



Universitat Autònoma de Barcelona

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  [http://cat.creativecommons.org/?page\\_id=184](http://cat.creativecommons.org/?page_id=184)

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**Departament de Telecomunicació i d'Enginyeria de Sistemes**

Escola d'Enginyeria — Universitat Autònoma de Barcelona

# Contributions to Intelligent Transportation Systems

Big data analytics for reliable and valuable data

**Guillem Boquet Pujadas, Ph.D. Dissertation, 2021**

Supervised by José López Vicario, Antoni Morell Pérez, Javier Serrano García



*A la meva mare,  
tres voltes rebel*



---

# Abstract

---

Transportation industry has entered the era of big data. Part of the data disseminated by connected vehicles and infrastructure is being exploited by Intelligent Transport Systems (ITS), advanced applications in which information and communication technologies are applied in the field of road transport traffic management. In the upcoming future, all road vehicles are likely to communicate with one another and the surrounding infrastructure, for example, to warn others about traffic incidents or poor road conditions. But, the connectivity and data analytics requirements for the envisaged use cases are far from covered.

Dedicated Short Range Communication (DSRC) is a higher layer standard based on the evolution of IEEE 802.11p Wi-Fi, one of the main technologies that support the first generation of vehicle-to-everything (V2X) communication. The first part of this dissertation addresses the improvement of IEEE 802.11p direct vehicular-to-infrastructure communication in the ITS data acquisition layer, which suffers from a well-known scalability problem. The analysis carried out concludes that the data dissemination of standardized protocols is not reliable enough to support safety applications that depend on ITS roadside units located in intersection areas. To solve this, novel infrastructure-oriented criteria is proposed to adapt the communication parameters and an intersection assistance protocol is designed in compliance with the standards to increase the reliability of the data acquisition layer up to the point where safety applications can be implemented.

As ITS data acquisition layer produces massive amounts of data, it requires data aggregation and processing in the data analytics and application layer to enable more advanced use cases, mission-critical applications that have the potential impact to reduce problems such as road safety, pollution, traffic congestion and transportation costs. The second part of the dissertation proposes a generative deep learning model that can be used in an unsupervised manner to solve multiple ITS challenges. Big data collected by ITS is exploited and transformed to an asset for safety applications and decision-making, without the need for additional knowledge nor labeled data. The model allows to efficiently compress traffic data and forecast, impute missing values, select the best data and models for a specific problem and detect anomalous traffic data at the same time.

The last part of the dissertation is motivated by the growing concern generated by the efficiency of ITS solutions and the large amount of data expected to be processed. The presented algorithm allows to automatically and efficiently derive the minimum expression architecture of the model that provides

## Abstract

---

maximal compressed representations that inform about the original traffic data. In this way, the performance of the subsequent ITS traffic forecasting system is not adversely affected, but benefits from data being represented with fewer dimensions, which is vitally important in the age of big data. The basis of the algorithm is taken from theoretical concepts of Information Theory applied to neural networks, going a step beyond the current available methods that are based on trial and error.

---

## Acknowledgements

---

Per acabar aquesta etapa m'agradaria donar les gràcies a aquelles persones que m'hi han acompanyat. En especial a José Vicario i Toni Morell per la seva ajuda, coneixement i comprensió. A Javier Serrano i el grup de recerca WIN (<http://win.uab.cat>) pel seu suport. A Pedro de Paco per obrir el camí. A les meves companyes i amigues de despatx, Ivan, Edith i Edwar. A la meua família, Montse, Dani, Aleix, Anna i Aris. A les meves amigues de fora de la universitat, entre elles, Anna, Gioia, Victor, Adri i Ori. Al grup de recerca WiNe (<http://wine.rdi.uoc.edu>) per deixar-me acabar d'escriure-la.

Espero haver-vos demostrat l'agraïment en persona. Tanmateix, aquí queda escrit per si un dia deixo de banda la humilitat i penso que tota aquesta feina l'he aconseguida sol.





---

# Contents

---

Abstract	iii
Acknowledgements	v
Contents	vii
List of Figures	ix
List of Tables	xi
List of Algorithms	xiii
<b>I Dissertation Summary</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Motivation . . . . .	6
1.2 Objectives . . . . .	8
1.3 Structure and Rationale . . . . .	9
1.4 Relevance of Publications . . . . .	11
<b>2 Enhancement of Vehicle-to-Infrastructure Communication in the ITS Data Acquisition Layer</b>	<b>13</b>
2.1 Related Work . . . . .	13
2.2 IEEE 802.11p Overview . . . . .	15
2.2.1 Scalability Problem . . . . .	15
2.3 Vehicle-to-Infrastructure Communication in Intersection Areas	16
2.3.1 Network Behavior . . . . .	17
2.3.2 Protocol Evaluation . . . . .	21
2.3.3 PHY & MAC Adaptation Criteria . . . . .	25
2.3.4 Intersection Assistance Protocol . . . . .	31
<b>3 Deep Learning Solution for Exploiting Traffic Data in the ITS Data Analytics Layer</b>	<b>37</b>
3.1 Related Work . . . . .	37
3.2 Road Traffic Forecast . . . . .	40
3.2.1 Forecasting Problem . . . . .	41

