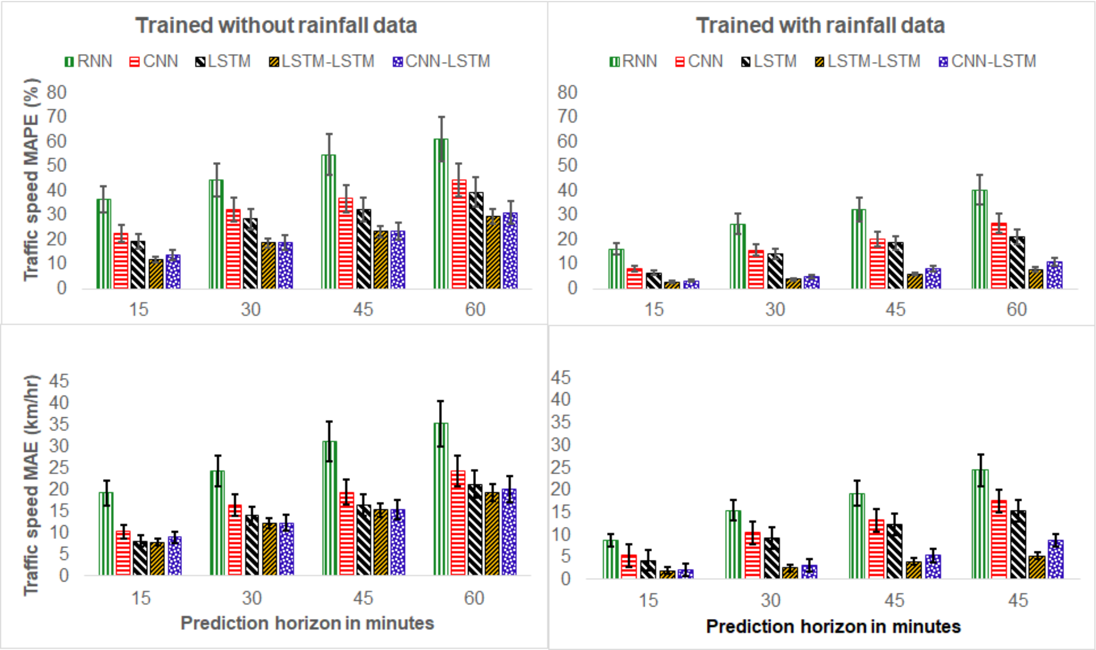


Figure D.6: MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{sub-arterial}$ Road Type. Error Bar Shows the Standard Deviation.

