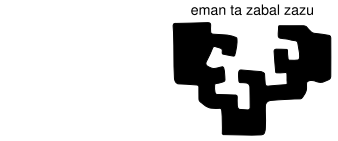
University of the Basque Country
UPV/EHU




Doctoral Thesis

---

# Beyond Short-term Traffic Forecasting Models: Navigating Through Data Availability Constraints

---

*Author:*
Eric L. Manibardo

*Supervisors:*
Prof. Dr. Ibai Laña
Prof. Dr. Javier Del Ser

*A Thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy*

*in the*

Department of Communications Engineering

January 26, 2024

"*The most inflated egos are often the most fragile*"

All Might

UNIVERSITY OF THE BASQUE COUNTRY UPV/EHU

# *Abstract*

Bilbao Faculty of Engineering
Department of Communications Engineering

Doctoral Degree

**Beyond Short-term Traffic Forecasting Models: Navigating Through Data Availability Constraints**

by Eric L. Manibardo

Short-term traffic forecasting supports route planning and decision making before traffic congestion occurs. Thanks to its direct application in real-world scenarios, short-term traffic forecasting remains as one of the hot topics within research on Intelligent Transportation Systems. During the last decade, researchers have heavily focused on proposing advanced and complex modeling techniques based on Deep Learning architectures. Motivated by the revolution Deep Learning has supposed to computer vision and natural language processing, authors continuously evaluate state-of-the-art methods on traffic forecasting datasets. However, published performance improvements are narrow. This Thesis conducts first a literature review on short-term traffic forecasting models, intending to shift the community research efforts beyond increasing the accuracy of traffic predictions. The experience accumulated in the course of the presented survey, allows drawing a road map of challenges an research opportunities for the years to come. Aiming to lead by example, several of the above challenges are directly addressed in this Thesis, in detail those that gravitate around different levels of data constraints.

Scholars rely on extensive datasets for adjusting proposed models, however, reality differs from these ideal experimental setups. Three levels of data availability are analyzed: 1) traffic measurements collected during a whole year; 2) traffic surveillance limited to a few weeks; 3) no traffic data available for a particular location. The first experimental setup aims to demonstrate that increasing the complexity of the models used for short-term traffic forecasting does not yield more accurate predictions in those scenarios where traffic recordings are accessible. The second case study explores how to learn traffic forecasting models under limited data holdouts, while maintaining a similar predictive performance regarding those models built without any data constraint. The last and most challenging scenario, delves into the characterization of sensorless locations. This research path has the potential of reducing the number of sensors permanently installed across a traffic network, by pairing sensorless road segments to those roads that share a similar traffic behavior.

# *Acknowledgements*

First of all I would like to say 'thank you' to Unai Irusta, may he rest in peace. As my professor, he was not only excellent during his teaching role, but also a friend that encouraged me to pursue new challenges outside pursuing my Master degree in Telecommunications Engineering. Without his support this project could not has been possible.

During my last years at the University of the Basque Country, I realized that becoming a data scientist could be more aligned with my professional goals and concerns. As fate would have it, one of my university professors turned out to be an expert in this field: Javier Del Ser (a.k.a. *Javi*). I managed to perform my Master Thesis project with Unai and Javi as directors, in which we developed an ECG classifier using Machine Learning techniques. Javi ended offering me a scholarship to do a doctoral thesis co-directed by the second pillar of this story: Ibai Laña. They have been in charge of providing me with the best conditions for pursuing this Thesis, while dealing with my intense and sometimes overwhelming personality. I wish to thank the major supports backing this Thesis for displaying such an excellent behavior as mentors. Now acting as friends, they were always willing to listen and give advice about my personal problems and life decisions. I can only be proud of them and brag about how lucky I was to have them as directors in what has been an enjoyable doctoral Thesis from start to finish.

The following acknowledgments may seem strange to those unfamiliar with the process involved in an industrial doctoral Thesis, but I want to thank DINYCON SISTEMAS and especially Roberto (i.e. its CEO) for believing in me when I was just a recent graduate and a bunch of promises. I also want to appreciate my years at TECNALIA, the research center responsible of supervising this Thesis. Here I was able to feel like home and hence work comfortably along with my teammates Alain, Aritz, Eneko, Esther, Iraide, Kakun, María, Sergio and Txus. I will like also to express my gratitude to Iñaki, who was always keen to welcome me in his lab when submission deadlines were just around the corner. In summary, all the people from TECNALIA have helped me every time I needed. Thanks to all of you.

From a personal perspective, discovering Japan has been a dream of mine. Its culture, landscape and gastronomy are, from my perspective, truly mesmerizing. I was lucky enough to live in Nagoya almost half a year, thanks to Takeda-sensei, who kindly offered me a place in his laboratory. I was even luckier to have the help of a Spanish-speaking teacher, Alex-sensei, who was able to support me during the countless administrative tasks required for pursuing a doctoral research stay in Japan. I am also especially fond of Sehun for understanding my homesickness, Atsushi for accompanying me to a sumo tournament and Quang for introducing me to Vietnamese cuisine. Finally thanks to Marcos for walking the streets of Tokyo with me. That journey holds a very special place in my memory.

Finally, I have to thank my parents for being an example of hard work. You have taught me that the most humble origin should not prevent me from dreaming big. Thanks to my uncle, aunt and cousin (a.k.a. *la prehma*) for being my shelter every Sunday. Thanks to Eguzkiñe for making me part of her family from day one. Thanks to my brother for understanding me without talking. I love having conversations with just looks and gestures. My last lines of gratitude are for my love, Noelia. She appeared in my life without warning, as a fallen star in the middle of the night. She soon became the sun that illuminates my path, which nowadays I can not conceive without her being beside me. Her unconditional support and cheerful personality provide me with permanent energy and happiness. I love you darling.

# Contents

**Abstract** **v**

**Acknowledgements** **vii**

**1 Introduction** **1**
1.1 Motivation and objectives . . . . . 2
1.2 Outline and contributions of the Thesis . . . . . 3
1.2.1 Chapter 2 . . . . . 4
1.2.2 Chapter 3 . . . . . 4
1.2.3 Chapter 4 . . . . . 4
1.2.4 Chapter 5 . . . . . 5
1.2.5 Chapter 6 . . . . . 5
1.3 Reading this Thesis . . . . . 5
1.3.1 Notes on the formulation . . . . . 6
1.3.2 Notes on the list of abbreviations . . . . . 7

**2 Background** **9**
2.1 Concepts and preliminaries . . . . . 9
2.1.1 Deep Learning . . . . . 9
2.1.2 Short-term traffic forecasting . . . . . 10
2.1.3 When Deep Learning meets traffic forecasting . . . . 13
2.2 Literature review . . . . . 15
2.2.1 Proposed taxonomy . . . . . 15
2.2.1.1 Characterizing the problem to solve . . . . 16
2.2.1.2 Categorizing Deep Learning architectures . 17
2.2.2 Understanding the popularity of Deep Learning . . . 20
2.3 Critical analysis . . . . . 22
2.3.1 When is a forecast considered to be long-term? . . . 23
2.3.2 Are traffic datasets correctly selected? . . . . . 24
2.3.3 Can models be trained with scarce data? . . . . . 25
2.3.4 Does contextual data yield any benefit? . . . . . 26
2.3.5 How is the data representation selected? . . . . . 26
2.3.6 Is feature extraction interesting for traffic data? . . . 27
2.3.7 What possibilities does data fusion offer? . . . . . 29
2.3.8 Are comparison studies well designed? . . . . . 29
2.4 Challenges and research opportunities . . . . . 31
2.4.1 Need for a centralized traffic data repository . . . . 32
2.4.2 New modeling techniques for traffic prediction . . . 33
2.4.3 Model actionability . . . . . 34

2.4.4 Understanding Deep Learning models . . . . . 35
2.4.5 Pseudo-real synthetic traffic data . . . . . 36
2.5 Summary and next steps of the Thesis . . . . . 37

**3 Evaluation of Traffic Forecasting Models 39**
3.1 Deep Learning for traffic forecasting . . . . . 40
3.1.1 Multi-Layer Perceptron . . . . . 40
3.1.2 Convolutional neural networks . . . . . 42
3.1.3 Recurrent neural networks . . . . . 43
3.1.4 Attention neural networks . . . . . 44
3.2 Randomization-based neural networks . . . . . 46
3.2.1 Extreme Learning Machine . . . . . 47
3.2.2 Random Vector Functional Link . . . . . 48
3.2.3 RVFL variants . . . . . 49
3.3 Description of the case study . . . . . 51
3.4 Materials and methods . . . . . 51
3.4.1 Data for evaluating model performance . . . . . 51
3.4.2 Considered data-driven methods . . . . . 53
3.4.3 Experiment design . . . . . 54
3.5 Experiments and results . . . . . 56
3.5.1 RQ3.1: Baseline performance benchmark . . . . . 56
3.5.1.1 Statistical analysis . . . . . 59
3.5.1.2 Insights distilled from the benchmark . . . 61
3.5.2 RQ3.2: Instability of random models . . . . . 63
3.6 Summary . . . . . 65

**4 Traffic Forecasting Models in Limited Data Regimes 67**
4.1 Model adaptation . . . . . 68
4.1.1 Transfer Learning . . . . . 69
4.1.1.1 Notations and definitions . . . . . 69
4.1.1.2 Transfer Learning in traffic forecasting . . . 70
4.1.2 Online Learning . . . . . 71
4.1.2.1 Online Learning in traffic forecasting . . . 71
4.2 Description of the case study . . . . . 72
4.3 Materials and methods . . . . . 72
4.3.1 Data for evaluating model adaptation . . . . . 73
4.3.2 Deep Learning architecture . . . . . 74
4.3.3 Experiment design . . . . . 75
4.4 Experiments and results . . . . . 78
4.4.1 RQ4.1: Transferring models with no updates . . . . 78
4.4.2 RQ4.2: Updating transferred models . . . . . 82
4.5 Summary . . . . . 83

**5 Traffic Characterization in the Absence of Data 85**
5.1 Sensorless characterization of traffic data . . . . . 87
5.1.1 Scarcity of urban traffic flow measurements . . . . . 87
5.1.2 Associating network design and traffic profiles . . . . 88
5.1.3 Graph representations for traffic forecasting . . . . . 89
5.1.4 Learning the relationship between roads . . . . . 90

5.2 Description of the case study . . . . . 91
5.3 Materials and methods . . . . . 92
5.3.1 Traffic data and graph representation . . . . . 92
5.3.2 Road feature embedding . . . . . 94
5.3.3 How to select a sensed road . . . . . 97
5.3.4 Traffic profiles and selection performance . . . . . 98
5.3.5 Approaches for generating synthetic samples . . . . 99
5.3.6 Association model and reference roads . . . . . 101
5.3.7 Sensor deployment optimization . . . . . 102
5.4 Experiments and results: Case-A . . . . . 103
5.4.1 RQ5.1: Can two similar roads be identified? . . . . . 105
5.4.2 RQ5.2: How does the similarity metric perform? . . 107
5.4.3 RQ5.3: Which is the best generating approach? . . . 110
5.4.4 Discussion and limitations of Case-A . . . . . 113
5.5 Experiments and results: Case-B . . . . . 114
5.5.1 RQ5.4: When to install a provisional sensor . . . . . 116
5.5.2 RQ5.5: Do the sensing masks generalize well? . . . . 119
5.5.3 RQ5.6: The relevance of reference roads . . . . . 121
5.6 Summary . . . . . 122

**6 Concluding Remarks** **125**
6.1 List of publications . . . . . 127
6.1.1 Other publications . . . . . 128
6.2 Future research lines . . . . . 129

**Bibliography** **131**

# List of Figures

1.1 Block diagram of the relationships between chapters. . . . . 6

2.1 Part (1/2) of traffic forecasting through the last decade . . 12
2.2 Part (2/2) of traffic forecasting through the last decade . . 13
2.3 Classification according to the problem to be solved . . . . 16
2.4 Classification according to the Deep Learning architecture . 18
2.5 Challenges and research opportunities . . . . . 32
2.6 QR code for accessing GitHub . . . . . 33

3.1 Architecture of a Multi Layer Perceptron . . . . . 41
3.2 Architecture of the different RVFL variants . . . . . 50
3.3 Considered data-driven methods . . . . . 53
3.4 Experiment design for Chapter 3 . . . . . 54
3.5 Short-term traffic forecasting performance benchmark . . . 57
3.6 Bayesian probabilities sampled via Monte Carlo . . . . . 60
3.7 Instability analysis of randomization-based methods . . . . 64

4.1 Location of the selected traffic sensors for Chapter 4 . . . . 74
4.2 Experiment design for Chapter 4 . . . . . 77
4.3 Part (1/2) of $R^2$ score evolution for offline settings . . . . . 78
4.4 Part (2/2) of $R^2$ score evolution for offline settings . . . . . 79
4.5 Part (1/2) of $R^2$ score evolution for online settings . . . . . 80
4.6 Part (2/2) of $R^2$ score evolution for online settings . . . . . 81
4.7 Road traffic flow comparison during a week of March 2017. 82

5.1 Location of the selected traffic sensors for Chapter 5 . . . . 93
5.2 Design of the road feature embedding . . . . . 97
5.3 Traffic profiles computed from weekdays or weekends only . 99
5.4 Sensor deployment optimization process . . . . . 102
5.5 Experimentation workflow for Case-A . . . . . 104
5.6 Part (1/2) of performance of the selection methods . . . . . 106
5.7 Part (2/2) of performance of the selection methods . . . . . 107
5.8 Embedding similarity against selection performance . . . . 108
5.9 Comparison of traffic profiles . . . . . 109
5.10 Detailed view of selected traffic sensors for Case-B . . . . . 115
5.11 Distribution of the solutions evolved by the algorithm . . . 117
5.12 Prediction performance comparison for the same weekday . 118
5.13 Validation and test error for every single week sensing mask. 120

# List of Tables

2.1 Surveys addressing traffic forecasting . . . . . 14

3.1 Traffic data sources for evaluating data-driven methods . . 52

4.1 Selected road segments for studying Transfer Learning. . . . 73

5.1 Mean and standard deviation of performance results . . . . 111
5.2 Summary on the results for the sensed locations . . . . . 113
5.3 Prediction performance benchmark for Case-B . . . . . 119
5.4 Forecasting performance using different reference roads . . . 121

# List of Abbreviations

| **Modeling Terms** | |
|---|---|
| **SLFN** | **S**ingle **L**ayer **F**eedforward **N**eural |
| **MLP** | **M**ulti **L**ayer **P**erceptron |
| **CNN** | **C**onvolutional **N**eural **N**etwork |
| **Conv1D** | **C**onvolutional **1D** |
| **RNN** | **C**onvolutional **N**eural **N**etwork |
| **LSTM** | **L**ong **S**hort **T**erm **M**emory |
| **ELM** | **E**xtreme **L**earning **M**achine |
| **RVFL** | **R**andom **V**ector **F**unctional **L**ink |
| **GNN** | **G**raph **N**eural **N**etwork |
| **GAN** | **G**enerative **A**dversarial **N**etwork |
| **ReLU** | **Re**ctified **L**inear **U**nit |
| **LV** | **L**atest **V**alue |

| **Traffic Terms** | |
|---|---|
| **ATR** | **A**utomatic **T**raffic **R**eader |
| **RCD** | **R**oadside **C**ar **D**ata |
| **FCD** | **F**loating **C**ar **D**ata |
| **POI** | **P**oint **o**f **I**nterest |
| **ITS** | **I**ntelligent **T**ransportation **S**ystems |

| **Performance Metrics** | |
|---|---|
| $\mathbf{R}^2$ | Coefficient of correlation |
| **MAE** | **M**ean **A**bsolute **E**rror |
| **RMSE** | **R**oot **M**ean **S**quared **E**rror |
| **NRMSE** | **N**ormalized **R**oot **M**ean **S**quared **E**rror |
| **CQV** | **C**oefficient of **Q**uartile **V**ariation |

| **Chapter 2** | |
|---|---|
| **PeMS** | Caltrans **Pe**rformance **M**easurement **S**ystem |
| **HA** | **H**istorical **A**verage |
| **ARIMA** | **A**uto **R**egressive **I**ntegrated **M**oving **A**verage |
| **XAI** | e**X**plainable **A**rtificial **I**ntelligence |
| **FRBS** | **F**uzzy **R**ule **B**ased **S**ystems |

| **Chapter 3** | |
|---|---|
| **LR** | Least-squares **L**inear **R**egressor |

| | |
|---|---|
| **KNN** | k **N**earest **N**eighbors |
| **DTR** | **D**ecision **T**ree **R**egressor |
| **SVR** | $\varepsilon$-**S**upport **V**ector **M**achine |
| **ADA** | **A**daboost |
| **RFR** | **R**andom **F**orest **R**egressor |
| **ETR** | **E**xtremely **R**andomized **T**ree |
| **GBR** | **G**radient **B**oosting **R**egressor |
| **XGBR** | e**X**treme **G**radient **B**oosting **R**egressor |
| **ATT** | **ATT**ention based neural network |
| **dELM** | deep **E**xtreme **L**earning **M**achine |
| **edELM** | ensemble deep **E**xtreme **L**earning **M**achine |

**Chapter 4**

| | |
|---|---|
| **TL** | **T**ransfer **L**earning |
| **OL** | **O**nline **L**earning |
| **DM** | **D**onor **M**odel |

**Chapter 5**

| | |
|---|---|
| **MFD** | **M**acroscopic **F**undamental **D**iagram |
| **uMFD** | upper bound **M**acroscopic **F**undamental **D**iagram |
| **SPBC** | **S**hortest **P**ath **B**etweenness **C**entrality |
| **RTP** | **R**epresentative **T**raffic **P**attern |
| **NSE** | **N**aïve **S**imilarity-based **E**stimation |

# Chapter 1

# Introduction

Transportation networks are the backbone of any thriving society, playing a pivotal role in its economic, social, and environmental progress [1]. By improving these networks, public authorities facilitate an efficient movement of goods and people, thereby fostering trade, enhancing accessibility to services and opportunities, and promoting social interaction. The concept of Intelligent Transportation Systems (ITS) refers to the application of advanced information and communication technologies to transportation infrastructure and vehicles with the aim of improving safety, efficiency, and sustainability [2]. Some examples of ITS applications and use cases are railway passenger train delay prediction [3], the airport gate assignment problem [4], adaptive control of traffic signaling in urban areas [5] and improvements in autonomous driving [6], to mention a few. The use and management of data represent the core of ITS systems, therefore their acquisition is crucial. In the context of road traffic networks, collecting traffic measurements involves deploying a wide array of sensors across different road segments to collect granular data about various parameters including vehicle speed or vehicle count (i.e. traffic flow). Thanks to these sensorized road networks, collected data allow developing data-driven models for predicting the traffic state. An accurate traffic state prediction, based on measurements of different nature (e.g. average speed, traffic flow, etc.), can be used to enhance traffic management and implement operational measures to relieve or prevent traffic congestion and its consequent implications [7], [8]. While numerous methodologies (e.g. statistical methods, time-series analysis, and Machine Learning algorithms) can be used for traffic data modeling, all of these approaches require reliable and high-quality data to produce accurate predictions.

Among the different purposes of traffic data modelling, *short-term traffic forecasting* refers to the process of predicting traffic conditions in the near future, typically within minutes to a few hours. This prediction is based on real-time and/or historical traffic data often collected from Automated Traffic Readers (ATRs). Traffic forecasting has been a demanded research topic from last decades, gathering a plethora of scientific publications every year, as can be seen in recent surveys on this topic [9]–[12]. Starting from the earlier attempts, in 1979 researchers proposed modeling traffic patterns using statistical models, such as ARIMA (i.e. AutoRegressive Integrated Moving Average) [13], [14]. However, these models assume

that the traffic behavior does not evolve over time, which makes the model obtain predictions from the common patterns while missing unusual traffic behaviors. In practice, the traffic state can rapidly change and fluctuate, hence a predictive model needs some kind of adaptation mechanism. The Kalman Filter algorithm was proposed as an alternative, due to its capacity to continuously update the selected state variable (e.g. traffic flow) [15]. Still, Kalman Filter is a recursive estimation algorithm designed to determine the state of a linear dynamic system under the influence of random noise. The traffic state is a complex phenomenon that evolves through time according to a *non-linear* dynamic process influenced by factors such as rush hour peaks, weather conditions, road incidents, and cultural habits.

The community soon realized that data-driven models, which can model non-linear processes, are the perfect choice towards building short-term traffic forecasting models [10]. These models leverage the knowledge encapsulated in data (i.e. its structure and pattern) for producing predictions of the future traffic state. As the name suggests, data quality plays a crucial role in the performance of data-driven models. For a model to provide useful predictions, data should be complete, free of noise and accurate (i.e. vehicle counts from a sensor should accurately reflect the number of vehicles that passed by). Since patterns are learned from collected data, a model that has access to different examples of traffic states will be likely to provide higher performance with respect to a second model that has learned only common traffic states. Still, the research society decided to improve traffic forecasting performance by exploring more powerful data-driven models.

## 1.1 Motivation and objectives

Although there are plenty of data-driven methods that can deliver accurate short-term traffic predictions, Deep Learning methods have monopolized the majority of publications of this type in recent years, becoming the reference for the research community when facing new forecasting problems [16], [17]. This predominance of Deep Learning methods for ITS problems is commonly justified by its theoretical ability to approximate any non-linear function [18], which is often the case of patterns underneath traffic time series [19]. As the epitome of model complexity, Deep Learning models have their own drawbacks in the form of the inability to understand their behavior [20], [21], the need for large quantities of data and the high computational demand, which makes Deep Learning models usually to demand powerful computational resources.

This Thesis aims to encourage new research efforts on data efficiency, or in other words, on how to exploit the most from different levels of data availability. Motivated by the above, this Thesis first produces a literature review on the use of Deep Learning for short-term traffic forecasting. The areas in which the use of Deep Learning can be justified are identified, as well as other scenarios where less computationally expensive data-driven methods provide similar or superior performance. Several research niches, open challenges and valuable research directions for the community are

provided thanks to the conclusions drawn from such study. Several of the detected challenges are addressed in this Thesis, aiming to spark the research effort of the traffic forecasting community to shift from model-based to data-based. As milestones to be completed, the following objectives have been established:

- **Surveying the traffic forecasting field**: a comprehensive review of the literature should expose that Deep Learning models are too complex for modeling the majority of traffic scenarios. Published works of the last decade are arranged according to new taxonomies, which have the potential for identifying new challenges and research opportunities unattended until date.
- **Producing a benchmark of traffic forecasting models**: performance between distinct modeling techniques can only be compared under identical scenarios. Authors often declare state-of-the-art results upon constrained datasets and specific conditions. A comprehensive benchmark is conducted for comparing modeling approaches upon a combination of contexts and traffic state variables.
- **Exploring alternative modeling methods**: the goal is to maintain *avant-garde* results for a family of models that can be trained faster and implemented on low capacity machines. Light-weight models are adjusted and compared those modeling approaches that furnish the prior benchmark.
- **Implementing models under data availability restrictions**: in the research context, public traffic datasets are available, however, in a real-world implementation traffic recordings are scarce and noisy. Reducing the amount of traffic data measurements required for developing capable forecasting methods has the potential of fastening the characterization of multiple road segments. Different techniques are analyzed on this behalf.
- **Characterizing sensorless road segments**: this implies producing a set of features that categorize a road segment without collecting traffic data. Expert knowledge from traffic managers can be exploited, so that two roads with a similar set of features do also share traffic behavior. Thus, the challenge is on how to distill such set of features so they represent the traffic behavior of a road segment.

These objectives are addressed in a sequential manner, since each objective arises from the conclusions distilled from pursuing the prior. Overall, this Thesis represents a journey through data availability constraints, where each chapter addresses a more challenging scenario regarding the amount of data available.

## 1.2 Outline and contributions of the Thesis

This Thesis starts with an extended state of the art analysis, which aims to provide the reader a comprehensive background of the short-term traffic

forecasting field along with the use of Deep Learning as its main modeling tool. The survey work, comprised in Chapter 2, intends to portray the current challenges and research opportunities for the traffic prediction task. The following chapters address some of the aforementioned challenges, presenting empirical experiments towards ensuring the veracity of the obtained results. The last chapter is dedicated to gathering conclusions and final thoughts drawn from the Thesis. A brief summary of each chapter is given below.

### 1.2.1 Chapter 2

This chapter presents a literature review focused on modeling techniques for short-term traffic forecasting. The motivation for conducting such a review is twofold: firstly, to provide the reader an extensive understanding of the current state of the traffic forecasting topic; secondly, to critically examine a research trend in terms of which real value recent advances have brought to this field. Novel Deep Learning methods from the last decade are categorized according to two different criteria. The target variable to be predicted and the context within data has been collected determines how traffic behaves and the difficulty of the task to be solved. Similarly, a fair comparison between models should keep into consideration the format of input data. Whether traffic data is arranged as a time series, as an image or expressed as a graph representation of the traffic network, only models of the same category should be compared. The conducted analysis permits to identify a set of challenges and opportunities, where several of them are addressed in the following chapters.

### 1.2.2 Chapter 3

This chapter produces a benchmark of different short-term traffic forecasting modeling techniques, intending not only to serve as a reference for performance comparisons, but also to propose good practices for prospective traffic forecasting studies. Four public access datasets provide real-world traffic data spanning all the combinations between urban/interurban roads and flow/speed as variables to predict. Several data-driven methods are selected, covering shallow, ensemble and Deep Learning methods. Modern data-driven models analyzed in the benchmark should provide a significant performance gain for justifying the increased model complexity with respect to other modeling approaches. Finally, less complex modeling techniques are proposed and added to the benchmark. Particularly, randomization neural networks are analyzed for the short-term traffic forecasting task.

### 1.2.3 Chapter 4

This chapter presents a methodology for developing traffic forecasting models under data availability restrictions. A case study is defined, where a target road segment that begins to be monitored using an ATR requires to be characterized as soon as possible. Hence, the goal is to implement

a model that provides accurate traffic predictions without the need of collecting data for a whole year (so all traffic patterns can be learned by the model). The inner weights of a model that predicts traffic at another similar location are transferred to the target model, in a similar fashion to Transfer Learning [22]. This accelerates the task of learning specific traffic behaviors to the point that only a month of data from target needs to be collected for producing a competitive model. Aiming to further improve the model's performance, the internal parameters of the model are continuously adapted using data collected by the ATR. This further improves the predictions capabilities under those scenarios not captured in the data used for building the model.

### 1.2.4 Chapter 5

This chapter defines a novel methodology for characterizing sensorless road segments. Without the need of any type of traffic data collected at the target location, the target road is associated with one monitored road. This association is carried out by comparing a variety of topographic and contextual characteristics, encapsulated as a set of numerical values. The so-called *road feature embedding* of two roads with similar traffic profiles must also share close numerical values. This collection of features merges topological traits such as the number of lanes and the road type, with other metrics that help to assess the centrality of the road segment (e.g. percentage of arbitrary routes passing through the target location). Obtained results demonstrate that is possible to find similar roads using the proposed method, and hence characterizing the traffic of a sensorless road segment.

### 1.2.5 Chapter 6

The final chapter of this Thesis summarizes the conclusions that have been reached throughout the course of several years of study. A list of contributions submitted to specialized journals and conferences is also given. Finally, open research opportunities are discussed, aiming at encouraging future research efforts in follow-up studies.

## 1.3 Reading this Thesis

No particular order needs to be followed for reading this Thesis, since each chapter provides the sufficient background to understand and assimilate its content. However, there is a topic that serve as a guiding thread for selecting the order in which chapters are arranged: how to approach different levels of data scarcity. Those readers unfamiliar to the short-term traffic forecasting field are recommended to start with Chapter 2, since it provides a comprehensive view of the field and hence, the necessary background for reading this Thesis. From such literature review, a set of research challenges are produced, motivating the following experimental chapters (i.e.

chapters 3, 4 and 5). This Thesis *navigates through data availability constraints*. Therefore, each experimental chapter addresses the prediction problem with an increased scarcity of traffic data, which makes a sequential reading highly recommended. Conclusions and insights are condensed at Chapter 6, a section of the Thesis that is intended to be revisited after finishing any other chapter, since it helps to understand how the results are related between different experiments. Figure 1.1 displays the structure in which this Thesis has been conceived.

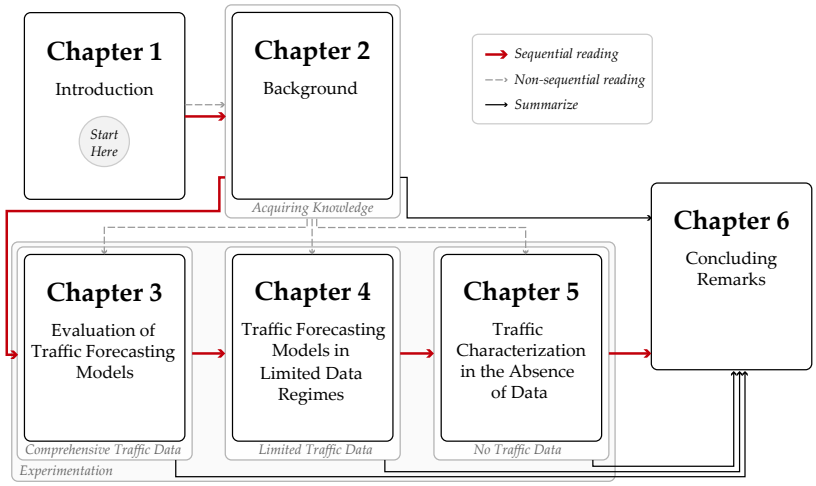



FIGURE 1.1: Block diagram of the relationships between chapters.

### 1.3.1 Notes on the Formulation

Some notation conventions have been established to describe the formulation of experimentation chapters. A list of the chosen notational principles is provided below, in order to facilitate comprehension:

- **Indexes**: denoted with a small Latin letter, usually starting with $i$ or $j$. Temporal indexes are specifically denoted with $t$.
- **Observations**: individual traffic readings are denoted with letter $o$. A superscript denotes the index of sensor, while a subscript refers to the temporal index e.g. $o_t^i$.
- **Vectors**: vectors of traffic readings, usually representing one day of observations are denoted with small bold letter, e.g. $\mathbf{o}$. Superscripts specify index of sensor and index of day they belong to e.g. $\mathbf{o}^{i,d}$. Vectors not related to traffic recordings are denoted using other Latin letters. For instance, $\mathbf{h}$ represents a set of hidden features produced by a hidden layer of a neural network.
- **Matrices**: matrices are represented by capital bold letters. For instance, $\mathbf{W}$ denotes the weight matrix of a neural network.

- **Datasets**: denoted with a calligraphic capital Latin letter, e.g. $\mathcal{D}$.
- **Functions**: denoted with small Latin letters, e.g. $f$.
- **Bars $\bar{}$ and hats $\widehat{}$**: These modifiers are added to observations or output vectors when they represent an average or an estimation respectively. For instance, Equation 3.21 refers to a real traffic state observation $o_t$, the predicted traffic state $\widehat{o}_t$ and the average traffic state among recorded observations $\bar{o}_t$.

### 1.3.2 Notes on the list of abbreviations

The number of acronyms in this Thesis is large partly due to the exploration of the state of the art on traffic forecasting, which yields a large number of methods, which in addition need to be put in a table, and thus their name must be abridged. Some of those methods are more common, and even used in the experimentation presented in this dissertation, while some others are used only once. For this reason, and trying to provide a list of acronyms that can be useful as a quick reference, the following criteria have been considered to include acronyms in it:

- Being a Thesis that addresses a myriad of data-driven methods, the modeling terms list is restricted to those that appear multiple times. If any term of this kind is used one time only, it is explained in place.
- All terms related to traffic (e.g. about traffic sensors), are included in the traffic terms list.
- The performance metrics employed in the obtained results are summarized in the corresponding list.
- The remaining abbreviations can be found on their corresponding chapters.

# Chapter 2

# Background

It is undeniable that the advent of Big Data has revolutionized most research fields [23]. Traffic forecasting is not an exception, where data-driven models have provided a huge performance increase on the prediction quality. In recent years, research efforts have been focused on developing complex Deep Learning architectures, leaving other emerging research topics unattended. This chapter analyzes the impact of this modeling approach on the literature. Singularities of Deep Learning architectures are introduced and explained, intending to shed light on why the research community has focused on improving prediction performance via modeling advances grounded on this family of models. A critical analysis is conducted over a literature review spanning the most recent years of research for short-term traffic forecasting using Deep Learning models. The above discussion is aimed at inspiring scholars to focus their research efforts on several challenges unattended until data. In summary, the primary goal of this chapter is to provoke a shift in current research efforts from model-centric to data-centric, as it is on the data constraints where usually real-world challenges emerge.

## 2.1 Concepts and preliminaries

Short-term traffic forecasting is one of the cornerstones for traffic management, as it is a reliable tool for regulating and maintaining traffic networks. On the other hand, Deep Learning comprehends a mixture of data-driven models with excellent results in many applications, which has stimulated a widespread adoption of this family of models for the short-term traffic forecasting task. With that in mind, the trajectory of both research fields and their relationships are reviewed in this section, which should provide a better understanding of how Deep Learning techniques have become dominant in the short-term traffic forecasting field.

### 2.1.1 Deep Learning

Machine Learning techniques provide a compendium of tools to develop data-based mathematical representations of real-world processes. These representations allow automatizing certain tasks or even predicting future states of the processes being modeled. As a subset of Machine Learning,

Deep Learning is inspired by the structure of human brains. The hierarchical composition of neural units, which are the fundamental building block of Deep Learning architectures, allows theoretically approximating any kind of non-linear function [24]. Since in nature there is an abundance of processes that can be modeled as non-linear functions, Deep Learning has quickly become the dominant approach in many applications. The capabilities of Deep Learning have been particularly relevant in natural language processing [25] and computer vision [26], among others, revolutionizing those fields. As a consequence, scholars are constantly applying these techniques to other areas of knowledge, seeking to extrapolate the benefits observed for these applications to other domains.

Deep Learning models, like other techniques belonging to different subsets of Machine Learning, can perform many tasks such as unsupervised learning, classification, or regression. Moreover, what makes them particularly relevant is their unique capabilities to automatically learn hierarchical features from data that are useful for the task under consideration. Classical Machine Learning methods are also called *shallow learning* methods because they cannot learn data representations directly from raw data. Feature extraction needs to be applied beforehand, often assisted by expert knowledge about the domain in which the problem is formulated. Deep Learning methods, however, can learn an implicit representation from raw data for a better understanding of the process to be modeled. This capability has been proven in certain cases to go beyond human reasoning. As a result, in fields dealing with complex, highly dimensional data, features discovered by Deep Learning methods lead to unprecedented performance with respect to the state of the art.

The other main capability of Deep Learning methods is their architectural flexibility, which suitability accommodates correlating together data of different nature (i.e. data fusion). Deep Learning flexible architectures allow for the different format data types to be merged, combining the information of multiple sources and extracting more knowledge about the process to model. Therefore, Deep Learning allows researchers to address complex learning problems, specially when dealing with highly-dimensional and diverse data.

### 2.1.2 Short-term traffic forecasting

The development of the short-term traffic forecasting field began when researchers started to apply time series forecasting methods to characterize traffic congestion measurements [13]. Back then, one popular approach relied on the assumption that the process that generated the traffic time series could be approximated using statistical methods like auto-regressive integrated moving average (ARIMA) [14], [27]. These predictive models were only capable of predicting a single target point of a road map.

With the beginning of the new millennium, the complexity of modeling techniques started to increase sharply, unleashing new research opportunities for the traffic forecasting realm. Vlahogianni et al. [10], who analyzed

short-term forecasting literature from 2004 to 2012, brought up that researchers are distancing themselves from what are considered classical statistical methods (i.e. auto-regressive models), drifting towards data-driven approaches [28]. The primary motivation for this shift remains on the ineffectiveness of classical methods to forecast while facing unstable conditions. The nature of the traffic is not stationary or linear, as a manifold of studies have hitherto shown [29]–[32]. Unfortunately, auto-regressive models tend to focus on the average behavior, so peaks and rapid fluctuations are generally missed [9]. Further into the review in [10], the literature analyzed therein inspected the scope of application, input and output data type, prediction horizon, and proposed technique of publications. Finally, challenges identified in this seminal review stressed out the overabundance of studies focused on freeways and interurban road traffic. Models for urban road traffic data were revealed to be less frequently studied. Furthermore, only a few solutions capable of predicting traffic simultaneously at different locations of the road network were known at the time [33]–[35], due to the scarcity of open-access traffic data for numerous points in a network, together with the high complexity of solving the interactions between the studied roads of the area.

After assimilating the criticism and challenges established in [10], another survey [11], published years thereafter, proposed new insights unattended until then. The newer literature review over the 2014-2016 period showed an increase in the number of publications focused on prediction at urban roads, which evinced that the research field covers nowadays most of possible geographic contexts of traffic prediction. Also in connection with the prospects in [11], there is also an increasing interest within the community in obtaining network-wide predictions, possibly promoted by the improvement in spatial data coverage and computing capacity achieved over the years [36], [37].

Among other points, [11] also underscored the need for establishing a unified set of metrics that permit to fairly compare performance between different models. Absolute error metrics provide interpretable values when comparing models for a single dataset, enabling a qualitative analysis of the error, as these express the error into traffic units (for instance, vehicles per hour). However, if the benchmark comprises several traffic datasets, relative error metrics should be considered for proper model comparison. This way the magnitude of the traffic unit does not affect the comparison study. Lastly, this survey highlighted an intrinsic problem of data-driven models: concept drift [38]. Since data-driven models acquire information from large data collections in order to extract traffic patterns and provide accurate predictions, performance is affected by exogenous non-planned events such as accidents, roads works or other circumstantial changes.

That same year, Ermagun et al. [39] analyzed the methodology and proposed methods for capturing spatial information over road networks. Their assumption is that present information of spatial relationships between road nodes should improve short-term predictive model performance. The study, which spans the period 1984-2016, offers an overview of the concerns of researchers in the field: 65.3% of revised works are concentrated

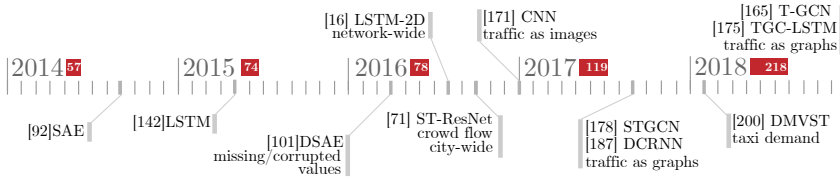



FIGURE 2.1: Part (1/2) of Deep Learning related milestones in short-term traffic forecasting during the last decade. Publications are ordered according to their publishing date. Horizontal red bars denote the number of works that were published each year concerning this topic (retrieved from Scopus in January 2024 with query terms: [`neural network` OR `deep learning` OR `deep neural network` OR `LSTM network` OR `deep spatio-temporal`] AND [`traffic prediction` OR `traffic forecasting` OR `traffic congestion` OR `traffic state prediction`], in title, abstract or keywords).

on traffic flow, 19.2% speed, and the remaining travel time. Likewise, only 26.5% chose urban zones as the implementation area, whereas the remainder are concentrated at freeways, confirming the postulated trend of Vlahogianni et al. in [10]. Finally, the survey concludes by encouraging the community to portray road networks as graphs [40], since they ease the representation of inter-nodal relationships and their subsequent use in modeling.

To round up this tour on the recent history of the field, in 2019 Angarita et al. [41] propose a general taxonomy for data-driven traffic forecasting models. The motivation of their work is not only to classify and revise learning models used to date, but also to categorize the approached traffic forecasting problems. In terms of data source type, data granularity, input and output nature, and overall scope. On the other hand, the reviewed models are sorted by preprocessing technique, type of in/out data, and step-ahead prediction. After analyzing the state of the art, they find no data-driven approach that suits all forecasting situations.

All the above surveys offer insights into the goals pursued by the field, as well as an outline of the opportunities and challenges that should be addressed in prospective studies. Vlahogianni et al. advocate for data-driven approaches, which were already gaining impulse at the time [10]. Posterior surveys confirmed this trend, and data-driven models prevail nowadays as the preferred option for short-term traffic modeling. The work of Laña et al. concludes that most possible geographic scopes are covered in the state of the art since, in the origins of the short-term traffic forecasting field, there was a shortage of publications based on urban traffic data [11]. In turn, Ermagun et al. grant importance to spatio-temporal relationships between nodes of traffic networks, which is one of the most exploited relationships to extract knowledge in the actual literature [39]. On a closing note, the taxonomy of Angarita et al. in [41] classifies traffic forecasting publications from a supervised learning perspective, which inspires in part the criteria later adopted in this work.

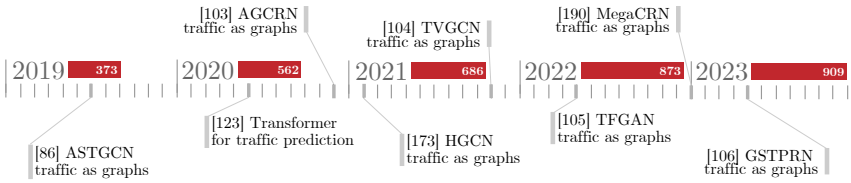



FIGURE 2.2: Part (2/2) of Deep Learning related milestones in short-term traffic forecasting during the last decade. Publications are ordered according to their publishing date. Horizontal red bars denote the number of works that were published each year concerning this topic (retrieved from Scopus in January 2024 with query terms: [`neural network` OR `deep learning` OR `deep neural network` OR `LSTM network` OR `deep spatio-temporal`] AND [`traffic prediction` OR `traffic forecasting` OR `traffic congestion` OR `traffic state prediction`], in title, abstract or keywords).

### 2.1.3 When Deep Learning meets traffic forecasting

As it can be concluded from the most recent surveys on short-term traffic forecasting, Deep Learning models have been applied in this research area mostly since the last decade. Figure 2.1 and Figure 2.2 depict a timeline with important milestones and achievements in short-term traffic forecasting approached via Deep Learning models. Among them, recent surveys that address short-term traffic forecasting in conjunction with Deep Learning methods are analyzed in this section (see Table 2.1), in order to highlight the need for the synthesis and investigation presented in this chapter.

Starting from [42], this work focuses on different Deep Learning architectures applied for short-term traffic forecasting, explaining their components and operation. A categorization of the reviewed models is presented, providing an overview of new modeling proposals. The second and third surveys [43], [44] analyze several Deep Learning methods for different transportation topics, including traffic signal control, autonomous driving and traffic state prediction. Therefore, the authors do not stress on the specific short-term traffic forecasting sub-domain, and only a few works concerning this topic are considered. A detailed review about Long Short-Term Memory-based Deep Learning models for short-term traffic forecasting is presented in [45]. The discussion is hence focused on this modeling technique and other approaches are not considered. Further away from the subject of short-term traffic forecasting, [46] revolves around spatio-temporal data mining as a general task that can be formulated in many application domains. Indeed, authors review Deep Learning models proposed for transportation and human mobility, but also take into account other unrelated topics like neuroscience and crime analysis. As a result, this survey only provides insights for some traffic forecasting solutions that benefit from spatio-temporal relationships. An introduction to the field of traffic forecasting is provided in [47]–[50], where the authors define the traffic prediction problem, summarize the state of the art on traffic prediction methods, and comment on the different Deep Learning architectures. However, the implications of using Deep Learning methods to solve traffic

TABLE 2.1: Published surveys that address short-term traffic forecasting based on Deep Learning methods. Column headers denote the citation reference of each publication. Rows correspond to different characteristics and content considered by the authors in each survey.

| **Survey** | [42] | [43] | [44] | [45] | [46] |
|---|---|---|---|---|---|
| **Considered period** | 1994 - 2018 | 1997 - 2017 | 1999 - 2018 | 2019 - 2022 | 2012 - 2018 |
| **# of reviewed works** | ~ 70 | ~ 15 | ~ 35 | ~ 80 | ~ 80 |
| **Target variable** | F, S, O | F, D | F, S, TT | - | - |
| **Context** | U, H | - | - | - | - |
| **Sensing technique** | - | - | - | - | - |
| **Temporal resolution** | Yes | - | - | - | - |
| **Dependencies** | ST, T | - | - | ST,T | ST |
| **Image representation** | - | - | - | - | - |
| **Graph representation** | - | - | - | - | Yes |
| **Coverage** | - | - | - | - | - |
| **# of steps ahead** | Yes | - | - | - | - |
| **Model type** | Yes | Yes | Yes | Yes | Yes |
| **Empirical study** | - | - | - | - | - |

| **Survey** | [47] | [48] | [49] | [50] | [51] |
|---|---|---|---|---|---|
| **Considered period** | 2015-2020 | 2014 - 2020 | 2016 - 2020 | 2015 - 2021 | 2014 - 2019 |
| **# of reviewed works** | ~ 65 | ~ 100 | ~ 55 | ~ 35 | ~ 40 |
| **Target variable** | F, S, D, O, TT | F, S, D, C, A | F, S, D, TT | F, S, TT | F, S |
| **Context** | - | - | - | - | - |
| **Sensing technique** | - | - | - | - | - |
| **Temporal resolution** | - | - | - | - | Yes |
| **Dependencies** | - | ST, T | ST, T | ST, T | ST, T |
| **Image representation** | - | - | - | - | - |
| **Graph representation** | Yes | Yes | - | - | - |
| **Coverage** | - | - | - | - | - |
| **# of steps ahead** | - | - | - | - | - |
| **Model type** | Yes | Yes | Yes | Yes | Yes |
| **Empirical study** | Yes | - | - | - | - |

| **Survey** | [52] | [12] | [53] | **This chapter** |
|---|---|---|---|---|
| **Considered period** | 2014 - 2019 | 2018 - 2020 | 2018 - 2020 | 2015 - 2023 |
| **# of reviewed works** | ~ 10 | ~ 210 | ~ 5 | ~ 165 |
| **Target variable** | F | F, S, D, O, TT, A | - | F, S, D, O, TT |
| **Context** | - | - | - | U, H |
| **Sensing technique** | Yes | - | - | Yes |
| **Temporal resolution** | - | - | - | Yes |
| **Dependencies** | - | - | ST, T | ST, T |
| **Image representation** | - | - | - | Yes |
| **Graph representation** | - | Yes | Yes | Yes |
| **Coverage** | - | - | - | Yes |
| **# of steps ahead** | - | - | - | Yes |
| **Model type** | Yes | Yes | Yes | - |
| **Empirical study** | - | - | Yes | Yes |

Note: The row "# of reviewed works" only takes into account publications related to short-term traffic forecasting based on Deep Learning methods. Any other unrelated reference has been filtered out and not accounted for in the reported quantities. F: Flow; S: Speed; D: Demand; O: Occupancy; TT: Travel Time; A: Accidents U: Urban; H: Highways/Freeways; ST: Spatio-temporal; T: Temporal.

forecasting problems are not discussed. The performance of recent Deep Learning methods is collected in a benchmark at [47], while [48] divides in five levels the complexity a Deep Learning architecture can present. Yet, in any of the above works insights are given in regard to whether Deep Learning performance is superior to those rendered by simpler learners. Next, both [51] and [52], provide an overview of existing Deep Learning methods for traffic flow forecasting. Future challenges for the research field are discussed in [51], such as a lack of well-established benchmark datasets, the inclusion of contextual data (e.g. weather data) and the development of graph-based modeling techniques. Finally, the current state of the art in graph neural networks applied to traffic forecasting is discussed in [12]. Studies reviewed in this survey are arranged by traffic graph type and the

composition of adjacency matrices, towards providing an overall picture of the trends in this specific research area. Further discussions about graph neural networks are presented in [53], where authors present a taxonomy for classifying these architecture in five categories. The performance of five graph-based methods (i.e. representative of each of the considered categories) is analyzed by collecting prediction metrics upon two datasets commonly used in the literature.

After analyzing the works summarized at Table 2.1, it can be concluded that they do not entirely provide a comprehensive, critical vision of the use of Deep Learning models for short-term traffic forecasting. Those who match the topic are restricted to an overview of the components of existing Deep Learning architectures, while the remaining ones revolve around general subjects like transportation or spatio-temporal data mining.

This chapter intend to go beyond an overview of recent Deep Learning techniques, towards answering other important questions such as *why?* and *what for?* Deep Learning models lead the majority of short-term traffic forecasting benchmarks, but often authors do not discuss the caveats related to their implementation. Some endemic features of Deep Learning do not comply with the requirements of traffic managers, including their computational complexity and black-box nature. Therefore, the adoption of such modeling techniques should be supported by other evidences and statements than a performance gain over other data-driven methods. Based on this rationale, this chapter does not elaborate on the different Deep Learning architectures used in the literature, but instead focuses on classifying it according to alternative criteria more aligned with the questions formulated above.

## 2.2 Literature review

In order to acquire a thorough understanding of the current use of Deep Learning techniques for short-term traffic forecasting, in this section a taxonomy for categorizing the published works during recent years is proposed. For this purpose, previous surveys serve as a starting point towards finding the common criteria that define these categories. A literature review covering the period 2015-2023 is performed subsequently as per the defined criteria.