## Contents

---

3.2.2	Probabilistic Approach . . . . .	41
3.3	Background on Probability . . . . .	41
3.3.1	Learning from Data . . . . .	41
3.3.2	A Maximum Likelihood Problem . . . . .	43
3.3.3	Latent Space of Traffic Data . . . . .	45
3.3.4	The Variational Inference Way . . . . .	46
3.3.5	Evidence Lower Bound Objective . . . . .	47
3.4	ITS Big Data Analytics Solution . . . . .	48
3.4.1	Neural Network Parametrization . . . . .	49
3.4.2	Variational Autoencoder . . . . .	49
3.4.3	Exact Model Definition . . . . .	52
3.4.4	Model Implementation . . . . .	54
3.5	Applicability in ITS Use Cases . . . . .	57
3.5.1	Missing Data Imputation . . . . .	59
3.5.2	Dimension Reduction . . . . .	62
3.5.3	Model & Data Selection . . . . .	64
3.5.4	Anomaly Detection . . . . .	68
<b>4</b>	<b>Theoretical-Tuning of Deep Learning Architectures of ITS Solutions</b> . . . . .	<b>71</b>
4.1	Related Work . . . . .	71
4.2	Information Perspective of the Autoencoder . . . . .	73
4.2.1	Autoencoder Role in ITS . . . . .	73
4.2.2	Mutual Information . . . . .	74
4.2.3	Data Processing Inequalities . . . . .	74
4.3	Information-Theoretic Tuning of the Architecture . . . . .	75
4.3.1	Information Plane . . . . .	75
4.3.2	Entropy of the Subspace as KPI . . . . .	77
4.3.3	The Sufficient Autoencoder Algorithm . . . . .	78
4.3.4	Mutual Information Estimation . . . . .	81
4.4	Experimentation . . . . .	82
4.4.1	Evaluation Model . . . . .	82
4.4.2	Information Plane Validation . . . . .	83
4.4.3	Algorithm Validation . . . . .	85
<b>5</b>	<b>Main Results of the Dissertation</b> . . . . .	<b>89</b>
5.1	Conclusions . . . . .	89
5.2	Future Lines of Research . . . . .	90
<b>II</b>	<b>Journal Publications (Appendix)</b> . . . . .	<b>93</b>
<b>A</b>	<b>Adaptive Beaconing for RSU-based Intersection Assis- tance Systems: Protocols Analysis and Enhancement</b> . . . . .	<b>95</b>
<b>B</b>	<b>A Variational Autoencoder Solution for Road Traffic Fore- casting Systems: Missing Data Imputation, Dimension Re- duction, Model Selection and Anomaly Detection</b> . . . . .	<b>97</b>
	<b>Bibliography</b> . . . . .	<b>99</b>

---

# List of Figures

---

1.1	Future interconnectivity of transportation . . . . .	4
1.2	Technology stack of the automotive industry. . . . .	5
1.3	Thesis structure and rationale of chapters . . . . .	10
2.1	Required position accuracies of representative ITS applications . . . . .	16
2.2	Urban traffic scenario in an intersection area . . . . .	16
2.3	Average number of vehicles within the coverage area of the RSU . . . . .	19
2.4	Position error at the RSU compared to the speed of vehicles . . . . .	19
2.5	Average number of packets dropped at the RSU . . . . .	20
2.6	Impact of shadowing on the average CBR computed at the RSU . . . . .	20
2.7	Impact of shadowing on the CCDF of the position error . . . . .	24
2.8	CCDF of the position error at the RSU for each protocol . . . . .	24
2.9	IASM state machine . . . . .	29
2.10	Spatial distribution of the position error using IASM . . . . .	30
2.11	IASM improvement on the position error . . . . .	30
2.12	IASM improvement over studied protocols . . . . .	30
2.13	Design framework of the intersection assistance protocol . . . . .	31
2.14	Position error of the intersection assistance protocol . . . . .	35
2.15	Spatial distribution of position error of IAP . . . . .	35
3.1	Mean and std. of speed measures for different days of the week . . . . .	42
3.2	Density estimation . . . . .	42
3.3	Graphical representation of the latent variable model . . . . .	45
3.4	A graph of the latent variable model . . . . .	47
3.5	The framework of the variational autoencoder . . . . .	50
3.6	Implemented deep neural network graph . . . . .	53
3.7	Feed forward pass of the network with the reparametrization trick . . . . .	55
3.8	Sensors location in England and California . . . . .	58
3.9	Schematic of the imputation procedure . . . . .	59
3.10	ITS traffic forecasting system considered for evaluation . . . . .	60
3.11	Forecasting system under evaluation for dimension reduction . . . . .	63
3.12	Traffic flow and speed samples projected to the latent space . . . . .	66
3.13	PCA visualization of one-day speed samples in the latent space . . . . .	67
3.14	PCA visualization with outlier detection . . . . .	69
4.1	Autoencoder-based traffic forecast steps . . . . .	72
4.2	Theoretical information plane of the autoencoder . . . . .	77