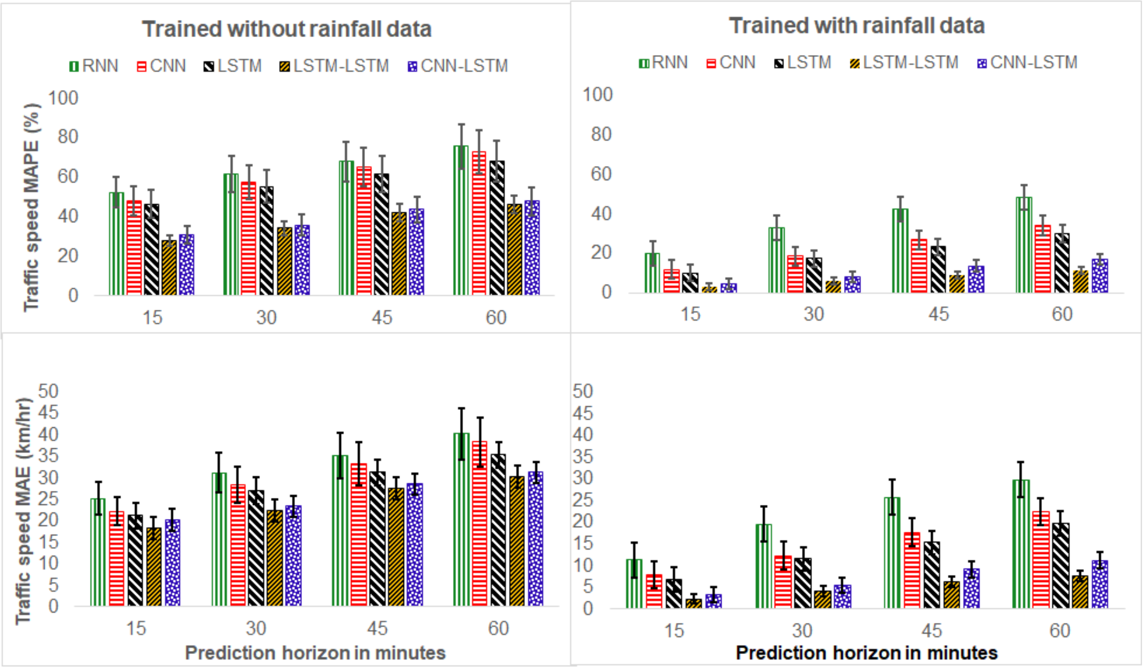


Figure D.7: MAPE (top) and MAE (bottom) of Traffic Speed Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{collector}$ Road Type. Error Bar Shows the Standard Deviation.

The traffic speed prediction error of the LSTM-LSTM model, when trained with and without rainfall data for 60-minute prediction horizon is 8% and 29%, respectively, for road type $R_{sub-arterial}$. For the similar prediction horizon and road type $R_{collector}$, the traffic speed performance (prediction error) of the LSTM-LSTM model when trained with and without rainfall data is 11% and 46%, respectively. The road type $R_{sub-arterial}$ and $R_{collector}$ depicts sub-arterial and collector roads' behavior under adverse weather conditions. For road type $R_{sub-arterial}$ and $R_{collector}$, we can observe that the target road traffic variables get affected due to rainfall at adjacent roads as water flows from one road segment to another. Similarly, waterlogging makes the situation worse. Therefore for road type $R_{sub-arterial}$ and $R_{collector}$ we can observe from the results that the performance of LSTM-LSTM and CNN-LSTM shows significant improvement over LSTM, CNN, and RNN models. For the 60-minute prediction horizon, when the model trained with rainfall data, the traffic speed prediction error of RNN is 40%, CNN is 26%, LSTM is 21%, LSTM-LSTM is 8%, and CNN-LSTM is 11%, for road type $R_{sub-arterial}$. For the similar prediction horizon and road type $R_{collector}$, the traffic speed prediction error of RNN is 48%, CNN is 34%, LSTM is 30%, LSTM-LSTM is 11%, and CNN-LSTM is 17%. From the results, we can observe that the prediction error increases for a larger prediction horizon because of error accumulation. In the case of non-hybrid models, this will reach a higher error for the larger prediction horizon than the hybrid models. The prediction error of the models for road type $R_{collector}$ is more than road type $R_{sub-arterial}$. $R_{collector}$ is the collector road that is most affected due to prolonged rainfall as it is highly prone to waterlog. The impact of rain on traffic stream variables is significant on $R_{collector}$ compared to other road types.

The traffic flow prediction error of the models for the scenario S2 for road type $R_{sub-arterial}$ and $R_{collector}$ is shown in figure D.8 and D.9, respectively. When trained with rainfall data, the model's traffic flow prediction error is less than the model when trained without rainfall data.