### 2.2.1 Proposed taxonomy

The proposed taxonomy follows two complementary strategies that recursively appear as such in the literature. The first criterion determines and characterizes the traffic forecasting problem to be solved, whereas the second criterion categorizes the Deep Learning method(s) in use for tackling it. These criteria are described below.

#### 2.2.1.1 Criterion 1: characterizing the problem to solve

Research activity of short-term traffic forecasting comprehends multiple combinations of traffic measurements, which can be combined to achieve predictions of increased quality. To illustrate the taxonomy based on the first criterion, a tree diagram is constructed (Figure 2.3), which represents the patterns existing in the field. Splits' order is chosen according to their effect on the proposed problem. This way, features that yield a major discrepancy for the addressed approach are placed at higher levels of the tree, and vice versa.

Following the above guidelines, the first split is made according to the nature of traffic measurements. After reviewing the short-term traffic forecasting literature, two main strategies can be discerned: forecasting `flow`, understood as the number of vehicles that pass through the location of interest during a time interval, and `speed`, defined as the average speed

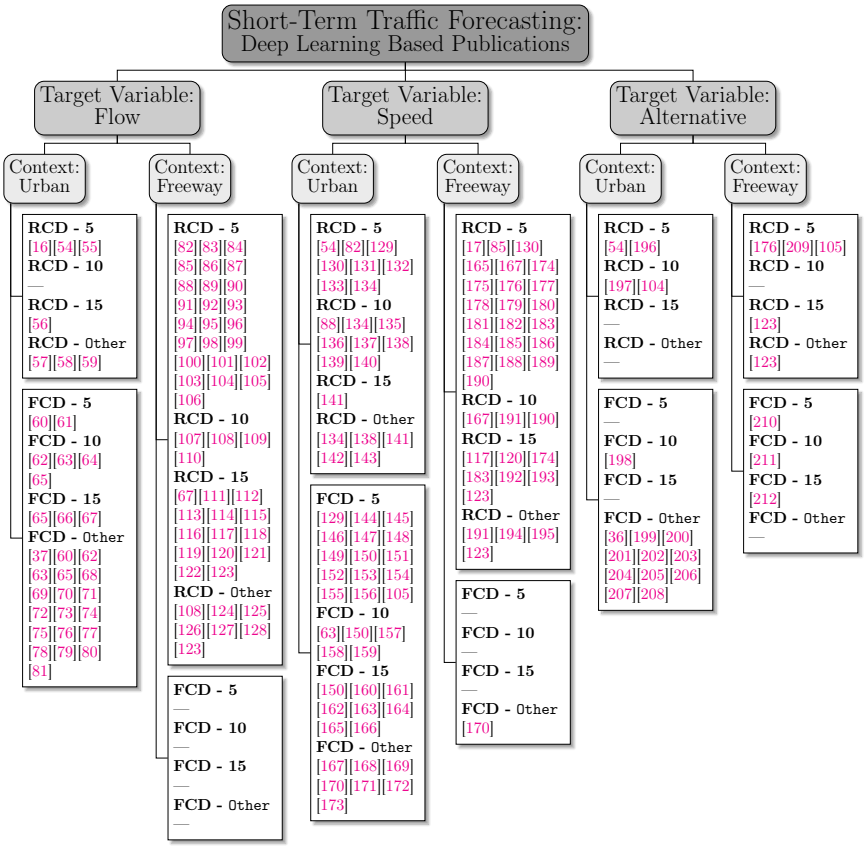



FIGURE 2.3: Contributions on Deep Learning based short-term traffic forecasting reported in the literature classified according to *Criterion 1*. From left to right, each branch level stands for nature of traffic measurements, traffic context, data collecting strategy, and temporal resolution in minutes. The `Other` keyword refers to other less used data temporal resolutions (e.g. 30 minutes.)

over a certain time period of all vehicles that traverse the target location. Other traffic measurements are travel time, occupancy, transport user demand (e.g. for taxis or bikes) and congestion level, all grouped under the category `alternative`, since the number of contributions that focus on these measurements is notably lower than the previous categories.

The second split in the tree considers the traffic context: `urban` or `freeway`. The different circumstances that occur in these contexts [213] generate more stable traffic patterns at highways, in contrast to urban routes, whose traffic flows are conditioned by traffic lights and signals, among other events.

The third split is set on how vehicular data are collected. Roadside sensing gathers measurements directly from road segments by using inductive loops, radar, or computer vision. On the other hand, GPS and other positioning sensing technologies allow tracking vehicle travel trajectory and speed by timestamped geolocalization measurements. These data collecting strategies are defined as Roadside Car Data (`RCD`) and Floating Car Data (`FCD`), respectively.

The last split addresses how the collected traffic data is aggregated. Sensors can feature different sampling frequencies, from a few seconds to several minutes. Since these sampling frequencies can impose – if high enough – a significant variability on the traffic measurement, the collected data is usually aggregated into lower temporal resolutions. Three prediction temporal resolutions `[5,10,15]` in minutes appear to be the most commonly used ones in the reviewed literature corpus. Additionally, the `Other` keyword appended at the labels of the third split refers to other less used data temporal resolutions, such as 30 minutes or 1 hour.

Before proceeding further, it is important to note that some publications may appear in multiple leaf nodes of the tree diagram. This is due to research work matching the criteria of different categories (for instance, if the proposed model predicts diverse traffic measurements, or if different kind of data sources are addressed).

#### 2.2.1.2 Criterion 2: categorizing Deep Learning architectures

Deep Learning architectures can be designed to adapt to diverse traffic scenarios. This design flexibility yields a heterogeneous mixture of modeling strategies. Under this premise, different features of Deep Learning methods are considered in this second criterion. A sunburst diagram (Figure 2.4) is selected to illustrate the different types of Deep Learning architectures proposed in the short-term traffic forecasting literature. The width of each angular sector is proportional to the number of research papers that fall within the category, relative to the total number of revised publications.

The most valuable information to predict the traffic state is usually that related to the target road. Previously collected data of the same road are in general good predictors of its short-term traffic profile. This statement is supported by the remarkable performance often offered by naïve methods such as the historical average [214], which computes the next traffic prediction value as the mean value of recent measurements at the considered point of the traffic network. On the other hand, historical

information of the surrounding areas (i.e. nearby roads) and measurements of downstream and upstream points of the same road have been lately incorporated to the input of the traffic forecasting model, as they can possess interesting correlations with the traffic of the target placement [215]. The spatio-temporal relationships between vicinity areas can provide better predictors of the traffic profile to be modeled [216], [217]. Those

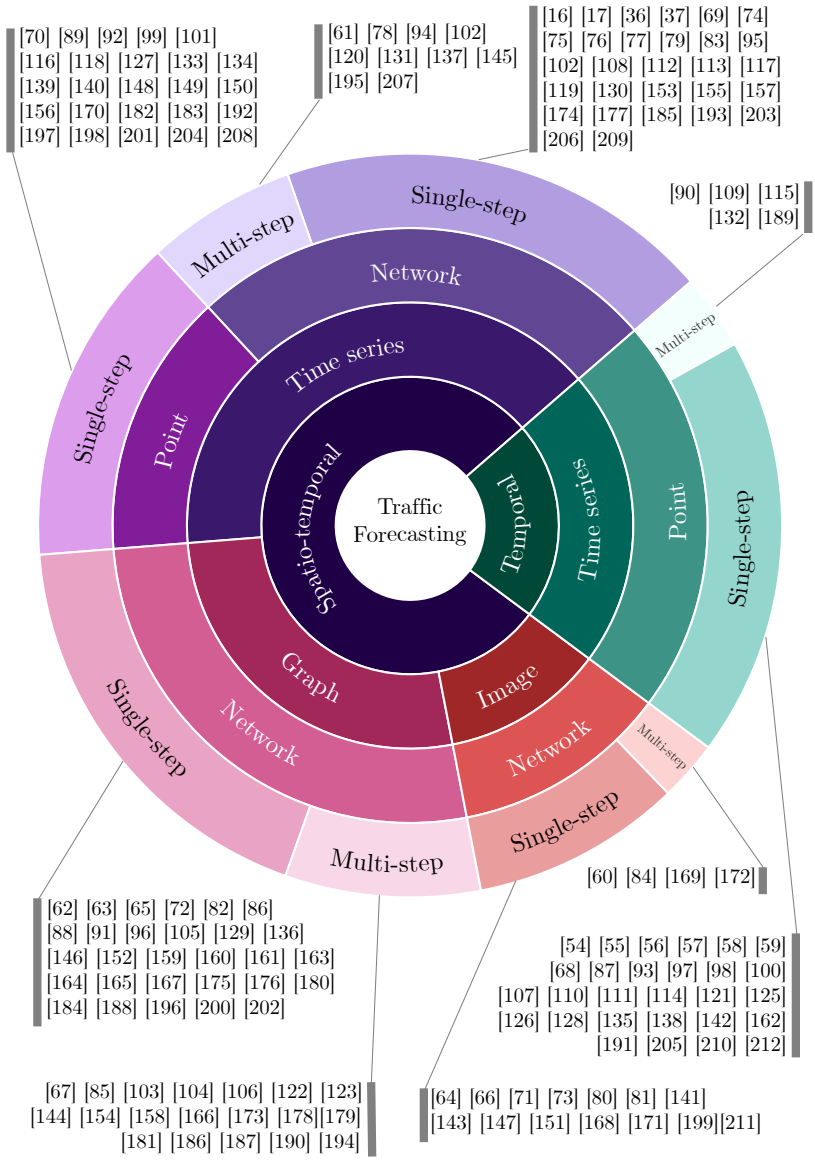



FIGURE 2.4: Deep Learning models for short-term traffic forecasting from examined works, classified according to *Criterion 2*. From inside out, each ring level stands for: considered dependencies, data representation format, range of coverage, and number of steps ahead prediction.

publications that feed the forecasting model exclusively with temporal data collected from the target road are categorized as `temporal`, whereas those that also resort to traffic measurements of other points in the same road network are categorized as `spatio-temporal`.

The next considered split is the format in which traffic measurements are expressed. Data related to traffic conditions are usually represented as time series, since their values are correlated through time [13]. Those publications that follow a traditional time series forecasting approach are cataloged as `time series`. Another possible approach consists of expressing the traffic state as an `image`. The great development in Deep Learning architectures (in particular convolutional networks) has led to a revolution in the image processing field [218]–[220]. In the context of traffic forecasting, the concept idea is to develop a model that predicts an image with traffic states (e.g. an image of a traffic network colored according to congestion levels). The predicted image can be transformed to express average speed, road congestion, and other traffic descriptors. Processing image representations of traffic networks allows predicting at once the traffic state at various roads of the network. The last considered format in this second split consists of expressing traffic data as graphs. Since traffic is restricted to road networks, it can be formulated as a `graph` modeling problem, where the structure of the road network is abstracted as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ [221]. In $\mathcal{G}$, $\mathcal{V}$ is a set of $N$ nodes representing road locations, whereas $\mathcal{E}$ is a set of edges representing the roads connecting such locations, and $\mathbf{A} \in \mathbb{R}^{N \times N}$ is an adjacency matrix, in which each element $a_{i,j}$ represents a numerical quantification of the *proximity* between nodes of the network in terms of traffic flow (e.g. the reachability from one node of the graph to another, or the intensity of traffic between them). This representation of a road network and its traffic, and the use of graph embedding techniques for their input to the Deep Learning models allow providing network-wide predictions and learn from the relationships between nodes. For instance, a graph representation of the traffic network is exploited in Chapter 5 for characterizing the traffic of sensorless locations.

Further along this second split, predictive models can be designed to forecast traffic state for one or multiple points of a traffic network. Those works that provide network-wide predictions are classified as `network`. In the case where models predict the traffic state of a single road, the research work at hand is labeled as `point`. Some studies predict different road congestion states simultaneously by using multiple models, but because the spatial coverage for each model remains to one road they are also cataloged as `point`.

The fourth considered split is the number of steps-ahead predicted by the model. For the simplest case, the model forecasts a single step-ahead point of the sequence (`single-step`), but there are models capable of predicting multiple steps ahead (`multi-step`). Multi-stage prediction consists of generating a multiple steps-ahead forecasts by using a single step-ahead model, which cyclically uses as input data the recently predicted values [222]. As this strategy employs single step-ahead models, the corresponding contributions are classified as `single-step`.

### 2.2.2 Understanding the popularity of Deep Learning

Once revised works have been categorized by the proposed problem and by the chosen Deep Learning approach, an in-depth literature review is performed, in order to objectively assess the trends followed by the community in this field of research.

A first inspection of the taxonomy depicted in Figure 2.3 reveals that the 5-minute temporal resolution positions itself as the most common in the reviewed literature. Almost half of the distinct data collections used by the reviewed papers gather data using 5 minutes sampling frequency. In addition, this trend is strengthened by the presence of Caltrans Performance Measurement System (PeMS) [223], which is by far the most popular traffic database, and also employs this sampling frequency. The 10 and 15 minute temporal resolution has less available original data collections, but sometimes the authors aggregate 5-minute data to obtain these resolutions, so the number of publications in this context increases slightly. Lastly, other temporal resolutions (denoted by the `Others` keyword) deserve a special mention. This group merges uncommon values from 2, 3, 6, or 16 minutes to 1 or 2 hours. Some of these temporal resolutions are acquired from data collections that have been utilized only once. The 30 and 60 minutes temporal resolutions are, however, adopted in many works, usually based on FCD from taxi flow or transport user demand. Transport user demand predictions (e.g. number of bikes expected to be rented during a time interval) usually employs low temporal resolutions, as these rates suffice for capturing the collective behavior of the population.

When focusing on traffic flow forecasting models, there is a clear tendency towards using RCD from freeways. The high cost of roadside sensors makes them to be typically deployed on critical road sections such as freeways, so there are more data sources of this kind than from urban arterials. However, since RCD is highly biased by the deployment location, its potential is limited when developing general-purpose traffic forecasting models. Interestingly, to the best of the author's knowledge there are not FCD based reviewed works that forecast freeway flow. FCD that captures flow measurements is mainly obtained from taxis and logistics services, or from passengers carrying cell phones in the vehicle. In the case of urban flow prediction, there are several published works, yet the majority of them are conducted over taxi or bike floating data. Since this sensing technique only captures a fraction of the circulating vehicles, FCD is usually utilized to predict flow values of certain vehicles type, and is hence not suitable for *general* flow forecasting problems. Research contributions are more balanced towards traffic speed prediction, covering all data type and granularity combinations, except for FCD at freeways, where only one work has been found [170]. PeMS and Los Angeles County highway dataset (METR-LA) [224] are the preferred option when looking for freeway speed RCD. For the speed prediction task, FCD provide reliable measurements since the average speed of the sensed vehicles (even though it is only a part of the vehicle fleet) can be considered as the average circulation speed on the road for an specific time interval.

Lastly, the `alternative` label blends together a mixture of works that predict traffic congestion [36], [196], [197], [209], [211], expected travel time [176], [201], [212], occupancy [54] and traffic performance index [210]. A special mention must be made to those works which predict service demand, understood as the number of vehicles necessary to cover a passenger demand. In this context, taxi demand is the most covered scope, probably due to the high data availability [199], [200], [202], [205], [206], [208]. There are also works focused on sharing bike demand [198], [207]. In either case, the `alternative` label covers different combinations of data types and temporal resolutions, so there is not a clear trend in this subset of contributions.

When the focus is placed on the employed methodology, Figure 2.4 unveils a clear increase of published studies that combine spatial and temporal information over recent years [39]. There are three times more works of this nature compared to those based only on temporal information. For a publication to be classified as `temporal`, the presented study can only take advantage of the knowledge from historical records at the point for which a predictions is issued. Therefore, the input format can only be classified as `time series`, since `image` and `graph` data representations always express information from multiple points of a traffic network. In turn, the number of publications based on temporal information is combined with those based on spatio-temporal information, it can be seen that more than half of the works formulate the input data as time series, which is the basic format to express traffic state.

As stated in the work of Ermagun et al. [39], the number of works based on graph theory [225] has increased notably in recent years. This trend is also highlighted at Figure 2.2, where every major contribution revolves around leveraging graph representations. Describing a traffic network as a graph adds spatio-temporal relational information between the different places where traffic state prediction is required, providing network-wide forecasts. For the remaining input formats, traffic representation as an image is the least chosen option, with about an eighth part of reviewed works. Some of these studies generate images from time series transformations of different points of the network expressed as matrices. Since the model is fed with images, even if they are a representation of multiple time series, these publications are classified as `image`. Graph based, image based, together with some time series based model works, represent more than half of revised publications dealing with network-wide coverage solutions. While these studies usually concentrate on performing simultaneous predictions for multiple traffic network points, publications classified as `point` often put their effort on other specific issues like traffic signal processing [125], [148], the exploration of new data sources [68], [132], the improvement of performance under particular situations [116], [182] or missing data [56], [140], [177].

Finally, single-step models represent the majority of existing publications, as it is in general an easier modeling task when compared to multi-step prediction. There is a surprisingly high amount of contributions that provide network-wide multi-step prediction, considering the difficulty of

predicting multiple steps-ahead of traffic state values for different locations simultaneously. Usually authors select constrained test holdouts comprising several weeks only. Network-wide models are overfitted to the data distribution selected for evaluation, hence impressive results are obtained. However, the lack of published results over a comprehensive traffic dataset (i.e. includes at least one year of traffic data) supports the hypothesis that this technology is far from being actionable.

## 2.3 Critical analysis

A critical look to the preceding literature review raises some questions about the suitability of Deep Learning techniques for the task of short-term traffic forecasting: is it always the best choice? In this section, the main aspects of this consideration are assessed trying to answer to eight questions, and examined towards opening a debate:

1. *When is a forecast considered to be long-term?* The concept short-term and long-term are often employed interchangeably, regardless of the true scope of the problem to be solved.

2. *Are traffic datasets correctly selected?* Novel modeling methods are usually tested over an specific traffic problem (e.g. average speed prediction), but compared to methods that were developed for a distinct traffic problem (e.g. flow forecasting).

3. *Can models be trained with scarce data?* As a byproduct of Deep Learning models being powerful modeling methods, they have parameters to adjust orders of magnitude above less complex methods, which can make them to learn details and noise if data is limited.

4. *Does contextual data yield any benefit?* Learning methods can use input data from multiple sources, making it interesting to analyze the impact of feeding contextual data into the model, such as day of the week or climate.

5. *How is the data representation selected?* Traffic data can be arranged not only as a time series, but also as an image or even as a graph representation of the traffic network. The motivation for selecting one format over another is discussed.

6. *Is feature extraction interesting for traffic data?* Deep Learning models are able to learn data representations from raw data, which can be useful in those cases where insights can not be produced without processing the data.

7. *What possibilities does data fusion offer?* Besides contextual data, Deep Learning models can learn from data abstractions such as text, images or graphs, a capacity other data-driven methods do not have.

8. *Are comparison studies well designed?* There is not a set of established guidelines in what refers to how an experimental setup should be designed, which allow authors to compare novel methods under favorable conditions.

### 2.3.1 When is a forecast considered to be long-term?

The use of Deep Learning techniques for traffic forecasting is relatively recent, regarding the traffic forecasting research topic [42]. However, the frontier between short and long-term predictions seems to remain largely ambiguous for many authors, thus jeopardizing the identification of Deep Learning models devised to tackle one or the other problem. This lack of consensus hinders the proper selection of modeling counterparts in benchmarks arising in the newer studies, often featuring an assorted mixture of short- and long-term approaches.

Authors of some related works establish the distinction between short- and long-term forecasting in terms of the prediction horizon, claiming that a prediction further than one hour ahead should be considered as long-term. This is by all means an unreliable consideration since, for a model where the time between consecutively arriving samples is one hour, a one-hour-ahead prediction problem would translate to a one-step-ahead forecasting task. There are other shared interpretations by which short-term forecasting is assumed to cover only the very first timesteps (usually no more than five steps) disregarding the temporal resolution of the time series at hand. However, for a fixed temporal resolution, models can be prepared to directly output one particular forecasting horizon (e.g. twelve-step-ahead). This case would entail some authors to classify it as long-term, while others would claim that it is short-term prediction, as the model is trained to forecast only that specific timestep.

Intending to homogenize the meaning of these concepts, the applicability of both approaches is herein clarified. Short-term predictions allow travelers to choose faster and more efficient routes, by avoiding bottlenecks. Likewise, local authorities can quickly respond and hopefully circumvent traffic congestion. They are, therefore, *operational* models [226], whose predictions are restricted to delimited geographical areas, since the interactions of the surroundings affect the traffic itself. On the other hand, long-term estimations allow traffic managers to prepare and implement *strategic* measures in case of predictable eventualities, such as sports events, weather conditions, road pricing, or general strikes [227]. The management of large areas (i.e. city-wide) may improve, for example, the design of road side infrastructure [228], eventually leading to more fluent traffic.

Based on this rationale, short-term models are usually built based on recent past observations of the target road and its vicinity to estimate their immediate subsequent ones. Here is where the distinction between approaches can be made: the model construction methodology. Long-term traffic estimation models seek different traffic patterns (e.g. typical daily traffic profiles), and decide which of these patterns suits best the traffic behavior of the selected road for the date under choice [229]. The

chosen pattern among all those elicited by the model becomes the prediction for the entire interval. Therefore, long-term estimation is, in general, less accurate and prone to larger errors in the presence of unexpected circumstances or when the selected output traffic pattern is inaccurate. By contrast, they provide a general idea of the expected behavior that can be used by traffic managers to decide strategic measures. Short-term forecasting models, on the other hand, issue their predictions by learning from recent past observations, obtaining more reliable forecasts as the model has access to better predictors for the target variable.

### 2.3.2 Are traffic datasets correctly selected?

The literature review presented in this chapter unveils another issue: the majority of publications select only one data source or multiple of the same scope (for instance, traffic collected in highways or urban areas, but not from both in the same study). This trend is observable by placing attention on duplicated citations at different leaves of the tree diagram in Figure 2.3. Benchmarks comprising datasets of different characteristics is a good practice that should be widely adopted for assessing the performance of newly proposed Deep Learning models. As addressed in Section 2.2.1, there are some characteristics of a traffic forecasting problem that can appreciably affect the model performance, namely data source type, data source context, predicted variable.

From the perspective of the data source type, RCD is an integrated count of any transportation vehicle that passes through the sensor location, while FCD is usually collected by vehicle types like taxis, buses, trucks, or bikes. The different way in which these two data types are gathered can impact severely on the time series behavior, leading to mismatches in the performance comparison. Besides the data type, the data collecting context is also relevant. Urban traffic is regulated by road signs and light traffics, leading to a particular driving behavior with higher data dispersion. On the other hand, freeway traffic forecasting is an easier task when compared to urban, since traffic profiles are usually more stable in the absence of traffic signs, pedestrians and other urban circumstances. Lastly, the different predicted variables (flow, speed, travel time) can express traffic congestion states but have different profiles and behaviors. Traffic speed measurements conform to a stable signal over time that exhibits scarce yet deep valleys when a traffic bottleneck occurs. By contrast, traffic flow measurements often show different kinds of daily patterns, where the difficulty resides in predicting sudden spikes.

To sum up, a Deep Learning architecture providing good performance results for a certain traffic data source could fail to generalize nicely to other traffic data sources with different characteristics. This behavior can be detected after testing a proposed Deep Learning method, along with a mixture of data-driven algorithms, to a collection of traffic data sources with varying characteristics. Otherwise, the novelty of the proposed model should be circumscribed to the characteristics of the traffic data source(s)

over which it has been tested, rather than claiming for a superior model for traffic forecasting in the wide sense.

### 2.3.3 Can models be trained with scarce data?

The ITS community has leaned towards Deep Learning based on the premise that these techniques can extract knowledge from unprocessed data more effectively than shallow learning methods. This mindset might be mistaken, as shallow learning models are advantageous in scarce data scenarios.

The main reason for it is that shallow learning models often require fewer parameters to be fit, leading to faster and less computationally demanding training processes, but also to less complex models. Since Deep Learning architectures have a potentially large number of trainable parameters, larger datasets are needed to prevent algorithms to learn the detail and noise in the training data (*overfitting*), to the extent that, unless properly counteracted, it negatively impacts on the performance of the model in real-life scenarios.

Therefore, in the context of scarce data, shallow learning methods may overcome the performance of Deep Learning models whenever the validation and test stages are designed and carried out correctly. Some of the works analyzed in the previous literature study consider very small periods of traffic data for training and testing. It could be thought that the results of these works are biased, since one could intuitively expect that the traffic behavior changes between months, weekdays, and daily hours [230].

As an example, if a forecasting model is trained over February data, and tested over measurements collected in March, both winter months have similar traffic behavior. This issue with a limited training data context is precisely the case where Deep Learning is prone to overfitting, leading to a higher yet biased performance on a test set. After enough training epochs, the model is good at the exposed scenario: forecasting traffic at winter, non-vacation months. This means that this Deep Learning model can be proficient forecasting in these highly specific circumstances, but will probably have trouble to generalize to other scenarios, rendering it useless. Since shallow learning methods usually have less trainable parameters, they can potentially outperform Deep Learning models in this scarce training data scenario, due to a less overfitting over data distribution.

In order to avoid overfitting of the model, the training samples - trainable parameters ratio should be maintained high, and the more trainable parameters of a model, the more training data should be required [231]. If this availability does not hold, the results of Deep Learning modeling experiments can be excellent due to overfitting, and be far from the good generalization properties sought for realizable traffic forecasting, which can lead to inconclusive insights.

### 2.3.4 Does contextual data yield any benefit?

The performance of predictive models can be improved with information that does not directly express the road traffic state. It is referred to as *contextual data*, since this data indicates temporal, meteorological, social, or other circumstances that can indirectly influence traffic profile. Calendar information [232], is commonly used as an additional source of knowledge [69], [113], [168], [199], supported by the intuition that traffic profile varies between workdays and weekends [233]. Another option is to provide the interval of the day, ensuring that the learning algorithm is able to correlate the temporal instant with traffic peaks [131], [144], [145], [209]. Weather has also been shown to affect drivers' behavior, eventually having an impact in the overall traffic [234]. Precipitations, wind, fog, and extreme temperatures are considered as model inputs in many traffic forecasting publications, intended to help predicting unusual traffic profiles [58], [69], [113], [176]. In this line, air pollution can be used as a congestion predictor, based on the idea that certain pollutant gases (for instance, *CO, $CO_2$, and NOx*) are expelled by exhaust systems. Therefore, air pollution should increase during traffic congestion and high occupancy periods, so models can benefit from this relationship [65], [235]. Lastly, other events like demonstrations, sports games, or accidents can be fed to forecasting models in order to identify uncommon traffic profiles [37], [71], [200], [209].

In what regards to Deep Learning models, the inclusion of previously described contextual data does not differ from its implementation with other Machine Learning models. These contextual data can be expressed as time series (e.g. temperature or concentration lever of some pollutant), or as a discrete sequence of finite values (for instance, calendar information). Just by increasing their input dimensionality, both Deep and Machine Learning models can append new sources of knowledge towards enhancing forecasting performance. However, within the bounds of network-wide traffic predictions, Deep Learning architectures stand out in the use of contextual data. The model can be fed with dedicated contextual data for each node of the traffic network, such as accidents or road cuts. This inherent capability of Deep Learning allows flexible solutions where contextual data serve as input only by demand at specific points of the neural network, avoiding output prediction noise due to high dimensionality inputs.

### 2.3.5 How is the data representation selected?

As previously explained, short-term forecasting models are usually built upon recent past traffic state observations. The most common option, as it can be observed in Figure 2.4, is to express traffic measurements as a vector for single road state prediction, or as a matrix for multiple-point prediction. Some researchers transform traffic time series into images, and estimate the images that best represent the network behavior at the time horizon for which the prediction is issued. Other authors instead design graph representations of the traffic network, aiming to learn from the spatial relationships between nodes.

However, the choice of data representation format does not always respond to a practical consideration. Sometimes, the actual contribution of a published work is to effectively adapt traffic forecasting tasks to image-based Deep Learning architectures. The methodology with which the traffic information is transformed into an image is the claimed cornerstone of the proposed learning method. However, this traffic representation does not add any valuable knowledge to the field, as it is just another way of expressing a time series. When describing a network as a matrix, its structure predetermines the connections between the analyzed roads that a Deep Learning architecture is able to model. Convolutional filters (which are commonly used for image processing) usually look for adjacent values to discover interesting high-dimensional features, so the same information arranged differently can produce contrasting performance results. Moreover, the complexity of an actual road network can hardly be represented only by the nodes that have sensors on them (which are the ones considered for any data-based study). Thus, the *picture* that represents the road network is distorted with regard to the actual road network. For a convolutional filter, the adjacency of two pixels has a particular meaning in the way they are processed, but this adjacency can have very different meanings within a network in terms of real adjacency. Hence, the claimed "spatial" awareness that this kind of methods provide must be handled with caution. Anyhow, traffic forecasting as an image can be interesting when the inputs are indeed images, for instance, screenshots from navigation services, satellite imagery, or other similar sources, as this is its original data format.

On the other hand, graph theory suits better for network representations, by providing node relationships (both directed and non-directed variants [225]), which are indeed supplementary information. The underlying structure of traffic data conforms a non-Euclidean space, as a traffic network can not be modeled in a multi-dimensional linear space without losing information (for instance, direction of the edges or values associated to nodes) [236]. It is for this reason that graph representations are best suited for network-wide forecasting models, where topological information of the traffic network can be fully exploited by the model. In the case where graph modeling is not an option (e.g. unclear node assignment), time series arranged as a matrix provides a flexible and straightforward format.

### 2.3.6 Is feature extraction interesting for traffic data?

As previously stated in Section 2.1.1, the most recognized capability of Deep Learning models is their ability to learn hierarchical data representations autonomously, overriding the need for handcrafting features from traffic data. As per many related studies, it is often argued that any non-Deep Learning based traffic prediction model potentially achieves a lower performance due to the fact that Deep Learning is able to model long-term dependencies in data (as opposed to handcrafted features). However, this point of view can be debatable.

Feature engineering is a difficult task that requires time, effort and domain knowledge from researchers. Nonetheless, the problem is that the predictive power of the produced features directly conditions the performance of prediction models. When input data is not self-descriptive and genuine features are not available, Deep Learning may outperform shallow learning due to its capability to learn from raw data. Nevertheless, traffic data used as inputs for traffic forecasting directly express traffic state. As an example, when the average speed of the road is available, the speed value determines if drivers are facing a *free-flow* traffic state or different severity levels of bottlenecks. The model only needs to interpret these values to output a proper prediction, and probably will not need any additional features.

Traffic observations can indeed be processed to obtain more complex and specific indicators [215], [237], but models are often trained upon raw traffic data. Thus, it could be said that the feature values automatically extracted by Deep Learning architectures in recurrent networks are in fact, the extraction of long-term patterns, since short-term dependencies can be modeled by a multi-layer perceptron or other basic models. Furthermore, given the nature of the data handled in traffic forecasting, in many occasions the expert knows the recurrence patterns in advance, which makes the feature learning capability of Deep Learning less relevant for the prediction task.

Nevertheless, for those researchers who still select Deep Learning as the modeling tool for short-term traffic forecasting, there are several insights that can be drawn from the reviewed studies, especially regarding the selection of the best neural network architecture for the case under study. First, the nature of the variable to be predicted, such as flow or speed, is not as important for deciding which architecture to use as the way these measurements are expressed. In the case of time series, the nature of the target variable can imprint distinct behavioral patterns in the modeled sequence of data points. Recurrent neural networks are indeed designed to address this type of data, particularly when dealing with long-term patterns, so they should be considered the starting point for any investigation where traffic prediction is formulated as a time series forecasting task. On the other hand, when modeling spatio-temporal data arranged as a collection of time series, a stacked hierarchy of convolutional and recurrent neural layers are usually adopted, as convolutional layers allow capturing temporal features over information collected in different locations. When traffic variables are transformed to image data, convolutional networks are often utilized to expand the feature space by correlating nearby pixels, which may produce high-quality descriptors of the traffic state. Finally, the prediction of traffic congestion from graph representations is still an immature research area, in which neural, convolutional, recurrent and attention based networks are already eliciting promising results [12].

In summary, automated feature extraction is a powerful feature of Deep Learning, but in the context of traffic forecasting it is not a deciding factor for selecting this modeling approach over other data-driven methods. Instead, the representation of traffic data is a key aspect to embrace the

modeling capability of Deep Learning models in both space and time. Therefore, new studies with modern Deep Learning architectures reported to the community should consider the literature analyzed in this chapter and the proposed guidelines for properly arguing their modeling choice, as for deciding the baseline models that should be included in the scenarios under consideration.

### 2.3.7 What possibilities does data fusion offer?

In addition to traffic recordings, other types of data sources may improve the prediction accuracy of traffic forecasting models. Beyond the feature mapping capacity of Deep Learning methods, a motivational driver for using these techniques should be its capability for in-model data fusion.

Data fusion is defined as the capacity for automatically or semiautomatically transform information from different sources into a representation of the modeled process [238]. In this context, there are some data abstractions that can not be processed by shallow learning methods. For instance, graph theory is able to model traffic network topology, and therefore the relationships between neighboring interconnected roads. Researchers take advantage of this representation via graph embedding layers to enhance the overall prediction performance of the model, as it can learn the traffic stream direction directly based on how the nodes of the graph are connected [82], [161], [186]. Another example is text data, which is often asynchronously generated. There are some works that use Twitter messages [132] or queries issued for the same destination in a navigation service as congestion predictors [166]. Images are also data representations that can be processed by Deep Learning architectures. Some studies arrange snapshots of network-wide traffic congestion maps as a time series, and resort to Deep Learning architectures for motion prediction to estimate the future trajectory of objects [64], [211]. Other works convert traffic speed time series from multiple points of a traffic network into a heatmap, where color expresses the speed value [141], [171]. All these examples illustrate the way in which data fusion capabilities can be used to take advantage of the Deep Learning methods potential.

Finally, complex neural architectures can assimilate on-demand specific data sources like weather or air pollution, by directly inserting these features at specific layers (generally after convolutional and recurrent layers, as these data do not need feature mapping). The model would use this information only when needed (e.g. during a special event like a football match), disabling these inputs during normal operation, to reduce model output noise. It does not seem that the traffic forecasting research community has taken advantage of this capability, which could be considered even more interesting for this particular field than its feature extraction competence.

### 2.3.8 Are comparison studies well designed?

The heterogeneity of methodological procedures for comparing traffic forecasting models is also visible in the literature review. For the comparison

to be useful for the community, methodologically principled comparisons should be performed. Otherwise, the reported results in upcoming literature might be misleading, and disguise the real performance of novel traffic forecasting methods. For instance, some works compare their proposed model to simpler Deep Learning architectures. Instead, other contributions choose a mixture of naïve, statistical, and Deep Learning models, but miss to include any kind of shallow learning method in the comparison. This variability of comparison methodologies make such studies inconclusive. In order to provide verifiable evidence of the performance improvement achieved by the proposed model, several baselines combined with state-of-the-art methods should be analyzed and compared to each other.

Starting with those methods without complexity, a few revised papers include a naïve model as a baseline. These low-complexity straightforward methods have two main representatives: latest value (LV) (also referred to as *persistence*) and historical average (HA) [214]. Since LV uses the most recently recorded traffic value as its prediction, no further calculation is required. On the other hand, HA consists of averaging past traffic data of the same interval of the day and weekday to produce the forecasting value of perform some sort of rolling average over the latter available values. This way, HA requires past sample values for computing the mean for every new prediction. In fact, HA should take into account the patterns that the expert knows in advance (for example, daily and night traffic patterns). Due to their low computational effort, at least one naïve method should be considered in the comparison study, as they establish the lowest performance expected to be surpassed by a more elaborated model. If a novel forecasting method performs slightly better, equal or even worse than naïve methods, the complexity introduced during training would render this method irrelevant to solve such forecasting task. Therefore, these naïve methods allow assessing the balance between the complexity of the proposed model and its achieved performance gap.

Some works revised in the literature analysis compare a novel Deep architecture against different statistical methods (for instance, an ARIMA model). These methods can be set as a performance baseline, but their parameter tuning should be fully guaranteed to ensure that the statistical model is properly fit to the traffic data. According to [28], the comparison between statistical and neural network models is unfair, as complex nonlinear models are compared to linear statistical models, drawing attention to performance metrics. Unfortunately, the study presented in Section 2.2 confirms that this malpractice still can be found in recent research. The aforementioned naïve methods also provide lower bounds for the performance of traffic forecasting models. As opposed to statistical methods, they do not have adjustable parameters, so naïve methods can provide a more reliable baseline for distinct traffic forecasting scenarios. Furthermore, the community could be overlooking other benefits carried by statistical methods, such as their ability to provide insights on the data and its structure.

Simple neural architectures should not be the only ones chosen for

comparing newer Deep Learning proposals (for example, stacked auto-encoders). The recent literature should be revised to elaborate comprehensive comparison studies, not only with basic Deep Learning architectures that presumably will perform worse than the proposed method, but also with the latest novel architectures, especially for spatial-temporal modeling (e.g. graph convolutional networks).

Finally, it should be highlighted that almost none of the revised works provides complexity measures for the models under comparison. Complexity is usually quantified by the number of internal parameters to be fit. Another well-established metric is the raw training time, always determined under identical conditions (i.e. same train data collection, computing resource and software framework). After building a performance benchmark, adding complexity measures should be mandatory for the sake of fairness in comparisons. With each passing year, it becomes more difficult to overcome the performance of previous proposals, narrowing the room for improvement between the latter and the emerging architectures. In this context, these measurements provide an objective tool to judge whether the complexity introduced in the novel traffic forecasting method compensates for the performance gain over the last dominating technique. Only in this way it can be verified whether the proposed model yields an effective and efficient improvement for traffic forecasting.

## 2.4 Challenges and research opportunities

The previous section is intended to bring some order to the current state of the art. Misleading concepts are clarified (e.g. short- vs long-term forecasting), some flaws are highlighted (e.g. the design of comparison studies), and the reasons that motivate the massive use of Deep Learning for traffic prediction are discussed. On the other hand, this section points out challenges that need to be faced, as well as research opportunities that should be explored by the community in years to come.

As new data processing and modeling techniques flourish in the community, emerging research paths arise to yield more precise and wider covering traffic forecasting models. The emblematic complexity of Deep Learning produces opaque models, where traffic managers can not understand the reasoning that leads the model to produce a congestion prediction. An interesting research niche could be model *actionability*, which revolves around making model implementations fast while maintaining prediction performance. A centralized repository of public code and model benchmarks can fight against ambiguous results that vertebrate some present publications. Finally, the utmost expression of traffic forecasting is not a prediction but a *road characterization*. Ideally road segments could be classified according to a collection of spatio-temporal features, such as its location inside the road network or its behavior according to the day of the week, towards offering traffic predictions without the need of collecting traffic measurements.

All the above challenges, as well as the research opportunities that should be explored by the community in years to come are portrayed in

Figure 2.5. It summarizes graphically a future vision of this research area, which is more in depth described hereafter.

### 2.4.1 Need for a centralized traffic data repository

The review of selected works has uncovered an increasing number and diversity of traffic data sources in use during recent years. The issue arises precisely by the number of available options. Even for a specific data source, different datasets can be furnished depending on the location of




Figure 2.5: Schematic overview of identified challenges and suggested research opportunities.

measurement, time intervals or aggregation rate, among other choices. Researchers often apply different preprocessing techniques (usually designed and implemented ad-hoc for the study) to prepare the data for better modeling performance due to more representative examples. For this reason, the ITS community has so far generated multiple versions of many data sources, leading to incongruities in benchmarks comprising state of the art solutions.

All these issues could be overcome if a single point of information was made available for the community: in short, a *centralized traffic data repository*. This repository would store different versions of traffic datasets in an uniform format, according to the different preprocessing techniques applied to the original traffic data sources. The repository would also publish a ranked list of the best performing models for each dataset and forecasting task, for the sake of fair comparisons between novel models. Researchers could reference datasets from third-party research works, and compare their newly proposed technique to previous ones. Interfaces enabling the submission of new data-based pipelines, datasets and results would also be unleashed for extending the coverage of this repository, including the source code producing the results published in the corresponding publication.

Definitely, the availability of a centralized repository would accelerate the understanding of the current status of the field, favoring the development of new and more reliable model comparisons. This idea is illustrated in Chapter 3, where a case study comprising several datasets and data-driven models is presented. This case study is a first step towards an ideal repository that integrates the presented performance benchmark and others scattered over the literature into a single point of information. For the sake of reproducibility, all the traffic data, experimental results and source code employed during this Thesis is accessible by from the QR code at Figure 2.6.

FIGURE 2.6: QR code for accessing https://github.com/Eric-L-Manibardo

### 2.4.2 New modeling techniques for traffic prediction

Another research path garnering a significant interest in recent times aims at the application of alternative data-based modeling approaches to traffic forecasting, mainly towards advancing over the state of the art in terms of design factors beyond the precision of their produced predictions (e.g. computational complexity of the underlying training process). This is the case of recent attempts at incorporating elements from Reservoir Computing

[239] and randomization-based Machine Learning to the traffic prediction realm, including Echo State Networks [240], Extreme Learning Machines [241], or more elaborated variants of these modeling alternatives [242], [243]. The extremely efficient learning procedure of these models makes them particularly appropriate for traffic forecasting over large datasets. On the other hand, the high parametric sensitivity of models currently utilized for traffic forecasting has also motivated the renaissance of bagging and boosting tree ensembles for the purpose, which are known to be more robust against the variability of their hyper-parameters and less prone to overfitting [244]–[246]. Finally, initial evidences of the applicability of automated Machine Learning tools for efficiently finding precise traffic forecasting models have been recently reported in [247].

All in all, there is little doubt that most discoveries and innovations in data-based modeling are nowadays related to Deep Learning. However, beyond the lessons and good practices exposed previously for embracing their use, more attention should be given to other modern modeling choices, such as the Generalized Operational Perceptron [248], Liquid State Machines [249], or models encompassing an hybridization of traffic flow models and Machine Learning techniques [250]. Likewise, other design objectives that do not relate strictly to the accuracy of issued predictions should be increasingly set under target, mostly considering the huge scales attained today by traffic data. A major shift towards efficiency is needed for data-based traffic forecasting models, making use of new evaluation metrics that take into account the amount of data and/or number of operations required for model training.

A randomization-based approach for traffic forecasting is explored in Chapter 3, motivated by its efficient training procedure. While Deep Learning models adjust their internal parameters applying a backpropagation of the loss gradients, randomization-based neural networks keep some parameters of their architecture fixed after a random initialization of their values. Only some weights are adjusted (those concerning the output layer), so that the training time is dramatically reduced. This increased efficiency ultimately enables traffic managers to deploy these models on computationally constrained/inexpensive machines.

### 2.4.3 Model actionability

The literature review has revealed that there is an evergoing race towards finding the best performing traffic forecasting model. However, model actionability should be the ultimate goal for works in the field, which has not exclusively to do with the precision of the forecasts [251].

If a data-driven model is split into sequential stages, a traffic forecasting scenario covers 1) data acquisition; 2) data preprocessing, towards building regression datasets; 3) a learning and validation phase, where a model is learned from such datasets; and 4) model testing, where the performance of the trained model is verified when predicting unseen traffic data. When one of these stages is granted too much relevance, important aspects in other phases of the data pipeline can be neglected. For instance, datasets

are sometimes composed of handpicked locations of the traffic network (i.e. the data source), coincidentally those with more stable patterns that could lead to unrealistically good model performance levels.

Additionally, traffic data might evolve over long time periods, which leads to the fifth and often overseen stage: model adaptation [252]. The idea of model adaptation is conceptually simple: traffic data is continuously fed to the model, which uses the new information to adapt to contextual changes affecting its learned knowledge [253], [254]. To this end, online learning techniques allow the model to be incrementally updated as it is fed with new data, whereas concept drift handling approaches permit to adapt the behavior of the forecasting model to changing data distributions. Although the literature provides specific publications about this topic [255]–[258], it remains as a largely uncharted research area in traffic forecasting.

Lastly, for a model to become fully actionable, confidence metrics should be appended to predictions, so that traffic managers can trust and assess the uncertainty associated to the traffic forecasts, and thus make better informed decisions. From a strategic point of view, confidence estimation in travel demand prediction has a solid research background [259]–[263], which helps design and scale properly road infrastructure. Confidence for long-term congestion predictions have also relevant contributions [229], [264]. However, there are no remarkable contributions on this matter for short-term traffic forecasting.

All in all, forecasting models are the bridge connecting raw data to reliable decisions for traffic management. This need for actionable decisions require far more insights that a single quantitative proof of the average precision achieved by forecasting models. Between the research opportunities that arise around model actionability, the Chapter 4 focus on model adaptation. The presented case study focuses on providing a methodology for the *fast deployment* of traffic forecasting models, where the goal is to deploy a top performance model without the need of collecting traffic data during long periods of time (i.e. a whole year). The obtained results certify that the performance of a model that is not fully adjusted but adapted to the traffic profile on a daily basis is similar to that of a model adjusted over an extensive training holdout. The difference between them is that the first model can be implemented after collecting a month of data, whereas the second needs to wait at least a whole year for an ATR to gather data of different traffic situations.

### 2.4.4 Understanding Deep Learning models

When trained, Deep Learning models are black-boxes that do not grant any chance for the general user to understand how their predictions are made [21], [265]. In the case of traffic operators, the reasons why a neural network produces a particular prediction are of utmost necessity for making informed decisions. In a situation of disagreement, in which the operator of the traffic network does not trust the model prediction, Deep Learning does not offer any means to explain the captured knowledge that

led to its forecasts. Similarly to other fields of knowledge (e.g. medical diagnosis), this lack of transparency of Deep Learning models makes it hard for humans to accept their predictions, who often opt for worse performing yet transparent alternatives (e.g. regression trees).

To the best of our knowledge, very few publications have tackled traffic forecasting from a eXplainable Artificial Intelligence (*XAI*) perspective. One example is [266], which studies the cause-effect relationship between nodes of a traffic network, attempting at learning how upstream and downstream traffic influence the traffic prediction at the target road. A model based on a stacked auto-encoder for missing and corrupt data imputation is presented in [101], where the features extracted by the first hidden layer are analyzed towards improving the interpretability of model decisions. In [89], authors develop an attention-based traffic forecasting model. Then, for a better understanding of the propagation mechanism learned by the model, they examine the evolution of these attention scores with respect to spatial and temporal input data. The last example is [267], where knowledge from two surrounding roads is studied by analyzing the importance of the traffic features (i.e. flow values from different timesteps of the time series from these roads) by using a post-hoc XAI technique.

Most cause-effect relationships in traffic data are studied theoretically [268], [269], without considering the complexity that comes from the use of Deep Learning techniques. Even with correct predictions, a model that is not understandable can be of no practical value for traffic managers willing to obtain insights beyond its predicted output. In recent years, the family of Fuzzy Rule Based Systems (FRBS) model has experienced a renaissance thanks to their envisaged relevance within the XAI paradigm [270]. FRBS learn a set of human-readable *if-then* rules defined on a fuzzy domain that best correlate the predictors and the target variable. These models, along with post-hoc XAI techniques specific to Deep Learning models, will be central for the acceptance of shallow and Deep Learning models in traffic management processes. Specifically, fuzzy rules built for explaining the knowledge captured by black-boxes, and other forms for visualizing local explanations of the produced forecasts will surely contribute to their use in practical deployments, further contributing to the actionability of their issued predictions.

### 2.4.5 Pseudo-real synthetic traffic data

The vast majority of learning methods selected by the ITS community attempt to model the conditional probability $P(y|\boldsymbol{x})$, where the desired output value $y$ (e.g. the traffic forecast) is conditioned by the input $\boldsymbol{x}$ (the predictor variables at the input of the forecasting model). On the other hand, *generative models* estimate $P(\boldsymbol{x}|y)$, as they try to learn the conditional distribution of data [271]. As their name suggests, these models can generate new synthetic data instances, opening an interesting research path towards augmenting the amount of traffic data with which models are trained.

Although researchers have access to traffic simulators like CORSIM [272], VISSIM [273], or SUMO [274], these tools serve a specific purpose: to provide simulated traffic environments with a concrete collection of features. Here, the fictional traffic network is designed and shaped by selecting parameters such as the number of vehicles, speed, road design, etc. Due to this tuning, the environment is conditioned by the investigation requirements and lose its realistic nature. On this line, generative models could provide synthetic data, that resemble real traffic networks. With this, scarce data sources from key locations could be extended, for scenarios where test holdout does not cover all possible traffic states.

In particular, Generative Adversarial Networks (GANs) [275] have demonstrated notable results at learning to synthesize new data instances that highly resemble real data. There are hundreds of publications reported in recent times using GANs for spatio-temporal data [276]. If these generative models provide notable results over an experimental setup, they will acquire a capital importance in traffic forecasting, especially in those scenarios with scarce data. Some recent achievements have already showcased the potential of GANs for this purpose [207], [277], paving the way towards massively incorporating these models for traffic forecasting under data availability constraints.