## List of Figures

---

4.3	Autoencoder-based traffic forecast model evaluated . . . . .	82
4.4	Estimated information planes of the encoder . . . . .	83
4.5	Estimated layer-wise mutual information . . . . .	84
4.6	RMSE increase of traffic forecast vs. amount of compressed data .	86

---

## List of Tables

---

1.1	Analysis of relevance of journal publications . . . . .	11
1.2	Analysis of relevance of main conference publications . . . . .	12
1.3	Analysis of relevance of publications derived from collaborations . . . . .	12
2.1	Summary of statistical performance of the studied protocols . . . . .	23
2.2	IEEE 802.11p 10 MHz channel data rates . . . . .	28
2.3	Statistical performance of the intersection assistance protocol . . . . .	35
3.1	Missing data imputation results . . . . .	61
3.2	Results of the forecast task with dimension reduction . . . . .	64
3.3	RMSE improvement of model selection . . . . .	67



---

## List of Algorithms

---

1	RSU procedure . . . . .	33
2	Vehicle procedure . . . . .	34
3	Estimation of the sufficient bottleneck layer dimension. . . . .	80
4	Monitoring of the entropy during training . . . . .	80





PART I

---

**Dissertation Summary**

---



# CHAPTER 1

---

## Introduction

---

A technology-driven ecosystem is emerging in the automotive industry. Autonomous vehicles, connected cars, electrification and shared mobility trends are driving a shift in the way the auto industry evolves. The underlying technological challenges facing the automotive industry describe the roles that different actors can take on in shaping the ecosystem to solve these challenges. Tech companies, from startups to global tech giants, transportation agencies, telecom operators and automakers will eventually merge as new use cases and customers will require tailored systems and solutions spanning the entire technology stack, including network access, connectivity devices, data management and applications. Enabling the use cases that the automotive industry promises will require significant investments in new capabilities, such as network infrastructure, data management platforms and edge computing power, plus considerable advances in research.

As we move toward an increasingly autonomous future, many of the use cases will rely on connectivity and thus increase the need for wireless capacity and reliability. All road vehicles are likely to communicate with one another and the surrounding infrastructure, for example, to warn others about traffic incidents or poor road conditions. Advanced driver assistance, platooning of vehicles and fully automated driving are key application areas that 6G aims to support with the first components to be implemented in the Third Generation Partnership Project (3GPP) Release 16 [Tat+20]. This is nothing new, as The 5G Automotive Association (5GAA) published an extended list of use cases in 2019 [5GA19]. Nonetheless, direct communication systems and networks lack the capacity to handle either the data traffic of autonomous vehicle fleets that communicate with each other in real time, nor the remote control of vehicles that requires high-bandwidth and low-latency networks, let alone critical safety applications in dense environments. Overall, today's communications infrastructure is not reliable enough for many of the use cases envisaged.

The transportation industry has entered the era of big data. Part of the data disseminated by vehicles and infrastructure is being exploited by Intelligent Transport Systems (ITS), advanced applications in which information and communication technologies are applied in the field of road transport traffic management. The idea of ITS was born in the 1980s by a small group of transportation professionals to recognize the impact of computing and communications techniques in the transportation field [WP00]. For the past decade, ITS has played a significant role in the global world and its applications

## 1. Introduction

---

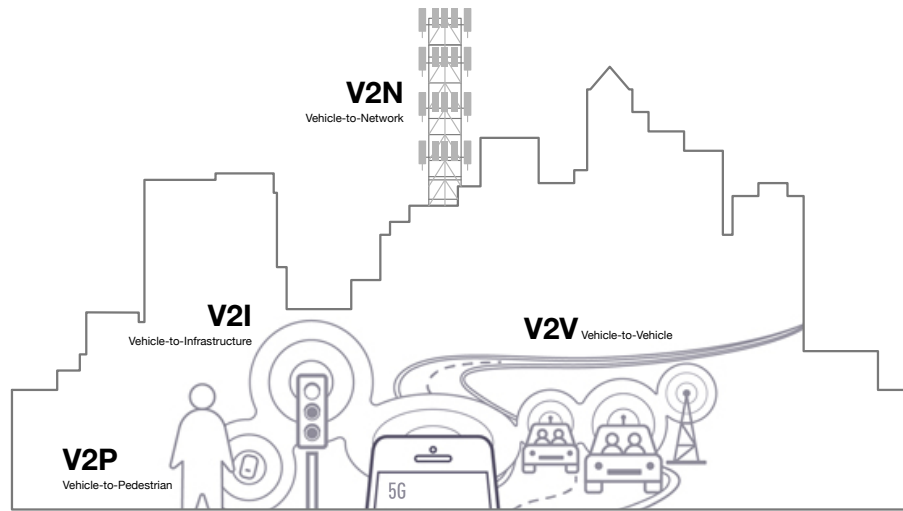


Figure 1.1: Future interconnectivity of transportation sustaining intelligent transportation systems. Vehicle-to-everything communication.