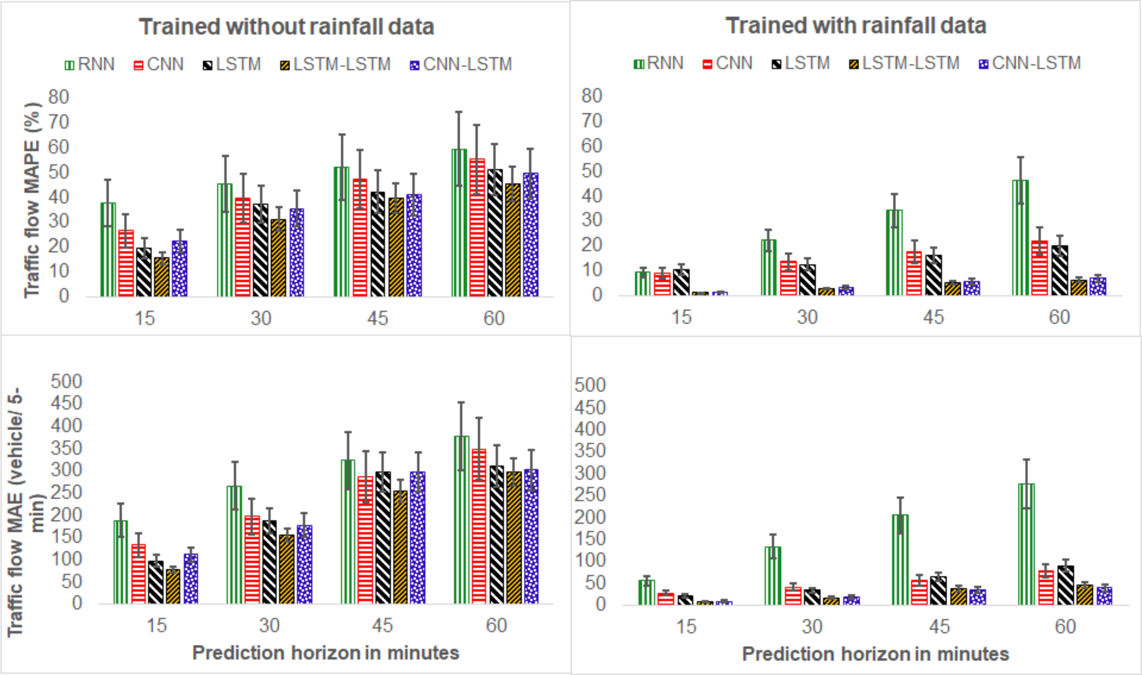


Figure D.8: MAPE (top) and MAE (bottom) of Traffic Flow Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{sub-arterial}$ Road Type. Error Bar Shows the Standard Deviation.