The last experimental chapter of this Thesis (i.e. Chapter 5) provides a research on the generation of pseudo-real traffic samples. The previously introduced GANs are compared with other generative approaches, as a part of a methodology for the sensorless estimation of traffic profiles. Aiming to characterize sensorless road segments, each target road is paired with a location that has an ATR installed. The criterion for establishing such pairing comes from comparing a set of topological and contextual traits. The real traffic measurements from the selected road serve as input data for the considered generative approaches that must provide pseudo-real traffic data for the target location.

## 2.5 Summary and next steps of the Thesis

Short-term traffic forecasting has been a topic of high interest for the last decades, which explains the major advancements obtained until date. From parametric models to the latest and most complex data-driven methods, nowadays traffic managers have access to models capable of predicting the traffic state on multiple points of a traffic network. However, not all publications in the traffic forecasting topic present remarkable contributions. The capability of Deep Learning to deliver good results has generated a prevalent inertia towards using Deep Learning models, without examining in depth their benefits and downsides. This chapter has focused on critically analyzing the state of the art in what refers to the use of Deep Learning for this particular Intelligent Transportation Systems research area.

Based on two different taxonomic criteria, a review of publications from recent years is performed. A posterior critical analysis is held to formulate questions and trigger a necessary debate about the issues of Deep Learning

for traffic forecasting. New challenges and research opportunities in road traffic forecasting are enumerated and discussed thoroughly, with the intention of inspiring and guiding future research efforts in this field. Following chapters of this Thesis address some of the exposed challenges: Chapter 3 presents a benchmark of the most popular traffic forecasting models employed in the literature over several traffic datasets. The coverage of the performance benchmark is further extended by exploring the capabilities of randomization-based methods, a kind of data-driven method that leverage a random initialization of its inner weights for reducing the training time. Chapter 4 focuses on model adaptation, motivated to accelerate the implementation of forecasting models under data availability restrictions. A limited data holdout demands further adaptation of the model for correctly predicting those traffic behaviors never experienced by the model. The Chapter 5, which addresses generative models, goes beyond any data constraint and aims to characterize a sensorless location without collecting traffic data on it. In summary, following chapters offer a journey through different levels of data availability constraints. For a topic where model complexity does not translate into remarkable improvements, this Thesis is intended to spark interest on data-centered solutions.

# Chapter 3

# Evaluation of Traffic Forecasting Models

The complex and dynamic nature of road traffic patterns is a persistent challenge in the realm of transportation systems. Recognizing and forecasting these patterns has massive implications for operational and strategic management of transportation networks, impacting every aspect from congestion management to infrastructural planning. The key to addressing this challenge lies in exploiting the copious amounts of data generated by the sensing equipment deployed in these networks and devising accurate and reliable forecasting models.

The motivation for this chapter is derived from the recognized necessity for a comprehensive performance benchmark of data-driven models in short-term road traffic forecasting. The proposal involves the generation of such benchmark, including the most popular data-driven models for single road traffic forecasting (i.e. according to the review of the literature conducted in Chapter 2). The original performance benchmark is further extended with a set of randomization-based neural networks, a family of models that might be of special interest for implementations with limited computational resources. This extension aims to broaden the scope of the initial benchmark and account for a greater diversity of approaches in the field. By doing so, it enables a more detailed comparison and understanding of various forecasting strategies, enhancing their applicability and informing future research and implementation decisions. Specifically, the main goals of this chapter can be summarized as follows:

1. Selecting a set of representative datasets for short-term traffic forecasting, aiming to cover all the common setup configurations.

2. Producing a road traffic forecasting performance benchmark for the representative Deep Learning architectures, along with several classical Machine Learning algorithms.

3. Expanding the above benchmark with a set of randomization-based neural networks, a modeling tool unexplored for traffic forecasting.

4. Discussing the stability of randomization-based models, often raised as argument against their use in real-world implementations.

## 3.1 Deep Learning for traffic forecasting

In this section an overview over the most commonly used Deep Learning architectures for traffic forecasting is presented. The goal is to provide the reader a general intuition over their functioning, specific traits and the reasons that motivate its use for short-term traffic forecasting. In detail, Section 3.1.1 introduces the Multi-Layer Perceptron, Section 3.1.2 the convolutional neural networks, Section 3.1.3 the recurrent neural networks, with an emphasis on Long Short-Term Memory cells, and Section 3.1.4 the attention neural networks.

### 3.1.1 Multi-Layer Perceptron

A Single-Layer Feedforward Neural (SLFN) architecture [278] constitutes a fundamental neural network configuration characterized by a single hidden layer interposed between the input and output layers. This architecture is distinguished by its simple yet powerful structure. The input layer receives data, which is then transformed by a set of adjustable parameters in the hidden layer. These parameters are typically represented as weights and biases. The transformed information or *hidden map* is further processed through an activation function and then provided as output. Despite its simplicity, SLFNs have demonstrated remarkable performance [279]. Furthermore, the training of SLFNs is often expedited due to their reduced architectural complexity, making them attractive for real-time applications and scenarios where computational efficiency is paramount.

A Multi-Layer Perceptron (MLP) [280], represents a more complex and versatile extension of the SLFN architecture. Unlike SLFNs, MLPs consist of multiple hidden layers between the input and output layers. These additional layers introduce a hierarchical structure to the network, allowing for more sophisticated feature extraction and representation learning. Each hidden layer in an MLP contains a set of neurons or units, and the network's behavior is determined by the interactions between these layers. This intricate architecture enables MLPs to capture intricate relationships in data, making them especially well-suited for complex tasks like image recognition and natural language processing.

The key advantage of MLPs over SLFNs lies in their ability to model non-linear and abstract features, which are often crucial for addressing real-world problems. However, it is important to note that the increased depth and complexity come at the cost of computational resources and potentially longer training times. As a result, the choice between SLFNs and MLPs depends on the specific problem at hand, with SLFNs favored for simpler tasks and MLPs offering a more comprehensive solution when dealing with intricate and high-dimensional data.

A MLP is composed of three primary components: input neurons, hidden layers, and output neurons (see Figure 3.1). The input neurons correspond to the features in the dataset, which are processed in the hidden layers to extract and learn complex patterns. The hidden layers can be single or multiple, with each layer consisting of a set of neurons. The architecture of a MLP is fully connected, meaning that every neuron in

one layer is connected to every neuron in the next layer. A MLP follows a mathematical formulation where each neuron in the network performs a weighted sum of its input, adds a bias, and then passes it through a non-linear activation function.

For a single neuron in a layer, the output is given by

$$z_j = \sum_{i=1}^{n} w_{ij} x_i + b_j, \tag{3.1}$$

where $x_i$ is the input feature of the $i^{th}$ neuron, $w_{ij}$ is the weight that connects neuron $i^{th}$ to neuron $j^{th}$ and $b_j$ is the bias to the $j^{th}$ neuron of the next layer. Computing the previous the Equation 3.1 gives as result the weighted sum of the inputs $z_j$. This is then passed through the activation function $f$

$$y_j = f(z_j), \tag{3.2}$$

where $y_j$ is the output of the $j^{th}$ neuron.

In the context of a complete layer, if $\mathbf{X}$ is denoted as the input vector, $\mathbf{W}$ as the weight matrix, $\mathbf{b}$ as the bias vector, and $f$ as the activation function (applied element-wise), then the outputs of a layer $\mathbf{Y}$ are given by

$$\mathbf{Y} = f(\mathbf{W}^T \mathbf{X} + \mathbf{B}), \tag{3.3}$$

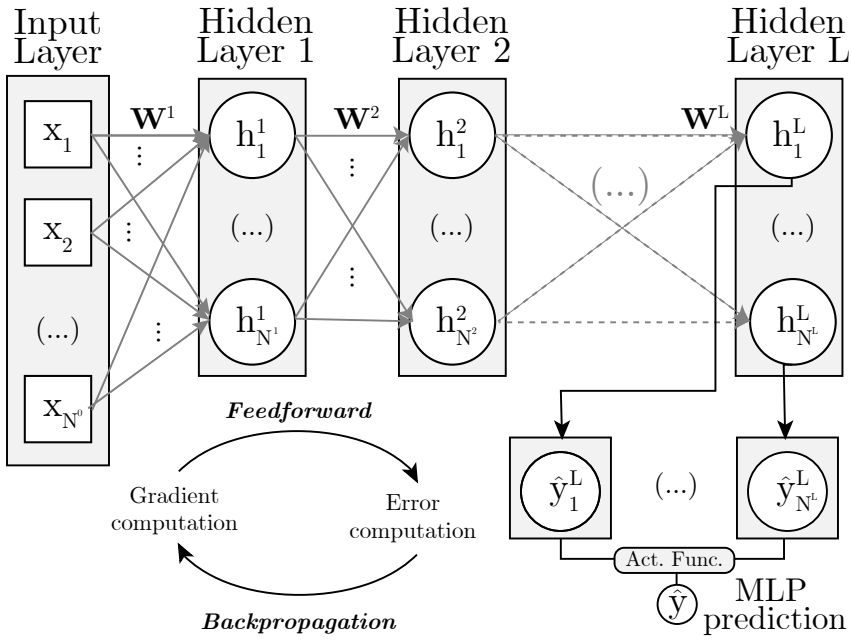



FIGURE 3.1: Architecture of a Multi Layer Perceptron composed by $L$ hidden layers and $N^\ell$ neurons per layer. The input $\mathbf{x}_t$ and a subset of the hidden features $\{\mathbf{h}_t^\ell\}_{\ell=1}^{L}$ are utilized in the output layer to compute the final prediction $\widehat{y}$.

where $T$ represents the transpose operation. The above operation is repeated for each layer in the MLP, with the output of one layer serving as the input to the next. Different layers can use different activation functions $f$. Common choices include the sigmoid function, the hyperbolic tangent, and the Rectified Linear Unit (ReLU) [281].

### 3.1.2 Convolutional neural networks

The MLP is a versatile structure that sets the foundation of various architecture designs. One of the most significant advancements built upon the MLP is the Convolutional Neural Network (CNN) [282]. A CNN extends the concept of hidden layers to include specialized convolutional layers, which have revolutionized tasks like image recognition and computer vision [283].

The mathematical operation known as *convolution* systematically scans through the input data (it was originally designed for images) using learnable filters or kernels. This operation captures local patterns and spatial relationships in the data, making CNNs highly effective at recognizing features within images. The hierarchical structure allows CNNs to automatically learn progressively more abstract features as the data flows through the network. This feature extraction mechanism is particularly advantageous in tasks like image classification. To accommodate one-dimensional data such as time series, signals, or sequences, a specialized architectural variant known as Convolutional 1D (Conv1D) was developed [284]. Conv1D layers, apply convolutional operations specifically tailored to one-dimensional data. As with images, by convolving learnable filters across the input sequences, Conv1D layers can identify patterns, features, and relationships within the data, offering an effective means for tasks like sequence classification, anomaly detection, and more.

The adaptation of convolutional principles to one-dimensional data illustrates the versatility and scalability of CNNs, allowing them to excel in a broader spectrum of applications beyond traditional image processing. Besides the success of standard CNNs on image processing, Conv1D layers have also achieved excellent results in time series prediction and signal identification [284]. Regarding time series prediction, there are published works about predicting how an electrocardiogram evolves [285] or forecasting wind speed and direction [286]. Several Conv1D layers are applied in [287] for solving a traffic prediction task, where authors propose an architecture partly composed by a concatenation of Conv1D layers that learns spatial features. Authors argument that this part of the network helps the model to learn the relationships between traffic of different but closely placed target roads.

A convolutional layer operates by applying multiple filters/kernels to the input data. In a single dimension convolutional layer, each filter is a small vector of weights which is passed over the input data, performing element-wise multiplication with the section of input it currently covers. These multiplied values are summed up to get a single value in the output feature map. This operation is mathematically described as *convolution*.

Given an input vector $\mathbf{x}$ of size $i$, a kernel $\mathbf{k}$ of size $m$, the convolution operation for a single element in the output feature map can be represented as

$$(\mathbf{x} * \mathbf{k})(i) = \sum_{m} \mathbf{x}(i-m) \cdot \mathbf{k}(m), \tag{3.4}$$

where $*$ denotes the convolution operation, $\mathbf{x}(i-m)$ and $\mathbf{k}(m)$ represent the elements at position $(m)$ in the input matrix and kernel respectively. The output feature map for a single filter is obtained by sliding this operation across the height and width of the input data. Convolutional layers apply several filters with different kernels, so distinct meaningful features can be computed.

### 3.1.3 Recurrent neural networks

While CNNs are designed to capture spatial features, a Recurrent Neural Network (RNN) is engineered to learn temporal features. In essence, RNNs are tailored for processing sequential data, making them appropriate for tasks where understanding the order and timing of events is vital. Unlike feedforward neural networks, RNNs incorporate loops within their architecture (hence the *recurrent* nomenclature), enabling them to maintain a form of memory about previous information encountered in a sequence. This recurrent nature grants them the ability to tackle diverse applications such as natural language processing, speech recognition, and time series forecasting. Just as CNNs are crucial for modeling spatial correlations, RNNs serve as a fundamental tool for temporal feature extraction, highlighting the complementarity of various neural network architectures in addressing multifaceted data analysis challenges.

The Long Short-Term Memory (LSTM) layer, a type of recurrent layer employed as component in RNNs, has garnered widespread adoption due to its remarkable ability to address the vanishing gradient problem, which often hampers traditional RNNs [288]. This advanced architecture introduces the concept of memory cells, which can retain information over extended sequences, thereby enabling the model to capture and store long-range dependencies in data. The unique architecture of LSTMs [289], with its three gating mechanisms (input, forget, and output gates), empowers the network to effectively manage and manipulate information flow, enhancing its capacity to learn intricate temporal patterns and relationships. This capability makes LSTMs a favored choice in various applications where preserving and understanding complex temporal dynamics is crucial [290].

The work of Han et al. [287] has been previosly commented regarding learning spatial dependencies. The presented architecture also implements a branch composed by LSTM layers, whose goal is to learn the time dependencies within the traffic signal. In the literature review conducted in Chapter 2, it is found that scholars follow a similar pattern: if there are several ATRs close to the target road, they use CNNs to learn the spatial relationships. This means learning how traffic readings from different locations are related for the same timestep. On the other hand, RNNs and in particular LSTMs are designed to learn how traffic readings from

a given location are related over time. Depending on the implementation details, one option may be more advantageous than the other or, as has been shown in [287], the two can be combined to get the most out from the input data.

In an LSTM cell (i.e. the basic unit of a LSTM layer), the information flow is regulated by three different types of gates: the forget gate, the input gate, and the output gate. Each gate involves a sigmoid activation function, which helps to control the amount of information passing through.

The forget gate is a sigmoid layer that decides which information to discard from the cell state. The equation for the forget gate is

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f), \tag{3.5}$$

where $\mathbf{W}_f$ is the weight matrix of size $N^\ell \times 2N^\ell$ that transform the concatenation of $[\mathbf{h}_{t-1}, \mathbf{x}_t]$ (operation noted as $[\cdot, \cdot]$), representing the hidden state of the LSTM from the previous timestep (size $N^\ell$) and the input of current timestep respectively (again of size $N^\ell$). The bias term $\mathbf{b}_f$ is added to the result before applying the sigmoid function $\sigma$, completing the computation of $\mathbf{f}_t$, which is the forget gate at timestep $t$, indicating how much of the previous cell state $C_{t-1}$ should be retained or forgotten.

The input gate is a combination of a sigmoid layer and a hyperbolic tangent layer (i.e. tanh layer). The sigmoid layer decides which values to update, while the tanh layer generates new candidate values that could be added to the state. The equations for the input gate are

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{3.6}$$

and

$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C), \tag{3.7}$$

where $\mathbf{i}_t$ and $\tilde{\mathbf{C}}_t$ correspond to the input gate and candidate cell state at timestep $t$, respectively. The updated cell state is computed by combining $\tilde{\mathbf{C}}_t$ with $\mathbf{C}_{t-1}$.

The output gate is a sigmoid layer which decides what the next hidden state should be. The equations for the output gate are

$$\mathbf{u}_t = \sigma(\mathbf{W}_u \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_u) \tag{3.8}$$

and

$$\mathbf{h}_t = \mathbf{u}_t \times \tanh(\mathbf{C}_t), \tag{3.9}$$

where $\mathbf{u}_t$ represents the output gate at timestep $t$.

Following the above equations, the LSTM cell iteratively updates and retains the useful information throughout the input sequence.

### 3.1.4 Attention neural networks

The so-called *attention mechanism* is the main component of Attention Neural Networks, which have collected excellent results in the fields of

natural language processing and computer vision [291]. Inspired by human visual attention and other cognitive mechanisms, it enables models to focus on specific parts of the input data when making predictions. Recurrent models typically process data along the symbol positions of the input and output sequences. The issue is that for long sequences, an standard RNN is not able to relate target information located at the beginning and end of the sequence. Unlike LSTM (which process the entire input sequence uniformly), the attention mechanism enables models to assign different levels of importance to distinct elements within the input. This dynamic and adaptive approach enhances the model's ability to capture intricate relationships and dependencies in the data.

In the context of traffic forecasting, the attention mechanism is usually combined with graph representations [86], [180]. Conceptually, the idea is to build a network that can learn which nodes of the graph (representing a traffic network) contribute and influence the traffic of a target road segment. However, it can also be implemented outside the graph-representation scope, for regular traffic modeling, where the input is represented by a timeseries. The work of Liu et al. [193] introduces a model that gathers the previously explained technologies into a single architecture. The input is conformed by traffic data collected at several closely placed road segments. A feature map is computed, using a CNN for analyzing the spatial relationships, and a RNN for the temporal components. This feature map is then processed by an attention model, producing a set of weights that gives importance to each value. In theory, the attention layer improves the performance of the output layers, which are in charge of producing the final prediction.

Attention weights are computed to determine how much emphasis each input element should receive. These weights are then used to create a context vector, capturing target information from the input sequence, which is incorporated into the output generation process. By dynamically adapting to the input sequence for each step of output, attention mechanisms are able to intelligently consider context. There are multiple types of attention mechanisms, including dot product attention [292], transformer attention [292] and additive attention [293]. The *additive attention* is explained below, as it has been selected due to its simplicity and straightforward computation.

Every attention mechanism is based on three components: query, key and value. The *query* represents the data to be predicted. For a traffic forecasting model, the query is the future traffic flow at the target road. The *key* is the associated historical data, such as the traffic flow values from previous instants. Finally, the *values* are the factors that influence the output, which in the context of traffic can be the time of the day, day of the week and so on. An attention mechanism allows the model to weigh the importance of different historical factors (keys) when forecasting the traffic flow (query), based on historical traffic flow values (values).

The additive attention mechanism causes the model to focus on different parts of the input sequence when producing an output sequence. This is achieved through a set of learnable parameters and a weighted

sum. Unlike other attention mechanisms that rely on dot products or multiplicative interactions, additive attention employs a learned function to compute attention weights.

The first step is to calculate the alignment score $s$, which is a real-valued variable $s \in \mathbb{R}$ for each query vector $\mathbf{q}$ and key vector $\mathbf{k}$ of size $N^q$ and $N^k$ respectively. The alignment scores are stored in a matrix $\mathbf{S}(q,k)$ of size $N^q \times N^k$, and its purpose is to express the similarity between the pair query-key. The scores are calculated by applying a trainable function to both the query and key, which is represented by the weight matrices $\mathbf{W}_q$ and $\mathbf{W}_k$ of size $N^\ell \times N^q$ and $N^\ell \times N^k$ respectively, being $N^\ell$ a tuning parameter of the layer. The obtained values are passed through a hyperbolic tangent activation function before being stored as $\mathbf{S}(q,k)$.

$$\mathbf{S}(q,k) = \tanh(\mathbf{W}_q \cdot \mathbf{q} + \mathbf{W}_k \cdot \mathbf{k}). \tag{3.10}$$

A softmax function is then applied to the resulting scores to obtain attention weights $\alpha$ that indicate the importance of each input element. This step normalizes the scores, ensuring that they sum up to 1.

$$\boldsymbol{\alpha}_{q,k} = \frac{e^{\mathbf{S}(q,k)}}{\sum_{q,k} e^{\mathbf{S}(q,k)}}. \tag{3.11}$$

The attention weights are used to compute a weighted sum of the input elements $\mathbf{h}$ received by the attention layer, which is then combined with $\mathbf{q}$ to generate the final output or *context vector*. This context vector $\mathbf{c}$ contains a condensed representation of the layer's input, emphasizing the most relevant information for the given task:

$$\mathbf{c}_{q,k} = \sum_{q,k} \boldsymbol{\alpha}_{q,k} \cdot \mathbf{h}. \tag{3.12}$$

The additive attention mechanism is usually placed after computing a feature map. This feature map can be the output from a CNN and/or a RNN. The attention layer will then give different degrees of importance for each value on the feature map so the output layers can improve on their predictions.

## 3.2 Randomization-based neural networks

Randomization-based neural networks are a class of artificial neural networks that leverage the principles of randomization in their design and operation. These networks have gained significant attention within the realm of Machine Learning due to their unique approach to information processing [294], [295]. In a randomization-based neural network, randomness is introduced at various stages of network architecture and/or training, departing from the deterministic nature of standard neural networks. This approach offers several advantages, including increased robustness, the potential for better generalization, and the ability to handle complex, non-linear relationships in data.

One of the fundamental distinctions of randomization-based neural networks lies in the initialization of network parameters. Instead of adopting deterministic initialization methods, such as the Glorot innitialization [296], randomization-based networks initialize weights and biases with random values drawn from specific distributions (e.g. Gaussian distribution). By introducing this stochastic element, these networks aim to break potential symmetries in the network, thus facilitating the discovery of diverse features and representations during training.

The SLFN can be argued as the baseline structure for the randomization-based models proposed in the literature [297]. Modifications range in a variaty of modifications, such as the presence of direct links between input and output [298], to applying bias for neurons [297]. The following architectures introduce several changes on the SLFN: Extreme Learning Machine (Section 3.2.1) and Random Vector Functional Link (Section 3.2.2). Further modifications around the original architecture of Random Vector Functional Link are discussed at Section 3.2.3.

### 3.2.1 Extreme Learning Machine

An Extreme Learning Machine (ELM) is a class of SLFN that originated in 2004 with the work of Huang et al. [299]. ELM deviates from conventional gradient-based training methodologies, offering an expedited and simplified learning process that holds the particular relevance of offering fast training procedures over constrained computational resources.

The learning speed of a SLFN is conditioned by: 1) the gradient-based learning algorithm, which finds a local minima in a iterative process; 2) the high number of parameters to be tuned using such algorithm. An ELM model tackles these issues by randomly setting most of the internal parameters and analytically determining the weights of the output layer. In detail, the weights that connect the input with the hidden layer are randomly assigned, usually drawn from uniform or Gaussian distributions. This characteristic substantiates the *extreme* nomenclature, as these random weights remain unaltered during the training process, setting ELM apart from its gradient-based counterparts. The learning procedure occurs only in the output layer, by applying a simple linear regression. This way optimal output weights can be computed using a closed-form equation, depart from the ELM's predictions and the actual target outputs.

The recent review performed by Wang et al. [300] compiles successful applications of ELM, usually those that have a real-time learning component. To mention a few, video analysis [301], chemistry [302], food safety monitoring [303] and cyber-attack detection [304]. Within the transportation research topic, a variety of research works have been conducted on real-time driver distraction detection [305], road surface temperature prediction [306] or predicting delays in railway networks [307]. Only a single work of those listed in [300] applies ELM for traffic forecasting [308]. However, the main goal for this research was not to provide the best forecasting method but to prove the feasibility of an ELM variant regarding other low

performance methods. Further along this line, the traffic data that compose the case study spans from April 2015 to May 2015, making the results biased and beneath any utility in real-world scenarios.

### 3.2.2 Random Vector Functional Link

The idea of randomizing some weights to reduce the complexity of the learning process is not new. The Random Vector Functional Link (RVFL) network was proposed ten years before ELM was developed [298]. The main difference between them is that the RVFL has direct input-output links, which for certain applications can improve performance [309].

In a ELM model, the output layer (which is in charge of computing the prediction) only has access to the data interpretation the hidden layer provides. The hidden map needs to supply a useful interpretation regarding the input values. However, for those problems where the raw input encapsulates knowledge that is interpretable without any further processing, a direct link between input and output should be beneficial. The rationale for this premise is that the hidden representation needs to provide the raw input values as part of the prior introduced hidden map for an ELM. Otherwise, the original knowledge cannot be interpreted by the output layer. On top of this, the direct link to the input value does not need to be trained, so the number of parameters to be adjusted between equivalent architectures (i.e. same number of parameters) remains constant.

A novel literature review concerning RVFL variants and its application has been conducted by Malik et al. [310]. This architecture and its variants are specially suited to be applied in time series forecasting problems where the raw data is already a good descriptor. Hence, [310] collects published works where RVFL has been successfully applied, such as electricity load prediction [311] , solar [312] and wind power [313] forecasting and crude oil price analysis [314]. Although there has been no attempt at applying RVFL to short-term traffic forecasting, the insights distilled from [310] indicates that RVFL should perform on top of an ELM architecture, just because the presence of direct links between input and output layers. Consequently, the architecture and functioning of RVFL and several of its variants are explained hereunder.

Inheriting the structure of a MLP, the general architecture of RVFL is composed of the standard building blocks of a neural network: the neural unit [315]. Thus, the main difference between MLP and RVFL variants lies in the learning algorithm adjusting their internal parameters. The first optimizes all internal parameters (i.e. output/hidden weights and biases) via gradient backpropagation, whereas the latter only adjusts the parameters of the output layer. These output weights $\mathbf{W}_o$ map a vectorized version of the information processed on the input through the first part of the model to the target variable to be predicted. Mathematically, assuming row vector notation and that the output is a single real-valued variable $y \in \mathbb{R}$ (regression):

$$\widehat{y}_t = \mathbf{s}_t \cdot \mathbf{W}_o^\top = [\mathbf{x}_t, \mathbf{h}_t] \cdot \mathbf{W}_o^\top, \tag{3.13}$$

where $t$ is an index that denotes the number of the data instance (in the context of time series forecasting, the time at which a prediction is issued), $[\cdot,\cdot]$ stands for vector concatenation, $\widehat{y}_t$ is the predicted value for the target, and $\mathbf{s}_t$ is a vector concatenating the input $\mathbf{x}_t$ and the hidden features $\mathbf{h}_t$. In the general form of a multilayered RVFL model (see Figure 3.2), the hidden features $\mathbf{h}_t^\ell$ computed for $\mathbf{x}_t$ at layer $\ell \in \{1,\ldots,L\}$ are given by:

$$\mathbf{h}_t^\ell = \mathbf{W}^\ell \cdot \left(\mathbf{h}_t^{\ell-1}\right)^\top, \tag{3.14}$$

where $\mathbf{h}_t^0 = \mathbf{x}_t$, and $\mathbf{W}^\ell$ is an intermediate $N^\ell \times N^{\ell-1}$ weight matrix, with $N^\ell$ denoting the number of neurons of each layer. Clearly, $N^0 = |\mathbf{x}_t|$, i.e., the number of input predictors upon which the forecast is made.

After a random initialization of $\mathbf{W}^\ell$ $\forall \ell \in \{1,\ldots,L\}$, their values are kept fixed, which set the stage for a fast and computationally affordable single-step optimization process to compute the values of $\mathbf{w}_o$ [309]. The $L_2$ norm regularized least square (or ridge regression) provide the $\mathbf{w}_o$ values by solving the following optimization problem:

$$\min_{\mathbf{w}_o} \sum_{t \in \mathcal{T}_{train}} ||y_t - \mathbf{s}_t \cdot \mathbf{w}_o^\top||_2^2 + \lambda ||\mathbf{w}_o||_2^2 \tag{3.15}$$

where $\lambda$ is the regularization parameter to be tuned, and $\mathcal{T}_{train}$ denotes the number of training examples. Despite not used in this chapter, RVFL can also be used for classification by replacing the ridge regression by a matrix pseudoinverse.

### 3.2.3 RVFL variants: deep and ensemble deep RVFL

The original RVFL algorithm (denoted as *shallow RVFL* in this study) is composed of a single hidden layer, i.e. $L = 1$. By stacking more hidden layers, a *RVFL network* can be obtained, in which the output layer receives a vector $\mathbf{s}_t$ composed by $\mathbf{x}_t$ (the input itself) and the features produced by the last layer of the hierarchy (e.g. $\mathbf{h}_t^L$). The purpose of the hidden layers is to generate a high-level interpretation of the original input values that supports these input features for computing the final output prediction. This feature representation is given by the random initialization of the architecture parameters.

Among several other variants proposed for classification tasks [316], [317], two new versions of the shallow RVFL were proposed in [318]: the Deep RVFL (dRVFL) and the Ensemble Deep RVFL (edRVFL) (see Figure 3.2). The dRVFL algorithm is an extension of a RVFL network where the data interpretation of all hidden layers is handled by the output layer, i.e.:

$$\mathbf{s}_t = [\mathbf{x}_t, \mathbf{h}_t^1, \ldots, \mathbf{h}_t^L], \tag{3.16}$$

i.e. not only the final feature representation is considered for computing the output, but all the intermediate $L$ hidden feature representations instead.

In the second RVFL variant (i.e. edRVFL), intermediate predictions issued from each feature vector produced by every layer are aggregated together to produce the predicted output $\widehat{y}_t$. Mathematically, the notation can be extended as in Expression (3.13) to define each of the intermediate predictions as:

$$\widehat{y}_t^{\ell} = \mathbf{w}_o^{\ell} \cdot \left(\mathbf{s}_t^{\ell}\right)^{\top} = \mathbf{w}_o^{\ell} \cdot [\mathbf{x}_t, \mathbf{h}_t^{\ell}]^{\top}, \tag{3.17}$$

which means that the intermediate prediction is modeled based on the feature representation provided by the $\ell$-th hidden layer. A separate vector of output weights $\mathbf{w}_o^{\ell}$ is then computed for every layer as per Expression (3.15), so that $L$ predictions $\{\widehat{y}_t^{\ell}\}_{\ell=1}^{L}$ are obtained for input $\mathbf{x}_t$. Finally, all such predictions are averaged to yield the finally predicted value:

$$\widehat{y}_t = \frac{1}{L} \sum_{\ell=1}^{L} \widehat{y}_t^{\ell}, \tag{3.18}$$

yet any other strategy for fusing intermediate predictions can be used (e.g. median value or a stacking ensemble).

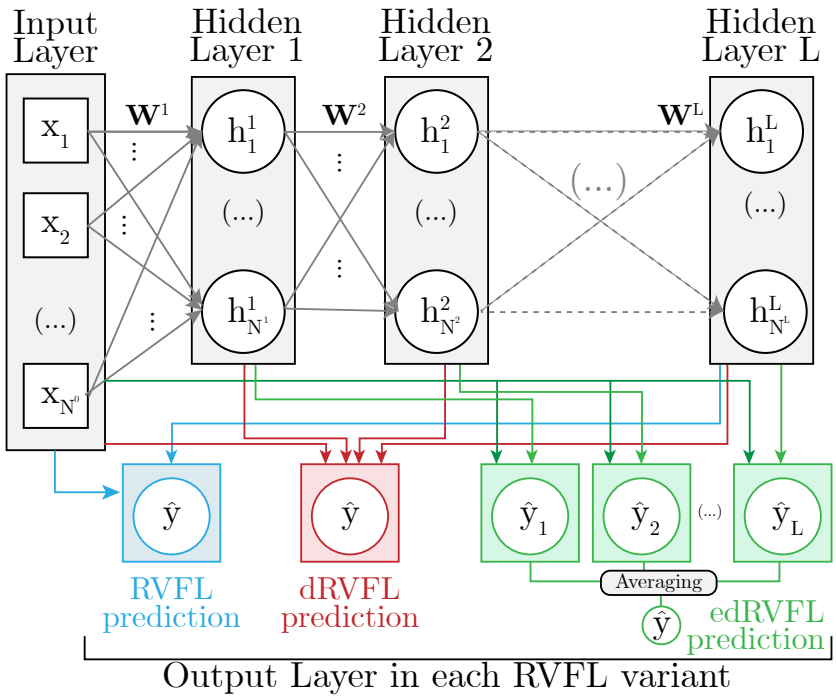



FIGURE 3.2: Architecture of the different RVFL variants covered in this chapter, departing from a generic multi-layer neural architecture with $L$ hidden layers and $N^{\ell}$ neurons per layer. The input $\mathbf{x}_t$ and a subset of the hidden features $\{\mathbf{h}_t^{\ell}\}_{\ell=1}^{L}$ are utilized in the output layer to compute the final prediction. Only the weights of the connections $\mathbf{W}^{\ell}$ are randomly initialized. Depending on the values used for the prediction, different RVFL variants can be defined.

## 3.3 Description of the case study

From the state-of-the-art analysis conducted in Chapter 2, it can be concluded that the application of Deep Learning methods to short-term traffic forecasting has been, to a point, questionable. In some cases, authors do not justify the high computational complexity inherent to their proposed method, nor do they compare it to less complex modeling alternatives. In turn, the configuration of the comparison studies and the lack of depth in the discussion and analysis of the obtained results do not often clarify whether newly proposed methods outperform the state of the art at the time of their publication. Motivated by this, a performance benchmark is conducted over a set of the most commonly employed data-driven models, intending to serve as a baseline for future contributions about modeling approaches for traffic forecasting.

One of the research opportunities highlighted in Chapter 2 revolves around new modeling techniques. Randomization-based neural networks offers fast-training times thanks to the inner weights being adjusted via ridge regression (see Section 3.2) instead of the standard backpropagation technique. The road traffic prediction capabilities of several RVFL variants and other randomization-based methods is evaluated and appended to the baseline performance benchmark. Additionally, the instability associated to the random nature of these models needs to be analyzed, since its the main aspect researchers use as argument for not implementing them in real-life scenarios.

In short, the above challenges can be formulated as the following Research Questions (RQ):

- RQ3.1: How do the distinct Deep Learning and randomization-based learning methods perform compared to other data-driven methods?
- RQ3.2: Should the instability of randomization-based models be the reason for discarding them in favor of other data-driven methods?

## 3.4 Materials and methods

The materials and methods employed in the case study are introduced hereafter. Section 3.4.1 describes the data employed for model assessment. Section 3.4.2 introduces the collection of data-driven methods selected for evaluation. Finally, the experiment design is presented in Section 3.4.3.

### 3.4.1 Data for evaluating model performance

Traffic forecasting setups encompass several variables including the type of traffic measurements the area under scope, the sensing technique, and the way data are aggregated. Aiming to emulate all possible scenarios is not feasible due to the vast number of potential setup combinations. Therefore, a representative subset has been selected, primarily focusing on traffic flow and speed forecasting, which has been most frequently addressed in the literature (see Figure 2.3).

The nature of collected traffic measurements largely defines the relationships between the traffic time series and congestion states. Although there are other traffic measurements such as travel time or occupancy, the most commonly used data sources contain *flow* and/or *speed* measurements. While both time series are related by the fundamental diagram of traffic flow [319], predicting speed is in general an easier task since, for most of the time, traffic circulates at the speed limit of the road (*free-flow*). It is, therefore, a more stable – hence, predictable – signal over time. Disruptions in the speed time series come in the form of valleys. Traffic congestion results in speed drops, directly related to the spikes of the flow time series. However, traffic flow has a wider dynamic value range, and in general undergoes multiple variations throughout the day. Events, weather, calendar, and other factors modify the traffic flow profile by narrowing or expanding flow spikes in time, or even removing them.

Analogously, traffic behavior also varies between highways and city roads. Freeways and other high nominal speed inter-urban roads provide stable patterns that barely changes between close locations. Since they act as the link between major cities in a regional transportation network, the traffic behavior is scarcely influenced by contextual factors, as opposed to the fluctuations that might appear in city road traffic. Urban trips are exposed to a manifold of factors such as roundabouts, pedestrian crossings or traffic lights. Drivers also introduce different behaviors in cities [320]. These aspects make data noisier and hence harder to predict. In contrast, highway traffic is not affected by such factors, so forecasting freeway traffic is in general much easier.

Table 3.1: Selected traffic data sources.

| Location | Target variable | Scope | Sensor | Time resolution | Year |
|---|---|---|---|---|---|
| Madrid [321] | Flow | Urban | RCD | 15 min | 2018 |
| California [223] | Flow | Freeway | RCD | 5 min | 2017 |
| New York [322] | Speed | Urban | RCD | 5 min | 2016 |
| Seattle [323] | Speed | Freeway | RCD | 5 min | 2015 |

Based on the above reasons, at least four datasets should be needed to cover all possible combinations of flow and speed forecasting over urban and highway areas. Table 3.1 summarizes the attributes of each selected data source according to the taxonomy defined in Chapter 2. All data sources gather traffic information by using roadside sensors. To the best of the author knowledge, no public FCD data source covers one complete year of data, which is a requirement to gauge the perform of a forecasting model throughout all seasons of the year. The temporal resolution is kept to the original value provided by the data repository. Every data source contain a set of sensed roads. Those that exhibit missing data are excluded. From the remaining, ten points of the traffic network are selected for building the traffic datasets employed for model assessment, always intending to provide the broadest spectrum of characteristics within the data source (e.g. number of lanes or speed limit).

### 3.4.2 Considered data-driven methods

A case study is conducted, which serves as an informed assessment of the effects of all the particularities of the Deep Learning methods previously described. To this end, the effectiveness of these techniques when predicting short-term traffic measurements is verified and compared to modeling techniques with less computational complexity. The forecasting methods that will compose the benchmark are selected from the most commonly used algorithms and architectures in the state of the art. Statistical methods are not included in this case study, since the naïve `LV` method already provides a performance baseline (i.e. prediction is set equal to the latest traffic value captured by a sensor). Inspired by revised works, a categorized list of learning methods is presented in Figure 3.3). Besides the RVFL variants presented in Section 3.2, ELM is also analyzed, aiming to spot performance differences due to the direct links between the input and ouput layer. In order to ensure a fair comparison of the performance of ELM to that of the corresponding RVFL counterparts, the benchmark includes a shallow ELM network with only one hidden layer, as well as an ELM network with multiple hidden layers.

- **Learning methods**
  - *Naïve*
    - `[LV]` Latest Value
  - *Shallow*
    - `[LR]` Least-squares Linear Regressor
    - `[KNN]` k Nearest Neighbors
    - `[DTR]` Decission Tree Regressor
    - `[SVR]` $\varepsilon$-Support Vector Machine
  - *Ensemble*
    - `[ADA]` Adaboost
    - `[RFR]` Random Forest Regressor
    - `[ETR]` Extremely Randomized Trees
    - `[GBR]` Gradient Boosting Regressor
    - `[XGBR]` Extreme Gradient Boosting Regressor
  - *Deep Learning*
    - `[MLP]` Multi Layer Perceptron
    - `[CNN]` Convolutional Neural Network
    - `[LSTM]` LSTM based Neural Network
    - `[CLSTM]` Convolutional-LSTM Neural Network
    - `[ATT]` Attention based Neural Network
  - *Random -based*
    - `[sRVFL]` shallow RVFL
    - `[RVFL]` RVFL network
    - `[dRVFL]` deep RVFL network
    - `[edRVFL]` ensemble deep RVFL network
    - `[sELM]` shallow ELM
    - `[ELM]` ELM network

FIGURE 3.3: Considered data-driven methods, grouped by its modeling nature.

### 3.4.3 Experiment design

The forecasting problem is formulated as a regression task, where the previous measurements of each target road collected at times $\{t-4,\ldots,t\}$ are used as features to predict the traffic measurement at the same location and time $t+h$. Four prediction horizons $h \in \{1,2,3,4\}$ are considered, so that a separate single-step prediction model is trained for each $h$ value and target location. Figure 3.4 describes the proposed experiment design. For each traffic data source, 10 points of the road network are selected, always choosing locations that offer diverse traffic profiles. Then, a regression dataset for each target placement is built, covering data of one year. The first three weeks of every month are used for model training, whereas the remaining days are kept for testing. This split criterion can be used to test the ability of the models to learn traffic profiles that vary between seasons and vacation days.

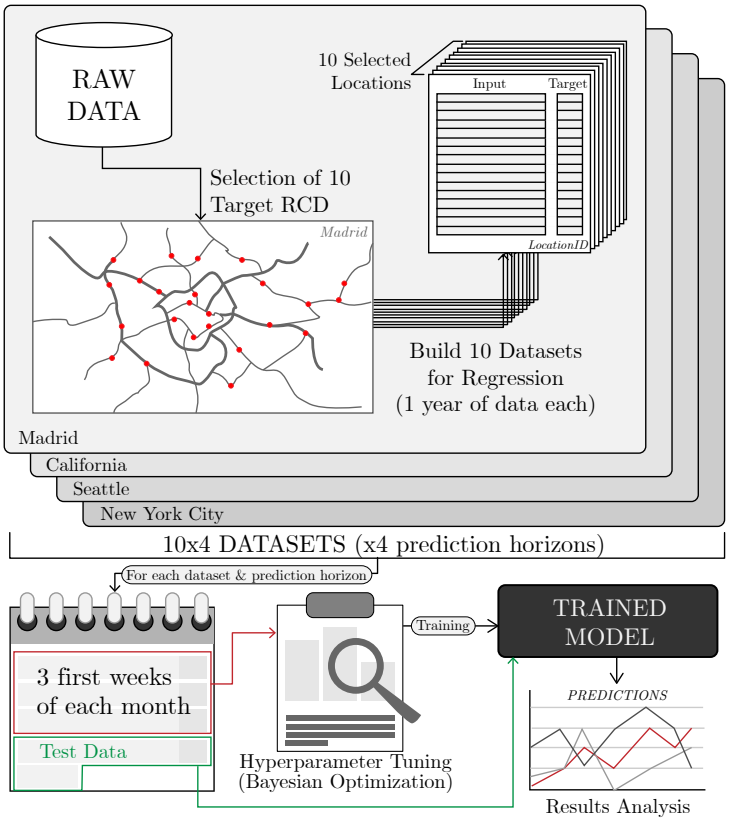



FIGURE 3.4: Experiment design used in this case study. After building regression datasets for each target location, training and testing data is reserved for every month along the year. Cross-validation provides measures to select the best hyper-parameter configuration for every model in the benchmark via Bayesian optimization. Finally, the optimized model learns from all available training data, and predictions are generated for all testing data.

In order to find the best hyper-parameter values for each regression model, three-fold cross-validation is performed: two weeks of every month are used for training, and the remaining ones of the reserved training data are used for validation. The average of the three validation scores (one per every partition) is used as the objective function of a Bayesian optimizer [324], which searches for the best hyper-parameter configuration efficiently based on the aforementioned objective function. After evaluating 30 possible configurations for each model, the best hyper-parameter configuration is set on the model at hand, which is trained over all training data. Once trained, model performance scores are computed over the data held for testing. This process reduces the chances to have a bias in the comparisons later discussed due to a bad hyper-parameter configuration of the models.

The purpose of the case study is to identify the model that best predicts the traffic signal for each of the prediction horizons. Some popular statistical metrics used to measure the performance of traffic forecasting models are Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) [11]. RMSE is a quadratic scoring rule that measures the average magnitude of the prediction error. Essentially, it describes how concentrated the data is around the line of best fit. By squaring the errors before they are averaged, RMSE gives a relatively high weight to large errors. This means RMSE is most useful when large errors are particularly undesirable. MAE, on the other hand, calculates the average of the absolute difference between the predicted and actual values. It is a linear scoring rule which means all individual differences are weighted equally in the average. MAE is less sensitive to outliers compared to RMSE, making it a more robust metric against the presence of outliers. However, these metrics are not best suited for a benchmark that gathers traffic variables of different scales. For instance, traffic flow might be in hundreds of vehicles, while speed might be in tens of kilometers per hour. RMSE and MAE are scale-dependent and could be larger for metrics with a larger scale, making it unfair when comparing the performance of models predicting different traffic metrics.

The equations for RMSE and MAE are:

$$\text{RMSE} \doteq \sqrt{\frac{1}{\mathcal{T}_{test}} \sum_{t \in \mathcal{T}_{test}} (o_t - \widehat{o}_t)^2} \tag{3.19}$$

and

$$\text{MAE} \doteq \frac{1}{\mathcal{T}_{test}} \sum_{t \in \mathcal{T}_{test}} |\, (o_t - \widehat{o}_t) \,| \tag{3.20}$$

where $\mathcal{T}_{test}$ denotes the set of time slots belonging to the test partition of the dataset at hand, $o_t$ denotes the real observed value at test time $t$ and $\widehat{o}_t$ the predicted one.

A third and often overlooked error metric is the coefficient of determination $R^2$ [325], which provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation

of outcomes explained by the model. In the context of short-term traffic forecasting, the use of $R^2$ as a performance metric has particular benefits. First, it is unit independent, allowing model performance to be compared across datasets of different nature. Secondly, as traffic data often exhibits strong temporal dependencies and potential non-linear relationships, the proportion of variance explained by a model provides insight into how much of the total information has been capture bythe model. A high $R^2$ means that the model is able to explain a large portion of the variance in traffic patterns, suggesting that the model has successfully captured underlying temporal dependencies and non-linear relationships in the data.

Bearing the above in mind, the $R^2$ score is computed over the testing data to measure the quality of predictions between real and predicted traffic measurements. This score is given by:

$$R^2 \doteq 1 - \frac{\sum_{t \in \mathcal{T}_{test}} (o_t - \widehat{o}_t)^2}{\sum_{t \in \mathcal{T}_{test}} (o_t - \bar{o}_t)^2}, \tag{3.21}$$

where $\bar{o}_t$ depicts the average of the observed values.

## 3.5 Experiments and results

The proposed Research Questions are addressed below. Specifically, Section 3.5.1 presents the obtained performance results for the considered data-driven methods, whereas Section 3.5.2 explores the instability of randomization-based neural networks.

### 3.5.1 RQ3.1: Baseline performance benchmark

The discussion begins with Figure 3.5, which displays the overall performance, computed as the mean $R^2$ score averaged over the 10 datasets of each data source, for every learning method and analyzed forecasting horizon $h$. As expected, the performance of the models degrades consistently as the prediction horizon increases. Traffic data corresponding to the California data source are stable, which can be appreciated by a simple visual inspection of their profiles: a high $R^2$ score is obtained for these datasets even when predicting four steps ahead ($h = 4$). As stated in Section 2.2.2, the PeMS data source is the most popular option for traffic congestion studies, especially when novel forecasting methods are presented. In this study, only datasets from District 4 have been collected (the so-called *Bay Area*), as data from other districts also provide stable traffic measurements, and District 4 is the most commonly selected sector among the revised literature.