go beyond highway traffic. ITS evolved in tandem with the automotive industry roadmap and its current data analysis workflow consists of four sequential stages: (i) data acquisition, which generally considers different sources; (ii) data preprocessing, the objective of which is to build consistent, complete and statistically robust data sets; (iii) data modeling, where you learn a model for different purposes; and (iv) model exploitation, which includes the definition of actions to be taken with respect to the knowledge provided by the models in real-life application scenarios [Lañ+21].

Inside the data acquisition layer, the connectivity requirements for the car of the future largely involve two types of vehicular communication. Network-based communication allows cars to use the cellular network to communicate with nearby vehicles, pedestrians and the infrastructure. Known as vehicle-to-network (V2N) communication, it has a much wider communication range compared to direct methods and uses commercially licensed spectrum from mobile network operators (MNOs). Direct communication allows vehicles to communicate directly with their surroundings without significantly relying on cellular networks. This type of communication includes vehicle-to-everything (V2X) communication, which will expand the range of knowledge of connected and automated vehicles with information received from neighboring vehicles, infrastructure or vulnerable road users. It includes Vehicle-to-vehicle (V2V) communication, where vehicles communicate with each other to issue warnings, avoid collisions or share immediate road and traffic conditions. Vehicle-to-infrastructure (V2I) communication, where vehicles communicate with nearby infrastructure such as traffic lights, road signs and other transportation infrastructure to further strengthen security measures. Figure 1.1 depicts the whole connect ecosystem. There are two main technologies that support the first generation of V2X communication. First, the Dedicated Short Range Communication (DSRC), which is a higher layer standard based on the evolution of IEEE 802.11p Wi-Fi. It is also the basis for the European standard for

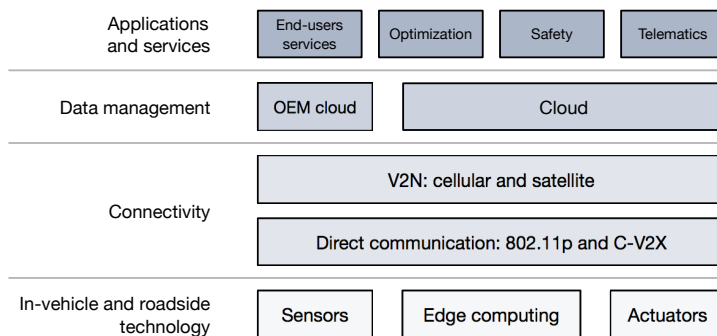


Figure 1.2: Technology stack of the automotive industry.

vehicular communication known as ETSI ITS-G5. Second, the Cellular vehicle-to-everything communication (C-V2X), which is a communication standard defined by 3GPP to use LTE and 5G NR for V2X communication. C-V2X was first specified as part of 3GPP version 14 in 2017 using 4G/LTE and will be further improved as part of 5G NR-based version 16. It is based on the side link PC5 or LTE radio interface, allowing direct V2V or V2I communications without transferring data over the cellular network. IEEE 802.11p and LTE-V2X are not interoperable, which has sparked an intense debate about the technology to implement, not exclusively based on technical but also regulatory and commercial aspects [MGS20]. Oblivious to this debate, the first part of this doctoral thesis addresses the improvement of the reliability of IEEE 802.11p direct vehicle-to-infrastructure communication seen from the point of view of the infrastructure.

Regardless of the type of communication, ubiquitous connectivity is the key to facilitate automation and autonomy among the cars on the road. The automotive tech-stack challenges extend far beyond connectivity, Figure 1.2. The proliferation of sensors generating new data streams and improvements in connectivity are enabling sets of new applications and services located at the cloud that can create new value pools, including more efficient car sharing, location-based marketing and intelligent driving. They are enabling ITS mission-critical applications that have the potential impact to reduce problems such as road safety, pollution, traffic congestion and transportation costs, not only because of the monetary cost but also because of the human cost that they produce. These applications will require new platforms capable of real-time analytics and strong integration with the hardware generating the data. If the aforementioned is achieved in a sustainable manner, it is safe to say that ITS will be a major component of tomorrow’s smart cities [Ull+20]. This has raised the concern of governments and companies about the design, analysis and control of information technologies applied to transport systems. Massive amounts of data require aggregation and processing in the data analytics and application layer of ITS to enable more advanced use cases. Deep learning (DL) solutions, as the new cutting-edge machine learning approach, have gained popularity in ITS because of its capability to flexibly address large amounts of data and model complex behavior. Among the series of data-driven solutions, deep learning models are considered one of the most promising models to tackle various