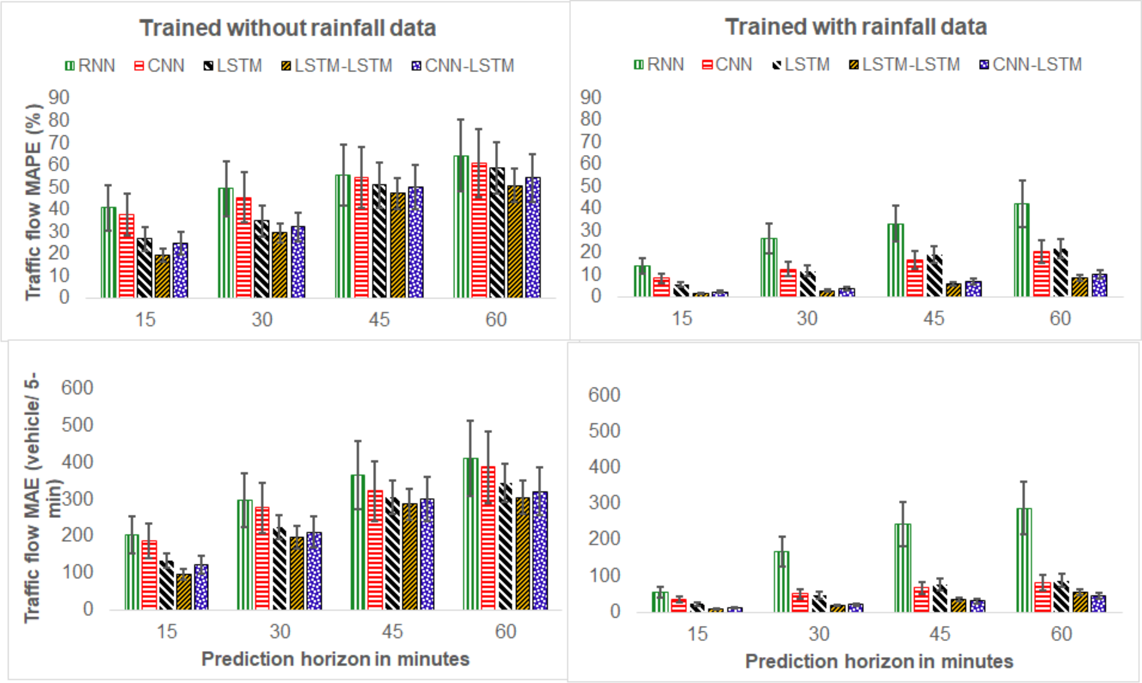


Figure D.9: MAPE (top) and MAE (bottom) of Traffic Flow Prediction when Model Trained Without Rainfall Data (left) and when Trained with Rainfall Data (right), for Scenario S2 and $R_{collector}$ Road Type. Error Bar Shows the Standard Deviation.

Therefore, for road type $R_{sub-arterial}$ and $R_{collector}$ significant improvement is observed in model performance with the inclusion of weather data. The prediction error of the models with inclusion of rainfall variable, is 46%, 21%, 20%, 6%, and 7% for RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM, respectively, for 60-minute prediction horizon. For similar prediction horizon and road type $R_{collector}$, the traffic flow prediction error with the inclusion of rainfall data is 52%, 20%, 21%, 8%, and 10% for RNN, CNN, LSTM, LSTM-LSTM, and CNN-LSTM, respectively.

The LSTM-LSTM and CNN-LSTM model's error improvement is significant over RNN, CNN, and LSTM model when trained with rainfall data for the small road network. But there is no significant difference is observed between the performance of the LSTM-LSTM and CNN-LSTM model.

With the results corresponding to a small road network, we conclude that hybrid models' performance in extracting the long-term weather dependency on traffic stream variables is better than the non-hybrid models. There is no significant difference between the CNN-LSTM, and LSTM-LSTM model's performance is observed. Also, due to error accumulation, the prediction error of the non-hybrid models is more than the hybrid models for a larger prediction horizon.

Appendix E Hyperparameters for the RNN, CNN, and LSTM Model for Large Road Network

Table E.1: Hyperparameters for the RNN Model for Large Road Network Scenario.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	12	2	[10,30]	10	0.5	3
	30	15	2	[20,40]	13	0.7	3
	45	18	3	[20,40,60]	16	0.7	5
	60	20	3	[30,40,70]	20	0.7	5
$R_{sub-arterial}$	15	12	2	[10,30]	10	0.5	3
	30	15	2	[20,40]	15	0.7	5
	45	20	3	[30,50,80]	18	0.7	5
	60	22	3	[30,60,100]	22	0.7	7
$R_{collector}$	15	15	2	[30,50]	12	0.5	3
	30	18	2	[40,60]	18	0.7	5
	45	22	3	[40,50,80]	20	0.7	7
	60	24	3	[40,60,110]	22	0.7	7