The nature of traffic measurements, jointly with the scope area of data sources, can suggest in advance how forecasting performance degrades when the prediction horizon $h$ is increased. Both in the city and in highways, drivers tend to maintain a nominal speed whenever possible, so time series drops suddenly. Thereby, only the last timestamps provide information on this phenomena [182]. Results for New York and Seattle data

sources corroborate this statement, where the performance degradation maintains a similarly decaying trend. In the case of flow data, traffic at urban roads can differ significantly depending on the selected location. Main roads maintain a nearly constant traffic flow as trucks, taxis, and

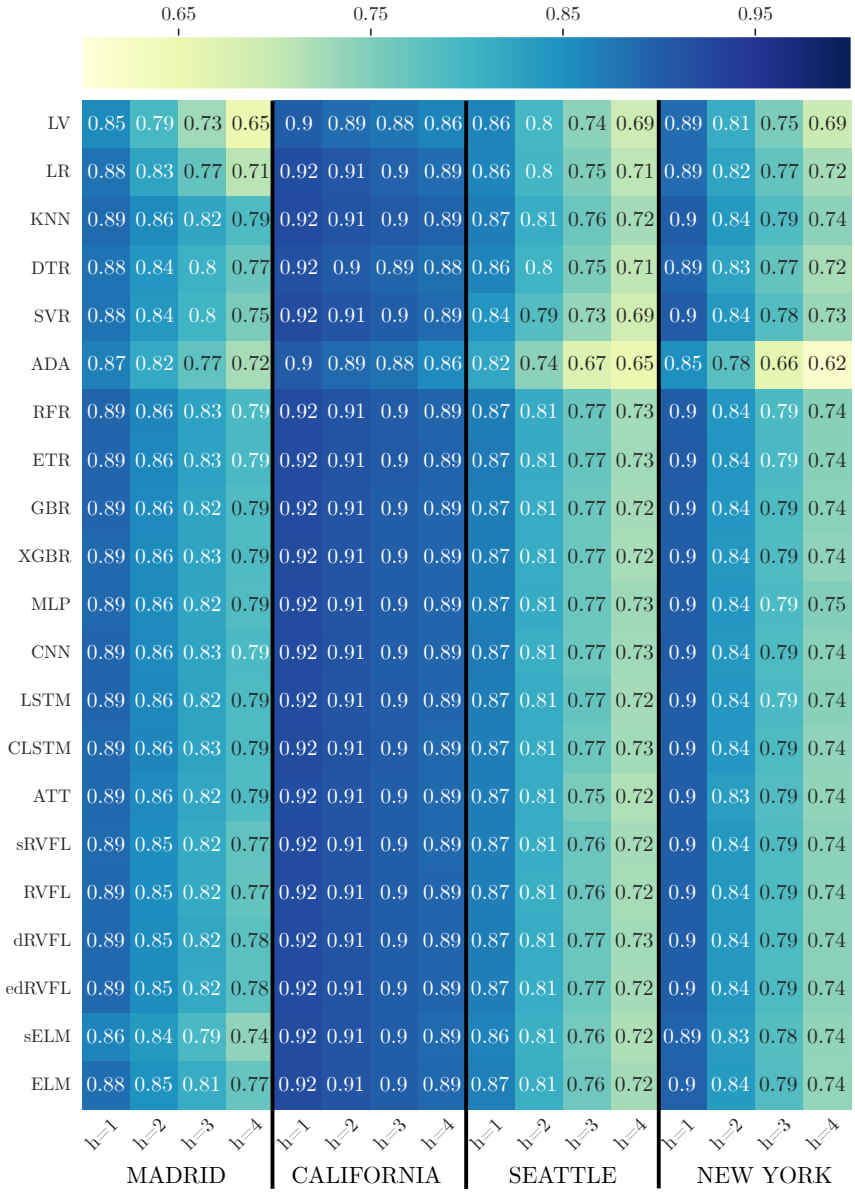



FIGURE 3.5: Performance benchmark. A heatmap shows the average $R^2$ test score obtained by each model and data source. Values are computed as the mean value of the test scores obtained for the 10 locations selected for each data source and value of the forecasting horizon. Columns stand for data source and forecasting horizon, while rows correspond to the considered forecasting models.

other basic services vehicles occupy the roads at night and early morning hours. This is not the case of special districts like the surroundings of universities, shopping malls and recreational areas, which impact on the traffic flow trends according to the schedules of their activities. Traffic flow at highways does not face these issues, degrading the forecasting performance more smoothly when increasing the prediction horizon, as it can be observed in the California test results.

With the focus set on the results of each model for the same collection of datasets, some of them render similar scores. At a first glance, the five Deep Learning architectures under consideration perform similarly to ensemble methods (except `ADA`). Theoretically, the `LSTM` model should provide more accurate predictions with regard to `CNN`. The input data contains information that evolve through time for a single road segment, so only temporal features can be distilled. Therefore, an architecture designed for such temporal dependencies should perform on top of an architecture that focuses on spatial relationships. More interestingly, the `CLSTM` model does not provide any advantage either, even that it combines both architectures. Finally, the `ATT` model provides the worst results for the considered Deep Learning models, probably due to its high internal complexity.

Shallow learning methods obtained a slightly lower $R^2$ score. Nevertheless, if the payoff for a minor performance degradation is a faster training time and less computational resource requirements, shallow learning methods should be taken into consideration. `SVR` is an exception, which holds by far, the longest optimization time among the analyzed methods. As long as researchers do not set iteration limit when searching the hyper plane combination that best fits the data distribution, `SVR` can demand long hyper-parameter optimization periods [326]. To end with, the relatively good forecasting performance of the naïve `LV` method for low values of the forecasting horizon $h$ imposes a narrow gap for improvement, as evinced by the negligible $R^2$ differences noted between models.

Finally, `ETR` has obtained the best score metrics among all analyzed methods. The ensemble nature of this method makes it better fit the training traffic data. Ensemble methods usually stand high in the ranks of every performance benchmark. By merging the outputs of several base learners, ensembles ensure that their overall performance does not get biased by potentially noisy training samples.

After analyzing the most commonly used models in the literature, the discussion shifts to randomization-based methods. The RVFL variants provide similar performance metrics between them but also with respect ensemble and Deep Learning models. This behavior is mainly explained by the direct links between input and output layer (see Figure 3.2). As previously explained, the self-descriptive nature of traffic measurements makes these input data act as high-quality predictors by themselves. Raw traffic information can (and *must*) be taken into consideration when furnishing the traffic forecast. Since all considered RVFL variants share this capability through their direct input connection, the test score remains almost identical for several of the analyzed scenarios.

The other considered randomization-based neural networks provide remarkable results, but always slightly below the performance of RVFL neural networks. In particular, `sELM` delivers the worst results of the benchmark after the naïve baseline model `LV` and the simple `LR`. The `ELM` neural network outperforms its shallow variant for every scenario, which makes sense since it has more hidden layers and therefore can elaborate more complex feature representations from the incoming traffic data. Given the absence of direct links between the input and output layer, the ELM neural networks can only rely on the self-crafted features, granting the advantage to those builds with more hidden layers.

Sharing the same number of hidden layers than the analyzed non-shallow randomization-based neural networks, the `MLP` provide similar results. Only at the $h = 3$ and $h = 4$ the performance starts to differ from that obtained by RVFL variants. However, the increased computational cost associated to adjusting the inner weights via back-propagation has to be kept into consideration. A faster and more affordable training phase can be interesting in multiple scenarios like those analyzed in the current case study, bearing in mind the narrow gap between the $R^2$ test scores reflected in the figure.

Nevertheless, the computational cost of adjusting several base learners to one data distribution can be unaffordable when deployed on devices undergoing severely restricted computational resources. In those circumstances, randomization-based neural networks approaches can achieve similar performance results but they also afford reduced computational requirements. Training time for all RVFL based methods is consistently under 0.2 seconds, while ensemble learning models require training times 10 to 20 times larger. In the case of Deep Learning based methods, the training time goes from 9 seconds for `MLP` to 95 seconds for `ATT`. This showcases the relevant time-consumption improvement that these approaches provide, particularly interesting for limited hardware environments. Training times were computed on an Intel Xeon Gold 5118 CPU, 512 GB RAM and 4 Tesla V100 GPU server.

The training phase for the `KNN` is often referred to as fast, since all it technically involves is storing the training dataset. Thus, the training time is essentially the time required to store (or sometimes index) the data, which is generally quick. However, KNN's real computational cost arises during the prediction phase. When predicting a new instance, KNN needs to compute distances to all points in the training set (or a significant subset if indexing structures are used), sort these distances, and then decide the label based on the majority class of the k-nearest points. This can be very slow, especially for large datasets.

#### 3.5.1.1 Statistical analysis

Given such small differences between the scores attained by the models, it is necessary to assess whether they are significant in the statistical sense. Traditionally standard null hypothesis testing has been adopted in this regard, including post-hoc tests and graphical representations (e.g. critical

distance plots [327]) to visually assess which counterparts in the benchmark are performing best with statistical significance. However, recently criticism has arisen around the use of these tests, due to their lack of interpretability and the sensitivity of their contributed statistical insights, and to the number of samples used for their computation.

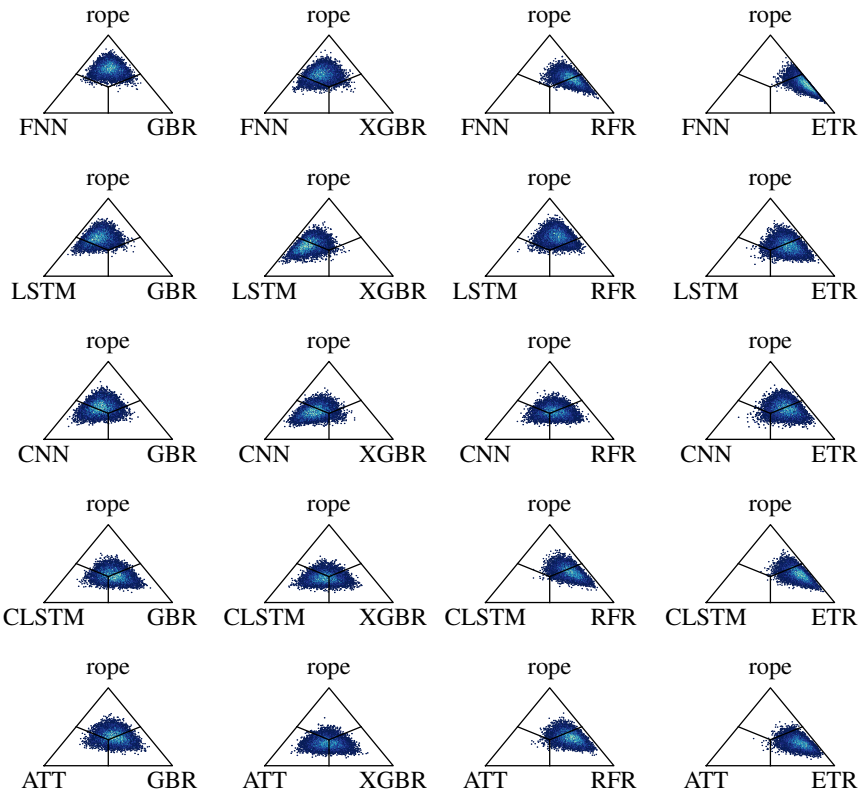



FIGURE 3.6: Bayesian probabilities sampled via Monte Carlo, and *rope* (i.e. absolute $R^2$ score differences under this value are considered to be equal). Rows correspond to Deep Learning models, whereas columns correspond to ensembles. Bright colors denote a higher probability of the fitted Gaussian distribution.

In this context, the seminal work by Benavoli et al in [328] exposed the drawbacks of standard hypothesis testing, and promoted the use of Bayesian analysis for multiple comparison analysis. Following this protocol, a Bayesian analysis between every (Deep Learning, ensemble) model pair is computed, which output is shown in Figure 3.6 (rows: Deep Learning models, columns: ensemble models). Bayesian analysis performed on every such pair allows computing the probability that one model outperforms another, based on the test results obtained by each of them over all locations, datasets and $h$ values. The obtained probability distribution can be sampled via Monte Carlo and displayed in barycentric coordinates, comprising two regions: one where the first model outperforms the second, and vice-versa. Additionally, a region of practical equivalence (where results can be considered to be statistically equivalent) can be set as per a

parameter called *rope*. This parameter indicates the minimum difference between the scores of both methods for them to be considered significantly different to each other. The value of *rope* depends on the task being solved. For example a forecasting error difference of one single car when predicting traffic flow at highways of 300 passing vehicles per analyzed interval can be ignored, as this margin does not affect a practical implementation of the predicting models.

The results of the Bayesian analysis depicted in Figure 3.6 reveals that `LSTM` and `CNN` have a slightly higher probability of providing better results than `GBR` and `XGBR` ensembles. However, the situation changes for `RFR` and `ETR`. The sampled probabilities of both ensembles when compared to Deep Learning variants are skewed towards the regions of practical equivalence (e.g. `RFR` versus `LSTM`) or towards the region where the ensemble performs better than the Deep Learning models (e.g. `ETR` versus `CLSTM`). Finally, the statistical analysis concludes that from the statistical point of view, there is no clear winner in the benchmark, nor any empirically supported reason for using Deep Learning based traffic forecasting models detrimentally to shallow modeling alternatives.

#### 3.5.1.2 Insights distilled from the benchmark

It can be concluded from the experimental results that Deep Learning models do not provide consistently better results than shallow modeling approaches. Furthermore, whenever hyper-parameters are properly tuned beforehand, ensemble methods outperform Deep Learning models in some cases. This fact demonstrates that parameter tuning should be mandatory in prospective studies to avoid unfair comparisons. Unfortunately, the hyper-parameter tuning stage is often neglected or mentioned very superficially, without the relevance it deserves.

Besides, the training complexity of this kind of algorithms is widely overlooked. The literature analysis presented in Chapter 2 unveils that short-term traffic forecasting publications are leaning towards more complex models on the understanding that their increased modeling power can improve the state of the art, often by narrow performance margins. However, such slight performance gaps do not translate into practical advantages for real traffic scenarios [162]. For a similar and sometimes even better result, classic Machine Learning techniques and randomization-based neural networks can perform as well as Deep Learning, but with less complexity and computational requirements.

It is also important to underscore the essential role of naïve methods when establishing the minimum complexity of the designed task (i.e. Figure 3.5). These baseline models should take part in any traffic forecasting benchmark. The task to be solved in Section 3.5.1 (i.e. predicting traffic state at a single road) is chosen on purpose to show that for simple tasks, complex models do not significantly improve the performance of a naïve model. The most meaningful information for the target to be predicted is made available at the input of every model (previous recent measurements collected at the target road). This is demonstrated by comparing the results of RVFL with those of ELM. Consequently there are no complex

relationships to be modeled, and ultimately, Deep Learning architectures can not provide better results than shallow learning methods. A lower performance boundary can also be established by means of autoregressive models, but they are very sensitive to parameter configuration. By contrast, the lack of parameters of naïve methods make them a better choice to ascertain the improvement margin that can be achieved by virtue of data-driven models.

Another relevant aspect is how train and test data are arranged. A common practice observed in the literature is that test data are carefully chosen in order to obtain the desired performance for the presented traffic forecasting method. Test data are often selected from short temporal intervals, with almost identical characteristics than the training data. This methodology neglects some of the basic notions of Machine Learning: whenever possible, test data should be different (yet following the same distribution) than training data to check the generalization capabilities of the developed model. Some of the papers analyzed in Chapter 2 reserve only one month of traffic data for training, and one week for testing. As a result of this partitioning criterion, the results can be misleading as learned traffic behavior can be identical to that present in the test subset, thereby generalizing poorly when modeling traffic belonging to other periods along the year.

In this context, different train/test partitioning choices are enabled by the amount of available data. In the best of circumstances, the data source covers at least two complete years, so researchers can train the model over the data collected in the first year, and check its generalization capabilities by testing over the data of the second year. Throughout the year, the traffic profile can change in some points of a traffic network due to e.g. road adjustments, extreme meteorological events or sociopolitical decisions. These circumstances generate unusual traffic daily patterns that modify the data distribution, inducing an additional level of difficulty for the learning and adaptation capabilities of data-based models. In this context, it is remarkable the fact that PeMS (arguably the most commonly used data source as it provides several years of traffic measurements), is not commonly utilized over the entire time span covered by this dataset.

The second option is to have only one complete year of traffic data. In this case, it is suggested arranging the data as depicted in Figure 3.4: three weeks of every month as train data, and the remaining days of every month for testing. This configuration allows the model to learn from different traffic patterns, so that authors can check if the model generalizes properly to unseen data using the test holdout and considering, at least, all traffic behaviors that can occur during the year for the location at hand.

The last case corresponds to a data source that does not cover an entire year. In this scenario, the generalization of the model's performance to the overall year cannot be fully guaranteed because, depending on the time range covered by the dataset, patterns learned by the model can only be used to produce forecasts for a short period of the year. Given the amount of traffic data available nowadays for experimentation, it should not be an issue for prospective works to find a public traffic data source that matches

the desired characteristics for the study, and also provides at least a full year of data.

### 3.5.2 RQ3.2: Instability of random models

Now the discussion centers on the instability associated to the random nature of the models at hand. The dispersion of the $R^2$ performance scores resulting from repeatedly training (each with a different random seed) the randomization-based neural networks under consideration is inspected. The random nature of the weight initialization process causes an statistical dispersion in the distribution of the performance metrics after several test runs. Therefore, the scope of this experiment is to numerically assess this dispersion.

Only `RVFL` and `ELM` are compared, due to their almost identical architecture. For each analyzed neural network, several number of hidden layers $L \in \{2, 4, 6, 8, 10\}$ are considered. The number of neurons per layer, which is kept equal for every hidden layer, varies in a discrete range of $N^\ell \in \{1, 10, 50, 100, 500, 1000\}$. To avoid a combinatorial explosion of simulated models, the number of neurons per layer is assummed equal across layers, e.g. $N^\ell = N^{\ell'}\ \forall \ell, \ell' \in \{1, \ldots, L\} : \ell \neq \ell'$. To determine the stability of these models, every combination of hidden layers and neurons per layer between the above ranges is tested by fitting a model for each training dataset and forecasting horizon $h \in \{1, 2, 3, 4\}$. This process is repeated 100 times, each with a different random seed, issuing 100 $R^2$ score measurements per configuration. Since training and test data is the same for every test run, the instability associated to the random nature of the initialization process can be isolated.

For the dispersion analysis the Coefficient of Quartile Variation (CQV) is selected, due to its capability to provide a relative measurement of the dispersion from the test performance metrics of a particular model and configuration (i.e. number of hidden layers and neurons per layer). The CQV of each dataset, model and configuration results from the first (25%) and third (75%) quartiles computed over the 100 $R^2$ score values obtained during the experimentation, namely:

$$CQV = \frac{Q_3(R^2) - Q_1(R^2)}{Q_3(R^2) + Q_1(R^2)}, \tag{3.22}$$

where $Q_n(X)$ denotes the $n$-th quartile of the probability distribution of variable $X$ estimated from a sample of realizations. In short, CQV values close to 1 stand for divergent quartiles and thereby, high statistical dispersion. On the contrary, low CQV values correspond to stable data distributions, since the gap between their first and third quartiles is narrow.

For each configuration, a $R^2$ performance distribution can be obtained from 100 test executions, where the shape of each distribution is directly related to the stability of the model under analysis. In this line, the selection of one data source or another only impacts on the median of the distribution, but is the architecture design what modifies the dispersion of

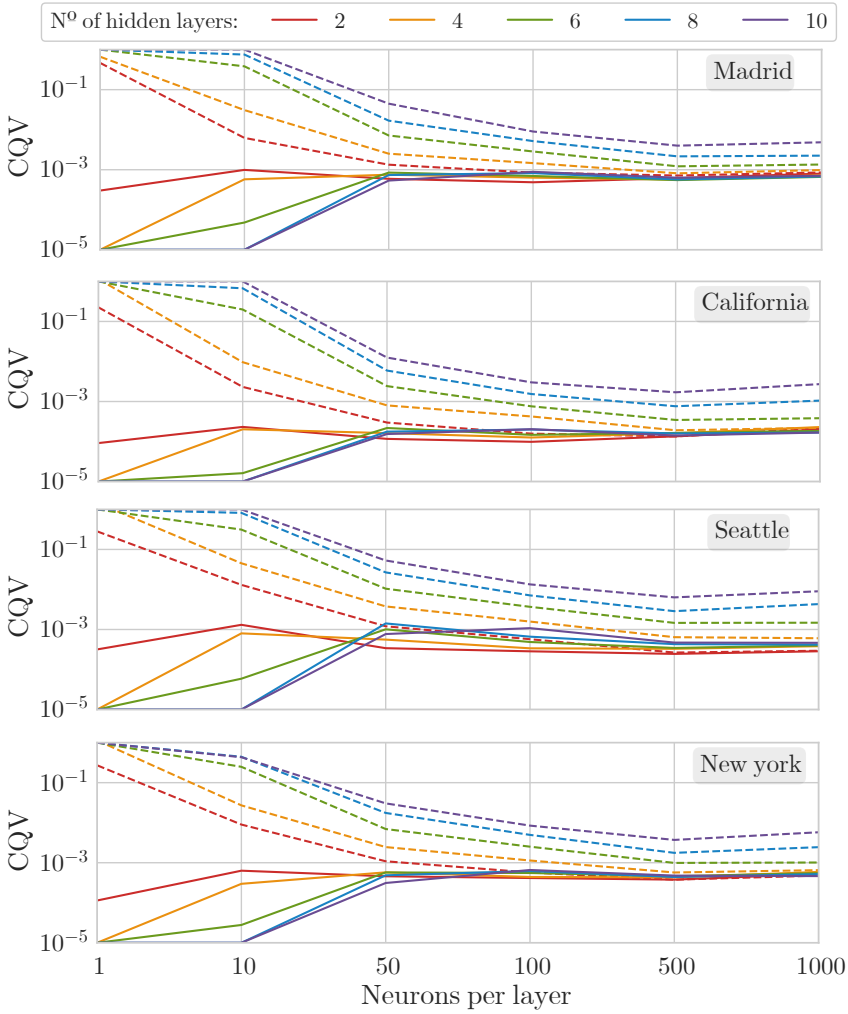



FIGURE 3.7: CQV of `RVFL` and `ELM` for $h = 1$. The CQV obtained for the 10 datasets of every data source is averaged and displayed for each subplot. Lines are drawn according to the number of neurons per layer (X axis) and the number of hidden layers (color of every curve in the plot). Continuous lines stand for RVFL models while dashed lines for ELM models.

the performance. Therefore, the same insights can be distilled from any of the subplots available in Figure 3.7. Only results for $h = 1$ are shown in the Figure 3.7, due to space limitations. Additionally, the different RVFL variants have obtained similar dispersion metrics, so only `RVFL` and `ELM` neural networks are displayed.

The two neural networks behave in opposite ways when the number of neurons per layer is small. On one hand, the direct link between the input and output layers endows RVFL architectures with stable results when they have a few neurons per layer. With such a small amount of

hidden features, the optimization process grants more attention to the input features (due to their self-descriptive nature), which are not affected by the random initialization of the architecture weights. Therefore, the predictive behavior remains roughly unaltered, providing a low dispersion for the test results. In the case of RVFL architectures with a high number of layers but few neurons, the incoming input values are overprocessed, providing a set of hidden features that are overly unrelated to the original traffic measurements. Consequently, the optimization process grant even less attention to these hidden features. In this way, the behavior of the model remains apparently unaltered, disregarding the initialization values of the neural parameters.

On the other hand, ELM only relies on hidden features when issuing its prediction. Those configurations with a low number of neurons per layer cannot produce a reliable set of hidden features that contain enough knowledge to perform a successful prediction. In these cases, the initialization of the weights produces a huge impact on the quality of the hidden features. In contrast to RVFL, ELM reduces the statistical dispersion of its results for architectures of a few hidden layers. The input values are processed multiple times after each hidden layer, so the impact of badly initialized weights produce more disparate hidden representations after a high amount of neural layers, regarding more restrained architectures.

Finally, both neural networks converge to similar CQV values for a high amount of neurons per layer, which implies that the dispersion of $R^2$ performance along executions is similar. RVFL increases the variability of its results as hidden features become more significant and therefore have an increased impact on the output prediction. In the case of ELM, a high number of neurons results in a likewise higher number of hidden features and hence, more chances for the output layer to be fed with informative features towards computing the final output prediction.

## 3.6 Summary

Thanks to its feature extraction capabilities, Deep Learning has become the preferred tool for modeling traffic congestion. However, this trait should not be what fuels a whole research path towards improving Deep Learning models for traffic forecasting, since traffic variables (e.g. average speed, flow) are already good predictors. The performance gain provided by novel Deep Learning models displayed in recent literature is not centered on innovative alterations of these models. On the contrary, authors select constrained time windows (e.g. one month of data) for building training and test holdouts that barely differ or leave out shallow methods that perform similarly (i.e. low complexity models) from comparison as tricks for distilling the desired insights. This chapter provides a comprehensive performance benchmark where different Deep Learning architectures are compared to less computationally demanding methods, aiming to set a baseline that ceases the above described practices. Obtained results strengthen the original hypothesis: Deep Learning models do not provide performance advantages when predicting isolated points of

the traffic network. Data is arranged as a time series, where the last traffic measurements serves as the model input. In this scenario, the traffic variable is always the best descriptor for predicting the next step of the time series, hence the complexity of a Deep Learning architecture can not provide better results than a shallow leaning method.

In an effort to find alternatives to the massive use of Deep Learning, this chapter also addresses randomization-based neural networks as a modeling tool for traffic forecasting. Performance results are obtained following the same experimental setup as in the original benchmark. This not only allows conclusions to be drawn about the predictive capabilities of shallow and Deep Learning methods, but also expands the scope of the performance benchmark. In detail, RVFL results in an interesting modeling option for implementations with low hardware resources. The direct connections between input and output layer give raise to a stable model, regarding the random initialization of the inner weights. Overall, RVFL should be considered for real-world scenarios where the traffic forecasting problem consist on predicting the state of a single sensorized road.

# Chapter 4

# Traffic Forecasting Models in Limited Data Regimes

The acquisition of comprehensive data is paramount for the creation of robust predictive models. In the previous chapter, forecasting models are built upon extensive traffic datasets spanning a complete year. This way, models can learn all the traffic patterns that might change due to seasonalities such as Christmas or the Easter holidays. This practice allows isolating their modeling capability in what refers to the algorithm's architecture. However, a key question faced by researchers and traffic managers pertains to the required duration for data collection, especially when historical datasets are absent. If traffic data from the target road are not available, the development of a traffic forecasting model must be postponed until enough traffic measurements are collected. An interest for modeling a non-sensorized road segment should not translate into delaying the model implementation a whole year so all the different data patterns are collected. With this issue in mind, this chapter provides a methodology towards the fast deployment of capable traffic forecasting models, reducing dramatically the time window scheduled for data acquisition. The core components of such methodology are Transfer Learning (TL) and Online Learning (OL). On one hand, TL allows leveraging the knowledge encapsulated in a model adjusted to other road segment, making the target model to start the learning process (i.e. from data captured at the target location) already being capable of producing traffic forecasts. On the other hand, OL enables updating the model to new traffic trends by learning from upcoming traffic measurements.

Accurate traffic forecasting models can be developed when data is available. However, real-world scenarios usually need for a capable model to be deployed as soon as possible. In this context, there is no time for collecting data through a whole year. Inspired by the above scenario, this chapter tackles traffic forecasting under a data availability constraint. The goal is to develop a model that performs similarly compared to a model that has been adjusted to a comprehensive data holdout, but without the need of waiting to collect such data. To reach this goal, both TL and OL techniques are exploited. The parameters of a traffic forecasting model adjusted to a sensorized location are transferred to the target forecasting model, so the basic prediction capabilities are already learned by the

model. A month of traffic data collected at the target location serve to adjust the inner weights of the target model, making it ready to being deployed. During operation, the model is further refined according to the incoming traffic flow measurements, so the concept drift between the original transferred context and the current traffic behavior can be addressed. As a result of this case study, the following contributions are achieved:

1. Allowing capable forecasting models to be deployed before collecting data a during a whole year.
2. Providing insights on which model to transfer depending on the characteristics of the road to be modeled.
3. Comparing several degrees of data availability constraints, intending to portray the best approach for each case.
4. Exposing the benefits of constantly updating a traffic forecasting model during its operation.

## 4.1 Model adaptation

In the realm of transportation research, plenty of traffic datasets are available to the public at no cost [223], [321]–[323]. Yet, when compared to other domains such as image recognition or natural language processing, the task of predicting road traffic as a time series poses unique challenges. Traffic patterns are not just a function of vehicular volume, but are influenced by a myriad of localized factors. The culture of a region, its meteorological patterns, local festivities, the intricacies of its road network, among others, play crucial roles in shaping traffic patterns. Consequently, to ensure a holistic understanding of these dynamics, a continuous monitoring period spanning a full year is essential [329]. Such an exhaustive time window ensures the encapsulation of all salient features, from regular weekday congestion patterns to anomalies arising during festive seasons or unforeseen disruptions.

Public traffic datasets serve many purposes, such as encouraging the development of novel modeling methods or providing a shared dataset towards furnishing a performance benchmark (as in Chapter 3). However, having access to traffic data collected at the area of interest plays a huge impact on real world implementations. For instance, a model fine-tuned on the sprawling highways of California [223] may be prone to provide innacurate forecasts on the bustling streets of the Tokyo urban area [330]. The inherent idiosyncrasies tied to each location make transferring models between distinct cities a non-trivial endeavor. Ideally, for a forecasting model to achieve optimal performance in predicting traffic for a specific location, it should be trained on data collected on that location. This tailored approach ensures that the model is cognizant of, and can account for, the unique traffic determinants of the area in question.

Bearing the above ideas in mind, a specific case may be conceived as follows: a municipal council may have strategically deployed permanent

sensors across certain segments of the city's transportation network. Over time, these sensors accumulate substantial data, eventually facilitating the development of robust forecasting models. Yet, when the need arises to formulate predictive models for new, previously unmonitored locations, there lies a challenge. To ensure a comprehensive understanding of the traffic dynamics at these new road segments, it is often imperative to acquire data over an extended duration, potentially spanning an entire year. Consequently, immediate deployment of predictive models for these new locations becomes a temporally constrained endeavor. On this particular but common scenario, the knowledge encapsulated in already fine-tuned models can be exploited, aiming to boost the performance of a model under development thanks to a technique known as *Transfer Learning* [331].

## 4.1.1 Transfer Learning

In tasks where there is a significant shortage of annotated data, the necessity to utilize knowledge from previously encountered tasks to address new, yet analogous, challenges has given rise to the TL paradigm. This approach seeks to enhance the performance of models in the target domain by leveraging insights derived from distinct yet related source domains. Consequently, this mitigates the reliance on vast amounts of data from the target domain to build effective models. From the last decades, several comprehensive overviews have revolved around TL methods and their applications [22], [332]–[334], hence demonstrating the repercussion of this technique in a multitude of diverse tasks.

### 4.1.1.1 Notations and definitions

In every TL setup, it must exist a source and target domain, denoted as $D_s$ and $D_t$ respectively [22]. Each domain is composed by a set of data employed as model input $\mathbf{X}$ and the values to be predicted $\mathbf{Y}$, producing $D_s = \{\mathbf{X_s}, \mathbf{Y_s}\}$ and $D_t = \{\mathbf{X_t}, \mathbf{Y_t}\}$ for the source and target domain. Likewise, the tasks to be solved in each domain are $T_s$ and $T_t$ respectively. Being $f_t(\cdot)$ the objective predictive function to be solved by a target model, TL aims to assist the learning of such function using the knowledge in $D_s$ and $T_s$, where $D_s \neq D_t$ or $T_s \neq T_t$.

Transfer Learning techniques can be applied to a variety of models, but especially for Deep Learning, their application is immediate. It is only necessary to choose which layers of the architecture to transfer to the target model. In detail, the first layers are commonly dedicated to elaborate a *feature map*, an alternative representation of the input values that should provide a better description of such values, so the task can be solved optimally. These first layers are usually the section to be transferred from $D_s$ to $D_t$, since is where most of the useful knowledge is encapsulated. For instance, the first layers of computer vision models learn features similar to Gabor filters and color blobs [335]. On the opposite, deeper layers are intended to interpret the feature map according to the particular traits of the target task.

The success of TL in Deep Learning architectures is attributed to the shared feature representations learned by models. These representations are often general enough and can be applied to related tasks. However, the key point in this process revolves around determining "what" to transfer [335]. This requires identifying the parts of knowledge from the source that can be beneficial for the target task. Distinguishing between source-specific knowledge and elements common to both domains is essential to ensure that the transferred knowledge is relevant. Transferring knowledge from a weakly related source may hinder the performance of the target model, a phenomenon referred to as *negative transfer* [336]. In the context of traffic forecasting, this translates to finding a $D_s$ close to $D_t$, so the traffic behavior at the target road segment does not discern severely.

#### 4.1.1.2 Transfer Learning in traffic forecasting

The work of Hu et al. [331] offers a nice case study about the capabilities of TL in a different but comparable domain (i.e. time series forecasting), where they need to predict wind speed at newly-built farms. Needless to say that sufficient historical data is not available for training an accurate model, so authors propose to transfer the predictive knowledge captured over older wind farms that have long-term records. The aforementioned problems also arise within the transportation domain. In [337] a similar approach to the one presented in this chapter is applied. Authors propose a case study where the goal is to develop models that can predict the spaces available at parking lots. The particularities of the distinct urban areas make these models to fail if sufficient data has not been collected. Here, TL boost model performance for those areas where enough data has not been collected yet. The schema for transferring knowledge follows the standard methodology described in this chapter: 1) a LSTM-based model is trained on $D_s$; 2) the weights of several initial layers are transferred to a target model, which replicates the architecture of the source model; 3) the weights of such layers are fixed (also referred to as *frozen* parameters); 4) remaining layers are adjusted according to data from $D_t$.

The closest contemporaneous work in which refer to the data to be predicted is [338]. The author investigates TL to provide speed data estimations using graph convolutional generative autoencoders (GCGA) [339]. Precisely both the traffic variable and the proposed TL mechanism deviate from the case study presented in this chapter. First, in [338] speed data is derived from GPS traces collected at several Chinese cities (i.e. Beijing as $D_s$ and Shanghai, Guangzhou and Shenzhen as $D_t$). Even that the traffic networks to be modeled belong to different cities, predicting speed primarily requires monitoring for anomalies or disruptions that could cause deviations from the nominal speed. Given the inherent stability around the nominal speed, target models can use this as a baseline, adjusting for observed or predicted anomalies. Secondly, the introduced TL mechanism is specific to the graph neural networks: only the topology related parameters of a GCGA model trained on $D_s$ are adjusted according to $D_t$. Therefore, the original GCGA model is adapted from a traffic network

topology to another, but the mechanisms entrusted with predicting speed are kept fixed.

As seen in the related work, a transferred model can only be refined according to data from $D_t$. Implementing a traffic forecasting model as soon as possible might be a constraint that limits the phase of data collecting. Under this data scarcity scenario, the transferred model can act as a precursor of a model adjusted through *Online Learning*, a technique that allows a model to be updated during its operation.

### 4.1.2 Online Learning

Online Learning refers to a model training approach where the algorithm incrementally updates and refines its predictions in response to new data points presented sequentially, rather than relying on a fixed, previously collected training data holdout [340]. In the evolving landscape of data-driven algorithms, models operating over continuously flowing data streams are often confronted with the challenge of changing data distributions, known as *concept drift* [38]. Drifts imply that predictive models trained over data become eventually obsolete, and do not adapt suitably to new distributions. Standard batch learning methods train on a static dataset, which makes them vulnerable to changes once training is completed.

#### 4.1.2.1 Online Learning in traffic forecasting

Few studies can be found mixing traffic data with OL [341], [342], possibly due to the relatively large time gap between arrival samples with respect to more traditional OL tasks, where consecutive instances turn up in typically less than one minute [343]. Niu et al. [341] propose an online route finding mechanism for smart cities supported by traffic flow predictions. As flow predictions evolve through time, the best route is constantly checked and updated towards reducing the expected time of arrival. However, the forecasting model does not benefit from an OL approach. On the opposite, Chen et al. [342] propose a traffic condition model that updates its inner weights during operation. Three major differences arise in regard to the experimentation presented in this chapter: 1) instead of predicting flow or speed, the model ranges its output between three congestion levels (i.e. free flow, slow traffic and impeded condition) which is an easier task to solve; 2) the proposed model is applied to traffic data collected from December 28, 2014 to February 3, 2015. This short time window reduce the credibility of the distilled insights, since the analyzed traffic patterns only occur during winter months; 3) the capabilities of TL where not considered.

By leveraging knowledge from a related source domain, TL provides an initial model for the target task. However, this preliminary model is typically far from optimal due to domain differences. The mere act of transferring knowledge does not always guarantee a perfect adaptation to the new environment, and sometimes, only a fraction of target domain data is available to adjust the model. To further enhance the performance

of a transferred model, $D_s$ can be interpreted as the "original concept" whereas $D_t$ represents a concept drift.

A work that can illustrate the above idea is [344], which revolves around the detection of concept drift in pedestrian flows. Authors provide several adaptation strategies towards dealing with the concept drift. The analyzed data comprises several years, including the Covid-19 pandemic, which generated a concept drift in the pedestrian flows at a Spanish city. The behavior of pedestrian flows dramatically changed in accordance with the new environmental and social aspects. The above works motivates the use of OL for model refinement, since the exposed concept drifts can be interpreted as if a model from $D_s$ was transferred to $D_t$, representing the before and after Covid-19 pandemic pedestrian flow profiles.

## 4.2 Description of the case study

Two major techniques are employed for model adaptation: Transfer Learning and Online Learning. Complementary to each other, the former focuses on reducing the amount of data needed to train a model, while the latter takes advantage of the latest available data to refine the model's behavior. The lack of research works addressing these techniques for traffic forecasting provides an interesting challenge to be solved, where the desired outcome is reducing the amount of time needed for deploying a capable traffic prediction model.

Motivated to analyze the effectiveness of the above approaches, the case study proposed in this chapter is divided into two stages: in the first stage, the performance of a transferred model is analyzed, while in the second stage, this model is updated with each new traffic measurement. This makes it possible to isolate the advantages of each technique in order to assess their value in a real scenario.

In this manner, the challenges to be solved during the experimentation can be summarized as the following Research Questions (RQ):

- RQ4.1: How do a transferred model behave when no updates are given to the model during its operation?
- RQ4.2: Do model updates provide significant performance gains to justify the increased computational effort?

## 4.3 Materials and methods

The following section describes the materials and methods employed during experimentation. The traffic data employed for performance assesment is introduced in Section 4.3.1. The architecture of the selected data-driven model is defined in Section 4.3.2. To finish, Section 4.3.3 proposes an experiment design dedicated to compare several data availability constraint scenarios towards discussing the best approach at each scenario.

### 4.3.1 Data for evaluating model adaptation

Data for this research work have been collected from a public repository maintained by the City Council of Madrid (Spain) [321]. From the public traffic data sources introduced in Chapter 3, Madrid is the one that provides the most challenging forecasting scenario, thanks to a combination of three factors: traffic variable, scope and time resolution. Traffic flow is a signal that presents cyclical patterns according to the weekly day, but it also can present spikes or valleys due to one-time events. On the other hand, the speed signal remains constant (i.e. nominal speed) and the model focuses on detecting anomalies that reduce traffic speed. Urban traffic has a myriad of points of interests (POIs) that can alter target variables, ranging from schools, hospitals and sports stadiums to nightlife and residential areas. On the opposite, the lack of external factors at interurban networks helps traffic measurements on highways to remain stable. Finally, data from Madrid is the only one among those introduced in the previous chapter that offers a time resolution of 15 minutes, allowing traffic to further evolve regarding a shorter time window such as 5 minutes.

Regarding the specific datasets arranged for the following case study, traffic flow data is aggregated in the form of 15-minute periods during 2017 and 2018 years as in [329]. Data was collected by sensors located in urban arterials (see Table 4.1), placed close to the main belt of the city, the so-called M-30 highway. These four locations have been selected (shown in Figure 4.1) towards considering different traffic profiles based on their number of lanes and proximity to Madrid center: Alcalá, Bravo Murillo, Doctor Esquerdo and García Noblejas streets. Still, TL works best when $D_s \sim D_t$, so selected road segments are either primary or secondary roads, where the main difference lies in the number of lanes.

Table 4.1: Selected road segments for studying Transfer Learning.

| **Street name** | **Road type** | **Number of lanes** |
|---|---|---|
| Alcalá | primary | 2 |
| Bravo Murillo | secondary | 3 |
| Doctor Esquerdo | primary | 4 |
| García Noblejas | primary | 3 |

Usually, bottleneck states are propagated downstream, in opposite direction to the traffic flow. Even so, there are some conditions where this transmission occurs upwards along the road [215]. Under this premise, the following scheme is proposed: if the flow value of a certain loop A is to be predicted at time slot $t$, features are defined as $\{t-5,\ldots,t-1\}$ instant flow values, recorded at the next four and previous four loops placed in the vicinity of A, along with $\{t-5,\ldots,t-1\}$ slot flow values from loop A itself. Consequently, a set of 45 historical input values are given to the model in order to predict the next flow value in a 15-minute interval. This modeling choice assumes that previous flow values from surroundings and target location itself, contain enough predictive information to build a proper short-term forecasting model.

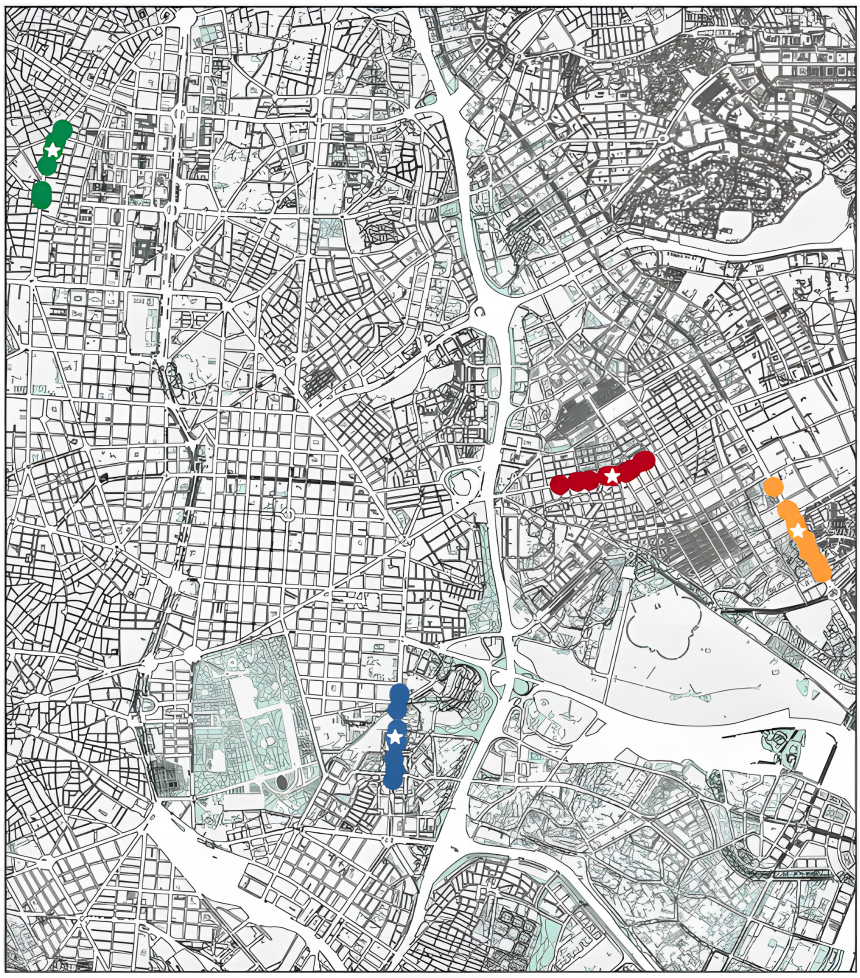


FIGURE 4.1: Location of the selected loops around the M-30 surrounding area. Colored markers are kept throughout the study: Alcalá (red), Bravo Murillo (green), Esquerdo (blue) and Noblejas (orange). The white star markers ☆ denote the loops where predictions are made.

### 4.3.2 Deep Learning architecture

Given that the focus of this chapter is placed on knowledge transfer and updating strategies, a Deep Learning architecture is selected for modeling traffic, thanks to the straightforward application of TL and OL techniques in these kind of architectures. As seen in Chapter 2, modeling trends in traffic forecasting that follows a time series approach are arguably monopolized by convolutional and/or recurrent neural networks [70], [169], [200]. Hence, a similar approach is followed. The proposed model receives an input of 9 vectors containing 5 values each, formed by $\{t-5,\ldots,t-1\}$ traffic flow values of each $4+1+4=9$ available loops for the road under analysis. The total of 45 feature values goes through 50 one-dimensional

convolutional filters, of kernel size equal to two steps. This process allows extracting high-dimensional features from input vectors, intending to encapsulate useful relationships between posterior and anterior traffic sensors. The convolutional layer output is fed to a stateful LSTM layer [16] composed by 75 memory cells, endowed with the role of discovering long temporal dependencies over time. Finally, a dense layer of 50 neurons selects the most significant output values, making prediction of the future traffic flow level. The Rectified Linear Unit (ReLu) serves as the activation function between layers [281]. The proposed standard Deep Learning architecture is applied to model every traffic profile analyzed in this study.

Although the results obtained in Chapter 3 (see Figure 3.5) support the hypothesis that for traffic data CNNs and RNNs do not yield any benefit regarding other modeling approaches, authors claim in the literature that these architectures can extract spatial and temporal features from the data [39]. Additionally, in this case study traffic data from nearby locations complement the input data, hence the convolutional layers could be able to distill the aforementioned spatial features, finding relationships among the considered locations.

### 4.3.3 Experiment design

The forecasting horizon is set to one time slot (i.e the next 15-minute interval). In addition, the disposal of historical traffic flow data for another three locations of the same city is assumed, so that the knowledge captured by a forecasting model in these locations will be exported to the model developed for the location of interest. In this fictional scenario, the release date for a traffic forecasting model is set to 01 January 2018. With this methodology in mind and depending on the data availability, three Possible Scenarios (PS) can occur:

- `PS1`: Only historical data at other locations is available. This means that no data has been collected at the target location, so the only option is to develop a model for another road segment, and transfer it to the location of interest.

- `PS2`: Historical data is available at other locations, along with a few weeks of historical data at the target road segment. Data at this location should be collected, for example, by deploying surface loops or any other type of temporal sensor. Then, the knowledge contributed by models learned from data of sensed roads (hereafter referred to as *donor models*) can be further specialized by learning from this temporal data collected at the target placement. A second approach might be to train a model from scratch, only using temporal gathered data.

- `PS3`: Historical data is available at the target road segment. A traffic forecasting model can be developed via standard batch learning.

In order to discover the right setting for every PS, the experimental setup follows the scheme described at Figure 4.2. The steps of this process are detailed next: firstly, four regression models are produced, one per

selected road described in Section 4.3.1. The goal is to provide high performance forecasting models, so full 2017 year historical data is fed to DL network for training. This way, the network is trained on examples of every day along the year, learning both usual traffic profiles from normal working periods, and special events such as Christmas or Easter holidays, where traffic profile changes no matter the weekday. One day sized batches are used for training for 10000 epochs without shuffling, to refresh the model as per the evolution of traffic profile sequentially. The same DL architecture (as described in Section 4.3.2) is used for all experiments.

At the `PS1`, data is not available at the target road segment, hence only information from other roads can be exploited. Knowledge from donor models are transferred to the target model. The testing phase is performed from January first, until the last day of 2018, covering all existing days. Now, if few weeks of historical data are collected by temporal surface loops like in the `PS2` (i.e. full month of January 2018), two new options emerge. The first one is to retrain transferred models by using data from the target road segment, making models to adapt from their original concepts to the actual one, throughout 2000 epoch. The reduced number of epochs are due to the lower amount of training data. The second option consists of directly train batch-wise a new model from scratch (i.e. all inner weights starts from a random initialization), by using the data from January 2018. The main drawback of both strategies is that the model release date would be delayed until February. Moreover, the initial hypothesis is that the examples that conform the training holdout, may have noticeable impact over the model behavior when facing the prediction during special-event days. In fact, at Spain (country from which data has been collected), the traffic profile of the first week of the year is quite unique, because of New Year's and Epiphany day (national festivity at January $6^{th}$), producing flow peaks at certain hours, when people start or end they holiday trips. Lastly, the `PS3` allows preparing a model drawn from 2017 data at target location. All the knowledge collected in a whole year should produce the highest quality possible model, because the dataset actually has traffic measurements for all different special events at the target road segment.

In addition to the previously explained tests, an online version of each model is also added to the comparison study, in order to assess how much performance can be improved with respect to its offline counterpart. On the offline configuration, the only source of knowledge comes from TL and batch learning. After that, the model's operation is limited to predicting the next traffic flow value. In contrast, OL imprints small updates to the model over time, by using the incoming real value samples as explained in Section 4.1.2. Under this paradigm, the performance is evaluated by comparing predicted value to real and then, the actual value serves as learning example for 1 epoch. It is important to perform only one gradient update per tested sample; otherwise, the adaptation of the weights of the model could become overly biased to just one instance.

As in Chapter 3, the coefficient of determination $R^2$ serves as regression metric (see Equation 3.21). This coefficient expresses the quality of the forecasting model, as it measures the variance between real and predicted

values. The metric is computed for each time slot $t$ averaged over a full week sized window, in order to show performance changes originated by traffic flow drifts.