## 1. Introduction

---

features of ITS [KNZ21]. Interestingly, many aspects of transportation systems are still uncertain, dynamic and highly non-linear: they are invariably complex due to the interaction between humans, vehicles, information technology and physical infrastructure. In this context, this doctoral thesis not only contributes to the improvement of ITS by making data acquisition more reliable: The second part of the thesis explores how to convert large amount of data traffic into a valuable asset. Assuming the infrastructure is able to reliably receive data, this thesis proposes an unsupervised DL solution for several ITS verticals such as missing data imputation, dimension reduction, model selection and anomaly detection. Furthermore, given the deep concern generated by the efficiency of solutions and the large amount of data expected, the last part of the thesis presents a methodology for the automatic definition of an efficient DL architecture.

### 1.1 Motivation

In the envisaged paradigm of vehicular communication, vehicular safety applications of ITS have strict requirements in terms of reliability and latency due to the critical nature of their mission. Information disseminated by vehicles within Vehicular Ad-hoc NETWORKS (VANETs) must be accurate, continuous and up-to-date to sustain those applications. To begin with, direct information exchange through latencies of the order of 100 ms is needed to facilitate the so-called cooperative awareness among vehicles to be able to meet safety requirements [5GA19]. Consequently, Cooperative Awareness Messages (CAM) are broadcasted on the standardized ETSI ITS-G5 Control Channel (CCH) in Europe, while Basic Safety Messages (BSM) are used in the same way by US standardization bodies [ETS14; SAE16a]. These messages (also called beacons) include basic information such as the position, speed or direction of the transmitting vehicle. Infrastructure-dependent safety applications rely on the periodic exchange of safety information between vehicles and road side units (RSU). In VANET literature, scalability and reliability of the network at intersections is recognized as a major problem because of its unique and severe characteristics that critically affect packet reception. The Intersection Collision Risk Warning (ICRW) application is considered as primary road safety application to detect potential vehicle collisions at road intersections relying on beacon processing in the RSU [5GA19]. Despite that, it had not yet been determined whether the information disseminated by the state-of-the-art IEEE 802.11p beaconing protocols was suitable enough for the implementation of specific RSU-based applications or to what extent they could sustain Intersection Assistance Systems (IAS) [Joe+16]. This motivated the following questions and the first part of the thesis.

#### **Research questions in ITS IEEE 802.11p data acquisition:**

- Is current adaptation criteria based on V2V metrics of beaconing protocols diminishing or enhancing performance of ITS RSU applications?
- What is the optimal design criteria that maximizes performance for RSU-based applications?

- When and how should the adaptation criteria of beacon protocols be adapted to support different applications or scenarios?

If data is reliable enough at the data acquisition layer, the amount of data traffic generated can be aggregated and processed at the data analytics and application layer, providing a broader perspective of the whole road traffic network. ITS are constantly generating, acquiring, and processing data in the form of speed, flow, density, etc. measures collected from different sources apart V2X communication (e.g., CAMs or BSMS received at RSU), such as loop detectors, cameras, etc. [Zhu+18]. Such amount of data must be processed somehow to create real value that causes social, environmental and economic profit. Traffic modelers with accumulated experience excel at uncovering patterns, extracting insight and performing complex reasoning based on the data they observe. How can we build artificial learning systems to do the same from big data? Data-driven approaches like DL have raised as a prominent solution as they are capable of mining information from messy and multi-dimensional traffic data sets with few modeling constraints. The intrinsic characteristics of road traffic still makes the forecast a challenging problem because of complex spatial dependency on road networks, non-linear temporal dynamics with changing road conditions and inherent difficulties of long-term forecasting. In addition to the forecasting problem, more challenges of equal magnitude arise from this context. To name a few, data quality, arterial and network level predictions, spatiotemporal predictions and model selection techniques are identified as some of the main current challenges in predicting future road traffic [Lañ+18b]. All models proposed in literature aim to solve only one of these challenges at the time, so the following questions were raised to motivate the second part of the thesis.

### **Research questions in ITS data analytics and application layer:**

- Does exist a unique solution to solve the major challenges of traffic forecasting in an unsupervised manner?
- Is it possible to learn the underlying structure that generates traffic data?
- If yes, can the learned model be used to generate new data, impute missing values, extract useful features, explore the data for a specific task and detect anomalous traffic?