Table E.2: Hyperparameters for the CNN Model for Large Road Network Scenario.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	12	2	$[(32,3\times3),(16,3\times3)]$	12	0.5	3
	30	15	2	$[(32,3\times3),(16,3\times3)]$	17	0.7	3
	45	18	2	$[(32,3\times3),(16,3\times3)]$	20	0.7	5
	60	22	2	$[(48,3\times3),(32,3\times3)]$	25	0.7	5
$R_{sub-arterial}$	15	12	2	$[(32,3\times3),(16,3\times3)]$	12	0.5	5
	30	15	2	$[(32,3\times3),(16,3\times3)]$	17	0.7	5
	45	18	2	$[(32,3\times3),(16,3\times3)]$	22	0.7	7
	60	22	3	$[(64,3\times3),(32,3\times3),(16,3\times3)]$	27	0.7	7
$R_{collector}$	15	15	2	$[(32,3\times3),(16,3\times3)]$	15	0.5	5
	30	18	2	$[(32,3\times3),(16,3\times3)]$	18	0.7	7
	45	20	2	$[(32,3\times3),(16,3\times3)]$	25	0.7	7
	60	24	3	$[(64,3\times3),(32,3\times3),(16,3\times3)]$	28	0.7	7

Table E.3: Hyperparameters for the LSTM Model for Large Road Network Scenario.

Road Type	PH	p	L	NPL	E	γ	Δ
$R_{arterial}$	15	15	2	[20,40]	18	0.6	3
	30	18	2	[30,70]	20	0.7	3
	45	20	3	[30,50,70]	25	0.7	5
	60	22	3	[50,80,100]	32	0.7	5
$R_{sub-arterial}$	15	18	2	[20,50]	20	0.6	3
	30	20	2	[30,80]	25	0.7	5
	45	22	3	[30,60,80]	30	0.7	5
	60	27	3	[50,80,110]	35	0.7	7
$R_{collector}$	15	20	2	[30,50]	18	0.6	5
	30	22	2	[40,80]	20	0.7	7
	45	27	3	[30,60,80]	35	0.7	7
	60	30	3	[50,80,110]	40	0.7	7

Appendix F Traffic Stream Variables Prediction in Presence of Snowfall

We examine the performance of the model for traffic stream variables prediction during snowfall. Therefore, to capture the snowfall effect on traffic stream variables, we collect traffic and weather data across Minneapolis, Minnesota, Twin city data. The traffic data of the 1-hour aggregation period is collected from the Minnesota Department Of Transportation (MnDOT) [110]. The area being studied is Trunk Highway 10 (TH10) in Minnesota. The weather data is collected from National Weather Service Twin Cities [111], the aggregation period is 1 hour. The duration of data is from 10 January 2019 to 20 February 2019. Figure F.1 shows the performance of the models trained with and without snowfall data for 60-minute and 120-minute prediction horizons. The traffic flow prediction error of the LSTM-LSTM model trained without and with snowfall data is 31% and 12%, respectively, for a 60-minute prediction horizon. We conclude that weather variables show a significant impact on traffic stream variables. Therefore, model prediction improves with the inclusion of weather data.

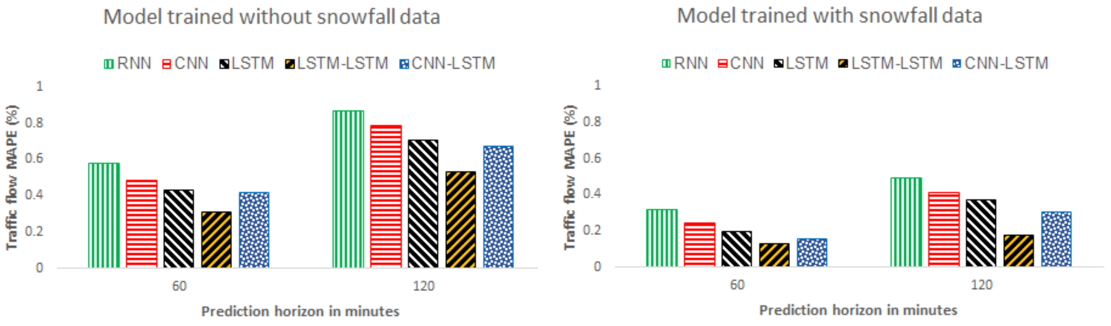


Figure F.1: Traffic Flow Prediction Performance of the Model Trained with and without Snowfall Data Corresponding to TH10 in Minnesota.