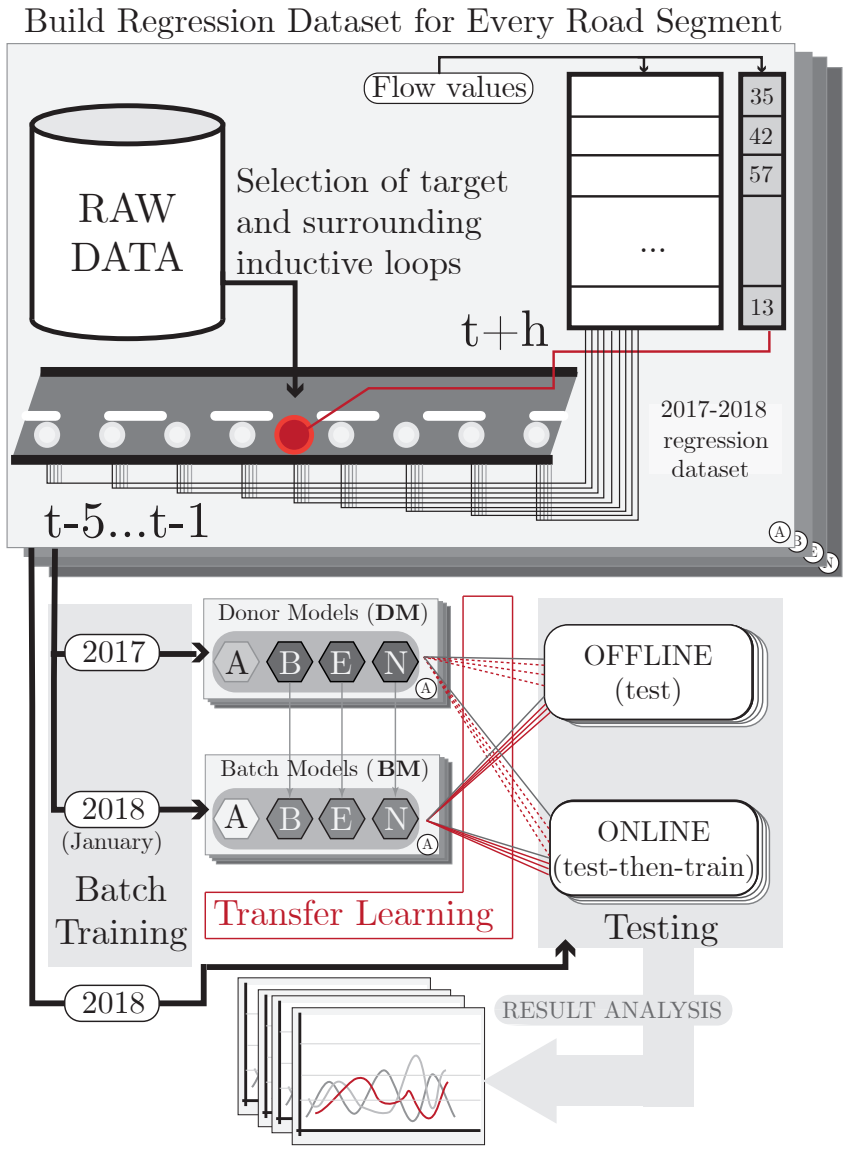



FIGURE 4.2: Experiment design used in this case study. Datasets for each year are built based on traffic flow values collected from selected ATRs of 4 different roads (A, B, E, and N, respectively). The dataset corresponding to 2017 contains information used for training donor models (DM), while the first month of 2018 is used to train a model from scratch and to re-train DM copies (only those from different placements with regard to target location). Then, all models are tested under offline and online settings, producing the results discussed in Section 4.4.

## 4.4 Experiments and results

A case study is conducted following the scheme illustrated at Figure 4.2. Towards easing the results analysis, the discussion is split in two: 1) no updates given to the model or offline approach (Section 4.4.1); 2) leveraging the information within incoming traffic data via small updates or online approach (Section 4.4.2).

### 4.4.1 RQ4.1: Transferring models with no updates

The discussion begins by commenting on Figure 4.3 and Figure 4.4, which depict the $R^2$ score comparison between different offline approaches. Colored identifiers are used for denoting the origin of the data employed for model training. In the cases where the target road segment and data source location do not match, TL techniques have been applied. The remaining colored line identifies a model trained with a data holdout covering the 2017 year collected at the target road (i.e. `PS3`). Two line styles are also employed: 1) dotted lines means that transferred models were implemented

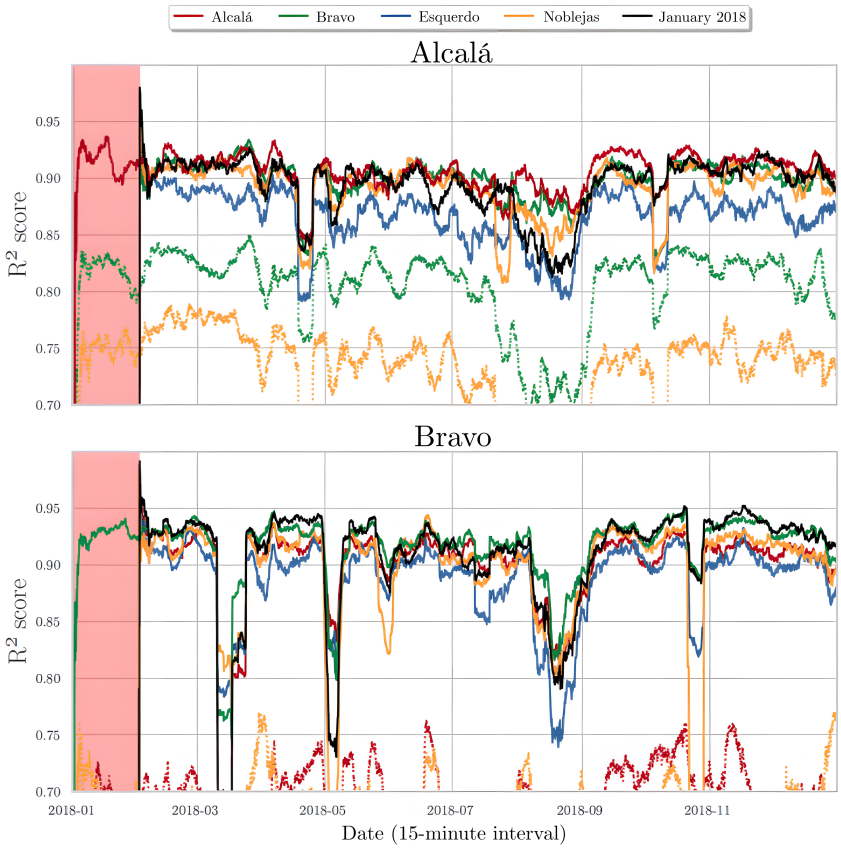



FIGURE 4.3: Part (1/2) of $R^2$ score evolution for offline settings.

directly at the release date with no updates, since data from target location is not available (i.e. `PS1`); 2) continuous lines pinpoint those transferred models that were also trained with data collected during January 2018 (i.e. first option for the `PS2`).

A red area denotes the waiting period before releasing the updated transferred model, which delays the implementation until February. To finish, the black line depicts a model trained only with data collected at the target road during the first month of 2018 (i.e. second option for the `PS2`). Again, the model implementation must be postponed one month.

As expected, the best performance case correspond to the `PS3`, where historical data at target location is available (for each subplot, the line that matches name with the target road segment). The model learns from different traffic flow events such as Easter or summer holidays, along with regular days, always from the target road. Therefore, this framework offers the most favorable conditions, positioning itself as a performance baseline.

Then, the transferred models are analyzed: those which are deployed at release date (`PS1`) and the ones which are also retrained using data from

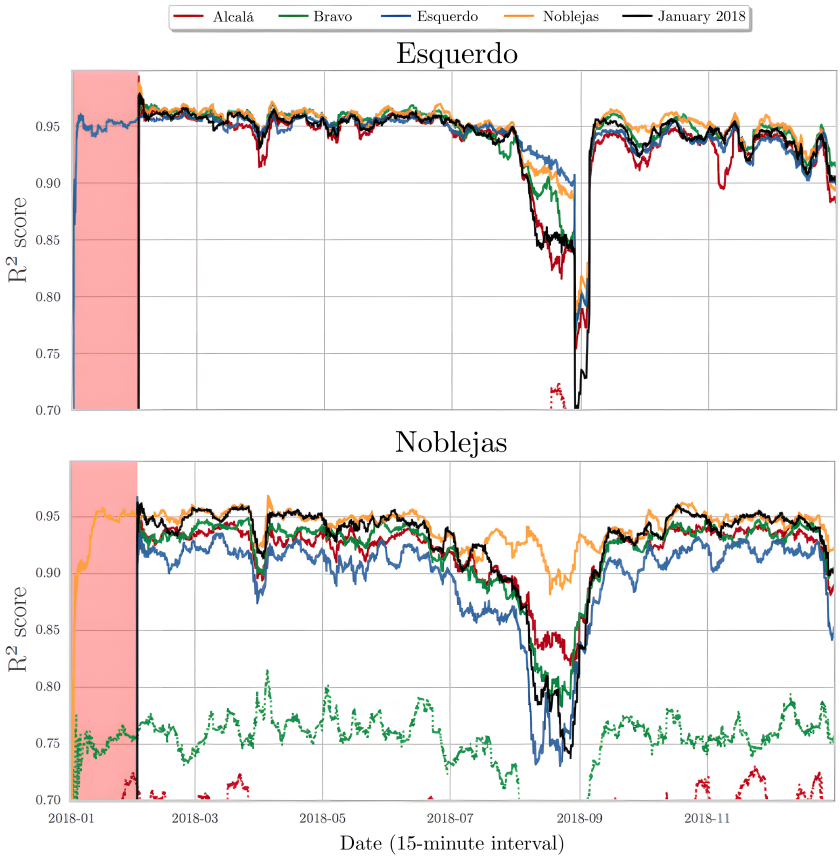



FIGURE 4.4: Part (2/2) of $R^2$ score evolution for offline settings.

January 2018 collected by temporal sensors at the target road. Focusing on dotted lines, it can be observed that different transferred models elicit distinct behaviors. For example, the transferred model based on Bravo Murillo works quite well for Alcalá and Noblejas datasets, bearing in mind that no updates were given to the model. However, this approach does not work in the opposite way, where Alcalá and Noblejas based models perform worse at Bravo Murillo road test. A feasible hypothesis is that during the 2017 year, Bravo Murillo road experienced some events that have had notable impact over the donor model, making it to be more prepared to forecast in the course of the test, where similar events can occur. On its part, transferred models trained with data from Esquerdo do not perform well over other datasets (it obtained negative results, so lines are out of the chart). If road traffic historical data is displayed (see Figure 4.7), traffic at Esquerdo exhibit larger car flow peaks when compared to other locations. The comparison between traffic measurements provide insights that help foreseeing how a donor model will behave when transferred. The abruptness of the spikes measured at Esquerdo explains why a transferred

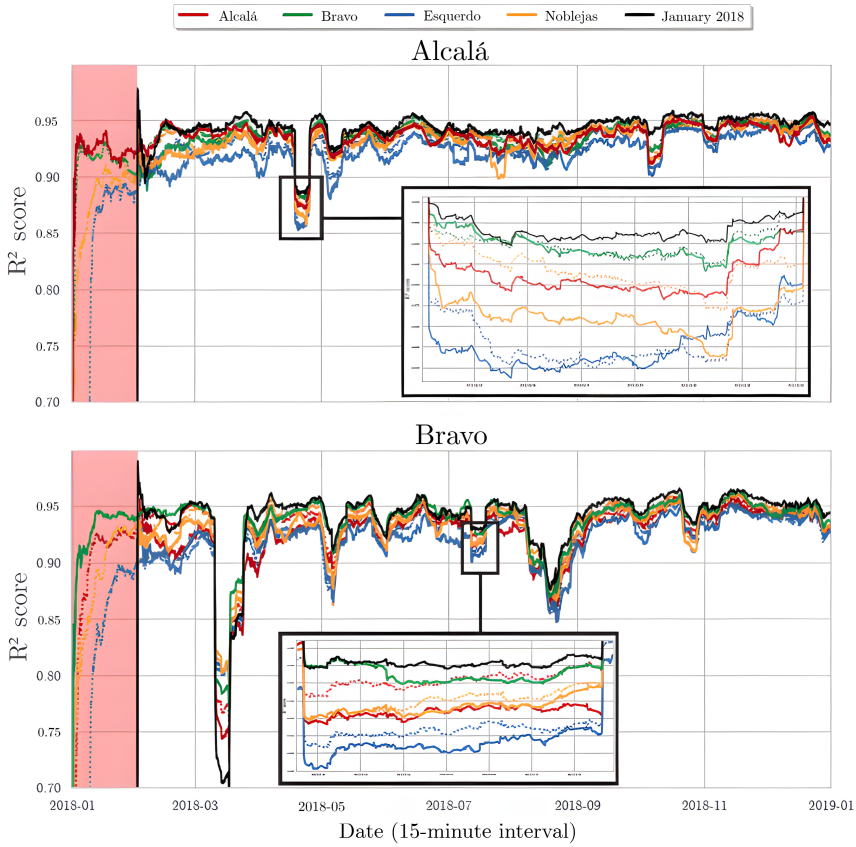



FIGURE 4.5: Part (1/2) of $R^2$ score evolution for online settings. Boxes show a close-up view of the obtained performance at certain areas.

model produces unuseful predictions (i.e. no blue dotted lines appear at Figure 4.3 or Figure 4.4).

Among the features presented at Table 4.1, only the number of lanes can provide a hint about the behavior a model can have, since no correlation between the road type and the traffic profile can be distilled from Figure 4.7. Alcalá is a primary road with two lanes with a similar traffic to Bravo Murillo: a secondary road with three lanes. For the Bravo Murillo road, having an extra lane could compensate being a part of the traffic network of less importance. However, Noblejas is a road that assembles both three lanes and being a primary road, but still manages to congregate similar traffic flows. Only the Esquerdo road exhibits a notable increase in the amount of vehicles, which could be justify by having four lanes.

Returning to the discussion over the performance of the transferred models that were trained with data from January, these models exhibit a predictive capabilities close to the baseline: after training over data at target location all of them improved greatly up to 90% $R^2$ score. Intuitively, the better their dotted counterparts are (i.e. `PS1`), the greater

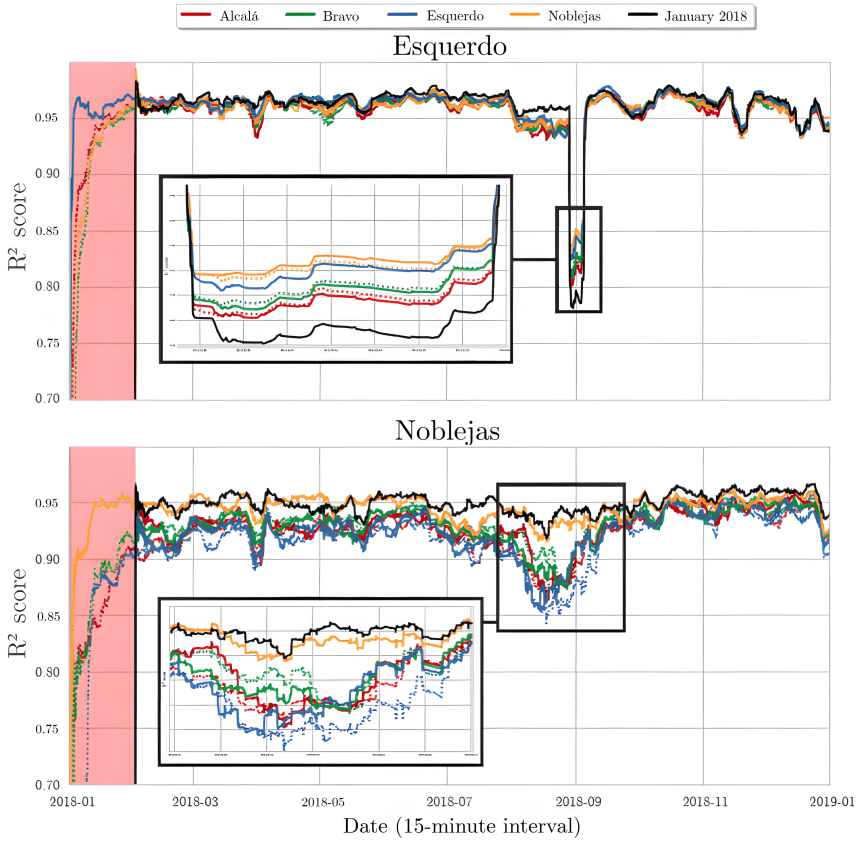



FIGURE 4.6: Part (2/2) of $R^2$ score evolution for online settings. Boxes show a close-up view of the obtained performance at certain areas.

performance is obtained due to the exclusive knowledge learned from other locations. However, if the black line is brought to the discussion, further insights emerge. During regular traffic periods, a model that only contain the knowledge encapsulated in the first month of 2018 performs slightly better, as model has only seen examples from target context. In contrast, when special events occur, like the last week of August, when people return from holidays, performance is contingent upon specific knowledge of each model. During the mentioned week, there is always a transferred model represented above the black line. With this, the importance of showing unique events to a model is highlighted. Only learning from such special events provides the necessary knowledge to deal with the arrival of other exceptional episodes.

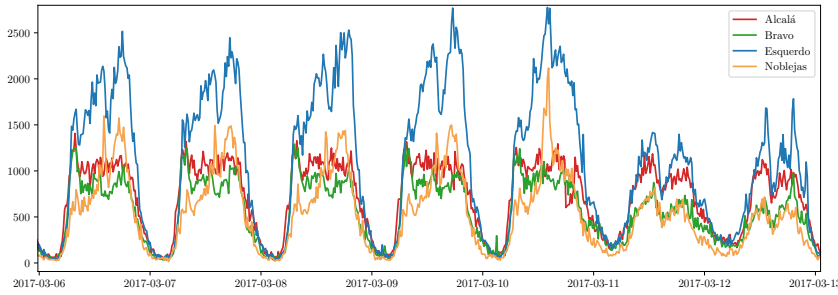



FIGURE 4.7: Road traffic flow comparison during a week of March 2017.

### 4.4.2 RQ4.2: Updating transferred models

Models can be updated via OL, only if a traffic sensor has been installed for collecting traffic data. Under the assumption that there are hardware resources capable of buffering an incoming data record and using it as an instance to update the model, predictions can be adapted to the prevailing traffic flow patterns. The performance of the models shown at Figure 4.3 and Figure 4.4 are now displayed for the online setting at Figure 4.5 and Figure 4.6.

As in the previous subsection, the discussion begins about the `PS3`, represented by those color lines that match the name of the test dataset. Even that results were already above 90% of $R^2$ score, predictions have further improve, with an increase up to 5 points for the target road of Alcalá.

The direct transfer strategy (i.e. implementing a donor model without adjusting its inner weights to $D_t$) followed in `PS1` has greatly benefited from the OL setting. Transferred models that were below 70% of $R^2$ now output predictions with a similar accuracy regarding their offline counterparts, catching up other models, and sometimes even performing better than their `PS2` equivalents (i.e. transferred models that were re-trained using temporal gathered data). The greatest benefit is that models can be implemented as of January $1^{st}$. During the first week of the year, no updates have been given to the model and the traffic patterns are very

specific, hence, low performance is expected. However, the remaining days of January have yielded results close to those provided the rest of the test holdout.

Models representing the PS2 have also seen their performance enhanced thanks to the assist provided by OL. Continuous lines are now more stable, which indicates an overall better prediction capability along the year. In detail, those models trained batch-wise only with data collected during January (represented by black lines) dominates the test benchmarks.

Overall, an increase of $R^2$ score has been achieved with respect to the offline setting but, what is more important, performance at special events, where offline setting struggles, has been greatly improved. The Easter and summer holidays are the periods where a model tend to perform worse. The close-up showcased by black boxes (Figure 4.5 and Figure 4.6) demonstrates that the reinforcement offered by OL procures more valuable knowledge for all models, letting them to actually presence additional special events, hence greatly improving performance. For the majority of main special events, is the model adjusted over January 2018 the one that dominates the benchmark. The only plausible explanation is that other models are overfitted to more common traffic profiles, due to the density of examples of this type in those training sets. On the other hand, models represented by black lines have been adjusted to a short but selective datasets that contains both seasonalities (e.g. Epiphany day) and working days.

## 4.5 Summary

A fast deployment of forecasting models is practicable thanks to both TL and OL. If a source domain is related to a target domain, a transferred model provide a better starting point than a random initialization of the inner weights. However, a small traffic data holdout from the target location is still needed, since the traffic patterns usually differ between points of the traffic network. In the presented analysis, the first month of the year encapsulates enough information for refining a transferred model to the point where predictions can be used in real world scenarios. Therefore, deploying a sensor during a month is enough for producing a capable model. If traffic is going to be monitored after the model implementation, traffic measurements can be fed into the model to further refine its operation. The constant updates gather not only examples from common traffic behaviors but also from special events. As a result, traffic flow predictions during such extraordinary periods translate into more accurate representations of the real traffic measurements.

Despite the above, the main insight distilled from the experimentation resonates with the expression «*quality over quantity*»: a small training dataset containing examples of both common and special traffic situations produces more capable models regarding extensive datasets comprised by a higher representation of working days. The traffic flow at a certain road segment can be represented by several data distributions, where each distribution represents different modes of operation (e.g. workdays, summer

holidays, etc). Following this idea, a model needs examples from every real distribution for learning such behaviors. If for instance, the traffic behavior during summer holidays is characteristic of that period, having a traffic sensor that collects such traffic data should be a priority towards developing effective forecasting models.

Experiments have been conducted over a secondary and a three primary roads. Neither the number of lanes or the road type could be related to the transferability of a donor model. Without this aid, selecting which donor model to transfer is not possible until the model is validated over the target road segment. Roads with similar traffic profiles need to be classified without comparing its traffic measurements, so no temporal sensors need to be deployed. A selection of topological features could serve as the criteria for producing such groups where ideally, multiple points of a traffic network have similar traffic profiles. This scenario can make donor models to be effective when transferred without the need of collecting data at target for model refinement. The above hypothesis is precisely the main motivation for conducting the experimentation of the following chapter: the sensorless characterization of road segments.

# Chapter 5

# Traffic Characterization in the Absence of Data

Top performing short-term traffic forecasting models use past traffic measurements from the target locations for computing predictions. Under this premise, installing a sensor at the road segment to be modeled is the only apparent choice to gather data from that location. However, in practice traffic data acquisition systems cannot be deployed over every link of a road, mainly due to the high costs of deployment and maintenance of the sensing equipment. On many occasions this issue has been addressed by deploying provisional sensors that provide measurements for certain locations of interest during a limited period of time. As seen in Chapter 4, a proper characterization of the traffic behavior under a variety of circumstances (e.g., events or holidays) requires real traffic measurements over more dilated periods. In summary, the most common scenario is that either it is not economically feasible to install sensors at all locations of interest or all target locations are characterized for a short period of time via provisional sensors.

If the goal is not to model traffic at a single road but the characterization of a traffic network, deploying sensors at all road segments might not be needed: only roads with particular traffic profiles need their traffic measurements to be collected. Assuming that the remaining sensorless locations share a similar traffic profile with one of the sensed roads, the real challenge is on how to establish links between instances from both spheres. Partly inspired by the findings of [345], in this chapter, a novel method that allows finding road segments that share a similar traffic behavior is presented. On this proposal, no traffic data is required, so the only information employed to make the selection has to be extracted from the design of the traffic network and other circumstantial information that extracted from its context. Under the concept of *road feature embedding*, a set of features that attempt to characterize each road segment is designed. These road feature embeddings encapsulate information about the network's topology, context of the area, and domain-specific knowledge about the behavior of traffic flows in similar urban areas. The main goal of this chapter is to *translate* this knowledge into a numeric vector (namely, a road feature embedding), so road segments of similar traffic profiles can

be found without comparing traffic measurements, but the aforementioned expert knowledge-driven features instead.

The major issue stemming from the above methodology is that a wrong association between roads might lead to an inaccurate characterization of the sensorless location. Intending to provide an alternative solution, further research is conducted shifting the goal to *when to sense*. The paradigm is reformulated by deploying a provisional sensor over the road whose traffic is to be characterized. A forecasting model is built by learning the relationship between the collected traffic measurements and the traffic of certain permanently sensed roads, which are chosen according to their road feature embedding. While obtained traffic predictions can be less accurate if compared to the forecasting models reported in the literature (see Chapter 2 for an state of the art overview and Chapter 3 for a comprehensive performance benchmark), the key design factor of the proposed framework is cost-efficiency. Non-sensed locations are now characterized by a model that has learned how traffic in one location relates to another one. Therefore, the new research question relies on how to reduce the time a provisional sensor needs to be installed for a proper road segment characterization.

In short, the presented hypothesis is that two roads with similar road feature embedding values should also share a comparable traffic profile. The feasibility of the proposed road feature embedding is firstly assessed by *characterizing traffic at sensorless locations without collecting any data*. Assuming that a temporary sensor can be deployed at the target location, the viablity of road feature embeddings for describing the traffic profile of a road segment is further evaluated by *estimating when to deploy a provisional sensor*, so the captured traffic data is sufficient for learning a model that can characterize the traffic flow at the target road segment. Due to the diverging nature of the task under evaluation, two experiments are conducted: **Case-A** and **Case-B**. Attending to the goal to be accomplished, they can be summarized as:

- **Case-A**: learning to model traffic without any data recordings.
- **Case-B**: learning to model traffic with data from a provisional sensor.

As a result of performing the above experiments, the following contributions are achieved:

1. Designing a set of features that describe road segments, based on their topological and contextual characteristics.
2. Analyzing if two locations of an urban traffic network with similar road feature embeddings do also exhibit a similar traffic behavior.
3. Comparing several generation methods for producing real-world like traffic samples at sensorless locations.
4. Developing a model that learns to relate traffic from different road segments, which enables to characterize the sensorless road segments.

5. Exploring how long a provisional sensor should be deployed, for maintaining the best predictive performance for the above model.

## 5.1 Sensorless characterization of traffic data

The sensorless characterization of road segments remains essentially unexplored [345]. Different topics concerning the latter are revisited, intending to provide a broad foundation before delving into the proposed solutions. The motivation for pursuing real-world synthetic traffic data instead of using simulation tools is discussed in Section 5.1.1. Published works that explore how to associate network design with traffic flow profile are commented in Section 5.1.2. The background that motivates the use of graph theory for representing road networks is introduced in Section 5.1.3. Finally, Section 5.1.4 comments on two novel works that propose to learn the relationship between traffic profiles of distinct locations.

### 5.1.1 Scarcity of urban traffic flow measurements

Traffic behavior is more complex to model in cities than at interurban roads, due to the multiple factors that can alter urban traffic (see Section 2.3.2). This fact, combined with the low number of public urban RCD datasets, clarifies why not so long ago, urban road traffic data was scarcely studied [10]. It is not until recent years when research papers on this context have seen its number increased [11]. However, the majority of urban traffic-focused manuscripts are conducted over taxi or bike FCD. The traffic flow of an entire urban network can not be predicted from these kinds of datasets. Since FCD is gathered from individuals, only a fraction of the total flow can be modeled. This is the reason why FCD is mainly used for speed or travel time forecasting (see Figure 2.3), where researchers can obtain the average speed in distinct segments of the traffic networks from the trajectory of just some vehicles.

The disposition of synthetic data can provide benefits to the field, by giving access to urban traffic data of similar characteristics regarding a target location. Synthetic data is rooted in the idea of producing artificial samples following a statistical distribution [346], which should be close to the real-world task to be modeled. Microscopic simulation tools, where the dynamics of each vehicle is modeled, such as SUMO [347], VISSIM [273], CORSIM [348], and MATSIM [349], provide complex scenarios that allow studying not only traffic congestion, but also protocols for traffic light switching, emissions, energy consumption and so on [350]. With such simulation tools, complex but still detailed flow traffic profiles for multiple road segments of a traffic network can be produced. Nevertheless, simulations mostly consider as influencing factors the weather, accidents reports or infrastructure change works. Other fundamentals aspects that concern the traffic behavior are not contemplated: 1) special events, road construction/restoration, etc., are difficult to include into simulations; 2) socio-cultural particularities of major communities also models the traffic behavior. Still, it is hard to embed this knowledge within the configuration

of simulation tools. The above appreciations serve as the motivation for characterizing urban roads not from simulations but from real-world traffic recordings.

A methodology for generating synthetic traffic samples for non-sensed locations is presented in [345] to cover the above issues. The authors exploit the knowledge that can be learned from neighboring sensed road segments. Deep Learning regression and GAN models are explored as data generation methods. It is a challenging task since there is not an explicit statistical distribution to be learned, due to the lack of real data at target locations. Likewise, traffic data exhibit a multi-modal nature, where not a single but multiple distributions or modes must be learned towards an accurate representation of data. In the context of traffic, these modes can be interpreted as the distribution followed by observed data under certain conditions (e.g., holidays and day of the week). Therefore, not only several distributions must be learned without the disposal of traffic data from the road segment of interest, but also the correct mode must be selected towards generating plausible synthetic data. This issue is addressed by using a set of conditions that groups traffic recordings according to the resemblance of their traffic patterns. Although the authors point out that mode selection can be improved by delving into the conditioning of generative systems, their main concern is to find a road segment with a similar traffic behavior. The generation systems will output divergent traffic patterns if the behavior of the selected roads is not close to the target location. Being exploratory research, a naïve criterion is adopted, where the closest ATRs available from the surroundings served as data sources. This approach entails speculation, where sometimes the selected data sources have similar traffic profiles, but in other cases traffic highly differs.

### 5.1.2 Associating network design and traffic profiles

The major challenge for a better design and usability of a traffic network is to understand the influence exerted by its structure. The topological design of a city map influences traffic behavior. Road occupation makes drivers choose distinct paths, pursuing a balance between selecting the shortest route and avoiding traffic congestion. In turn, simple street layouts facilitate for drivers to foresee other drivers' actions, so they can adapt their driving style without sudden braking or other similar abrupt movements. Nonetheless, with the inexorable expansion of the city population, the same applies to their infrastructure needs [351], where additional lanes and routes are appended to the existing layout, promoting traffic flow and hence, traffic congestion.

Some published works address this issue, seeking the relationship between traffic profiles and the design of the road itself. Wen et al. [352] elaborate on the idea that numerous turning directions at crossroads can result in a hindrance for fluent traffic flow. By using collected data from the city of Taipei, Taiwan, they manage to identify the most congested segments of the metropolis (i.e. business districts and industrial areas). On the same line, Wang et al. [353] conducted a novel study in the city of

Shenyang, China, about the relationship between traffic and the proximity of certain POIs (i.e. bus stations, schools, and hospitals). They could only demonstrate a correlation of traffic flow with the number of lanes. Despite these results, authors warned about the confidence of the presented conclusions, calling for more case studies over other cities towards a better understanding of the considered relationships. A whole research line stems from Geroliminis and Daganzo [354], where the first Macroscopic Fundamental Diagram (MFD) was obtained from a real environment (i.e. the city of Yokohama, Japan). The introduced MFD provides a description of the ideal network dynamic performance, drawing a curve that defines the critical point where an increase in vehicle density leads to a decrease in vehicle flow. As classified by Ambühl et al. [355], in later years distinct versions of the MFD have come out, but for the substance of this manuscript, the most relevant is what they define as theoretical upper bound MFD (uMFD). Just from the network topology and the traffic control policies, it is possible to simulate the uMFD, towards a better understanding of the theoretical traffic flow ceiling or, in other words, to foresee which is the maximum capacity of the network during an ideal driver behavior. In this line, several road network designs can be studied for selecting the one with higher capacity. Laval and Castrillón [356] proposed and stochastic approximation for computing the uMFD without the need for traffic flow measurements, just only from the block length and the traffic lights timings. A comparison with the empirically obtained uMFD from the city of Yokohama validates the method. Still, this technique does not have into consideration the particularities of other specific transport means (e.g., bus fleets). The latest advances in this investigation line are covered at the introduction of [357] and [358], in the case of further reading needs.

### 5.1.3 Graph representations for traffic forecasting

As a subset of Machine Learning, Deep Learning [24] has gained momentum in recent years, thanks to the impressive results obtained in several fields such as natural language processing [359] or computer vision [360]. The excellent modeling capabilities of Deep Learning architectures expanded the competence of multiple research fields, though these improvements were temporally constrained to Euclidean data (i.e. data that can be expressed in a $m$-dimensional Euclidean space) [361].

The superior modeling capabilities of Deep Learning made authors concentrate their research efforts on solving traffic forecasting problems with this family of learning models. Different Deep Learning architectures have been proposed over the last years: from basic convolutional and recurrent networks to more complex and deep architectures such as those based on the attention mechanism [292] (a more comprehensive analysis of related efforts to date is provided in Section 3.1). State-of-the-art architectures from other fields were applied to traffic forecasting, expecting similar levels of predictive performance. However, as demonstrated in Chapter 3, Deep Learning architectures do not necessarily outperform conventional Machine Learning methods (e.g. ensemble learning). The traffic state at

previous instants $\{t-n,\ldots,t\}$ contains enough valuable information given the high persistence of the traffic time series to be predicted. Even a naïve model where the latest recorded traffic state value serves as the predicted traffic value can be shown to yield a sufficiently good forecast for close time intervals. In summary, the feature learning capability of Deep Learning models is not differential for this specific application scenario, and poses further problems such as the interpretability of the knowledge captured by such black-box models once they have learned to forecast traffic [21].

It was not until the upsurge of *geometric Deep Learning* [236] when multiple techniques arose towards developing new applications concerning non-Euclidean data: those tasks where data can be described as a graph [362] or as a manifold [363]. As for the subject of this thesis, traffic networks can be easily defined as graphs, where the road segments are the edges, and the crossroads are the nodes (although some authors prefer the opposite representation). This way, an abstraction of the traffic network can be defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ is a set of $N$ nodes representing the junction of road segments, $\mathcal{E}$ is a set of edges symbolizing those road segments, and $\mathbf{A}$ is a binary adjacency matrix in which element $a_{i,j}$ represents the *reachability* between nodes $i$ and $j$ of the traffic network: node $j$ can be reached from node $i$ if $a_{i,j} = 1$.

Bearing the above definitions in mind, graph neural networks (GNNs) are a type of Deep Learning architecture that is specifically designed to exploit the information represented by graph data structures [364]. From the development of novel GNNs, researchers achieved to apply this technology towards traffic forecasting [12], as the traffic state (e.g., flow or mean speed) can be easily encapsulated as node features. This way, each node of the graph has a collection of features, composed by its traffic state of previous time instants. According to the reachability defined by $\mathbf{A}$, the GNN can perform operations with not only the traffic features of one node but also merging the information of its neighbor nodes.

Given the ability of graph representations to condense the characteristics of a road network into a non-Euclidean space, the surroundings of a road segment should be able to be expressed as a collection of graph-based features that conforms to intuition and domain knowledge about behavioral patterns of traffic flows in urban networks with different topologies. For this reason, the road segment similarity finding method proposed in Section 5.3.2 is partially based on features distilled from a graph representation which is intended to portray the topology and context of the neighboring.

### 5.1.4 Learning the relationship between roads

If a provisional sensor collects traffic data for a large period, a data-based traffic model can be developed. One option is to build the model in a long-term fashion, where traffic patterns are computed from historic records. Theoretically, all existing profiles can be addressed from a whole year of traffic data, which is the minimum recommended period for optimal results. The work of [344] introduces a long-term estimation framework

based on clustering. The training holdout is split into daily records and then grouped into several clusters according to their flow values. Then a classifier is trained to learn the correlation between the representative pattern of the cluster and associated calendar features (e.g. day of the week and holidays). An alternative approach is presented in [365] where authors have developed a long-term traffic prediction model based on long short-term memory cells. A hard attention mechanism ensures that traffic patterns are learned and not forgotten. These traffic patterns improve the predictive performance on future dates according to the periodicity of the traffic flow. Still, the system needs a constant supply of traffic observations, so further modifications should be made to its structure for computing traffic estimations from an static data holdout.

The case study `B` presented in Section 5.5 occurs on the same context: a forecasting model is built from a traffic holdout collected by a provisional sensor. However, a short-term approach is employed for modeling traffic. The target data collected by the provisional sensor enable an algorithm to learn the relationship between such traffic and the one from other permanently monitored roads. The road feature embedding criteria described in Section 5.3.2 assist the major design problem about how to select which sensed locations to use as input predictors to the traffic model. The other main concern, namely, when and for how long to deploy a provisional sensor at the target location, can be tackled as an optimization problem and solved efficiently via evolutionary heuristics. This problem aims at reducing the associated costs of the sensor maintenance while still maintaining a high predictive performance.

## 5.2 Description of the case study

It can be inferred from the literature review that a comparison between two roads of a traffic network can be performed by contrasting their topological features. Likewise, graphs have been proven to be a competent traffic network representation in previous literature. Depart from the findings and research directions reported in [345], which emphasized that in order to improve the performance of generative methods for traffic data, a better criterion for selecting similar road segments should be conceived. The selection of the right traffic dataset originates another dilemma, since synthetic data produced by a generative model could be no more accurate than taking as prediction the traffic record of the desired date from the selected dataset. Aiming to condensate these concerns, several Research Questions (RQ) are formulated, where the following are explored by performing the Case-A:

- RQ5.1: Can two road segments with similar traffic profiles be identified from a set of topological features, without the need of a sensor?
- RQ5.2: Is there any relationship between road feature embedding similarity and the performance of the presented selection method?

- RQ5.3: Which is the best approach for generating synthetic data in a sensorless road segment?

If provisional sensors are available, an alternative modeling technique is proposed, where a model learns to associate the traffic from the target location to the one collected by the ATR at the selected sensed road. The traffic collected by the provisional sensor directly impacts the performance of the proposed model, since the model will learn correlations from such a limited data holdout. Hence, the key question shifts now to *when to sense*. New Research Questions are formulated, but this time the hypotheses are evaluated by conducting the Case-B:

- RQ5.4: When should a provisional sensor be installed for developing a model that learns the relationship between traffic of two roads?
- RQ5.5: Do the obtained sensing masks generalize well so the target road segment can be properly characterized?
- RQ5.6: Should reference roads be similar to the target location to produce reliable traffic estimations?

## 5.3 Materials and methods

The materials and methods employed in both Case-A and Case-B are introduced hereafter. Section 5.3.1 describes the traffic data used to simulate a real-world scenario and illustrates how to generate a graph representation of a traffic map. The components of the proposed road feature embedding are defined in Section 5.3.2, and the metric that measures similar road feature embeddings is detailed in Section 5.3.3. Section 5.3.4 describes the concept of traffic profile, which characterizes the traffic flow of a road segment and enables comparing different locations within the traffic network. The considered approaches for generating synthetic traffic data are described in Section 5.3.5. How to learn the relationship between traffic of several roads is explained in Section 5.3.6. Finally, an optimization process for estimating the best timing for deploying a provisional sensor is defined in Section 5.3.7.

### 5.3.1 Traffic data and graph representation

Several years of traffic records from the city of Madrid (Spain) have been retrieved from the public repository available at [321]. The ATRs installed at multiple road segments of the city yield high-quality traffic measurements in the form of flow, speed, and occupation. The time span of the selected datasets for validating the design of the road feature embeddings covers the 2018 and 2019 years. Four consecutive years (i.e. 2016 to 2019) of flow observations are utilized in the experimental setup of Section 5.5: the first two years are used for selecting the optimal sensing mask (see Figure 5.4), while the remaining years simulate a real implementation where traffic samples are collected from the target road according to the sensing mask, and the whole next year is used for evaluating the model's test

performance. In a real implementation, the model could be applied right after the provisional sensor is removed from the target location.

The location of the ATRs is displayed in Figure 5.1. Here, it can be observed that all considered locations correspond to secondary, tertiary, or residential roads. Motorways and primary roads are excluded on purpose, as they are meant to be the infrastructure for in/out city journeys. The traffic behavior on these driveways is expected to be differentiated regarding the traffic flow of the lower-rank roads contemplated for this investigation.

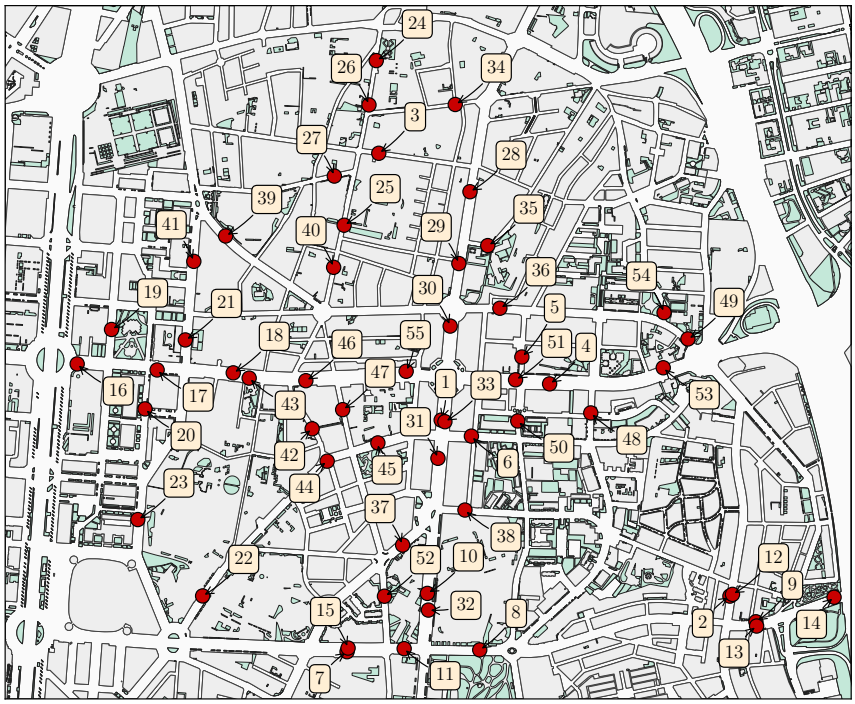



FIGURE 5.1: Location of the deployed sensors in the neighborhood of Chamartín in Madrid, Spain. Several road types are considered. Namely: secondary, tertiary, and residential roads, along with road links between distinct road types.

The traffic network graph representation $\mathcal{G}$ is computed via the OSMnx Python package [366], which allows obtaining geospatial data from OpenStreetMap [367]. From the coordinates of each ATR, a directed graph is built within a 2km radius. Such distance has been established after a systematic search but must be long enough so every influential street is captured. This way, the road segments conform the graph edges $\mathcal{E}$, whereas the nodes $\mathcal{V}$ represent the crossroads. The road length, maximum speed, and travel time of the road (computed from the previous characteristics) are assigned as edge attributes for $\mathcal{E}$. Since the graph is obtained from the coordinates of an ATR, and this one is placed in a road segment, the focal point of the graph would be an edge. However, for the sake of an easier graph processing, the ATR should be placed as the central node of

the network. To this end, the corresponding edge is split into two smaller segments which are, at the same time, connected to an artificial central node. In the following, the artificial node is considered as the focal point of the graph, which shares the coordinates of the ATR, and is ultimately referred to as *central node*.

### 5.3.2 Road feature embedding

A road feature embedding is designed to compare street segments. Some of the features that conform the embedding are based on a graph representation. However, the graph representation $\mathcal{G}$ is too wide to represent only the particularities of the target road. Therefore, an ego-graph $e\mathcal{G}$ [368] is obtained by pruning all nodes in the original graph $\mathcal{G}$, that need more than $N$ hops to be reached from the central node. The parameter $N$ regulates the relevance of the surroundings during road segment comparisons. Low values for $N$ will make the system to focus on the topological aspects of the target street, while high $N$ values can make ego-graphs similar to each other, only because distant road segments regarding the central node are overlaid. This last case neglects the particular traits of the target road, which can produce non desirable resemblances. With these concerns in mind and after a grid search, a value of $N = 5$ is set for this investigation. This parameter can be tuned according to the traffic map and feedback delivered by experts in traffic management with experience in the urban area under study. Complex traffic network designs might need higher values of $N$, since road segments are usually shorter, and hence small surfaces can be well represented by $e\mathcal{G}$. Conversely, the characterization of long road segments might require low $N$ values to avoid the overlap of similar graph representations.

From the ego-graph centered in the target location, a set of features is arranged. The NetworkX Python package [369] provides the methods used to compute them. A first subset of such features is meant to express the *centrality* of the central node and its neighbor nodes. As its name suggests, these metrics are intended to quantify the prominence of an agent (node or edge) in a network [370]. This concept is motivated from the data networks domain, which often shares properties with traffic networks [371]. Among the existing centrality measures available in the state-of-the-art [372], the *shortest path betweenness centrality* ($SPBC$) has been selected, which accounts for the count of all pairs shortest paths that pass through a node $v$. Mathematically:

$$SPBC(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}, \tag{5.1}$$

where $\mathcal{V}$ is the set of considered nodes (neighbors of $v$), $\sigma(s,t)$ is the number of shortest $(s,t)$ paths, and $\sigma(s,t|v)$ is the total of those paths passing through node $v$, as defined in [373].

Since in a graph representation crossroads serve as the edge split criterion, the streets represented in $\mathcal{E}$ are usually divided into multiple edges.

Some long road segments are partitioned several times, while others with a reduced number of crossroads are represented by fewer, yet longer road segments. Upon this principle, the shortest path is computed by comparing the travel time assuming free-flow speed, instead of the number of hops (which is the default metric). From Expression 5.1, the following metrics are computed:

1. *SPBC of the central node*: Higher values should correlate with high flow profiles.
2. *Maximum SPBC among neighbors*: It searches for other more essential nodes in $e\mathcal{G}$, regarding the central node.
3. *Median SPBC among neighbors*; It helps figure out the distribution of centrality, among nodes in $e\mathcal{G}$.

Other centrality measures have been considered but excluded, as they do not reflect well the traffic profile of a road segment, or more precisely, it was intended to get rid of those features that could be ambiguous (road similarity or disparity could be argued with the same value). It is the case of *degree*, which in the context of traffic networks is the number of links a crossroad has. Multiple connections to other road segments do not directly imply that a crossroad is heavily used, as many of these links can be scarcely frequented streets. In turn, a low-degree arterial road can be an essential milestone for numerous paths that go across the area. The *closeness* of a node is the distance to all other connected nodes in the graph, or in other words, it measures how long it will take to spread information (or vehicles in this context), from the central node to all other nodes sequentially. While closeness is oriented to measure the broadcast capabilities of the nodes, it is a centrality measure focused on information networks. In contrast to data packages, vehicles are treated as individuals with no capacity for duplication. Instead of granting importance to the reachability of all other nodes, only certain points of the network should be interesting to be close to, as drivers are prone to searching for higher-rank road segments (e.g., primary roads). Departing from this idea, new embedding features are defined further on.

The neighborhood of Chamartín is enclosed by a primary road at the west, and by a motorway at the east (see Figure 5.1). In a large city like Madrid, the population usually travels in/out of the district through the fastest route, which generally compromises high-rank roads. The travel time towards reaching the closest primary road and/or motorway can elicit insightful features. This concept can be expressed as:

$$TT(v, type) = \min \Delta(v, e_{type}) \forall e_{type} \in \mathcal{E}, \tag{5.2}$$

where $TT(v, type)$ is the expected travel time for a vehicle in node $v$, towards reaching the closest road of certain rank $type$ (in terms of travel time), and $\Delta(v, e_{type})$ is the expected travel time from $v$ to the closest node of a street segment $e$ of a certain rank $type$ belonging to a set of edges $\mathcal{E}$.

This time, instead of using the ego-graph, the overall graph $\mathcal{G}$ is employed, as high rank roads could not fall into $e\mathcal{G}$. From Equation 5.2 the following features are obtained:

4. *Travel time to the nearest motorway*: Motorways can be oriented to trips out of town.

5. *Travel time to the nearest primary road*: Primary roads can be oriented to trips to the city center.

During rush hours, flow spikes can be narrow or wider depending on the overall flow. In the case of dormitory towns, most drivers go out of the city in the morning and return home in the evening, while major cities oppositely receive traffic. These previous features are tailored towards characterizing the events that impact traffic profile during rush hours.

Additionally to the features obtained from analyzing the graph, two extra features based on road characteristics are included for characterizing the road segments: the *road type* and the *number of lanes*. Each road type is encoded as an ascending scale where zero represents residential roads and the unit value corresponds to a motorway. Intermediate road types (namely tertiary, secondary, and primary) are encoded as in-between numbers, ensuring that all values are uniformly distributed. In contrast, the number of lanes is directly expressed by its original value. However, it is worth mentioning that although the NetworkX Python package provides methods for obtaining this road feature, there are some aggregation errors (i.e. merging the lanes of both directions), and missing values (generally in residential and tertiary single lane roads). Under this premise, each of the ATRs considered for this investigation has been manually inspected via Google Street View, to verify and rewrite the number of lanes if necessary. In those cases where the number of lanes varies, the predominant number of lanes across the entire length of the road segment is selected.

The above topological features are meant to give some notion about the traffic behavior in the studied road segments. Roads are classified in classes or ranks towards denoting their importance within the road network as a whole. Their structure, capacity, speed limits and even lane width varies between high and lower-rank roads. The number of lanes is just one specific parameter that defines the road structure. Nonetheless, it can produce more insights about the traffic flow regarding other road design features, such as the speed limit, which in the context of urban networks is most of the time equal or below 50 kph. Still, even if road networks are designed bearing in mind the expected traffic demand, real daily traffic profiles might not correlate with the expectations [374]. Bearing this in mind, the last features that conform to the road feature embedding are defined as:

6. *Road type*: Higher rank road types are expected to portray higher daily traffic flow profiles.

7. *Number of lanes*: Directly defines the road capacity and, henceforth the contemplated traffic flow.

For the sake of a better understanding of the concepts beneath each topological and contextual feature, Figure 5.2 illustrates how such features are extracted from a road segment.