Without a doubt, all players in the automotive industry can use this data to improve, but storing it comes at a high cost. Realizing the true potential of ITS requires ultra-low latency and reliable data analytics solutions that can combine a heterogeneous mix of data stemming from the ITS network and its environment in real-time. Such data analytics capabilities should be provided by efficient data processing techniques capable of avoiding the curse of dimensionality and whose communication and computing latency are low. Despite that, most DL traffic forecasting ITS solutions overlook the importance of scalability and efficiency under the big data paradigm in which they are intended to work, also leaving aside or completely disregarding the operational aspects for the applicability of such models in ITS environment. The trend found in the literature is based on trial and error methods, that is, defining the architecture as a hyper-parameter that needs to be optimized using exhaustive search approaches. This is in contrast to the recent suggestion of the information bottleneck (IB) method as



## 1. Introduction

---

the theoretical basis for DL, a technique in information theory designed to find the best balance between precision and complexity [ST17; TZ15]. The advances of IB theory coupled with the need to find a method based on theoretical concepts so that a practitioner can efficiently find the right architecture is what motivated the third part of the thesis and raised the following questions.

### Research questions in deep learning model tuning:

- Can DL solutions be theoretically tuned according to ITS needs instead of using brute force approaches?
- Is there an efficient way to automatically select the minimum architecture that offers the best performance?
- Is the evolution of information-theoretic quantities during training an indicator of the performance of the forecasting network?

## 1.2 Objectives

The three high-level objectives of the thesis are presented below.

### Objective I

Given the first set of research questions, the thesis aims to develop a V2I protocol to support critical RSU-based applications in difficult environments like road intersections. Towards that goal, the first step is to clarify which type of applications can sustain the information that the RSU receives from direct communication of standardized beaconing protocols. Then, derive a protocol taking advantage of the analysis and insights retrieved.

- Develop an IEEE 802.11p V2I beaconing protocol to support RSU mission-critical applications that require low position error with high reliability in road intersections.

### Objective II

Given the second set of research questions and that DL excels at discovering non-linear patterns from big data in a flexible way: The framework developed shall merge the recent advances in DL with ITS traffic forecasting systems' needs. As a contribution, the framework shall be a unique solution to several future major challenges of traffic forecast identified by Laña et al. [Lañ+18b], formulating the traffic forecasting problem as a latent variable model and resolving it using DL. It should learn the latent variable model of the traffic data, that is, learn useful forecasting features and also meaningful characteristics. The model learned should enhance ITS preprocessing stages and serve as a tool for traffic modelers to improve model development and decision-making.

- Develop a unique model for ITS to extract knowledge from traffic data to enhance traffic forecast, missing value imputation, model and data selection and anomaly detection.

**Objective III**

Given the third set of research questions, ITS solution of Objective II shall be aligned to the efficiency needs of ITS. Therefore, the aim should be to avoid using exhaustive search to define the architecture, thus develop a methodology based on information-theoretical concepts. The resulting architecture should be the minimum one that can provide maximum data compression to avoid forecasting with excessive number of features, which is computational inefficient and undertakes the risk of overfitting.



Develop an efficient methodology that automatically defines the minimum-expression architecture of ITS solution of Objective II that can provide maximum data compression without diminishing the accuracy of the subsequent forecasting system.

**1.3 Structure and Rationale**

This is a thesis written by compendium of works. As such the annexed papers represent the core of the thesis which include the thorough descriptions, discussions, mathematical developments and results. This thesis is structured in two Parts. Part I contains a dissertation summary based on the co-authored journal papers and conference papers contributions. Part II is divided in Appendix A and Appendix B, containing the two published journals [Boq+18b] and [Boq+20], respectively.

The dissertation summary is meant to give an overall logical envelope by discussing the relevant related work in the literature and the main findings pointing the reader to the relevant papers where complete results can be found. Its logical structure is summarized in Figure 1.3, where four sequential stages of the ITS design workflow are depicted: data acquisition, data preprocessing, data modeling and model exploitation. These stages are related to the data acquisition, data analytics and application layers of ITS, which are the scope of Chapter 2, Chapter 3 and Chapter 4, respectively. More specifically, Part I contains:

**Chapter 1** has introduced the doctoral thesis. The main motivation, framework and objectives pursued have been presented. It also introduces the overall structure of the thesis and provides a brief analysis on the relevance of the publications annexed and co-authored.

**Chapter 2** is aligned with Objective I. Section 2.1 starts with novelty and related work discussion. Section 2.2 provides the reader with an overview of IEEE 802.11p technology and its scalability problem. Section 2.3 contains the main results: an analysis of the standardized beaconing protocols, new infrastructure-oriented adaptation criteria for communication parameters and a new communication protocol to support critical applications at intersections. The work summarized in this chapter is the result of the paper contributions [Boq+17] (in collaboration with German Aerospace Center, DLR), [Boq+18a] and journal article [Boq+18b] (Appendix A).