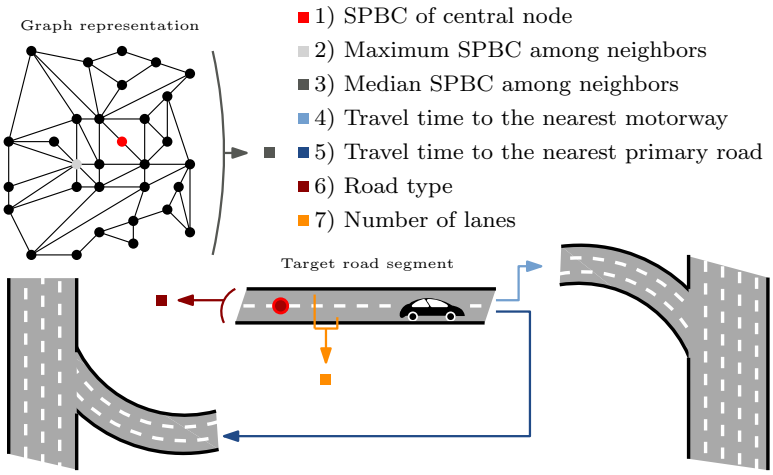



FIGURE 5.2: Graphical representation of the topological and contextual factors that conform the road feature embedding.

While some features (i.e. those based on centrality, road type, and the number of lanes) can be useful for any target area, features 4) and 5) are developed ad-hoc based on discussions held with experts in traffic analysis. The area considered in the case study (see Figure 5.1) is surrounded by a motorway and a primary road. These two arterials can be regarded as the major in- and out- traffic gates flowing into and out of the district, so their proximity to them (in terms of travel time) might be an interesting trait for characterizing road segments.

### 5.3.3 How to select a sensed road

Road feature embeddings are intended to portray the characteristics of the road segments. Therefore, two locations with a close traffic profile, should share similar values for the features that compose their respective embeddings. After a feature normalization, road feature embeddings can be represented as single points in a N-dimensional space. From this, by selecting the road feature embedding of the target road as origin, the Euclidean distance to any of the other road feature embeddings can be computed [375]:

$$D(\mathbf{u}, \mathbf{v}) = \sum_{n=1}^{N} \sqrt{|(u_n - v_n)|^2}, \tag{5.3}$$

where $D(\mathbf{u}, \mathbf{v})$ is the Euclidean distance between two N-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$, whereas $u_n$ and $v_n$ are the features for the corresponding vector in position $n \in \{1, \ldots, N\}$. Since similar road feature embeddings should likewise have (according to the presented hypothesis) close values of their

feature values, the minimal Euclidean distance among all available sensed road segments can be found.

After selecting one road via the road feature embedding method, a second location is selected according to the coordinates of the ATRs and the target location. This way, the performance of both geographical (i.e. baseline from [345]) and road feature embedding selection methods can be analyzed.

### 5.3.4 Traffic profiles and selection performance

The main objective is to present a technique that allows finding similar roads in terms of traffic profile, without the need to compare the real traffic recordings. However, for the sake of evaluating the performance of the road selection, the traffic profiles of both sensed and target roads must be set side by side and compared. Every road segment receives interactions from the surroundings, which uniquely condition its traffic profile, making perfect matches between target and selected locations unattainable. Still, as this technique is intended to provide an alternative real traffic data source drawn from an analogous road for the target location, it is mandatory to pursue suchlike traffic profiles.

A traffic profile is a flow pattern that expresses the traffic behavior at a certain location of the traffic network. At a road segment, traffic flow varies with regard to the time of the day, the day of the week, holidays, and so on [329], making it arduous to condense all the traffic information in a single traffic pattern. Even so, a traffic profile must be computed in order to characterize distinct points of the city. Daily traffic flow metrics are more likely to be similar for two road segments that possess analogous traffic patterns and vice-versa. Henceforth, the comparison of traffic profiles for each of the 55 considered road segments is employed as a performance gauge for the proposed system.

Traffic profiles are computed from the median traffic flow. Since traffic highly differs between daily hours, the traffic flow for a certain timestamp is computed as the median of all recordings at that specific time. The median operator filters extreme flow values that can occur during special events such as Christmas, the beginning and return of the summer holidays, etc. The mean, on the other hand, would produce unrealistic traffic profiles, triggered by the aforementioned outliers. When it comes to filtering atypical values, only weekdays are considered for computing the traffic pattern, leaving out the traffic recordings from weekends. Traffic profile at weekends has fewer particularities than during weekdays, where usually the traffic profile is more restrained, as can be seen in Figure 5.3. In addition to this, the weekends-only traffic patterns usually share a similar shape, to the point that with a proper scaling factor, the majority of these flow patterns could be obtained. By comparison, weekdays-only traffic patterns are more insightful towards characterizing distinct points of a traffic network, as they draw silhouettes that differ on the width and location of spikes. Additionally, Figure 5.3 also showcases that the standard

deviation for weekends, obtained from all the flow recordings for a certain timestamp, is more uniform across the daytime. Putting all together, by excluding weekends in the calculation, more distinctive traffic patterns would be obtained. Therefore, the traffic profile is obtained exclusively from weekdays as:

$$\mathbf{y}_t^i = median(\mathbf{o}_t^{i,d}) \ \forall d \in \mathcal{D} : \ d \text{ is weekday}, \tag{5.4}$$

where $\mathbf{y}_t^i$ is the traffic profile at a timestamp $t$ for a certain ATR $i$ whose traffic metrics are contained in dataset $\mathcal{D}$, $median(\cdot)$ denotes median statistic, and $\mathbf{o}_t^{i,d}$ denote the traffic flow observable during day $d \in \mathcal{D}$ at a timestamp $t$.

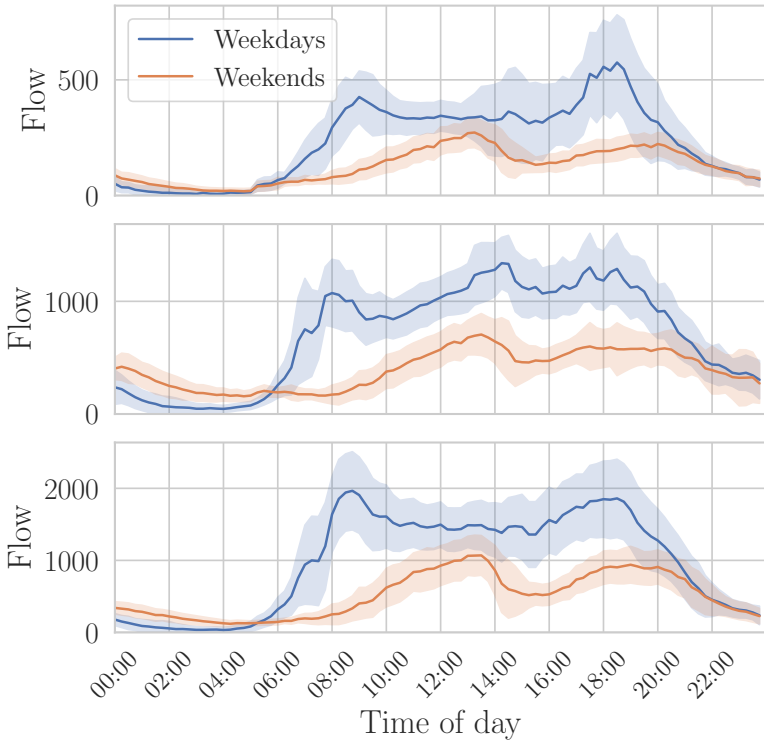



FIGURE 5.3: Traffic profiles computed from weekdays or weekends only, for three of the considered road segments. The continuous line represents the median flow value for each timestamp during a two-year period. Standard deviation is also displayed for each timestamp.

### 5.3.5 Approaches for generating synthetic samples

From the traffic data of the selected sensed road segment, synthetic samples can be generated for the target location. In the experimental framework of [345], the authors select the two closest locations with an ATR installed for generating data at the target location. This criterion does not ensure

similar traffic profiles at the selected points of the traffic network, so a GAN is employed towards mixing the information of such locations. This way, a more stable solution is sought. On the contrary, the road feature embedding approach is designed to identify road segments with expected comparable traffic patterns. Under this premise, the first candidate to be chosen should be a more accurate substitute for the target location, regarding second and third proposals. Therefore, only one sensed road is employed for generating data.

Within the context of this work, GAN-based solutions could not be the best approach. The GAN-based approach presented in [345] loses its purpose when a single traffic dataset conforms the data source, as there is not information to be combined. Several data generation approaches are suggested in this section, which are further analyzed considering their complexity (i.e. computational effort), and prediction performance.

Traffic data usually follows several distributions or modes. One distribution could define how traffic behaves on workdays, whereas other distribution can model traffic under extreme weather conditions (e.g., heavy rain or snowfall). Thus, characterizing properly all modes of a sensed road segment is a complex task, which deserves its own research line [255]. Nevertheless, when generating traffic data, it is critical to select the correct distribution from which to compute the synthetic sample. Without identifying data distributions, a generation query could be answered with any plausible traffic profile from the target road, which will provoke inaccurate estimations.

Operation modes have been reduced to 14, as per defined in [345], obtained from the combination of weekdays and holidays. Since this criterion is merely based on calendar features, data can be labeled without the need of flow-based features. Every generation approach is adjusted using as training data the selected dataset from the road feature embedding system. The performance of each generative approach is then analyzed by employing the traffic recordings of the target location as test data. A traffic pattern is generated for all days in the test dataset (giving a total of 730 days). Henceforth, for each of the 55 considered target locations, 730 error metrics are computed, by comparing the real traffic with the generated traffic pattern.

The first generation approach revolves around training a conditional GAN [376], conditioned with the 14 classes defined above. The goal is not to produce a top performance generation method, but to analyze distinct generation approaches. Under this premise, the *approach* Ⓐ proposed in [345] has already showcased a prominent performance for this task, so it is selected as the generation method. A ReLU layer is added at the end of the generator model, so no negative traffic flow can be generated. The generative model receives as input the class of the desired day to be generated and a noise vector. Conceptually, this solution is based under the concept of learning several data distributions and taking as output a random sample of the selected mode.

The second generation approach inherits concepts from the long-term estimation framework presented in [344]. Here, traffic data is grouped

by flow similarity into clusters. Each cluster disposes of a Representative Traffic Pattern (RTP), computed as the median of all the traffic samples within the cluster. From the value of certain calendar features, incoming days to be predicted are assigned to a cluster, and the aforementioned RTP serve as the traffic pattern that is going to be used as generated data. The framework is modified, so traffic data is grouped according to the aforementioned 14 classes. Given a date, that particular day is associated to one cluster. The RTP of the designated cluster is taken as the generated traffic pattern.

The last and third generation approach is the simplest and more easily interpretable by a user without any background on Machine Learning. If the target and selected road segments are considered similar, the traffic flow at a specific date should be similar in both locations. Under this premise, a Naïve Similarity-based Estimation (NSE) method is proposed, where the generated traffic samples for the target location are taken from the real traffic measurements at the selected road segment, for the same date. This approach is constrained to the disposal of traffic data for the date to be predicted. Hence, no traffic data can be generated for future dates, denying the prospective use of the system.

### 5.3.6 Association model and reference roads

Assuming that traffic measurements from the target road can be collected using a provisional sensor, a novel and intuitively more stable approach for generating traffic samples can be produced. The proposed traffic forecasting model is designed to predict traffic at a target location by using data collected from several permanently sensed road segments. Therefore, the model learns the relationship between the traffic of the reference roads and the target location. Data from three reference roads are used to feed the model intending to provide a more stable input to the system (regarding a single reference road). The model receives as inputs the previous traffic state measurements collected at times $\{t-4,\ldots,t\}$ at the reference roads, towards predicting the traffic flow at $t+1$ when queried at time $t$. An ETR model [377] is selected due to his outstanding performance exhibited in Chapter 3.

At this point it is important to bear in mind that the criteria for selecting reference roads might certainly affect the prediction performance. Since the proposed model learns how traffic relates between reference and target roads, a high correlation between such traffic observations should ease the computation of plausible traffic forecasts. However, before deploying a provisional sensor at the target location, reference roads must be selected, as their traffic data are used for computing the period of the traffic surveillance at the target. The road feature embeddings of all available sensed locations are compared to the road feature embedding of the target road. The most similar ones according to their road feature embeddings will serve as the reference road.

### 5.3.7 Sensor deployment optimization

The cost of traffic surveillance relates directly to the time a sensor is deployed [378]. Moreover, if the deployment time is reduced, the same sensor can collect data from multiple locations throughout the year. Motivated by the above, a sensor deployment optimization problem is formulated, whose goal is to minimize two a priori conflicting objectives: 1) to reduce the time a sensor is deployed; and 2) to minimize the forecasting error of the model that is trained over the data collected by the sensor. This problem represents a real-world implementation where traffic is monitored at the target location during a period of time, while still aiming for high-quality forecasts.

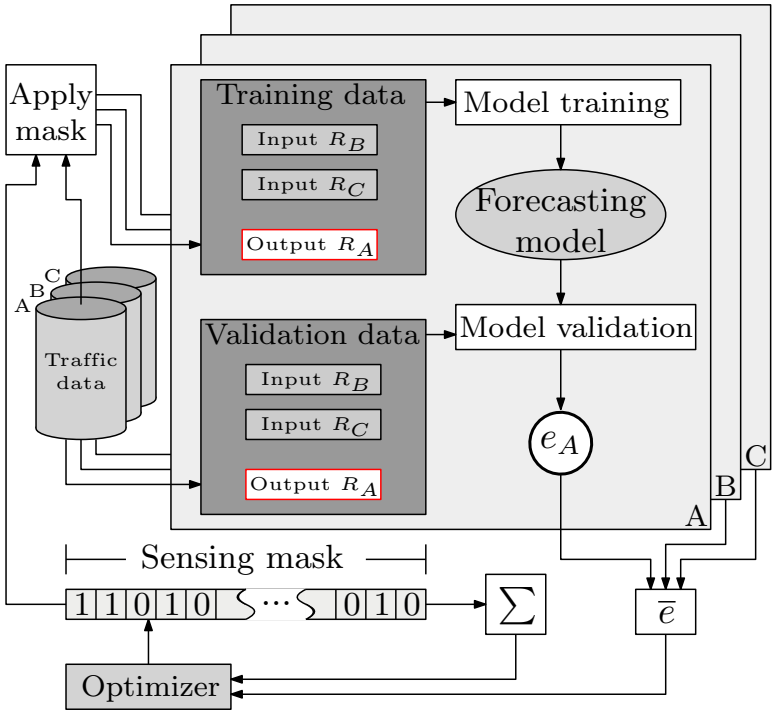



FIGURE 5.4: Sensor deployment optimization process. Each reference road $R$ is denoted by an alphabetic identifier $X$ (e.g. A, B and C). A forecasting model targeting one reference road is learned from a training dataset that simulates the traffic data that would be collected according to the sensing mask. Its performance is computed over a validation holdout, producing an error metric $e_X$. The process is repeated by targeting the remaining reference roads. Finally, the optimizer searches for the sensing mask that minimizes both the number of sensed weeks and the mean prediction error.

Figure 5.4 depicts the design of the sensor deployment optimization framework. It is to be expected that as the sensing time decreases, the performance of the predictive model will also decline. With such opposed

objectives a meta-heuristic multi-objective algorithm is proposed for optimizing both goals. In detail, the algorithm should be designed for computing the Pareto trade-off between traffic forecasting performance and the time a sensor is deployed. For each evaluation of the algorithm, a *sensing mask* is randomly generated. The sensing mask is a binary array that selects the dates within a year for the target sensor to be deployed. In detail, the year is split into weeks, so the minimum amount of time a sensor could be deployed is one week. This minimum measurement duration is motivated by lower periods (e.g., one day) not being feasible from a real implementation perspective.

Since no target traffic data is available, traffic records from the reference roads are used to validate the performance of any candidate sensing mask produced by the solver. This way, weeks during the first year of traffic data are selected according to the sensing mask. Data of these chosen weeks are then used to estimate the performance generalization of a forecasting model that correlates the traffic of one reference road with the traffic recordings of the other two reference roads. The performance of the forecasting model is validated with another data holdout, which is comprised of a whole year of traffic measurements. The process is executed three times, so that every reference road serves as the target location in a sort of *leave-one-reference-road-out* cross-validation. The mean average validation error and the number of days the sensor should be deployed are fed to the optimizer, which searches for those combinations that best balance (in the Pareto sense) between both objectives. As a result, the optimization algorithm produces multiple sensing masks, each achieving a different share between the predictive performance of the model and its cost as per the number of weeks for the deployment.

## 5.4 Experiments and results: Case-A

Figure 5.5 represents the workflow of this experiment, which is designed to provide an overview of all the processes to be explained. First, available real-world traffic data is obtained from the data source at hand, along with the coordinates of every sensed and non-sensed road segment (Section 5.3.1). These coordinates are used to compute a set of features which are intended to portray the context and topology of such locations: a road feature embedding (Section 5.3.2). By comparing these embeddings, a sensed road is selected (Section 5.3.3). Similarly, the nearest sensed road is selected by inspecting the coordinates of every location (i.e. geographical selection method). The performance of both selection methods is analyzed by comparing the traffic profile of the target and selected roads (Section 5.3.4). Finally, three generative methods are introduced, towards analyzing the quality of the produced synthetic traffic data (Section 5.3.5). Algorithm 1 summarizes the data and steps involved in a generic implementation of the proposed methodology for the sensorless estimation of road traffic profiles.

Towards assessing quantitatively the performance of the different traffic generation methods, error metrics must be computed. This can be

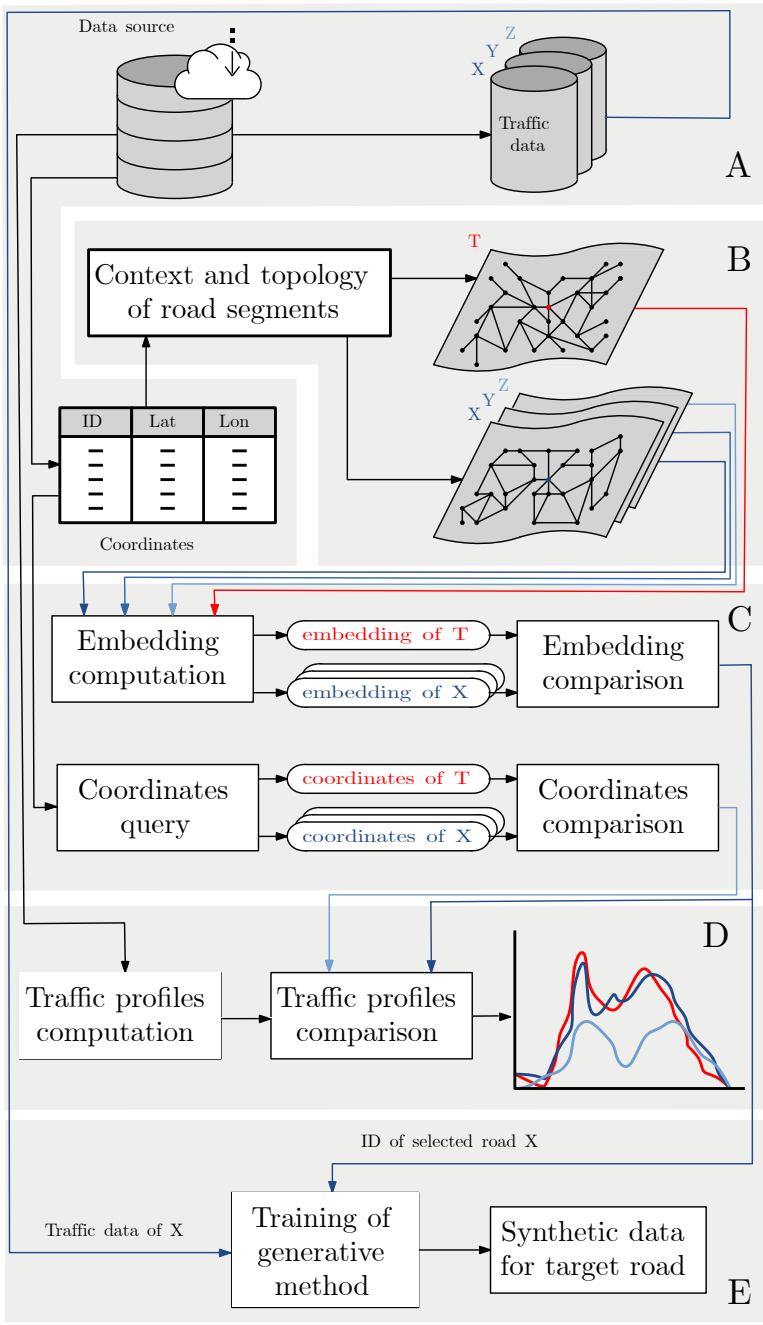



FIGURE 5.5: Experimentation workflow for Case-A. Capital black letters within each shaded region denote the conceptual blocks into which the methodology is divided. The red T stands for *target road segment*, whereas X, Y and Z represent those locations considered as *sensed road segments*. The colored dot represents the *central node* of each ego graph.

**Algorithm 1:** Methodology for generating traffic samples at a sensorless road segment depart from a set of sensed locations.

**Input:** Coordinates $C$ of target $t$ and sensed road segments $\mathbf{s} = \{s_1, \ldots, s_n\}$, historic records of traffic flows $\mathbf{F}[s]$, ego-graph $e\mathcal{G}$, road feature embedding $\mathbf{E} = [f_1, \ldots, f_7]$ composed by features $f$, generative model $m$, and date of day to be predicted $d$

**Output:** Traffic data estimation at target location

**For** $t$ *and* $\mathbf{s}$
  Compute $e\mathcal{G}$ according to $C$
  Compute graph-based features $\mathbf{E_g}[e\mathcal{G}] = [f_1, \ldots, f_5]$, Expressions (5.1) and (5.2)
  Obtain topological features $\mathbf{E_r}[C] = [f_6, f_7]$
  Arrange all features $\mathbf{E} = [\mathbf{E_g}[e\mathcal{G}], \mathbf{E_r}[C]]$
  Standardize $\mathbf{E}$
**end**
Find $s$ that minimizes $d(\mathbf{E}[t], \mathbf{E}[s])$, Expression (5.3)
Train $m$ with $\mathbf{F}[s]$
Estimate traffic flow at the target location $t$ as $m[d]$

achieved by comparing the traffic profiles of two road segments. RMSE has been chosen as error metric, due to its explainability (see Equation 3.19). Obtained deviations can be interpreted as the number of vehicles that might be under/over predicted. Given a desired traffic pattern, the RMSE measures the performance of the selection system.

Being RMSE a scale dependent error metric, its normalized version or nRMSE is also employed in this section. There is no single normalization criterion in the literature; as such, commonly adopted options imply dividing the measure by the mean, standard deviation or the range of the data. The chosen approach is to normalize the RMSE by the mean flow of all weekdays from target location. In the context of this chapter, the standard deviation expresses how much the flow can vary among distinct days and conditions, so a normalization using this metric will be more punishing for those road segments with a fluctuating flow. Similarly, the range of data is also discarded towards normalizing the obtained error. The majority of roads have certain time periods where traffic flow is close to zero (i.e. early morning hours). This entails that the range of data is almost equal to the maximum flow value of the traffic profile. By normalizing the RMSE with the mean flow value, extreme values are smoothed, producing a more robust metric.

### 5.4.1 RQ5.1: Can two similar roads be identified?

As previously introduced, the performance metrics of the road segment selection methods are obtained by comparing traffic profiles. For each target road, two performance metrics are calculated: one for the embedding selection method; a second one for the geographical selection method. Both

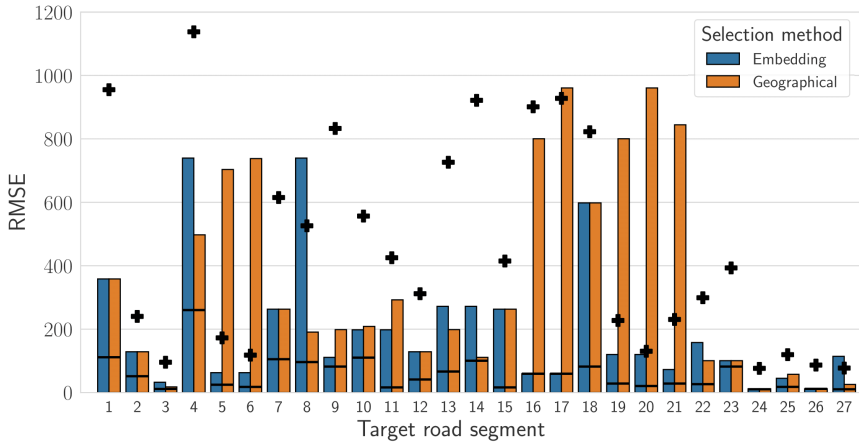



FIGURE 5.6: Part (1/2) of performance of the selection methods for every considered road segment. Black lines showcase the lowest RMSE that would be obtained by selecting the most similar location among sensed ones if traffic measurements were available for comparison. Black crosses denote the mean flow value for the target road (considering only weekdays).

metrics can be directly compared since they are computed using the same traffic profile as the ideal generated pattern (i.e. traffic pattern from the sensorless location). Additionally, the RMSE that would be obtained from the best fitting traffic profile among sensed road segments is also considered. This metric is intended to gauge the lower boundary error that could be obtained. To conclude, the mean flow value of all weekdays (since traffic profile calculation also excludes weekends) of the target road, portrays an upper boundary error. Putting all this together, Figure 5.6 and Figure 5.7 present the obtained results.

From a total of 55 considered road segments, comparing their road feature embeddings produces a better selection for 30 cases. The geographical criterion excels the proposed system in 12 cases. For the rest of the analyzed target roads, both selection methods output the same location (13 draws). Furthermore, a closer look reveals that the selection system based on road feature embedding produces more restrained errors, whereas the average committed error is far greater for the geographical method. The nearest sensed road segment does not have to share any topological relationship regarding the target location (e.g., road type or number of lanes).

Both Figure 5.6 and Figure 5.7 also include the mean flow value for each target road segment, computed from weekdays only. Although there is not a consensus that relates the average traffic flow with the minimum accuracy for a system to be appealing, these values help to showcase the dimensions of each scenario. The same committed error has a different impact on the quality of the selection. When comparing traffic profiles, a difference of 50 vehicles is meaningful for a 100 average flow road. However, if the mean traffic flow is around 1000 vehicles, the performance of the system would be acceptable.

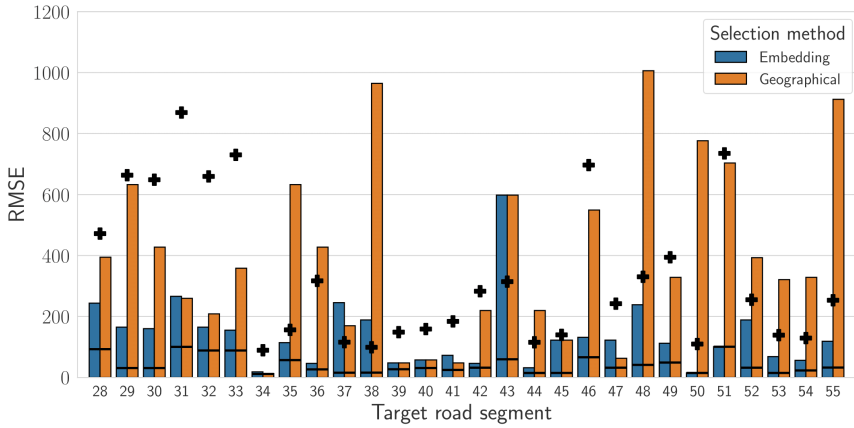



FIGURE 5.7: Part (2/2) of performance of the selection methods for every considered road segment. Black lines showcase the lowest RMSE that would be obtained by selecting the most similar location among sensed ones if traffic measurements were available for comparison. Black crosses denote the mean flow value for the target road (considering only weekdays).

If traffic measurements were available at target location, the black horizontal lines denote the error that would be obtained by selecting the road segment with the most similar traffic profile. Therefore, it is the lowest error that can be obtained with the current set of sensed road segments. This metric helps to contextualize the performance of the selection criteria. If the best selection performance that can be obtained from the current set of sensed locations is not acceptable, it means that either the target location has a peculiar traffic profile or more sensors should be deployed towards collecting traffic data at an analogous location. While the disposal of more sensed locations can enhance the performance of the selection method, the key point should be to gain the ability to spot those road segments of the network with unusual traffic profiles. This way, only a representative road segment for each type of traffic profile would be needed to be sensed, opening the gates to more restricted budget plans.

### 5.4.2 RQ5.2: How does the similarity metric perform?

Being the Euclidean distance the criterion for searching the most similar road feature embedding, it is worth studying if there is any kind of relationship between this metric and the performance of the system. In order to extract insights from comparing the results of every road segment, a scale independent metric is needed. Therefore, the performance of each selection procedure is measured by the nRMSE score.

Along with the embedding selection method, the embedding similarity that would be obtained from the locations with the most similar traffic profile is also analyzed (i.e. those that produce the lowest RMSE possible). By contrast, the geographical selection method is not considered for this

topic, as this criterion is unpredictable: the closest sensed location does not have to possess a similar road feature embedding.

Road feature embeddings are compared by computing the Euclidean distance between a couple of embeddings. Since the seven considered features (see Figure 5.2) are normalized to the range (0,1), from Expression (5.3) the maximum Euclidean distance can be computed as $\sqrt{7 \cdot (0+1)^2}$. This way, two identical road feature embeddings would produce an Euclidean distance of zero, whereas the most dissimilar couple of road feature embeddings would be at an Euclidean distance of $\sqrt{7}$. From this, the embedding similarity can be expressed as a percentage.

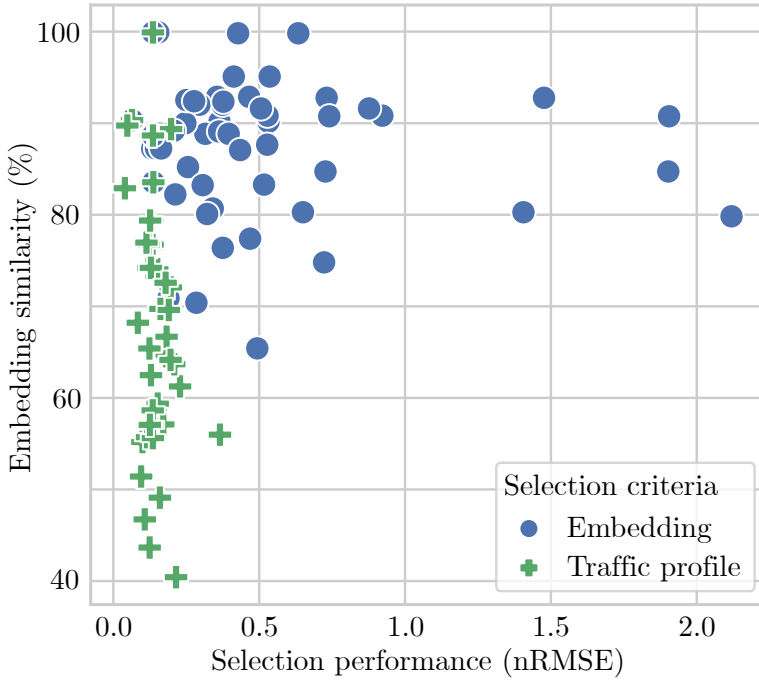



FIGURE 5.8: Road feature embedding similarity against selection performance for each of the analyzed road segments. In green, the selected location is the one with the most similar embedding. In blue, the selected location is the one with the most similar traffic profile (so traffic measurements at target location would be needed for this comparison).

Figure 5.8 shows the relationship between selection performance and embedding similarity. In addition to the road segments with the most similar road feature embedding, for each target location, the road feature embeddings of the sensed locations with the most similar traffic profiles are also represented.

No correlation between embedding similarity and selection performance is found. On one hand, the performance of the road feature embedding selection method is not optimal. There are a few cases where even with an embedding similarity above 80% the selection performance is above the unit (i.e. RMSE above the mean flow of target location). On the other

hand, the selection performance of the traffic profile selection criterion demonstrates that for the majority of considered target locations there is a sensed road segment with a comparable traffic pattern. However, it should be point out that some sensed locations exhibit a low embedding similarity while still producing a high selection performance. Therefore, analogous traffic profiles can exist at road segments of distinct contextual and topological characteristics. Under this premise, optimal selection performance could not be possible to achieve for every considered target location, as unrelated roads can also produce a similar flow.

To finish this second discussion, it is worth pointing out those cases with an almost identical road feature embedding (i.e. similarity near to 100%). As some sensors are deployed geographically close to each other, their graph representation is similar. Road type and number of lanes is also shared. Sometimes, there are two sensed road segments with a direct connection between them. In other cases, both directions of the same avenue are present in the data collection. Sensors that share most of their context and surrounding network, also share most of their embedding feature values. This fact might produce sub-optimal road selections, where almost identical points of the network have distinct traffic behaviors.

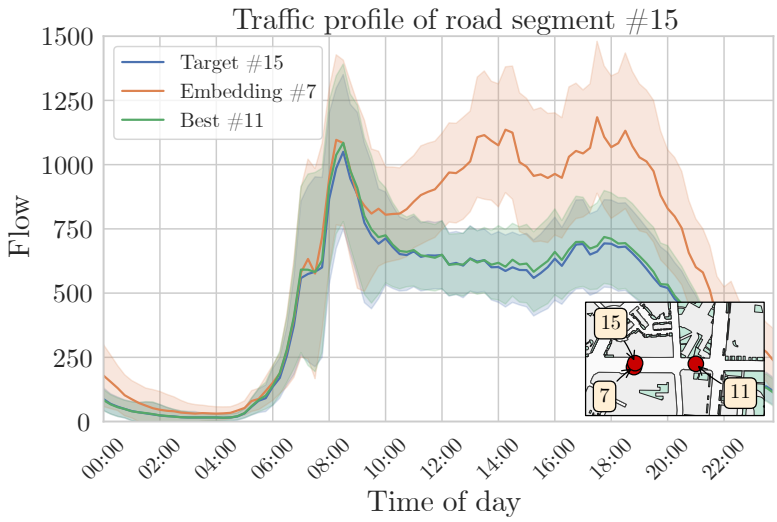



FIGURE 5.9: Comparison of traffic profiles. In blue, the target road. In orange, the traffic pattern of the most similar road according to its road feature embedding. In green, the best fitting traffic profile among sensed locations.

Figure 5.9 illustrates the above issue. Road segment #15 and #7 are placed in the same avenue, just in opposite directions (see Figure 5.1). As can be appreciated however, their traffic profiles highly differ after the morning rush hour. Being both primary roads, #15 heads up to the city center, whereas #7 conducts traffic towards the outskirts. Madrid is a city that receives daily traffic from the surrounding towns, in the form of workers going to complete a work shift. They leave the city center when returning home, which is why a high flow is maintained at #7 until

evening. Among sensed locations, the road segment #11 is located in the same avenue and direction than #15, producing an analogous traffic pattern (colored in green). Clearly, #11 is a good representation of the traffic flow of #15. Still, the road feature embedding of #7 is closer to #15, which is why a high error is obtained.

Nevertheless, in the context of this investigation the target location is not sensed, so no traffic measurements are available at that point of the traffic network. Without traffic data, only topological and contextual data can be used as criteria for selecting one sensed road. From the example of Figure 5.9 it could be justified the need for an additional feature towards representing if a sensed road shares avenue and direction with the target location. However, road segments are connected by intersections that can highly alter the traffic profile (both adding or subtracting flow). Without any information of the influence of such intersections the previous statement is not straightforward. This is the reason for not modeling this aspect in the road feature embedding.

### 5.4.3 RQ5.3: Which is the best generating approach?

For each target location, traffic data is generated for a total of 730 days from the 2018-2019 year period. This process is replicated for the three considered generation approaches: *GAN*, *Cluster* and *NSE*. Synthetic samples are compared to the real traffic flow measurements, thus obtaining a set of error scores.

Table 5.1 shows the mean and standard deviation of the 730 error values for each target location and generative approach. The normalized version of the RMSE allows comparing performance between considered road segments. Figure 5.6 and Figure 5.7 represent the selection performance but can also be interpreted as an estimation of the generative performance for the same time period (if the selection error is normalized by the mean flow). The obtained normalized selection performance is close to the best generative performance obtained for the same location. This fact demonstrates the viability of weekday-only traffic profiles for characterizing road segments. However, since the computed traffic profiles are smoothed representations of the traffic behavior (i.e. no weekends or holidays are represented), the particularities of the daily traffic produce slightly worse performance results for the generative approaches.

High standard deviation values are reported concerning the mean performance error. This aspect is justified both by the size of the traffic datasets and by the analyzed task itself. Two whole years compose the test holdout. Special days like holidays and summer vacations can highly differ between the selected road segment and the target location, even if during the rest of the year they share a similar traffic profile. Likewise, not prediction methods are analyzed, but generative approaches instead. As synthetic data is not built from the traffic recordings of the target location, higher error metrics should be expected regarding short-term forecasting methods where the input features of the model is the past traffic state at target location.

TABLE 5.1: Mean and standard deviation of performance results expressed as NRMSE for all considered generative approaches for every target road segment.

| | GAN | Cluster | NSE | Road type | Best |
|---|---|---|---|---|---|
| **1** | 0.50 ± 0.17 | 0.44 ± 0.18 | **0.38 ± 0.16** | Secondary | NSE |
| **2** | 0.78 ± 0.31 | **0.59 ± 0.21** | 0.61 ± 0.22 | Tertiary | Cluster |
| **3** | 0.60 ± 0.18 | **0.51 ± 0.18** | **0.49 ± 0.13** | Residential | Cluster, NSE |
| **4** | **0.60 ± 0.19** | 0.66 ± 0.21 | 0.64 ± 0.17 | Secondary | GAN |
| **5** | 0.59 ± 0.21 | 0.55 ± 0.22 | **0.49 ± 0.17** | Residential | NSE |
| **6** | 0.70 ± 0.28 | **0.66 ± 0.23** | 0.71 ±0.25 | Residential | Cluster |
| **7** | 0.53 ± 0.18 | 0.49 ± 0.18 | **0.45 ± 0.14** | Secondary | NSE |
| **8** | **1.31 ± 0.30** | 1.42 ± 0.40 | 1.40 ± 0.38 | Secondary | GAN |
| **9** | 0.43 ± 0.12 | 0.30 ± 0.16 | **0.21 ± 0.11** | Secondary | NSE |
| **10** | 0.50 ± 0.19 | 0.45 ± 0.18 | **0.37 ± 0.15** | Secondary | NSE |
| **11** | 0.70 ± 0.26 | 0.56 ± 0.25 | **0.48 ± 0.20** | Secondary | NSE |
| **12** | 0.53 ± 0.18 | 0.51 ± 0.19 | **0.46 ± 0.17** | Tertiary | NSE |
| **13** | 0.54 ± 0.16 | 0.44 ± 0.19 | **0.44 ± 0.14** | Secondary | NSE |
| **14** | 0.44 ± 0.17 | 0.39 ± 0.14 | **0.34 ± 0.11** | Secondary | NSE |
| **15** | 0.78 ± 0.26 | 0.69 ± 0.26 | **0.66 ± 0.21** | Secondary | NSE |
| **16** | 0.40 ± 0.15 | 0.29 ± 0.16 | **0.14 ± 0.10** | Secondary | NSE |
| **17** | 0.42 ± 0.12 | 0.28 ± 0.13 | **0.14 ± 0.10** | Secondary | NSE |
| **18** | 0.75 ± 0.21 | 0.75 ± 0.22 | **0.74 ± 0.18** | Secondary | NSE |
| **19** | **0.52 ± 0.16** | 0.57 ± 0.20 | **0.55 ± 0.15** | Residential | GAN, NSE |
| **20** | 0.97 ± 0.27 | **0.95 ± 0.29** | 0.95 ± 0.27 | Residential | Cluster |
| **21** | 0.56 ± 0.15 | 0.43 ± 0.18 | **0.41 ± 0.16** | Tertiary | NSE |
| **22** | 0.65 ± 0.23 | 0.59 ± 0.25 | **0.57 ± 0.21** | Tertiary | NSE |
| **23** | 0.52 ± 0.29 | 0.45 ± 0.30 | **0.40 ± 0.31** | Tertiary | NSE |
| **24** | 0.73 ± 0.26 | 0.45 ± 0.21 | **0.26 ± 0.16** | Tertiary | NSE |
| **25** | 0.60 ± 0.21 | 0.50 ± 0.20 | **0.40 ± 0.14** | Tertiary | NSE |
| **26** | 0.56 ± 0.24 | 0.49 ± 0.24 | **0.23 ± 0.14** | Tertiary | NSE |
| **27** | 1.97 ± 0.30 | **1.82 ± 0.29** | 1.87 ± 0.39 | Residential | Cluster |
| **28** | 0.65 ± 0.19 | **0.59 ± 0.21** | **0.53 ± 0.13** | Secondary | Cluster, NSE |
| **29** | 0.48 ± 0.17 | **0.35 ± 0.10** | **0.33 ± 0.11** | Secondary | Cluster, NSE |
| **30** | 0.48 ± 0.18 | **0.34 ± 0.13** | **0.31 ± 0.13** | Secondary | Cluster, NSE |
| **31** | 0.44 ± 0.14 | 0.40 ± 0.17 | **0.30 ± 0.15** | Secondary | NSE |
| **32** | 0.44 ± 0.12 | 0.42 ± 0.15 | **0.34 ± 0.12** | Secondary | NSE |
| **33** | 0.39 ± 0.10 | 0.37 ± 0.13 | **0.27 ± 0.09** | Secondary | NSE |
| **34** | 0.50 ± 0.12 | 0.45 ± 0.16 | **0.36 ± 0.12** | Residential | NSE |
| **35** | **0.85 ± 0.20** | 0.85 ± 0.20 | **0.84 ± 0.18** | Residential | GAN, NSE |
| **36** | 0.50 ± 0.18 | 0.40 ± 0.15 | **0.36 ± 0.18** | Tertiary | NSE |
| **37** | 2.22 ± 0.61 | **2.12 ± 0.51** | 2.25 ± 0.70 | Tertiary | Cluster |
| **38** | 2.36 ± 0.51 | **1.94 ± 0.43** | 2.00 ± 0.53 | Residential | Cluster |
| **39** | 0.80 ± 0.17 | **0.46 ± 0.16** | 0.49 ± 0.25 | Residential | Cluster |
| **40** | 0.56 ± 0.22 | 0.52 ± 0.20 | **0.45 ± 0.16** | Tertiary | NSE |
| **41** | 0.66 ± 0.19 | 0.52 ± 0.22 | **0.51 ± 0.20** | Tertiary | NSE |
| **42** | 0.53 ± 0.22 | **0.43 ± 0.22** | **0.40 ± 0.21** | Tertiary | Cluster, NSE |
| **43** | **1.99 ± 0.65** | 2.03 ± 0.39 | 2.06 ± 0.50 | Secondary | GAN |
| **44** | 0.61 ± 0.56 | 0.58 ± 0.58 | **0.55 ± 0.58** | Residential | NSE |
| **45** | 0.98 ± 0.27 | 0.90 ± 0.26 | **0.89 ± 0.24** | Residential | NSE |
| **46** | 0.48 ± 0.15 | 0.32 ± 0.14 | **0.27 ± 0.10** | Secondary | NSE |
| **47** | **0.52 ± 0.17** | 0.56 ± 0.18 | **0.52 ± 0.14** | Residential | GAN, NSE |
| **48** | **0.84 ± 0.25** | 0.85 ± 0.24 | **0.84 ± 0.22** | Residential | GAN, NSE |
| **49** | 0.45 ± 0.17 | 0.40 ± 0.17 | **0.29 ± 0.12** | Tertiary | NSE |
| **50** | 0.54 ± 0.24 | **0.44 ± 0.26** | 0.62 ± 0.59 | Residential | Cluster |
| **51** | 0.41 ± 0.13 | 0.28 ± 0.15 | **0.21 ± 0.13** | Secondary | NSE |
| **52** | **0.74 ± 0.20** | 0.77 ± 0.24 | **0.75 ± 0.20** | Residential | GAN, NSE |
| **53** | 0.66 ± 0.20 | **0.58 ± 0.19** | **0.58 ± 0.15** | Secondary | Cluster, NSE |
| **54** | 0.61 ± 0.22 | **0.52 ± 0.22** | **0.50 ± 0.19** | Residential | Cluster, NSE |
| **55** | **0.56 ± 0.17** | 0.58 ± 0.18 | 0.56 ± 0.15 | Residential | GAN |

Note: displayed results are rounded due to space constraints. For performance assessment, all decimals are considered.

Due to the high standard deviations exposed above, the statistical significance of the differences found between the GAN, Cluster and NSE approaches is further inspected by ranking them considering the outcomes of a hypothesis test. Specifically, for every target road segment, a global Friedman test [379] is performed for repeated measurements to ascertain whether any significant differences exist over the RMSE results obtained with every approach in comparison. If such significance holds at level $\alpha = 0.05$, a Nemenyi post hoc test for unreplicated blocked data [380] is performed over the RMSE results of every pair of generative approaches, so that wins, losses and ties among the three comparison counterparts can be resolved taking into account the statistical significance of the differences in the means of the results.

The NSE approach dominates the benchmark, where for 78% of the analyzed cases, is declared as the best by the Nemenyi test. Furthermore, the standard deviation denotes less statistical dispersion for the committed error. Even so, results are close among the three considered generation approaches. Model complexity and computational resources might be properties of concern for selecting one approach over the others. Coincidentally, performance decrease as the complexity of the generation approaches increases. In line with this, the inconveniences of the NSE method are twofold: 1) the need for traffic measurements for the desired date to be generated; 2) the inability for generating traffic for future dates. The selected data source for this investigation has been selected explicitly to not have missing data, so traffic recordings are available for the NSE generative approach. Likewise, aggregation errors and other data anomalies have been cleaned. Otherwise, the NSE method would have output these corrupted data as synthetic samples, producing inaccurate predictions. The second concern might not be critical for some implementations, as the system is aimed at data generation. However, due to the nature of traffic data, synthetic samples can also serve as future traffic estimations, so the inability to be used for traffic prediction might be a hindrance for the NSE approach. Nevertheless, both issues emanate from the nature of the technique itself, thus being unbridgeable. At this juncture, the use of the Cluster method is encouraged. While still being simple to implement, delivered results are close to the NSE approach without the inconveniences of it.

The Cluster approach could provide enhanced performance with a dedicated set of grouping criteria. For the sake of simplicity and not delving into the particularities of the traffic behavior at the city of Madrid, an approach based on weekdays and holidays only has been presented. Still, how to group available training samples is the key element towards optimal performance. Daily traffic patterns can be grouped considering additional criteria such as weather or events (for instance, football matches). Another option is to perform an autonomous search of clusters with tools like DBSCAN or K-means [381], [382], which find traffic samples with a similar flow traffic pattern, without considering other criteria. A final innovative approach could be focusing on the search of atypical traffic samples sizes. Traffic flow follows a daily pattern, but this profile can also be fragmented

into shorter segments (e.g., hourly segments). While early morning flow is often shared among all days, the flow spike observed during rush hours can present distinct shapes. In this way, partial daily segments could produce further clusters, providing more accurate predictions. In conclusion, the goal should be to produce clusters where all samples have a similar traffic distribution.

Table 5.2: Summary on the results for the sensed locations

| | **Secondary** | **Tertiary** | **Residential** |
|---|---|---|---|
| **# of sensed locations** | 23 | 14 | 18 |
| **# GAN as the best approach** | 3 | 0 | 6 |
| **# Cluster as the best approach** | 4 | 3 | 8 |
| **# NSE as the best approach** | 20 | 12 | 11 |
| **Mean nRMSE** | 0.50 | 0.53 | 0.75 |

The contents of Table 5.2 are extracted from Table 5.1. It is intended to summarize certain statistics drawn from the road type of the target locations. Among considered road segments, road types are distributed as follows: 42% secondary, 25% tertiary, and 33% residential. However, the mean performance error is not evenly distributed, where the synthetic data for residential roads is more different than real flow measurements. Coincidentally, whereas the NSE approach is the preferred option for secondary and tertiary roads, residential roads do not rely on NSE and also select the best generative approach the GAN and Cluster methods. From Figure 5.6 and Figure 5.7 it can be seen that the mean flow of residential roads is usually below 200 vehicles. As the nRMSE is normalized by this last metric, it narrows the room for generation mistakes. This fact leads to the real bottleneck towards a leading performance: the selection of the sensed road segment. Of course, the same error can be committed for other road types, but after normalization, the nRMSE for residential roads is going to be higher. When an appropriate sensed road segment is selected, the NSE approach is most of the time the method that delivers the best performance. This behavior is motivated by the functioning of the generation method, where real traffic flow measurements serve as synthetic data. In contrast, the GAN and Cluster approaches output a smoothed traffic pattern. If the target road segments are analogous to the selected sensed location, the details present in the NSE output provide a high-quality prediction. On the opposite, after a poor choice of sensed road segment, a less disordered traffic pattern performs better.

In essence, the generative approach is not what determines the quality of the synthetic samples, but the selection of the road segment that provides the training data.

### 5.4.4 Discussion and limitations of Case-A

Several conclusions can be drawn from the above experimentation. To begin with, it has been verified that the tailored selection of the segment providing traffic data for the estimation at the target location outperforms

a naïve selection method based on geographical proximity. This stresses on the importance of topological and graph-based features to choose, in an informed fashion, which traffic data to use along the estimation process. This has been noted to be more decisive than the approach used to generate the traffic profile in the target location. As a matter of fact, results have revealed that a naïve estimation method based on assigning the traffic measurements collected in the other most similar location for the same date performs relatively better than other more sophisticated and less interpretable model-based approaches, further highlighting the significance of our proposed road feature embeddings for its competitive performance and its inherent algorithmic transparency.

Despite the higher trustworthiness derived from its inception from domain-specific knowledge and intuition, an evident limitation of considering only topological and graph-based features in the estimation of traffic profiles is that other exogenous factors are not explicitly considered along the process. Instead, aspects such as sociological habits and cultural traits are assumed to be embedded in the traffic data itself, as well as in the experience of traffic experts from which road features were formulated. This implicitness may not affect when estimating data *inside* a city, or a neighborhood, but may conversely hinder the transferability of traffic profiles across different cities even if strongly similar road feature embeddings are discovered. In other words, estimating traffic profiles at one city using data collected by sensors of another city might not be straightforward due to the aforementioned factors.

## 5.5 Experiments and results: Case-B

The prior experiment proposes a solution towards modeling traffic without any data recordings. Even that the road feature embedding method outperforms a naïve criterion such as selecting the closest ATR (in geographical distance), the obtained results elucidate the importance of selecting roads that share similar traffic profiles. However, the disposal of a provisional sensor offers the opportunity to make the latter a less crucial task.

As explained in Section 5.3.6, the idea is to build a model that learns how traffic relate between a target and a sensed location. In order to reduce the time a provisional sensor should be placed at the target road segment, the sensor deployment optimization framework introduced in Section 5.3.7 is implemented. Four consecutive years (i.e. 2016 to 2019) of flow observations are utilized in Case-B: the first two years are used for selecting the optimal sensing mask (see Figure 5.4), while the remaining years simulate a real implementation where traffic samples are collected from the target road according to the sensing mask, and the whole next year is used for evaluating the model's test performance. In a real implementation, the model could be applied right after the provisional sensor is removed from the target location.

The setup considers a subset of 55 road segments located in the neighborhood of Chamartín. Out of these segments, one has been randomly

selected as the target location. The reference roads that will serve as input for the forecasting model are selected by comparing the road feature embedding of all road segments. Figure 5.10 displays the position of such reference roads along with the location of the target road. While the road segments available at the data collection exhibit distinct road types and number of lanes, the four considered locations are single-lane residential roads.



FIGURE 5.10: Location of deployed sensors. The target location is marked with a white star, while red dots represent permanent sensors deployed at the selected reference roads.

The optimization process is applied over the traffic data collected at the three reference roads during 2016 and 2017. A total of 500 evaluations are computed, which provides a collection of results where each sensing mask has an associated validation score. The number of evaluations is obtained by a grid search, where less evaluations could cause the system to not find enough non-dominant solutions, thus only partially modeling the distribution of results. Likewise, more evaluations lead to a greater computational effort for the system, without providing a better understanding of the expected performance.

The RMSE serves again as the error metric, due to its straightforward interpretability. Since the RMSE value depends on the average flow of the target road to be modeled, the validation RMSE that would be produced if no sensing mask was applied is also computed; in other words, if the provisional sensor would be deployed during a whole year. This metric serves as a baseline of the model performance. Theoretically, the forecasting model should perform best in this case since examples of all distinct traffic patterns over the year are fed and can be learned by the model. For this reason, the validation score of each sensing mask produced by the optimization algorithm is normalized by this baseline metric. Finally,

the multi-objective evolutionary algorithm NSGA-II implemented in the jMetalPy library [383] is used to evolve and refine sensing masks as per their Pareto dominance over the space spanned by the two objectives established in the framework (i.e. 'optimizer' block in Figure 5.4). This solver is configured with a population size equal to 100, simplex crossover (SPX) with probability 0.9 and bit flip mutation with probability 1/52 (according to the number of weeks present in a year). Since the purpose of the work is not to evaluate the statistical variability of the results of this meta-heuristic solver, and for the sake of a better focused experimental analysis,the discussion will be held on the results of a single run of the algorithm, leaving any further insights on the matter for future work.

### 5.5.1 RQ5.4: When to install a provisional sensor

The discussion begins by inspecting Figure 5.11, which depicts the non-dominated solutions (sensing masks) that approximate the Pareto trade-off between validation score and the number of training weeks of the target location shown in Figure 5.10. This estimated Pareto front exposes the trade-off between the sensing time and performance that articulates the hypothesis of this case study. However, for this particular scenario there is no reason for deploying the provisional sensor for more than one week since the validation score barely changes. According to the results, there are even some sensing masks that perform better than if sensing was performed for the whole year. Still, these improvements are so narrow that they can be attributed to the random nature of the learning algorithm in use. In the following experiments, the single week sensing mask provided by the optimization process is employed for simulating the traffic data holdout that would be obtained during a real implementation.

A question that naturally arises from the above results is whether sensing the target location with a provisional sensor gives rise to a severe performance degradation with respect to other sensing strategies. To the best of the author's knowledge, there is no prior work dealing with this same scenario whose proposal could be adopted for comparison purposes. Therefore, a performance benchmark is designed according to the sensing approach at the target location:

1. Provisional sensor according to a sensing mask.
2. Provisional sensor for a whole year.
3. Permanent sensor and data holdout according to the sensing mask.
4. Permanent sensor and a whole year of data as holdout.

The first and second approaches only differ in the amount of available data for training, while both implement the framework described in Section 5.3.6. On the other hand, approaches 3) and 4) use a standard short-term forecasting approach, where the last traffic states from target collected at times $\{t-4, \ldots, t\}$ are used as features to predict the traffic flow at the same location and time $t+1$. As for approaches 1) and 2), the ETR algorithm

is employed for building a model. Since the 2016 and 2017 years are used for selecting a sensing mask, the performance benchmark is built upon the following two years of data, where 2018 is the training subset, and 2019 is used for evaluating the test performance. Finally, a *naïve* method is also appended to the benchmark as a performance baseline. The method consists of providing a traffic forecast for $t+1$ the traffic observation at the previous instant $t$. Therefore, this approach also implies a permanent deployment of a sensor on the target road.

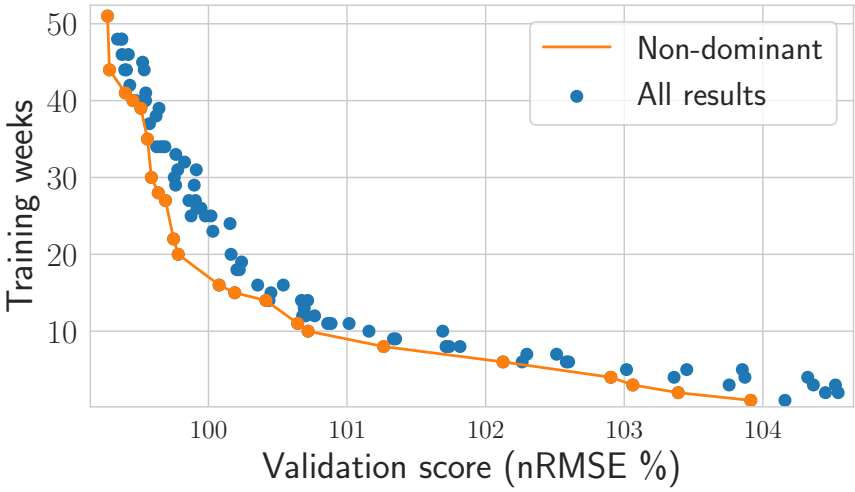



FIGURE 5.11: Distribution of the solutions (sensing masks) evolved by the proposed framework. The validation score has been normalized by the performance that would be obtained if the whole year is sensed. The Pareto front, in orange, is conformed by the non-dominated results. The remaining solutions are highlighted in blue.

The average predictive performance for every approach is displayed in Table 5.3. The forecasting models are fit upon a training dataset, collected either during a single week (i.e. according to the selected sensing mask) or the whole year 2018. For model assessment, the RMSE score is computed for every day of the 2019 year, according to Equation (3.19). The performance levels attained by approaches 3 and 4 are better than those of the proposed approach. Since the sensor is permanently installed at the target road, in these scenarios the model builds its predictions upon recent observations, which are far more related to the traffic state at the target road than the recent past traffic measurements of the reference roads. Traffic fluctuates so little between consecutive observations (strong short-term traffic persistence) that the naïve method exhibits an analogous performance than short-term forecasting approaches. During model training, the algorithm learns that by providing the traffic at $t$ as the prediction for $t+1$, the error is close to the lowest possible.

This behavior can be also appreciated in Figure 5.12, where dashed lines represent the estimated traffic for both approaches 3 and 4. At some timestamps, there is an offset of one interval between the forecast and

the real traffic flow at target. Bearing this in mind, it is not surprising that these approaches render a similar performance than that of the naïve model. As distilled from the traffic forecasting performance benchmark of Chapter 3, short-term traffic models do not necessarily enhance the prediction quality for short forecasting horizons. Only when the forecasting horizon is extended, the baseline performance of the naïve method can be surpassed by more complex data-based models.

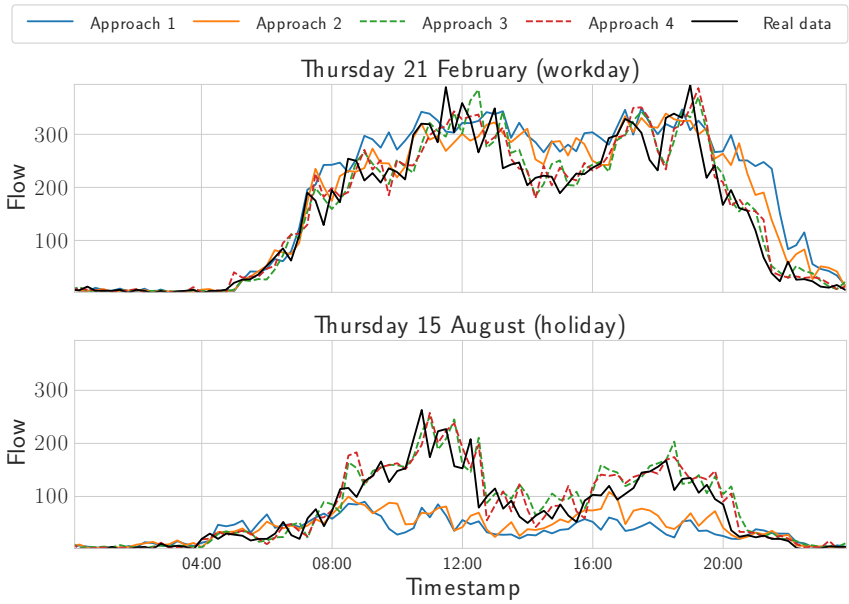



FIGURE 5.12: Prediction performance comparison for the same weekday. Forecasting models based on reference roads are depicted by continuous lines, whereas standard short-term predictions models are represented by dashed lines. The real traffic observation for that day is represented by a black line.

Similar to the short-term models based on past observations, approaches 1 and 2 behave closely to each other in terms of predictive performance, even though the training dataset is strongly limited for approach 1. This insight was already anticipated in Chapter 4, where a short-term forecasting model is shown to perform well with just a month of training traffic data. That training holdout is specially designed towards containing traffic observations from both workdays and holidays. However, the optimization process described in Section 5.3.7 returns a single week sensing mask: from October 29th to November 4th of 2018, which is a very unique week. Monday, Tuesday, and Wednesday are workdays, but Thursday is a national holiday. By extension, Friday also provides an unusual traffic profile, since many citizens in Madrid book holidays on that day so that they can rest for four days in a row. The weekend traffic behaves as usual. In summary, distinct traffic patterns are observed in a single week, which provides an optimal time window for sensing.

Table 5.3: Prediction performance benchmark for the considered approaches. Error metrics are computed as the mean RMSE for the 2019 year holdout.

| Approach | Sensor deployment | Training data | RMSE |
|---|---|---|---|
| (Proposal) 1 | Temporal | One week | 65.27 |
| 2 | Temporal | Whole year | 60.30 |
| 3 | Permanent | One week | 38.29 |
| 4 | Permanent | Whole year | 35.50 |
| Naïve | Permanent | - | 38.18 |

As illustrated in Figure 5.12, during workdays (which are the most common day category), models based on the traffic data collected in reference roads perform similarly to those models that require the maintenance of permanent sensor at the target location. Only during special events such as summer or Christmas vacations, the proposed approach fails to provide high-quality predictions. Errors held for special days are responsible for the performance variation that can be noticed in Table 5.3. The performance improvement obtained by deploying a provisional sensor during the whole year (approach 2) is too narrow to be considered as a cost-efficient option. The relationship between the traffic of reference roads and the target road is stable enough during *most of the year* to be properly modeled with a short-length sensing window.

Summarizing, the main findings reached so far are that 1) a reference road-based forecasting model can be developed within a week if data has been collected at the right moment of the year; and that 2) due to the short sensing time, multiple road segments could be modeled with a single provisional sensor.

### 5.5.2 RQ5.5: Do the sensing masks generalize well?

The optimization process introduced in Section 5.3.7 evaluates the sensing mask by following a *leave-one-reference-road-out* cross-validation strategy, namely, by computing the average performance that would be obtained if one of the reference roads was set as the target location. However, without data from the target location itself, it cannot be assumed that the selected sensing mask is optimal and generalizes well to target locations unseen during the framework's operation.

To shed light on this matter, single-week sensing masks are analyzed all over the year, producing 52 RMSE values, each quantifying the expected validation performance when applying that sensing mask. Likewise, the test performance of the forecasting model after measuring traffic for every single week sensing mask is computed. In this way, a correlation can be analyzed for the same sensing mask between the validation and the test performance metrics. To avoid drawing conclusions from a case that could be singular, the experiment is repeated by considering that each of the 55 road segments in the neighborhood of Chamartín are considered the target location where the novel traffic estimation must be accomplished.

The validation and test error for every sensing mask is displayed in Figure 5.13. The majority of curves exhibit the same shape: high error peaks gather at the three main holiday seasons in Spain (Christmas at the first and last week of the year, Easter week at the end of March or beginning of April, and summer vacations). The reason behind the performance offset between validation and test results is that the first has been computed with the 2018 year data, whereas test performance values have been computed over the next year. The Easter week starts on a different date every year, so the traffic shape drifts slightly between consecutive years. Focusing on the summer vacations, the error increases as the sensing week reaches the middle of August, following the same distribution of the holiday bookings for the season. The good results reported with a single-week sensing mask and the consistent traffic behavior exhibited by the blue lines in Figure 5.13 suggests that the time window can be set according to the results obtained for other target locations. A few orange curves do not follow the common pattern, displaying a chaotic behavior. Reference roads are selected according to the similarity of their road feature embeddings and the one of the target road. However, this do not make the selected reference roads similar among each other. Therefore, during evaluation the obtained error might possess high values if the learned relationship between traffic profiles do not remain constant throughout the validation holdout.

For the majority of the cases under analysis, the best time windows for deploying a provisional sensor are condensed during non-vacation weeks. This insight can be used for accelerating the traffic modeling of multiple target road segments, where a provisional sensor is physically moved to another target road every week during non-vacation seasons.

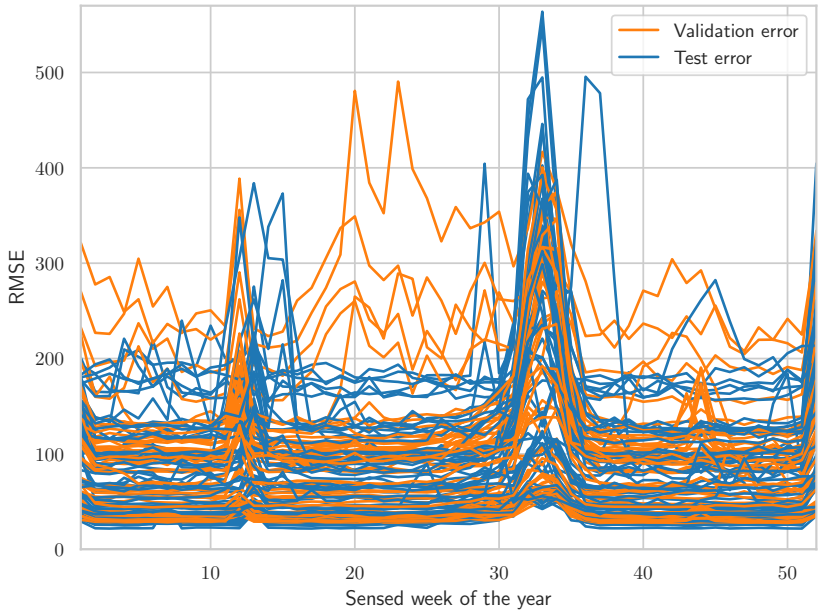



FIGURE 5.13: Validation and test error for every single week sensing mask.

### 5.5.3 RQ5.6: The relevance of reference roads

Reference roads are selected according to their road feature embeddings. However, in the lack of such information, reference roads should be chosen uniformly at random. To ascertain whether an embedding-based selection makes any difference in terms of traffic estimation, the performance analysis of the case study is repeated (i.e. the same target location), but now setting 3 new reference roads from outside the Chamartín neighborhood. The traffic profile from these reference roads could not maintain a constant relationship regarding the real traffic at the target location. Departing from the findings of RQ4, the sensing mask selected during the case study seems the right choice for every set of reference roads. Therefore, the performance of the forecasting model is computed using the new reference roads and the same sensing mask. This way the performance can be compared with the results displayed in Table 5.3. Additionally, the effects of restricting the input data to a single reference road are analyzed.

TABLE 5.4: Forecasting performance using different locations as reference roads. Input data refers to those roads selected by their road feature embedding similarity ($E$) and chosen randomly ($R$).

| Input data | Road type | Mean flow | RMSE |
|---|---|---|---|
| Target | Residential | 151 | - |
| E1 | Residential | 68 | 73.76 |
| E2 | Residential | 80 | 66.65 |
| E3 | Residential | 105 | 106.64 |
| E1, E2, E3 | Residential | 84 | 65.27 |
| R1 | Tertiary | 1063 | 68.46 |
| R2 | Residential | 132 | 67.26 |
| R3 | Primary | 403 | 71.06 |
| R1, R2, R3 | Mixed | 533 | **62.09** |

Table 5.4 summarizes the results obtained to answer RQ5. Reference roads selected via road feature embeddings are denoted as $E1, E2, E3$, whereas reference roads selected uniformly at random are referred to as $R1, R2, R3$. Surprisingly, the best-performing model is the one built upon the traffic observations of the randomly selected roads. These roads are far enough so that their traffic is not directly related to that of the target road. Different road types conform to this reference set, whereas those selected by the road feature embeddings share the same one. The road type defines the role a road segment plays within the traffic network. Hence, the traffic behavior can be expected to be similar between roads of the same category. Another feature that was initially borne in mind is the mean flow. The reference roads selected via road feature embeddings provide a close flow average to the target. However, two of the randomly selected reference roads exhibit higher flow values, while still performs well for the problem under study.

When focusing on those models which a single reference road, the performance is slightly worse than when more data is available as input. If more reference roads are available, the model has more possibilities of finding a more reliable relationship. Still, the performance is acceptable for a scenario where only one location has a permanent sensor deployed. The model based on $E3$ exhibits worse performance than the rest of the analogous models. The reason behind this result is that traffic collected at $E3$ experienced a sudden increase during the summer months. The model learned to relate traffic during the last week of October (according to the selected sensing mask), so this abrupt change of the traffic behavior causes the predictions to fail during those months, hence increasing the average RMSE reported in the table.

On a concluding note, these experiments underscore that in the presented framework, what matters is that the relationship between the traffic of reference roads and that of the target location remains stable for most of the time. The dynamic range of the flow measurements does not influence the forecasting performance, since the model learns to relate traffic between locations. Therefore, the selection criteria for the reference roads should not be focused on the average flow, but rather on the seasonality of the traffic data observed in the scenario under consideration.

## 5.6 Summary

This chapter has elaborated on the problem of estimating road traffic data over a location of an urban road network without any deployed sensor nor prior collected traffic data in the location whatsoever. Under these circumstances, it becomes necessary to resort to other sources of information to estimate the traffic profiles occurring at the target location. For this purpose,it has been proposed to exploit topological and graph-based information around the location of interest to find other *similar* road segments. The overarching idea is that traffic data collected over those similar segments can be used to estimate the traffic in the target location. To compute the similarity between any two road segments, each location is characterized in the form of road feature embeddings, built upon a set of topological and graph-based features. The definitions of these features rely on intuition and domain-specific knowledge about the traffic dynamics in the location of interest, which ultimately adds to the overall trustworthiness of traffic estimations by decision makers for which they are produced. Such road embeddings are used together with a measure of distance to compute the similarity between the segments at hand, so that the location closest to the target can be identified and used for traffic estimation. It is important to note that the challenging assumptions from which this chapter departs (unavailability of any other source of information beyond the road network) do not hinder the generalization of these developments to other scenarios with further information available for the traffic estimation process. Nevertheless, features within the road feature embedding should be designed according to the knowledge of traffic managers regarding the target area, in a similar process to the one in Madrid (Spain) showcased

in the first case study. This has herein focused on the traffic map of a major city, but urban areas in the countryside connected by interurban networks might require further investigations to yield new topological and graph-based features tailored for such particular scenarios.

Besides the limitation just exposed, other research lines rooted on the insights distilled from Section 5.4 are projected for the near future. New and more insightful road embedding features should be investigated, potentially using meta-learning methods to infer them automatically from data (e.g., symbolic regression, evolutionary programming with graph-based primitives). In partial connection with this, the obtained results help to foresee that a generalized distance metric – for instance, a weighted Euclidean distance – could be evolved via an evolutionary wrapper towards optimally tuning the importance of every feature of the embedding in the value of the distance. Finally, other sources of information that can be retrieved from geographical information systems can be valuable inputs to be considered in our road feature embeddings, without drifting away from the assumed starting point of this work. As such, hospitals and police stations can regulate the average speed of the surrounding roads under the legal limit, whereas entertainment venues surrounded by stores and restaurants often have a high road occupation percentage on holidays and weekends. The *Santiago Bernabeu* stadium is located inside the area of the case study. Football matches affect the traffic of the surrounding roads, where the intensity of its influence is inversely related to the distance to the stadium [384]. The referred to as POIs comprise any kind of business, service center or important area that might influence the traffic behavior. Consequently, future work will embrace such POI-based information (when available) to produce better road feature embeddings that leverage even further the existence of expert knowledge about the factors affecting the traffic at the target location.

Moving to the second case study (Section 5.5), a novel framework for traffic prediction over non-permanently-sensed roads has been proposed. The framework builds reliable predictions without the need for continuously collecting traffic measurements at the target location. Provisional traffic sensing allows monitoring multiple locations with the same sensor, hence providing a cost-effective alternative to the installation of permanent sensors in different locations of the road network. In doing so, the framework formulates an optimization problem to decide when to deploy a provisional sensor based on the minimization of the expected prediction error and the number of weeks for which the sensor's deployment is needed. The obtained results elucidate that one week of target data suffices for model training. Nevertheless, this insight might be coincidental and a by-product of the case study under analysis. Therefore, the experiments have also examined in depth whether the optimality of the sensing mask evolved by the proposed framework conforms to intuition. This is confirmed as validation and test performance levels seem to correlate well and the selected sensing mask correspond to a special week with a diversity of daily traffic patterns. A final study has been performed about the relevance of the reference roads being similar to the target location. The

study concludes that the dynamic range of the traffic flow is not the most relevant feature, but rather to keep a high correlation between the traffic observations of reference roads and the target location.

Only one target location has been considered for testing a model that learns to relate traffic from two or more road segments. During real implementations numerous target roads should be expected. Further research could be focused on how to optimize the deployment of provisional sensors at multiple targets. Obtained results demonstrate that with a single sensor traffic over several roads could be modeled, since the obtained error is similar for various sensing masks. However, these results have been computed for single-week sensing periods, simplifying the cardinality of possible schedules for sensing each target road. Future research will further investigate whether the provisional sensing schedule can be optimized at an urban scale, exploiting similarities and differences between sensed and non-sensed locations, and maintaining cost-effectiveness as one of its design objectives.

# Chapter 6

# Concluding Remarks

Traffic forecasting, being one of the research areas that vertebrates the application of Intelligent Transportation Systems, produces excellent results nowadays. However, accurate traffic predictions are constrained by a key factor: data availability. State-of-the-art performing models can only be produced upon comprehensive traffic datasets. Furthermore, instead of exploring how to develop top-performing models under different degrees of data constraints, the research community has focused on improving the forecasting accuracy by developing more complex models. These proposed models are usually evaluated over a limited set of public traffic datasets, hence results are biased and overfitted to such data collections.

This Thesis has aimed at exploring new techniques that allow characterizing the road traffic under distinct levels of data availability. This *characterization* should be understood as being able to predict and understand the traffic behavior of a certain location on a date of interest. The degrees of data availability constraints considered in this work can be summarized as: 1) having access to historic traffic data from at least one complete year; 2) collecting traffic data during few weeks only; 3) no traffic measurements from the target location. Several contributions and findings have been produced while pursuing these challenges, which can be further summarized into the following blocks.

- **State of the Art of the Field**

During the last decade, research on traffic forecasting has focused on developing complex Deep Learning models. Even surveys on this topic circumvent this issue and centralize on less insightful questions around modeling, such as analyzing their components, their performance or new promising methods that should be considered in future work. Before performing a critical analysis about this current research trend, an independent review of the literature is conducted. Over 165 published works addressing short-term traffic forecasting with Deep Learning models were thoroughly classified and analyzed, according to two novel criteria, namely: 1) the nature of the proposed problem; 2) the type of Deep Learning architecture employed for modeling the traffic. The analysis serves not only to confirm the above research trend, but also to highlight common pitfalls encountered in the literature. For instance, scholars mix the concepts of short- and long-term forecasting. Traffic datasets employed for model assessment

are not properly selected, since they do not usually cover all the scenarios that can happen throughout a year. Besides, comparison studies are not properly design, since authors usually compare novel models with not properly adjusted or out-of-date models. The final contribution distilled from the presented survey is drawing a set of challenges unattended until date. Some of these research opportunities inspire the subsequent chapters of this Thesis, aiming to motivate new research efforts on such endeavors.

- **A Benchmark for Short-term Traffic Forecasting Models**

One of the main insights distilled from the review of the state of the art is that research should not be heavily focused on improving the predictive performance of traffic models. Excluding network-wide forecasting models that leverage the graph representation of traffic maps, the forecasting performance is close to its limit. The most common approach for predicting traffic is to build a model that learns to estimate the short-term traffic state depart from previous flow measurements. Over the last decade, complex Deep Learning architectures have been proposed, fueled by the promise of producing a more representative set of features regarding the original input data. However, the most useful knowledge is already accessible from the latest traffic samples recorded by an ATR. The baseline for the presented short-term traffic forecasting benchmark, is a naïve model that use the latest traffic flow measurement as prediction for next timestep. The excellent results yielded by the naïve model set a narrow margin for the hypothetical improvement a new more complex modeling technique could provide. As could be expected, a comparison between several Deep Learning architectures and shallow learning methods provided similar predictive performance. The major insight distilled from the benchmark is that for a simple problem as it is the short-term traffic forecasting of a permanently sensed road, the modeling technique is not the key factor but data quality instead. A more extensive preprocessing of the training data has the potential to uncover data patterns that aid more reliable traffic predictions. Motivated to inspire new research lines, alternative modeling methods have been appended to the benchmark. In detail, the analyzed randomization based approaches have produced similar results in comparison to more complex and computationally expensive architectures. This enables implementing predictive models in affordable machines.

- **Building Forecasting Models From Limited Data Holdouts**

Traffic samples can be collected at any location of interest by installing ATRs. However, monetary and time constraints are part of traffic modeling projects, making traffic surveillance limited in many occasions. If traffic data can only be collected during a restricted time window, a top performing model can not be delivered using standard modeling techniques. In Chapter 4, a case study is proposed, representing an scenario where data from a target road segment is limited. The key point during this circumstance is to make the most of the available traffic data. With this idea in mind, the weights learned by other traffic models are transferred,

making the target model able to predict traffic without learning from target traffic data. However, without any other adjustment, the predictions will be adapted to a different traffic behavior. It is on this stage when the traffic data collected by the provisional sensor comes into play, allowing to adjust the model to the target traffic profile. Still, if traffic was not monitored long enough for capturing all the traffic behaviors that can occur in the target road segment, the model can be updated during operation. For doing this, a permanent sensor needs to be installed in the location of interest, so every new traffic measurement is fed to the model. Obtained results validate the aforementioned approaches for two possible scenarios. If traffic is temporarily monitored, transferring the knowledge from other forecasting model complements the lack of traffic data, enabling to build a capable model. On the opposite, if a permanent sensor is intended to be installed on the location of interest, mixing transfer learning with online learning allows to further extend the capabilities of the target model, without the need of collecting traffic data during an extensive time period.

- **Characterizing Sensorless Road Segments**

The goal for a traffic forecasting model is to characterize a road segment and ultimately a whole traffic network. A minimum amount of data collected at the location of interest is needed for building such a model, since models learn from the structure and patterns encapsulated in traffic data. The natural progression for this Thesis has been to explore an scenario where traffic data is not available, which means to investigate how to characterize a sensorless location. The key concept proposed to solve this task is to elaborate a *road feature embedding*, which is a set of values computed from the topological and contextual traits of each road segment. Experiments are conducted under the hypothesis that roads of similar context (e.g. road type or centrality with respect to a given graph) share similar traffic behaviors. Promising results have been obtained in this endeavor, since analogous traffic profiles could be found between pairs of selected roads. A new research area stems from this finding, since until now, installing ATRs or at least collecting data via temporary sensors were the only known means for characterizing road segments. In theory, this technology allows exploring in which roads should ATRs be installed, so the rest of the traffic network can be characterized by pairing the remaining sensorless roads to those permanently monitored.

## 6.1 List of publications

As a result of the research conducted while pursuing this doctoral degree, several contributions were published in journals and conferences of the traffic forecasting area, which are listed below:

- **Journal publications**
  - E. L. Manibardo, I. Laña and J. Del Ser, "Deep learning for road traffic forecasting: Does it make a difference?", in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no 7, p. 6164-6188, 2021.

    **JCR** 9.551 5/40 **Q1** *Transportation Science and Technology*
  - E. L. Manibardo, I. Laña, E. Villar-Rodriguez and J. Del Ser, "A Graph-Based Methodology for the Sensorless Estimation of Road Traffic Profiles", in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no 8, p. 8701-8715, 2023.

    **JCR** 8.5 4/40 **Q1** *Transportation Science and Technology*
- **Conference publications**
  - E. L. Manibardo, I. Laña, J. L. Lobo and J. Del Ser, "New perspectives on the use of online learning for congestion level prediction over traffic data", in International Joint Conference on Neural Networks (IJCNN), p. 1-8, 2020.
  - E. L. Manibardo, I. Laña and J. Del Ser, "Transfer learning and online learning for traffic forecasting under different data availability conditions: Alternatives and pitfalls", in IEEE International Conference on Intelligent Transportation Systems (ITSC), p. 1-6, 2020.
  - E. L. Manibardo, I. Laña and J. Del Ser, "Random vector functional link networks for road traffic forecasting: Performance comparison and stability analysis", in International Joint Conference on Neural Networks (IJCNN), p. 1-8, 2021.
  - E. L. Manibardo, I. Laña and J. Del Ser, "Change detection and adaptation strategies for long-term estimation of pedestrian flows",in IEEE International Conference on Intelligent Transportation Systems (ITSC), p. 1-8, 2021.
  - E. L. Manibardo, I. Laña and J. Del Ser, "What to sense when there is no sensor: Ex-novo traffic flow estimation for non-sensed roads", International Conference on Intelligent Transportation Systems (ITSC), p. 1-8, 2022.

### 6.1.1 Other publications

Besides, the author has also collaborated in the research yielding the following publications:

- **Conference publications**
  - E. L. Manibardo, I. Laña, J. Del Ser, A. Carballo and K. Takeda, "Expert-driven rule-based refinement of semantic segmentation maps for autonomous vehicles", IEEE Intelligent Vehicles Symposium (IV), p. 1-8, 2023.

– E. L. Manibardo, I. Laña and J. Del Ser, "Multi-step Ahead Visual Trajectory Prediction for Object Tracking using Echo State Networks", International Conference on Intelligent Transportation Systems (ITSC), p. 1-8, *to appear*.

## 6.2 Future research lines

Those reviews conducted upon the short-term traffic forecasting research topic focuses on the modeling approaches presented in the literature. This clarifies why the research opportunities proposed in such surveys are centered in developing more complex traffic forecasting models. The original motivation for conducting this Thesis was to break with such a trend. Therefore, during the development of this Thesis, new questions and challenges have emerged. While some of them inspire the studies conducted in the experimental chapters, several research opportunities remain unsolved. Similarly, investigating on them can make new questions arise, reason for which topics addressed in previous chapters also appear in the list below:

- Alternative modeling techniques: data-driven models extend beyond the shallow models or Deep Learning architectures that can be easily found in the literature. During Chapter 2, FRBS models [270] have been introduced as an alternative modeling approach that has gained popularity, thanks to its interpretable functioning. Instead of encoding knowledge in the form of numerical parameters as in most data-driven models, FRBS models learn human-readable *if-then* rules. Control operations related to the ITS field already benefit from FRBS, such as traffic signal control systems [385]. However, to the best of the author's knowledge, the use of FRBS models for traffic forecasting has declined during the last decade. Further research should be conducted on this matter, towards determining if the prediction performance gain of not interpretable models (e.g. Deep Learning) justifies dropping FRBS as modeling tool.

- Understanding the behavior of Graph Neural Networks: one challenge unattended in this Thesis is about explaining the behavior of Deep Learning models. The rationale for this decision is that the benchmark presented in Chapter 3 showcases that Deep Learning models do not perform on top of other less-complex and explainable data-driven methods such as Random Forest [386]. However, graph-based Deep Learning architectures are the most promising approach for tackling network-wide traffic predictions when enough time resolution in the traffic flows is available. A high density of installed ATRs needs to be deployed and maintained, otherwise the GNNs does not have enough traffic information from the network for performing quality predictions. This enables learning the correlation among nodes, which is a meaningful and valuable resource for modeling traffic. In this scenario, it makes sense to use this family of methods, since GNNs are able to leverage the information present in a graph representation of a traffic network to improve the prediction quality. The near future in network-wide traffic predictions

will be dominated by this technology, so efforts should be made towards understanding how the design of the graph representation impacts the model output. Likewise, scholars should concentrate on understanding the meaning of those features distilled from the graph representation. One research option could be to merge GNN with FRBS, exploiting the best of both techniques: the performance of GNNs and the explainability of FRBS.

- Characterizing road segments without traffic data: the whole Chapter 5 is dedicated to investigate this novel idea. Although progress has been made, results have demonstrated that characterizing traffic at sensorless locations need further research efforts. More precisely, the proposed *road feature embedding* represents an initial effort of would should be a more complex vector comprising hundred of values with the potential of precisely describing the traffic behavior. New features could be engineered by comparing the topology and connections of those roads that share similar traffic behaviors. On this basis, the behavior of a road should not be condensed in a single traffic profile as prescribed in this Thesis, but all traffic measurements collected by an ATR should be analyzed. Finally, a data-driven method could be trained to learn how road feature embeddings relates to traffic profiles, instead of assisting the selection of similar road segments via naïve approaches such as the Euclidean distance.

- Foundation models for traffic forecasting: foundations models are powerful and versatile Machine Learning models trained on vast amounts data, intending to be used as pre-trained models for a wide range of downstream tasks. One application of this technology heavily used in modern days is ChatGPT [387], which is a state-of-the-art language model that excels in natural language understanding and generation for conversational applications. Another cutting-edge model is DALL-E [388], capable of generating images from textual descriptions. Inspired by this revolutionary technology, Garza and Mergenthaler-Canseco have created TimeGPT-1 [389], the first time series foundation model capable of building predictions for time series never seen by the model. The key point that makes foundation models so powerful is the vast amount of data used for training. In the scope of traffic forecasting, a foundational model could be produced by specializing it to the desired task and then comparing the computational effort and forecasting performance respect to adjusting an ad-hoc data-driven model to the target location.

# Bibliography


[1] T. Litman, "Transportation cost and benefit analysis," *Victoria Transport Policy Institute*, vol. 31, no. 1, pp. 1–19, 2009.

[2] A. Perallos, U. Hernandez-Jayo, E. Onieva, and I. J. G. Zuazola, *Intelligent Transport Systems: Technologies and Applications*. John Wiley & Sons, 2015.

[3] M. Yaghini, M. M. Khoshraftar, and M. Seyedabadi, "Railway passenger train delay prediction via neural network model," *Journal of Advanced Transportation*, vol. 47, no. 3, pp. 355–368, 2013.

[4] J. Xu and G. Bailey, "The airport gate assignment problem: Mathematical model and a tabu search algorithm," in *Annual Hawaii International Conference on System Sciences*, IEEE, 2001, p. 10.

[5] P. Mannion, J. Duggan, and E. Howley, "An experimental review of reinforcement learning algorithms for adaptive traffic signal control," in *Autonomic Road Transport Support Systems*, Springer, 2016, pp. 47–66.

[6] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, "Deep Learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2020.

[7] J. I. Levy, J. J. Buonocore, and K. Von Stackelberg, "Evaluation of the public health impacts of traffic congestion: A health risk assessment," *Environmental health*, vol. 9, no. 1, p. 65, 2010.

[8] J. Jin and P. Rafferty, "Does congestion negatively affect income growth and employment growth? Empirical evidence from US metropolitan regions," *Transport Policy*, vol. 55, pp. 1–8, 2017.

[9] 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.

[10] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[11] I. Laña, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road traffic forecasting: Recent advances and new challenges," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 93–109, 2018.

[12] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Systems with Applications*, vol. 207, p. 117 921, 2022.

[13] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. Transportation Research Board, 1979, pp. 1–9.

[14] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transportation Research Record*, no. 773, 1980.

[15] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transportation Research Part B: Methodological*, vol. 18, no. 1, pp. 1–11, 1984.

[16] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "LSTM network: A Deep Learning approach for short-term traffic forecast," *Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[17] N. G. Polson and V. O. Sokolov, "Deep Learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017.

[18] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *International Conference on Learning Representations*, pp. 1–22, 2013.

[19] A. S. Nair, J.-C. Liu, L. Rilett, and S. Gupta, "Non-linear analysis of traffic flow," in *Intelligent Transportation Systems Conference*, IEEE, 2001, pp. 681–685.

[20] D. Gunning, "Explainable artificial intelligence (XAI)," *Defense Advanced Research Projects Agency (DARPA), Web*, vol. 2, no. 2, pp. 1–36, 2017.

[21] A. Barredo-Arrieta, N. Díaz-Rodríguez, J. Del Ser, *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[22] S. J. Pan and Q. Yang, "A survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[23] F. Suchanek and G. Weikum, "Knowledge harvesting in the big-data era," in *International Conference on Management of Data*, 2013, pp. 933–938.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of Deep Learning for natural language processing," *Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2020.

[26] L. Liu, W. Ouyang, X. Wang, *et al.*, "Deep Learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[27] H Kawashima, "Long term prediction of traffic flow," *IFAC Proceedings Volumes*, vol. 20, no. 3, pp. 75–82, 1987.

[28] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[29] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Statistical methods for detecting non-linearity and non-stationarity in univariate short-term time-series of traffic volume," *Transportation Research Part C: Emerging Technologies*, vol. 14, no. 5, pp. 351–367, 2006.

[30] E. Vlahogianni and M. Karlaftis, "Temporal aggregation in traffic data: Implications for statistical characteristics and model choice," *Transportation Letters*, vol. 3, no. 1, pp. 37–49, 2011.

[31] Y. Kamarianakis, H. O. Gao, and P. Prastacos, "Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 821–840, 2010.

[32] J. Tang, Y. Wang, H. Wang, S. Zhang, and F. Liu, "Dynamic analysis of traffic time series at different temporal scales: A complex networks approach," *Physica A: Statistical Mechanics and its Applications*, vol. 405, pp. 303–315, 2014.

[33] T. Cheng, J. Haworth, and J. Wang, "Spatio-temporal autocorrelation of road network data," *Journal of Geographical Systems*, vol. 14, no. 4, pp. 389–413, 2012.

[34] Y. Kamarianakis, W. Shen, and L. Wynter, "Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO," *Applied Stochastic Models in Business and Industry*, vol. 28, no. 4, pp. 297–315, 2012.

[35] S. Sun, R. Huang, and Y. Gao, "Network-scale traffic modeling and forecasting with graphical LASSO and neural networks," *Journal of Transportation Engineering*, vol. 138, no. 11, pp. 1358–1367, 2012.

[36] X. Ma, H. Yu, Y. Wang, and Y. Wang, "Large-scale transportation network congestion evolution prediction using Deep Learning theory," *PloS one*, vol. 10, no. 3, e0119044, 2015.

[37] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *International Conference on Advances in Geographic Information Systems*, 2016, pp. 1–4.

[38] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014.

[39] A. Ermagun and D. Levinson, "Spatiotemporal traffic forecasting: Review and proposed directions," *Transport Reviews*, vol. 38, no. 6, pp. 786–814, 2018.

[40] J.-P. Rodrigue, *The geography of transport systems*. Taylor & Francis, 2020, pp. 1–480.

[41] J. S. Angarita-Zapata, A. D. Masegosa, and I. Triguero, "A taxonomy of traffic forecasting regression problems from a supervised learning perspective," *IEEE Access*, vol. 7, pp. 68 185–68 205, 2019.

[42] L. N. Do, N. Taherifar, and H. L. Vu, "Survey of neural network-based models for short-term traffic state prediction," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 1, e1285, 2019.

[43] H. Nguyen, L.-M. Kieu, T. Wen, and C. Cai, "Deep Learning methods in transportation domain: A review," *Intelligent Transport Systems*, vol. 12, no. 9, pp. 998–1004, 2018.

[44] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via Deep Learning: A survey," *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 144–163, 2019.

[45] A. Khan, M. M. Fouda, D.-T. Do, A. Almaleh, and A. U. Rahman, "Short-term traffic prediction using Deep Learning long short-term memory: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 11, pp. 94 371–94 391, 2023.

[46] S. Wang, J. Cao, and P. Yu, "Deep Learning for spatio-temporal data mining: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3681–3700, 2020.

[47] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep Learning on traffic prediction: Methods, analysis, and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4927–4943, 2021.

[48] K. Lee, M. Eo, E. Jung, Y. Yoon, and W. Rhee, "Short-term traffic prediction with deep neural networks: A survey," *IEEE Access*, vol. 9, pp. 54 739–54 756, 2021.

[49] H. Yuan and G. Li, "A survey of traffic prediction: From spatio-temporal data to intelligent transportation," *Data Science and Engineering*, vol. 6, pp. 63–85, 2021.

[50] Y. Hou, X. Zheng, C. Han, W. Wei, R. Scherer, and D. Połap, "Deep Learning methods in short-term traffic prediction: A survey," *Information Technology & Control*, vol. 51, no. 1, pp. 139–157, 2022.

[51] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Transactions on Knowledge and Data Engineering*, pp. 3795–3796, 2023.

[52] A. Gobezie and M. S. Fufa, "Machine Learning and Deep Learning models for traffic flow prediction: A survey," *Research Square preprint*, 2020.

[53] K.-H. N. Bui, J. Cho, and H. Yi, "Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues," *Applied Intelligence*, vol. 52, no. 3, pp. 2763–2774, 2022.

[54] H. Yi and K.-H. N. Bui, "VDS data-based Deep Learning approach for traffic forecasting using LSTM network," in *AAAI Conference on Artificial Intelligence*, Springer, 2019, pp. 547–558.

[55] G Albertengo and W Hassan, "Short-term urban traffic forecasting using Deep Learning," *Annals of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, pp. 3–10, 2018.

[56] T. Pamuła, "Impact of data loss for prediction of traffic flow on an urban road using neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1000–1009, 2018.

[57] K.-H. N. Bui, H. Yi, H. Jung, and J. Seo, "Big data analytics-based urban traffic prediction using Deep Learning in ITS," in *International Conference on Artificial Intelligence*, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing, 2019, pp. 270–273.

[58] A. E. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "DeepPRESIMM: Integrating Deep Learning with microsimulation for traffic prediction," in *International Conference on Systems, Man and Cybernetics*, IEEE, 2019, pp. 4257–4262.

[59] L. Liu and R.-C. Chen, "A novel passenger flow prediction model using Deep Learning methods," *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 74–91, 2017.

[60] A. Zonoozi, J.-j. Kim, X.-L. Li, and G. Cong, "Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns," in *IJCAI*, vol. 18, 2018, pp. 3732–3738.

[61] A. Sudo, T.-H. Teng, H. C. Lau, and Y. Sekimoto, "Predicting indoor crowd density using column-structured deep neural network," in *Workshop on Prediction of Human Mobility*, 2017, pp. 1–7.

[62] Y. Zhang, T. Cheng, and Y. Ren, "A graph Deep Learning method for short-term traffic forecasting on large road networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 10, pp. 877–896, 2019.

[63] Q. Zhang, Q. Jin, J. Chang, S. Xiang, and C. Pan, "Kernel-weighted graph convolutional network: A Deep Learning approach for traffic forecasting," in *International Conference on Pattern Recognition*, IEEE, 2018, pp. 1018–1023.

[64] T. Jia and P. Yan, "Predicting citywide road traffic flow using deep spatio-temporal neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3101–3111, 2020.

[65] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, "Deep Learning architecture for short-term passenger flow forecasting in urban rail transit," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7004–7014, 2020.

[66] 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.

[67] Y. Huang, Y. Weng, S. Yu, and X. Chen, "Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting," in *International Conference On Trust, Security And Privacy In Computing And Communications*, IEEE, 2019, pp. 678–685.

[68] V. Hassija, V. Gupta, S. Garg, and V. Chamola, "Traffic jam probability estimation based on blockchain and deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 3919–3928, 2021.

[69] Y. Ren, H. Chen, Y. Han, T. Cheng, Y. Zhang, and G. Chen, "A hybrid integrated Deep Learning model for the prediction of citywide spatio-temporal flow volumes," *International Journal of Geographical Information Science*, vol. 34, no. 4, pp. 802–823, 2020.

[70] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A Deep Learning prediction," in *AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5668–5675.

[71] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 1655–1661.

[72] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask Deep Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2019.

[73] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, "Cross-city Transfer Learning for deep spatio-temporal prediction," in *International Joint Conference on Artificial Intelligence*, AAAI Press, 2019, pp. 93–99.

[74] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3913–3926, 2019.

[75] Z. Duan, K. Zhang, Z. Chen, *et al.*, "Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time," *IEEE Access*, vol. 7, pp. 127 816–127 832, 2019.

[76] W. Li, W. Tao, J. Qiu, X. Liu, X. Zhou, and Z. Pan, "Densely connected convolutional networks with attention LSTM for crowd flows prediction," *IEEE Access*, vol. 7, pp. 140 488–140 498, 2019.

[77] L. Mourad, H. Qi, Y. Shen, and B. Yin, "ASTIR: Spatio-temporal data mining for crowd flow prediction," *IEEE Access*, vol. 7, pp. 159–165, 2019.

[78] Y. Zhou, H. Chen, J. Li, Y. Wu, J. Wu, and L. Chen, "ST-Attn: Spatial-temporal attention mechanism for multi-step citywide crowd flow prediction," in *International Conference on Data Mining Workshops*, IEEE, 2019, pp. 609–614.

[79] C. Chen, K. Li, S. G. Teo, *et al.*, "Exploiting spatio-temporal correlations with multiple 3D convolutional neural networks for citywide vehicle flow prediction," in *International Conference on Data Mining*, IEEE, 2018, pp. 893–898.

[80] B. Wang, Z. Yan, J. Lu, G. Zhang, and T. Li, "Explore uncertainty in residual networks for crowds flow prediction," in *International Joint Conference on Neural Networks*, IEEE, 2018, pp. 1–7.

[81] Z. Duan, Y. Yang, K. Zhang, Y. Ni, and S. Bajgain, "Improved deep hybrid networks for urban traffic flow prediction using trajectory data," *IEEE Access*, vol. 6, pp. 31 820–31 827, 2018.

[82] K. Guo, Y. Hu, Z. Qian, *et al.*, "Optimized graph convolution recurrent neural network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1138–1149, 2021.