**Chapter 3** is aligned with Objective II. Likewise, Section 3.1 starts with novelty and related work discussion. Section 3.2 presents the traffic problem from

# 1. Introduction

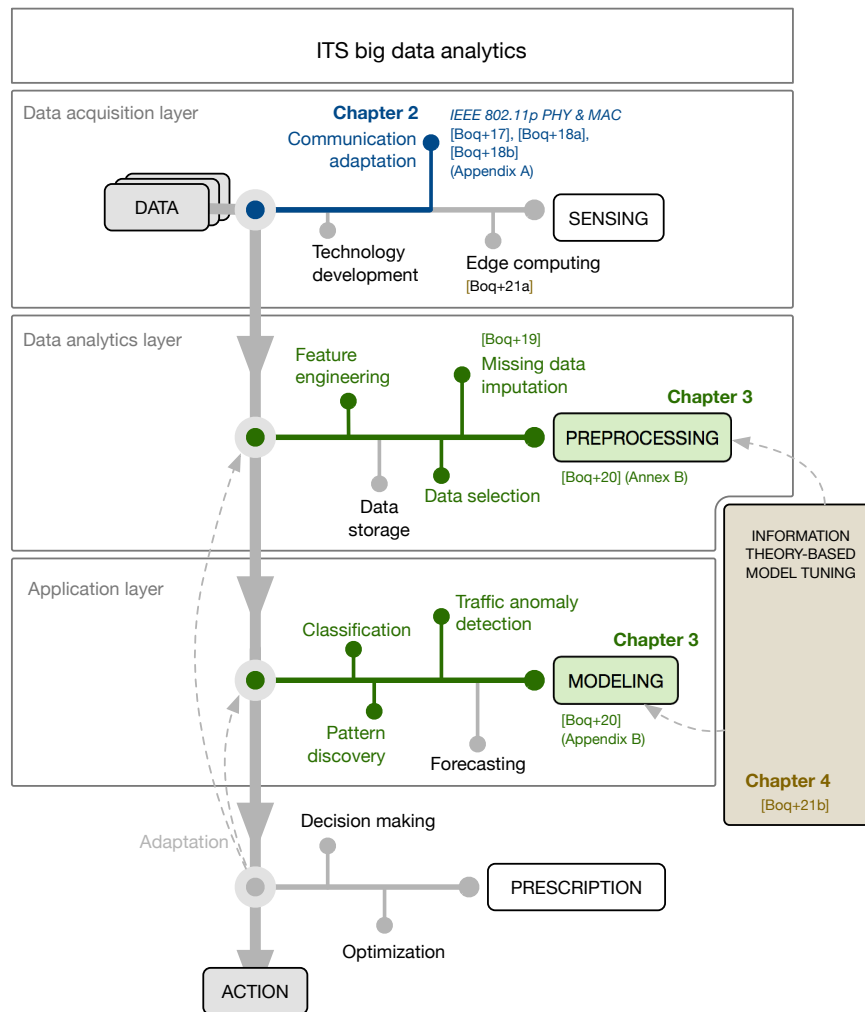


Figure 1.3: Thesis structure and rationale of chapters. Thesis' chapters and publications are related to ITS design workflow for data-based modeling, which at the same time are divided into the ITS big data analytics layers. Fields addressed in the thesis are highlighted in color. This figure is an adaptation of the workflow shown in [Lañ+21].

## 1.4. Relevance of Publications

Table 1.1: Analysis of relevance of journal publications [Boq+18b] and [Boq+20]. The abbreviation Ref. indicates the corresponding appendix and Cit. stands for Citations.

Title	Ref.	Journal	Q	IF	Cit.	Year
Adaptive Beaconing for RSU-based Intersection Assistance Systems: Protocols Analysis and Enhancement	A	Vehicular Communications, ELSEVIER	Q1	4.71	8	2018
A Variational Autoencoder Solution for Road Traffic Forecasting Systems: Missing Data Imputation, Dimension Reduction, Model Selection and Anomaly Detection	B	Transportation Research Part C: Emerging Technologies, ELSEVIER	Q1	6.08	17	2020

a probabilistic point of view. Section 3.3 provides the reader with the necessary background and context. The solution is presented in Section 3.4, deriving the model and analyzing the main motivations for the choices made. Section 3.5 presents the main results obtained in missing data imputation, dimension reduction and model selection, also exploring anomalous traffic detection. The work summarized in this chapter is the result of the conference paper [Boq+19] and journal article [Boq+20] (Appendix B).

**Chapter 4** is aligned with Objective III. Section 4.1 starts with novelty and related work discussion. Section 4.2 briefly reviews information-theoretic concepts applied to an autoencoder, a specific DL architecture. Section 4.3 contains the main results, the proposed algorithm and the basis on which the proposal is based. Part of the work presented in this chapter is the result of the paper contribution [Boq+21b].

**Chapter 5** discusses the main results of the doctoral thesis and proposes future lines of research.

## 1.4 Relevance of Publications

The analysis in Table 1.1 shows that the publications annexed to this thesis have been published in journals of the first quartile (Q1) for the engineering category, and have been referenced by other articles in the literature. The analysis of the relevance of the publications is based on the data reported by each journal, the JCR analysis application of FECYT and Google Scholar. The quartile and impact factor (IF) values refer to the year of publication. Table 1.2 analyzes the authored conference publications directly related to the thesis. Additionally, the analysis in Table 1.3 shows publications done in collaboration that are related to the thesis, where the author is not necessarily listed as the first author.