[83] K. Pholsena, L. Pan, and Z. Zheng, "Mode decomposition based Deep Learning model for multi-section traffic prediction," *World Wide Web*, vol. 23, pp. 2513–2527, 2020.

[84] X. Dai, R. Fu, E. Zhao, *et al.*, "DeepTrend 2.0: A light-weighted multi-scale traffic prediction model using detrending," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 142–157, 2019.

[85] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting," *Transportation Research Record*, vol. 2674, no. 9, pp. 473–488, 2020.

[86] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 922–929.

[87] D. Impedovo, V. Dentamaro, G. Pirlo, and L. Sarcinella, "TrafficWave: Generative Deep Learning architecture for vehicular traffic flow prediction," *Applied Sciences*, vol. 9, no. 24, p. 5504, 2019.

[88] Y. Zhang, T. Cheng, Y. Ren, and K. Xie, "A novel residual graph convolution Deep Learning model for short-term network-based traffic forecasting," *International Journal of Geographical Information Science*, vol. 34, no. 5, pp. 969–995, 2020.

[89] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid Deep Learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.

[90] Z. Wang, R. Zhu, M. Zheng, X. Jia, R. Wang, and T. Li, "A regularized LSTM network for short-term traffic flow prediction," in *International Conference on Information Science and Control Engineering*, IEEE, 2019, pp. 100–105.

[91] W. Chen, L. Chen, Y. Xie, W. Cao, Y. Gao, and X. Feng, "Multi-range attentive bicomponent graph convolutional network for traffic forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 3529–3536.

[92] 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.

[93] X. Dai, R. Fu, Y. Lin, L. Li, and F.-Y. Wang, "DeepTrend: A deep hierarchical neural network for traffic flow prediction," *arXiv preprint arXiv:1707.03213*, 2017.

[94] R. Asadi and A. C. Regan, "A spatio-temporal decomposition based deep neural network for time series forecasting," *Applied Soft Computing*, vol. 87, p. 105 963, 2020.

[95] R. Asadi and A. Regan, "A convolutional recurrent autoencoder for spatio-temporal missing data imputation," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 206–212.

[96] T. Wu, F. Chen, and Y. Wan, "Graph attention LSTM network: A new model for traffic flow forecasting," in *International Conference on Information Science and Control Engineering*, IEEE, 2018, pp. 241–245.

[97] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Youth Academic Annual Conference of Chinese Association of Automation*, IEEE, 2016, pp. 324–328.

[98] S. Du, T. Li, X. Gong, Y. Yang, and S. J. Horng, "Traffic flow forecasting based on hybrid Deep Learning framework," in *International Conference on Intelligent Systems and Knowledge Engineering*, IEEE, 2017, pp. 1–6.

[99] D. Kang, Y. Lv, and Y.-y. Chen, "Short-term traffic flow prediction with LSTM recurrent neural network," in *International Conference on Intelligent Transportation Systems*, IEEE, 2017, pp. 1–6.

[100] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with Conv-LSTM," in *International Conference on Wireless Communications and Signal Processing*, IEEE, 2017, pp. 1–6.

[101] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of Deep Learning for traffic data imputation," *Transportation research part C: emerging technologies*, vol. 72, pp. 168–181, 2016.

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

[103] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 804–17 815, 2020.

[104] Y. Wang, S. Fang, C. Zhang, S. Xiang, and C. Pan, "TVGCN: Time-variant graph convolutional network for traffic forecasting," *Neurocomputing*, vol. 471, pp. 118–129, 2021.

[105] A. Khaled, A. M. T. Elsir, and Y. Shen, "TFGAN: Traffic forecasting using generative adversarial network with multi-graph convolutional network," *Knowledge-Based Systems*, vol. 249, p. 108 990, 2022.

[106] Y. Chen, K. Li, C. K. Yeo, and K. Li, "Traffic forecasting with graph spatial-temporal position recurrent network," *Neural Networks*, vol. 162, pp. 340–349, 2023.

[107] L. Cai, M. Lei, S. Zhang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune LSTM network for short-term traffic flow forecasting," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 2, p. 023 135, 2020.

[108] Z. Abbas, A. Al-Shishtawy, S. Girdzijauskas, and V. Vlassov, "Short-term traffic prediction using long short-term memory neural networks," in *International Congress on Big Data*, IEEE, 2018, pp. 57–65.

[109] B. Zhao and X. Zhang, "A parallel-RES GRU architecture and its application to road network traffic flow forecasting," in *International Conference on Big Data Technologies*, 2018, pp. 79–83.

[110] Y. Jia, J. Wu, and M. Xu, "Traffic flow prediction with rainfall impact using a Deep Learning method," *Journal of advanced transportation*, vol. 2017, pp. 1–10, 2017.

[111] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *International Conference on Smart City*, IEEE, 2015, pp. 153–158.

[112] D. Yang, H.-M. Yang, P. Wang, and S.-J. Li, "MSAE: A multitask learning approach for traffic flow prediction using deep neural network," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, vol. 1, Springer, 2020, pp. 153–161.

[113] D. Yang, S. Li, Z. Peng, P. Wang, J. Wang, and H. Yang, "MF-CNN: Traffic flow prediction using convolutional neural network and multi-features fusion," *IEICE Transactions on Information and Systems*, vol. 102, no. 8, pp. 1526–1536, 2019.

[114] W. Zhao, Y. Gao, T. Ji, X. Wan, F. Ye, and G. Bai, "Deep temporal convolutional networks for short-term traffic flow forecasting," *IEEE Access*, vol. 7, pp. 114 496–114 507, 2019.

[115] S. Du, T. Li, Y. Yang, X. Gong, and S.-J. Horng, "An LSTM based encoder-decoder model for multistep traffic flow prediction," in *International Joint Conference on Neural Networks*, IEEE, 2019, pp. 1–8.

[116] E. L. Manibardo, I. Laña, and J. Del Ser, "Transfer Learning and Online Learning for traffic forecasting under different data availability conditions: Alternatives and pitfalls," in *International Conference on Intelligent Transportation Systems*, IEEE, 2020, pp. 1–6.

[117] K. Zhang, L. Wu, Z. Zhu, and J. Deng, "A multitask learning model for traffic flow and speed forecasting," *IEEE Access*, vol. 8, pp. 80 707–80 715, 2020.

[118] S. Du, T. Li, X. Gong, and S.-J. Horng, "A hybrid method for traffic flow forecasting using multimodal Deep Learning," *International Journal of Computational Intelligence Systems*, vol. 13, no. 1, pp. 85–97, 2020.

[119] Y. Zhang and G. Huang, "Traffic flow prediction model based on deep belief network and genetic algorithm," *Intelligent Transport Systems*, vol. 12, no. 6, pp. 533–541, 2018.

[120] M. Elhenawy and H. Rakha, "Stretch-wide traffic state prediction using discriminatively pre-trained deep neural networks," in *International Conference on Intelligent Transportation Systems*, IEEE, 2016, pp. 1065–1070.

[121] 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.

[122] M. Xu, W. Dai, C. Liu, *et al.*, "Spatial-temporal transformer networks for traffic flow forecasting," *arXiv preprint arXiv:2001.02908*, 2020.

[123] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, vol. 24, no. 3, pp. 736–755, 2020.

[124] Y. Liang, Z. Cui, Y. Tian, H. Chen, and Y. Wang, "A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation," *Transportation Research Record*, vol. 2672, no. 45, pp. 87–105, 2018.

[125] S. Kolidakis, G. Botzoris, V. Profillidis, and P. Lemonakis, "Road traffic forecasting: A hybrid approach combining artificial neural network with singular spectrum analysis," *Economic Analysis and Policy*, vol. 64, pp. 159–171, 2019.

[126] L. Mou, P. Zhao, H. Xie, and Y. Chen, "T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction," *IEEE Access*, vol. 7, pp. 98 053–98 060, 2019.

[127] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: A GRU-based Deep Learning approach," *Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, 2018.

[128] H.-F. Yang, T. S. Dillon, and Y.-P. P. Chen, "Optimized structure of the traffic flow forecasting model with a Deep Learning approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2371–2381, 2016.

[129] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 890–897.

[130] S. George and A. K. Santra, "An improved long short-term memory networks with Takagi-Sugeno fuzzy for traffic speed prediction considering abnormal traffic situation," *Computational Intelligence*, vol. 36, no. 3, pp. 964–993, 2020.

[131] C. Song, H. Lee, C. Kang, W. Lee, Y. B. Kim, and S. W. Cha, "Traffic speed prediction under weekday using convolutional neural networks concepts," in *Intelligent Vehicles Symposium*, IEEE, 2017, pp. 1293–1298.

[132] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A Deep Learning model for urban traffic flow prediction with traffic events mined from twitter," *World Wide Web*, vol. 24, no. 4, pp. 1345–1368, 2021.

[133] D. Liu, L. Tang, G. Shen, and X. Han, "Traffic speed prediction: An attention-based method," *Sensors*, vol. 19, no. 18, p. 3836, 2019.

[134] G. Shen, C. Chen, Q. Pan, S. Shen, and Z. Liu, "Research on traffic speed prediction by temporal clustering analysis and convolutional neural network with deformable kernels," *IEEE Access*, vol. 6, pp. 51 756–51 765, 2018.

[135] A. C. Piazzi and T. Tettamanti, "LSTM approach for spatial extension of traffic sensor points in urban road network," *European Association for Research in Transportation*,

[136] T. Zhang, J. Jin, H. Yang, H. Guo, and X. Ma, "Link speed prediction for signalized urban traffic network using a hybrid Deep Learning approach," in *Intelligent Transportation Systems Conference*, IEEE, 2019, pp. 2195–2200.

[137] S. Zhang, L. Zhou, X. Chen, L. Zhang, L. Li, and M. Li, "Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 10, pp. 1132–1147, 2020.

[138] Y. Jia, J. Wu, and Y. Du, "Traffic speed prediction using Deep Learning method," in *International Conference on Intelligent Transportation Systems*, IEEE, 2016, pp. 1217–1222.

[139] L. Li, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Traffic speed prediction for intelligent transportation system based on a deep feature fusion model," *Journal of Intelligent Transportation Systems*, vol. 23, no. 6, pp. 605–616, 2019.

[140] L. Li, B. Du, Y. Wang, L. Qin, and H. Tan, "Estimation of missing values in heterogeneous traffic data: Application of multimodal Deep Learning model," *Knowledge-Based Systems*, vol. 194, p. 592, 2020.

[141] Y. Kim, P. Wang, Y. Zhu, and L. Mihaylova, "A capsule network for traffic speed prediction in complex road networks," in *Sensor Data Fusion: Trends, Solutions, Applications*, IEEE, 2018, pp. 1–6.

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

[143] F. Sun, A. Dubey, and J. White, "Dxnat: Deep neural networks for explaining non-recurring traffic congestion," in *IEEE International Conference on Big Data*, IEEE, 2017, pp. 2141–2150.

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

[145] J. Guo, Y. Liu, Q. Yang, Y. Wang, and S. Fang, "GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model," *Transportmetrica A: Transport Science*, vol. 17, no. 2, pp. 190–211, 2021.

[146] Y. Shin and Y. Yoon, "Incorporating dynamicity of transportation network with multi-weight traffic graph convolutional network for traffic forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2082–2092, 2020.

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

[148] J. Cao, X. Guan, N. Zhang, X. Wang, and H. Wu, "A hybrid Deep Learning-based traffic forecasting approach integrating adjacency filtering and frequency decomposition," *IEEE Access*, vol. 8, pp. 81 735–81 746, 2020.

[149] G. Fusco, C. Colombaroni, and N. Isaenko, "Comparative analysis of implicit models for real-time short-term traffic predictions," *Intelligent Transport Systems*, vol. 10, no. 4, pp. 270–278, 2016.

[150] X. Yang, Y. Yuan, and Z. Liu, "Short-term traffic speed prediction of urban road with multi-source data," *IEEE Access*, vol. 8, pp. 87 541–87 551, 2020.

[151] L. Han, K. Zheng, L. Zhao, X. Wang, and X. Shen, "Short-term traffic prediction based on DeepCluster in large-scale road networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 12, pp. 12 301–12 313, 2019.

[152] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1720–1730.

[153] S. Sun, J. Chen, and J. Sun, "Traffic congestion prediction based on GPS trajectory data," *International Journal of Distributed Sensor Networks*, vol. 15, no. 5, 2019.

[154] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A Deep Learning approach considering spatio-temporal dependencies," *Transportation research part C: emerging technologies*, vol. 105, pp. 297–322, 2019.

[155] B. Liao, J. Zhang, M. Cai, *et al.*, "Dest-ResNet: A deep spatiotemporal residual network for hotspot traffic speed prediction," in *International Conference on Multimedia*, 2018, pp. 1883–1891.

[156] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong, "Traffic speed prediction and congestion source exploration: A Deep Learning method," in *International Conference on Data Mining*, IEEE, 2016, pp. 499–508.

[157] 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.

[158] Y. Zhang, S. Wang, B. Chen, and J. Cao, "GCGAN: Generative adversarial nets with graph CNN for network-scale traffic prediction," in *International Joint Conference on Neural Networks*, IEEE, 2019, pp. 1–8.

[159] Y. Zhang, S. Wang, B. Chen, J. Cao, and Z. Huang, "TrafficGAN: Network-scale deep traffic prediction with generative adversarial nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 219–230, 2019.

[160] N. Zhang, X. Guan, J. Cao, X. Wang, and H. Wu, "Wavelet-HST: A wavelet-based higher-order spatio-temporal framework for urban traffic speed prediction," *IEEE Access*, vol. 7, pp. 118 446–118 458, 2019.

[161] Y. Lee, H. Jeon, and K. Sohn, "Predicting short-term traffic speed using a deep neural network to accommodate citywide spatio - temporal correlations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1435–1448, 2021.

[162] C. Bratsas, K. Koupidis, J.-M. Salanova, K. Giannakopoulos, A. Kaloudis, and G. Aifadopoulou, "A comparison of Machine Learning methods for the prediction of traffic speed in urban places," *Sustainability*, vol. 12, no. 1, p. 142, 2020.

[163] Z. Liu, M. Huang, Z. Ye, and K. Wu, "DeepRTP: A deep spatio-temporal residual network for regional traffic prediction," in *International Conference on Mobile Ad-Hoc and Sensor Networks*, IEEE, 2019, pp. 291–296.

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

[165] L. Zhao, Y. Song, C. Zhang, *et al.*, "T-GCN a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[166] B. Liao, J. Zhang, C. Wu, *et al.*, "Deep sequence learning with auxiliary information for traffic prediction," in *International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 537–546.

[167] K. Chen, F. Chen, B. Lai, *et al.*, "Dynamic spatio-temporal graph-based CNNs for traffic flow prediction," *IEEE Access*, vol. 8, pp. 136–145, 2020.

[168] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, "Predicting citywide crowd flows using deep spatio-temporal residual networks," *Artificial Intelligence*, vol. 259, pp. 147–166, 2018.

[169] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, 2017.

[170] W. Elleuch, A. Wali, and A. M. Alimi, "Neural congestion prediction system for trip modelling in heterogeneous spatio-temporal patterns," *International Journal of Systems Science*, vol. 51, no. 8, pp. 1373–1391, 2020.

[171] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[172] X. Ma, H. Zhong, Y. Li, J. Ma, Z. Cui, and Y. Wang, "Forecasting transportation network speed using deep capsule networks with nested LSTM models," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[173] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical graph convolution network for traffic forecasting," in *AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 151–159.

[174] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, "A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection," *Transportation Research Part C: Emerging Technologies*, vol. 115, p. 102 622, 2020.

[175] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A Deep Learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.

[176] S. Ryu and D. Kim, "Intelligent highway traffic forecast based on Deep Learning and restructured road models," in *Computer Software and Applications Conference*, IEEE, vol. 2, 2019, pp. 110–114.

[177] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for forecasting network-wide traffic state with missing values," *Transportation Research Part C: Emerging Technologies*, vol. 118, p. 102 674, 2020.

[178] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A Deep Learning framework for traffic forecasting," in *International Joint Conference on Artificial Intelligence*, 2018, pp. 1–6.

[179] L. Wei, Z. Yu, Z. Jin, *et al.*, "Dual graph for traffic forecasting," *IEEE Access*, 2019.

[180] C. Zhang, J. James, and Y. Liu, "Spatial-temporal graph attention networks: A Deep Learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019.

[181] S. Shleifer, C. McCreery, and V. Chitters, "Incrementally improving graph wavenet performance on traffic prediction," *arXiv preprint arXiv:1912.07390*, 2019.

[182] E. L. Manibardo, I. Laña, J. L. Lobo, and J. Del Ser, "New perspectives on the use of Online Learning for congestion level prediction over traffic data," in *International Joint Conference on Neural Networks*, IEEE, 2020.

[183] X. Yang, Y. Zou, J. Tang, J. Liang, and M. Ijaz, "Evaluation of short-term freeway speed prediction based on periodic analysis using statistical models and Machine Learning models," *Journal of Advanced Transportation*, vol. 2020, 2020.

[184] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *International Joint Conference on Artificial Intelligence*, AAAI Press, 2019, pp. 1907–1913.

[185] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep Learning: A generic approach for extreme condition traffic forecasting," in *Conference on Data Mining*, SIAM, 2017, pp. 777–785.

[186] X. Wang, X. Guan, J. Cao, N. Zhang, and H. Wu, "Forecast network-wide traffic states for multiple steps ahead: A Deep Learning approach considering dynamic non-local spatial correlation and non-stationary temporal dependency," *Transportation Research Part C: Emerging Technologies*, vol. 119, p. 102 763, 2020.

[187] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018, pp. 1–16.

[188] Y. Sun, Y. Wang, K. Fu, Z. Wang, C. Zhang, and J. Ye, "Constructing geographic and long-term temporal graph for traffic forecasting," in *International Conference on Pattern Recognition*, IEEE, 2021, pp. 3483–3490.

[189] A. Fandango and R. P. Wiegand, "Towards investigation of iterative strategy for data mining of short-term traffic flow with recurrent neural networks," in *International Conference on Information System and Data Mining*, 2018, pp. 65–69.

[190] R. Jiang, Z. Wang, J. Yong, *et al.*, "Spatio-temporal meta-graph learning for traffic forecasting," in *AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 8078–8086.

[191] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using Deep Learning method," *Intelligent Transport Systems*, vol. 11, no. 9, pp. 531–536, 2017.

[192] Y. Adu-Gyamfi and M. Zhao, "Traffic speed prediction for urban arterial roads using deep neural networks," in *International Conference on Transportation and Development: Traffic and Freight Operations and Rail and Public Transit*, American Society of Civil Engineers Reston, VA, 2018, pp. 85–96.

[193] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 11, pp. 999–1016, 2018.

[194] T. Epelbaum, F. Gamboa, J.-M. Loubes, and J. Martin, "Deep Learning applied to road traffic speed forecasting," *arXiv preprint arXiv:1710.08266*, 2017.

[195] Z. He, C.-Y. Chow, and J.-D. Zhang, "STANN: A spatio–temporal attentive neural network for traffic prediction," *IEEE Access*, vol. 7, pp. 4795–4806, 2018.

[196] X. Cheng, R. Zhang, J. Zhou, and W. Xu, "Deeptransport: Learning spatial-temporal dependency for traffic condition forecasting," in *International Joint Conference on Neural Networks*, IEEE, 2018, pp. 1–8.

[197] R. Toncharoen and M. Piantanakulchai, "Traffic state prediction using convolutional neural network," in *International Joint Conference on Computer Science and Software Engineering*, IEEE, 2018, pp. 1–6.

[198] C. Xu, J. Ji, and P. Liu, "The station-free sharing bike demand forecasting with a Deep Learning approach and large-scale datasets," *Transportation research part C: emerging technologies*, vol. 95, pp. 47–60, 2018.

[199] W. Jiang and L. Zhang, "Geospatial data to images: A deep-learning framework for traffic forecasting," *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 52–64, 2018.

[200] H. Yao, F. Wu, J. Ke, *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *AAAI Conference on Artificial Intelligence*, vol. 32, 2018, pp. 1–8.

[201] M. Abdollahi, T. Khaleghi, and K. Yang, "An integrated feature learning approach using Deep Learning for travel time prediction," *Expert Systems with Applications*, vol. 139, p. 112 864, 2020.

[202] Z. Chen, B. Zhao, Y. Wang, Z. Duan, and X. Zhao, "Multitask learning and GCN-based taxi demand prediction for a traffic road network," *Sensors*, vol. 20, no. 13, p. 3776, 2020.

[203] Y. Sun, Y. Wang, K. Fu, *et al.*, "FMA-ETA: Estimating travel time entirely based on FFN with attention," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2021, pp. 3355–3359.

[204] X. Ning, L. Yao, X. Wang, B. Benatallah, F. Salim, and P. D. Haghighi, "Predicting citywide passenger demand via reinforcement learning from spatio-temporal dynamics," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2018, pp. 19–28.

[205] F. Rodrigues, I. Markou, and F. C. Pereira, "Combining time-series and textual data for taxi demand prediction in event areas: A Deep Learning approach," *Information Fusion*, vol. 49, pp. 120–129, 2019.

[206] D. Wang, Y. Yang, and S. Ning, "DeepSTCL: A deep spatio-temporal ConvLSTM for travel demand prediction," in *International Joint Conference on Neural Networks*, IEEE, 2018, pp. 1–8.

[207] D. Saxena and J. Cao, "D-GAN: Deep generative adversarial nets for spatio-temporal prediction," *arXiv preprint arXiv:1907.08556*, 2019.

[208] S. Liao, L. Zhou, X. Di, B. Yuan, and J. Xiong, "Large-scale short-term urban taxi demand forecasting using Deep Learning," in *Asia and South Pacific Design Automation Conference*, IEEE, 2018, pp. 1–5.

[209] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, "Scalable deep traffic flow neural networks for urban traffic congestion prediction," in *International Joint Conference on Neural Networks*, IEEE, 2017, pp. 2251–2258.

[210] H. Yi, K.-H. N. Bui, and H. Jung, "Implementing a Deep Learning framework for short term traffic flow prediction," in *International Conference on Web Intelligence, Mining and Semantics*, 2019, pp. 1–8.

[211] S. Zhang, Y. Yao, J. Hu, Y. Zhao, S. Li, and J. Hu, "Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks," *Sensors*, vol. 19, no. 10, p. 2229, 2019.

[212] X. Ran, Z. Shan, Y. Fang, and C. Lin, "An LSTM-based method with attention mechanism for travel time prediction," *Sensors*, vol. -19, no. 4, p. 861, 2019.

[213] R. Barlovic, "Traffic jams: Cluster formation in low-dimensional cellular automata models for highway and city traffic," Ph.D. dissertation, Standort Duisburg university, 2003.

[214] C. I. Van Hinsbergen, F. Sanders, *et al.*, "Short term traffic prediction models," in *World Congress on Intelligent Transport Systems*, 2007, pp. 1–18.

[215] M. J. Cassidy and R. L. Bertini, "Some traffic features at freeway bottlenecks," *Transportation Research Part B: Methodological*, vol. 33, no. 1, pp. 25–42, 1999.

[216] Y. Yue and A. G.-O. Yeh, "Spatiotemporal traffic-flow dependency and short-term traffic forecasting," *Environment and Planning B: Planning and Design*, vol. 35, no. 5, pp. 762–771, 2008.

[217] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *International Conference on Knowledge Discovery and Data Mining*, 2011, pp. 1010–1018.

[218] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[219] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[220] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.

[221] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2017, pp. 1–12.

[222] H. Cheng, P.-N. Tan, J. Gao, and J. Scripps, "Multistep-ahead time series prediction," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2006, pp. 765–774.

[223] *Caltrans, Performance Measurement System*, http://pems.dot.ca.gov, Accessed: 2023-11-06.

[224] H. V. Jagadish, J. Gehrke, A. Labrinidis, *et al.*, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[225] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 2013, vol. 184.

[226] P. Næss and A. Strand, "Traffic forecasting at 'strategic', 'tactical' and 'operational' level: A differentiated methodology is necessary," *disP-The Planning Review*, vol. 51, no. 2, pp. 41–48, 2015.

[227] C Lamboley, J. Santucci, and M Danech-Pajouh, "24 or 48 hour advance traffic forecast in urban and periurban environments: The example of Paris," in *World Congress on Intelligent Transport Systems*, 1997.

[228] K. Jha, N. Sinha, S. S. Arkatkar, and A. K. Sarkar, "A comparative study on application of time series analysis for traffic forecasting in India: Prospects and limitations," *Current Science*, pp. 373–385, 2016.

[229] I. Laña, E. Villar-Rodriguez, U. Etxegarai, I. Oregi, and J. Del Ser, "A question of trust: Statistical characterization of long-term traffic estimations for their improved actionability," in *Intelligent Transportation Systems Conference*, IEEE, 2019, pp. 1922–1928.

[230] J. N. Ivan, W. M. Eldessouki, M Zhao, and F. Guo, "Estimating link traffic volumes by month, day of week and time of day," Tech. Rep., 2002.

[231] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine Learning algorithm validation with a limited sample size," *PloS one*, vol. 14, no. 11, e0224365, 2019.

[232] W. Laverty and I. Kelly, "Cyclical calendar and lunar patterns in automobile property accidents and injury accidents," *Perceptual and motor skills*, vol. 86, no. 1, pp. 299–302, 1998.

[233] W. Weijermars and E. C. van Berkum, "Daily flow profiles of urban traffic," *WIT Transactions on The Built Environment*, vol. 75, pp. 1–10, 2004.

[234] T. H. Maze, M. Agarwal, and G. Burchett, "Whether weather matters to traffic demand, traffic safety, and traffic operations and flow," *Transportation research record*, vol. 1948, no. 1, pp. 170–176, 2006.

[235] F. M. Awan, R. Minerva, and N. Crespi, "Improving road traffic forecasting using air pollution and atmospheric data: Experiments based on LSTM recurrent neural networks," *Sensors*, vol. 20, no. 13, p. 3749, 2020.

[236] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric Deep Learning: Going beyond Euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[237] R. L. Bertini and M. T. Leal, "Empirical study of traffic features at a freeway lane drop," *Journal of Transportation Engineering*, vol. 131, no. 6, pp. 397–407, 2005.

[238] T. Meng, X. Jing, Z. Yan, and W. Pedrycz, "A survey on Machine Learning for data fusion," *Information Fusion*, vol. 57, pp. 115–129, 2020.

[239] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[240] F. Yang, C. Wang, X. Zuo, R. Zhong, and F. Xiang, "Short-term traffic flow prediction based on echo state networks," *Adv. Inf. Sci. Serv. Sci.*, vol. 4, no. 9, pp. 269–277, 2012.

[241] J. Lou, Y. Jiang, Q. Shen, R. Wang, and Z. Li, "Probabilistic regularized extreme learning for robust modeling of traffic flow forecasting," *IEEE Transactions on Neural Networks and Learning Systems, in press*, vol. 34, no. 4, pp. 1732–1741, 2023.

[242] J. Del Ser, I. Laña, M. N. Bilbao, and E. I. Vlahogianni, "Road traffic forecasting using stacking ensembles of echo state networks," in *Intelligent Transportation Systems Conference*, IEEE, 2019, pp. 2591 –2597.

[243] J. Del Ser, I. Laña, E. L. Manibardo, *et al.*, "Deep echo state networks for short-term traffic forecasting: Performance comparison and statistical assessment," in *Intelligent Transportation Systems Conference*, IEEE, 2020.

[244] S. Yang, J. Wu, Y. Du, Y. He, and X. Chen, "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *Journal of Sensors*, vol. 2017, pp. 1–16, 2017.

[245] Z. Lu, J. Xia, M. Wang, Q. Nie, and J. Ou, "Short-term traffic flow forecasting via multi-regime modeling and ensemble learning," *Applied Sciences*, vol. 10, no. 1, p. 356, 2020.

[246] W. Li, C. Yang, and S. E. Jabari, "Short-term traffic forecasting using high-resolution traffic data," in *International Conference on Intelligent Transportation Systems*, IEEE, 2020, pp. 1–6.

[247] J. S. Angarita-Zapata, A. D. Masegosa, and I. Triguero, "Evaluating automated Machine Learning on supervised regression traffic forecasting problems," in *Computational Intelligence in Emerging Technologies for Engineering Applications*, Springer, 2020, pp. 187–204.

[248] D. T. Tran, S. Kiranyaz, M. Gabbouj, and A. Iosifidis, "Heterogeneous multilayer generalized operational perceptron," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 3, pp. 710–724, 2019.

[249] W. Maass, "Liquid state machines: Motivation, theory, and applications," in *Computability in context: computation and logic in the real world*, World Scientific, 2011, pp. 275–296.

[250] Z. Zhang, Y. Yuan, and X. Yang, "A hybrid Machine Learning approach for freeway traffic speed estimation," *Transportation Research Record*, vol. 2674, no. 10, pp. 68–78, 2020.

[251] I. Laña, J. J. Sanchez-Medina, E. I. Vlahogianni, and J. Del Ser, "From data to actions in Intelligent Transportation Systems: A prescription of functional requirements for model actionability," *Sensors*, vol. 21, no. 4, p. 1121, 2021.

[252] L. A. M. Matias, "On improving operational planning and control in public transportation networks using streaming data: A Machine Learning approach," Ph.D. dissertation, Universidade do Porto (Portugal), 2015.

[253] C. Buchanan, *Traffic in Towns: A study of the long term problems of traffic in urban areas*. Routledge, 2015.

[254] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *International Conference on Advances in Geographic Information Systems*, 2013, pp. 344–353.

[255] I. Laña, J. L. Lobo, E. Capecci, J. Del Ser, and N. Kasabov, "Adaptive long-term traffic state estimation with evolving spiking neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 126–144, 2019.

[256] R. Mena Yedra, J. Casas Vilaró, T. Djukic, and R. Gavaldà Mestre, "Improving adaptation and interpretability of a short-term traffic forecasting system," in *Australasian Transport Research Forum*, 2017, pp. 1–15.

[257] T. Wu, K. Xie, D. Xinpin, and G. Song, "A online boosting approach for traffic flow forecasting under abnormal conditions," in *International Conference on Fuzzy Systems and Knowledge Discovery*, IEEE, 2012, pp. 2555–2559.

[258] M. J. Procopio, J. Mulligan, and G. Grudic, "Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments," *Journal of Field Robotics*, vol. 26, no. 2, pp. 145–175, 2009.

[259] M. S. Nicolaisen and P. A. Driscoll, "Ex-post evaluations of demand forecast accuracy: A literature review," *Transport Reviews*, vol. 34, no. 4, pp. 540–557, 2014.

[260] P. Parthasarathi and D. Levinson, "Post-construction evaluation of traffic forecast accuracy," *Transport Policy*, vol. 17, no. 6, pp. 428–443, 2010.

[261] C. Yang, A. Chen, X. Xu, and S. Wong, "Sensitivity-based uncertainty analysis of a combined travel demand model," *Transportation Research Part B: Methodological*, vol. 57, pp. 225–244, 2013.

[262] S. Rasouli and H. J. Timmermans, "Using ensembles of decision trees to predict transport mode choice decisions: Effects on predictive success and uncertainty estimates," *European Journal of Transport and Infrastructure Research*, vol. 14, no. 4, pp. 412–424, 2014.

[263] M. Welde and J. Odeck, "Do planners get it right? The accuracy of travel demand forecasting in Norway," *European Journal of Transport and Infrastructure Research*, vol. 11, no. 1, pp. 1–16, 2011.

[264] A. Matas, J.-L. Raymond, and A. Ruiz, "Traffic forecasts under uncertainty and capacity constraints," *Transportation*, vol. 39, no. 1, pp. 1–17, 2012.

[265] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[266] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.

[267] A. Barredo-Arrieta, I. Laña, and J. Del Ser, "What lies beneath: A note on the explainability of black-box Machine Learning models for road traffic forecasting," in *International Conference on Intelligent Transportation Systems*, IEEE, 2019, pp. 2232–2237.

[268] B. S. Kerner, "Congested traffic flow: Observations and theory," *Transportation Research Record*, vol. 1678, no. 1, pp. 160–167, 1999.

[269] M. Treiber and D. Helbing, "Explanation of observed features of self-organization in traffic flow," *arXiv preprint cond-mat/9901239*, 1999.

[270] A. Fernandez, F. Herrera, O. Cordon, M. J. del Jesus, and F. Marcelloni, "Evolutionary fuzzy systems for explainable artificial intelligence: Why, when, what for, and where to?" *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 69–81, 2019.

[271] J.-H. Xue and D. M. Titterington, "Comment on "on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes"," *Neural processing letters*, vol. 28, no. 3, p. 169, 2008.

[272] A. Halati, H. Lieu, and S. Walker, "CORSIM corridor traffic simulation model," in *Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities Urban Transportation Division, ASCE; Highway Division.*, 1997.

[273] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," in *Fundamentals of traffic simulation*, Springer, 2010, pp. 63–93.

[274] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO simulation of urban mobility: An overview," in *International Conference on Advances in System Simulation*, ThinkMind, 2011.

[275] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[276] N. Gao, H. Xue, W. Shao, *et al.*, "Generative adversarial networks for spatio-temporal data: A survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 2, pp. 1–25, 2022.

[277] Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan, "Multivariate time series imputation with generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 1596–1607.

[278] T. Matias, F. Souza, R. Araújo, and C. H. Antunes, "Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine," *Neurocomputing*, vol. 129, pp. 428–436, 2014.

[279] J. Xie, S.-Y. Liu, and J.-X. Chen, "A framework for distributed semi-supervised learning using single-layer feedforward networks," *Machine Intelligence Research*, vol. 19, no. 1, pp. 63–74, 2022.

[280] H. Ramchoun, Y. Ghanou, M. Ettaouil, and M. A. Janati Idrissi, "Multilayer perceptron: Architecture optimization and training," vol. 4, no. 1, pp. 26–30, 2016.

[281] A. F. Agarap, "Deep Learning using rectified linear units (ReLU)," *arXiv preprint arXiv:1803.08375*, pp. 1–7, 2018.

[282] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *International Conference on Engineering and Technology*, IEEE, 2017, pp. 1–6.

[283] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, vol. -33, no. 12, pp. 6999–7019, 2022.

[284] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci, and M. Gabbouj, "1-D convolutional neural networks for signal processing applications," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2019, pp. 8360–8364.

[285] U. Erdenebayar, H. Kim, J.-U. Park, D. Kang, and K.-J. Lee, "Automatic prediction of atrial fibrillation based on convolutional neural network using a short-term normal electrocardiogram signal," *Journal of Korean medical science*, vol. 34, no. 7, pp. 1–10, 2019.

[286] S. Harbola and V. Coors, "One dimensional convolutional neural network architectures for wind prediction," *Energy Conversion and Management*, vol. 195, pp. 70–75, 2019.

[287] D. Han, J. Chen, and J. Sun, "A parallel spatiotemporal Deep Learning network for highway traffic flow forecasting," *International Journal of Distributed Sensor Networks*, vol. 15, no. 2, p. 1542, 2019.

[288] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, PMLR, 2013, pp. 1310–1318.

[289] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132 306, 2020.

[290] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *The European Physical Journal Special Topics*, vol. 228, no. 10, pp. 2313–2324, 2019.

[291] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of Deep Learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[292] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, p. 11, 2017.

[293] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015, pp. 1–15.

[294] L. Zhang and P. N. Suganthan, "A survey of randomized algorithms for training neural networks," *Information Sciences*, vol. 364, pp. 146–155, 2016.

[295] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019.

[296] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[297] W. F. Schmidt, M. A. Kraaijveld, R. P. Duin, *et al.*, "Feed forward neural networks with random weights," in *International Conference on Pattern Recognition*, IEEE Computer Society Press, 1992, p. 1.

[298] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[299] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *International Joint Conference on Neural Networks*, IEEE, vol. 2, 2004, pp. 985–990.

[300] J. Wang, S. Lu, S.-H. Wang, and Y.-D. Zhang, "A review on extreme learning machine," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 41 611–41 660, 2022.

[301] Y. Yi, J. Dai, C. Wang, *et al.*, "An effective framework using spatial correlation and extreme learning machine for moving cast shadow detection," *Applied Sciences*, vol. 9, no. 23, p. 5042, 2019.

[302] Z. Geng, J. Dong, J. Chen, and Y. Han, "A new self-organizing extreme learning machine soft sensor model and its applications in complicated chemical processes," *Engineering Applications of Artificial Intelligence*, vol. 62, pp. 38–50, 2017.

[303] Z. Geng, S. Zhao, G. Tao, and Y. Han, "Early warning modeling and analysis based on analytic hierarchy process integrated extreme learning machine (AHP-ELM): Application to food safety," *Food Control*, vol. 78, pp. 33–42, 2017.

[304] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," *Applied Soft Computing*, vol. 72, pp. 79–89, 2018.

[305] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised Machine Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1108–1120, 2015.

[306] B. Liu, S. Yan, H. You, *et al.*, "An ensembled RBF extreme learning machine to forecast road surface temperature," in *International Conference on Machine Learning and Applications*, IEEE, 2017, pp. 977–980.

[307] L. Oneto, E. Fumeo, G. Clerico, *et al.*, "Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 10, pp. 2754–2767, 2017.

[308] Y.-m. Xing, X.-j. Ban, and R. Liu, "A short-term traffic flow prediction method based on kernel extreme learning machine," in *International Conference on Big Data and Smart Computing*, IEEE, 2018, pp. 533–536.

[309] L. Zhang and P. N. Suganthan, "A comprehensive evaluation of random vector functional link networks," *Information Sciences*, vol. 367, pp. 1094–1105, 2016.

[310] A. K. Malik, R. Gao, M. Ganaie, M. Tanveer, and P. N. Suganthan, "Random vector functional link network: Recent developments, applications, and future directions," *Applied Soft Computing*, vol. 143, p. 110 377, 2023.

[311] X. Qiu, P. N. Suganthan, and G. A. Amaratunga, "Electricity load demand time series forecasting with empirical mode decomposition based random vector functional link network," in *International Conference on Systems, Man, and Cybernetics*, IEEE, 2016, pp. 001 394–001 399.

[312] A. Aggarwal and M. Tripathi, "Short-term solar power forecasting using random vector functional link (RVFL) network," in *Ambient Communications and Computer Systems*, Springer, 2018, pp. 29–39.

[313] X. Qiu, Y. Ren, P. N. Suganthan, and G. A. Amaratunga, "Short-term wind power ramp forecasting with empirical mode decomposition based ensemble learning techniques," in *Symposium Series on Computational Intelligence*, IEEE, 2017, pp. 1–8.

[314] X. Qiu, P. N. Suganthan, and A. G. Amaratunga, "Ensemble incremental random vector functional link network for short-term crude oil price forecasting," in *Symposium Series on Computational Intelligence*, IEEE, 2018, pp. 1758–1763.

[315] D. Husmeier, "Random vector functional link networks," in *Neural Networks for Conditional Probability Estimation*, 1999, pp. 87–97.

[316] M. Ganaie, M Tanveer, and P. Suganthan, "Minimum variance embedded random vector functional link network," in *International Conference on Neural Information Processing*, Springer, 2020, bibrangedash 412–419.

[317] M Tanveer, M. Ganaie, and P. Suganthan, "Ensemble of classification models with weighted functional link network," *Applied Soft Computing*, vol. 107, p. 107 322, 2021.

[318] Q. Shi, R. Katuwal, P. N. Suganthan, and M. Tanveer, "Random vector functional link neural network based ensemble Deep Learning," *Pattern Recognition*, vol. 117, p. 107 978, 2021.

[319] N. Geroliminis and J. Sun, "Properties of a well-defined macroscopic fundamental diagram for urban traffic," *Transportation Research Part B: Methodological*, vol. 45, no. 3, pp. 605–617, 2011.

[320] F. K. Adamidis, E. G. Mantouka, and E. I. Vlahogianni, "Effects of controlling aggressive driving behavior on network-wide traffic flow and emissions," *International Journal of Transportation Science and Technology*, vol. 9, no. 3, pp. 263–276, 2020.

[321] *Madrid Open Data Portal*, http://datos.madrid.es, Accessed: 2023-11-06.

[322] *NYC Real Time Traffic Speed Data Feed*, https://www.kaggle.com/crailtap/nyc-real-time-traffic-speed-data-feed, Accessed: 2023-11-06.

[323] *Seattle Inductive Loop Detector Dataset*, https://github.com/zhiyongc/Seattle-Loop-Data, Accessed: 2023-11-06.

[324] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A Python library for optimizing the hyperparameters of Machine Learning algorithms," in *Python in Science Conference*, Citeseer, 2013, pp. 13–20.

[325] A. Di Bucchianico, "Coefficient of determination (R2)," *Encyclopedia of statistics in quality and reliability*, 2008.

[326] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[327] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[328] A. Benavoli, G. Corani, J. Demšar, and M. Zaffalon, "Time for a change: A tutorial for comparing multiple classifiers through bayesian analysis," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2653–2688, 2017.

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

[330] H. Zhou and H. Gao, "The impact of urban morphology on urban transportation mode: A case study of Tokyo," *Case Studies on Transport Policy*, vol. 8, no. 1, pp. 197–205, 2020.

[331] Q. Hu, R. Zhang, and Y. Zhou, "Transfer Learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–95, 2016.

[332] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer Learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[333] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of Transfer Learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

[334] F. Zhuang, Z. Qi, K. Duan, *et al.*, "A comprehensive survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[335] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[336] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 293–11 302.

[337] H. Chen, G. Chen, Q. Lu, and L. Peng, "MMSE-based optimized transfer strategy for transfer prediction of parking data," in *IEEE Intelligent Transportation Systems Conference*, 2019, pp. 407–412.

[338] J. James, "Online traffic speed estimation for urban road networks with few data: A Transfer Learning approach," in *Intelligent Transportation Systems Conference*, IEEE, 2019, pp. 4024–4029.

[339] J. J. Q. Yu and J. Gu, "Real-time traffic speed estimation with graph convolutional generative autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3940–3951, 2019.

[340] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, "Online Learning: A comprehensive survey," *Neurocomputing*, vol. 459, pp. 249–289, 2021.

[341] X. Niu, Y. Zhu, Q. Cao, X. Zhang, W. Xie, and K. Zheng, "An online-traffic-prediction based route finding mechanism for smart city," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 970 256, 2015.

[342] Y. Chen, Y. Lv, Z. Li, and F.-Y. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *IEEE International Conference on Intelligent Transportation Systems*, 2016, pp. 132–137.

[343] M. Seufert, P. Casas, N. Wehner, L. Gang, and K. Li, "Stream-based Machine Learning for real-time qoe analysis of encrypted video streaming traffic," in *Conference on Innovation in Clouds, Internet and Networks*, IEEE, 2019, pp. 76–81.

[344] E. L. Manibardo, I. Laña, and J. Del Ser, "Change detection and adaptation strategies for long-term estimation of pedestrian flows," in *International Intelligent Transportation Systems Conference*, - IEEE, 2021, pp. 1867–1874.

[345] I. Laña, I. Oregi, and J. Del Ser, "Soft sensing methods for the generation of plausible traffic data in sensor-less locations," in *International Intelligent Transportation Systems Conference*, IEEE, 2021, pp. 3183–3189.

[346] T. Brinkhoff, "Generating traffic data," *IEEE Computer Society Technical Committee on Data Engineering*, vol. 26, no. 2, pp. 19–25, 2003.

[347] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-simulation of urban mobility," *International Journal on Advances in Systems and Measurements*, vol. 5, no. 3&4, 2012.

[348] L. E. Owen, Y. Zhang, L. Rao, and G. McHale, "Traffic flow simulation using CORSIM," in *2000 Winter Simulation Conference Proceedings*, IEEE, vol. 2, 2000, pp. 1143–1147.

[349] K. W Axhausen, A. Horni, and K. Nagel, *The multi-agent transport simulation MATSim*. Ubiquity Press, 2016.

[350] P. A. Lopez, M. Behrisch, L. Bieker-Walz, *et al.*, "Microscopic traffic simulation using SUMO," in *International Conference on Intelligent Transportation Systems*, IEEE, 2018, pp. 2575–2582.

[351] M. K. Reilly, M. P. O'Mara, and K. C. Seto, "From Bangalore to the Bay area: Comparing transportation and activity accessibility as drivers of urban growth," *Landscape and Urban Planning*, vol. 92, no. 1, pp. 24–33, 2009.

[352] T.-H. Wen, P.-C. Lai, *et al.*, "Understanding the topological characteristics and flow complexity of urban traffic congestion," *Physica A: Statistical Mechanics and its Applications*, vol. 473, pp. 166–177, 2017.

[353] S. Wang, D. Yu, X. Ma, and X. Xing, "Analyzing urban traffic demand distribution and the correlation between traffic flow and the built environment based on detector data and POIs," *European Transport Research Review*, vol. 10, no. 2, pp. 1–17, 2018.

[354] N. Geroliminis and C. F. Daganzo, "Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 759–770, 2008.

[355] L. Ambühl, A. Loder, L. Leclercq, and M. Menendez, "Disentangling the city traffic rhythms: A longitudinal analysis of MFD patterns over a year," *Transportation Research Part C: Emerging Technologies*, vol. 126, p. 103 065, 2021.

[356] J. A. Laval and F. Castrillón, "Stochastic approximations for the macroscopic fundamental diagram of urban networks," *Transportation Research Procedia*, vol. 7, pp. 615–630, 2015.

[357] W. Wong, S. Wong, and H. X. Liu, "Network topological effects on the macroscopic fundamental diagram," *Transportmetrica B: Transport Dynamics*, vol. 9, no. 1, pp. 376–398, 2021.

[358] I. I. Sirmatel and N. Geroliminis, "Stabilization of city-scale road traffic networks via macroscopic fundamental diagram-based model predictive perimeter control," *Control Engineering Practice*, vol. 109, p. 104 750, 2021.

[359] S. Landolt, T. Wambsganss, and M. Söllner, "A taxonomy for Deep Learning in natural language processing," International Conference on System Sciences, 2021.

[360] M. Hassaballah and A. I. Awad, *Deep Learning in computer vision: principles and applications*. CRC Press, 2020.

[361] A. Darmochwał, "The Euclidean space," *Formalized Mathematics*, vol. 2, no. 4, pp. 599–603, 1991.

[362] F. Harary, *Graph theory*. CRC Press, 2018.

[363] J. R. Munkres, *Analysis on manifolds*. CRC Press, 2018.

[364] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2020.

[365] Z. Wang, X. Su, and Z. Ding, "Long-term traffic prediction based on LSTM encoder-decoder architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6561–6571, 2020.

[366] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.

[367] OpenStreetMap contributors, *Planet dump retrieved from https: planet .osm.org*, https://www.openstreetmap.org, 2017.

[368] D. Sandfelder, P. Vijayan, and W. L. Hamilton, "Ego-GNNs: Exploiting ego structures in graph neural networks," in *International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2021, pp. 8523–8527.

[369] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos National Lab, Tech. Rep., 2008.

[370] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994.

[371] P. Holme, "Congestion and centrality in traffic flow on complex networks," *Advances in Complex Systems*, vol. 6, no. 02, pp. 163–176, 2003.

[372] T. W. Valente, K. Coronges, C. Lakon, and E. Costenbader, "How correlated are network centrality measures?" *Connections*, vol. 28, no. 1, p. 16, 2008.

[373] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[374] D. Braess, A. Nagurney, and T. Wakolbinger, "On a paradox of traffic planning," *Transportation Science*, vol. 39, no. 4, pp. 446–450, 2005.

[375] J. Tabak, *Geometry: the language of space and form*. Infobase Publishing, 2014.

[376] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[377] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[378] B. Han, X. Yu, and E. Kwon, "A self-sensing carbon nanotube/cement composite for traffic monitoring," *Nanotechnology*, vol. 20, no. 44, p. 445 501, 2009.

[379] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[380] P. B. Nemenyi, *Distribution-free multiple comparisons*. Princeton University, 1963.

[381] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future," in *International Conference on the Applications of Digital Information and Web Technologies*, IEEE, 2014, pp. 232–238.

[382] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[383] A. Benítez-Hidalgo, A. J. Nebro, J. García-Nieto, I. Oregi, and J. Del Ser, "JMetalPy: A python framework for multi-objective optimization with metaheuristics," *Swarm and Evolutionary Computation*, p. 100 598, 2019.

[384] I. I. Olabarrieta and I. Laña, "Effect of soccer games on traffic, study case: Madrid," in *International Conference on Intelligent Transportation Systems*, IEEE, 2020, pp. 1–5.

[385] S. Prontri, P. Wuttidittachotti, and S. Thajchayapong, "Traffic signal control using fuzzy logic," in *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, IEEE, 2015, pp. 1–6.

[386] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[387] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[388] A. Ramesh, M. Pavlov, G. Goh, *et al.*, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8821–8831.

[389] A. Garza and M. Mergenthaler-Canseco, "TimeGPT-1," *arXiv preprint arXiv:2310.03589*, 2023.