## 1. Introduction

---

Table 1.2: Analysis of relevance of conference publications [Boq+17], [Boq+18a], [Boq+19] and [Boq+21b], respectively.

Title	Conference	Cit.	Year
Trajectory Prediction to Avoid Channel Congestion in V2I Communications	Personal, Indoor, and Mobile Radio Communications (PIMRC)	8	2017
Analysis of Adaptive Beaconing Protocols for Intersection Assistance Systems	Wireless On-demand Network Systems and Services (WONS)	1	2018
Missing Data in Traffic Estimation: A Variational Autoencoder Imputation Method	International Conference on Acoustics, Speech and Signal Processing (ICASSP)	14	2019
Theoretical Tuning of the Autoencoder Bottleneck Layer Dimension: A Mutual Information-based Algorithm	European Signal Processing Conference (EUSIPCO)	0	2021

Table 1.3: Analysis of relevance of publications derived from collaborations [Cor+17], [Pis+18], [Mac+19] and [Boq+21a], respectively.

Title	Journal/Conf.	Q	IF	Cit.	Year
Autonomous Car Parking System through a Cooperative Vehicular Positioning Network	Sensors, MDPI	Q1	3.28	25	2017
VAIMA: a V2V based Intersection Traffic Management Algorithm	Wireless On-demand Network Systems and Services (WONS)	–	–	6	2018
Novel Imputing Method and Deep Learning Techniques for Early Prediction of Sepsis in Intensive Care Units	Computing in Cardiology (CinC)	–	–	1	2019
Offline Training for Memristor-based Neural Networks	European Signal Processing Conference (EUSIPCO)	–	–	0	2021

## CHAPTER 2

---

# Enhancement of Vehicle-to-Infrastructure Communication in the ITS Data Acquisition Layer

---

This chapter covers Objective I of the thesis by proposing an IEEE 802.11p communication protocol that adapts the communication parameters of vehicles to guarantee reliable and updated data for infrastructure safety applications. Specifically, Section 2.1 starts with novelty and related work discussion. Section 2.2 provides the reader with a brief overview of IEEE 802.11p technology and its scalability problem. Section 2.3.1 critically analyzes the behavior of the vehicular network in a dense intersection scenario. Section 2.3.2 evaluates the state-of-the-art beaconing protocols considered for standardization from the point of view of roadside infrastructure. Section 2.3.3 derives infrastructure-oriented adaptation criteria for communication parameters. Section 2.3.4 proposes a communication protocol for Intersection Assistance Systems (IAS).



“Develop an IEEE 802.11p V2I beaconing protocol to support RSU mission-critical applications that require low position error with high reliability in road intersections.”

### 2.1 Related Work

Adaptive beaconing protocols are proposed by standardization bodies and researchers to improve the overall VANET performance, mainly adapting frequency and power transmission to different criteria such as channel load, traffic density, dynamics of vehicles or application requirements, to name a few. Several authors put an effort into summarize adaptive beaconing into three surveys: [Sep+11], [Gha+13] and [Sha+16], while the current European Decentralized Congestion Control (DCC) is standardized in [ETS11] and the U.S. standard in [SAE16b] (hereafter referred as USA DCC). Adaptive beaconing protocols can be divided depending on their approach into message frequency control, transmit power control or hybrid based approaches. Also, depending on their aim, they can be divided into congestion control protocols, those aiming

## 2. Contributions in ITS Data Acquisition

---

to control channel congestion, and awareness control protocols, those that aim to fulfill application requirements.

The most relevant trend being followed is to adapt beacon frequency as a function of channel load so as not to exceed a threshold considered optimal with respect to the throughput of the channel, which in turn leaves capacity to receive messages that promptly inform of specific events. In addition, the vast majority of them are designed based on the fairness postulation, that is, all vehicles must achieve the same performance and the same opportunities within the network. One of these examples is LIMERIC [KBR11] that jointly with PULSAR [Tie+11] is considered by ETSI to be included in the ITS-G5 vehicular standard together with their DCC mechanism and CAM triggering conditions [ETS11; ETS14]. However, some challenges still remain unresolved, for instances, moderately adaptive approaches like the ETSI's DCC do not perform well considering network dynamics caused by shadowing. So, Sommer et al. [Som+15] proposed DynB, which aims to be stable under heavy network congestion and to be able to quickly react to density changes. Traditionally awareness control protocols have been designed and evaluated separately from congestion control protocols. Therefore, Sepulcre et al. [Sep+16] proposed INTERN, which integrates a congestion control process as a function of the channel load and an awareness control process aiming to adapt the power to the minimum necessary so that the messages are received with certain reliability at an individual warning distance. Kloiber et al. [Klo+16] proposed an awareness control protocol that provides different levels of awareness-quality at different ranges, while accounting for correlated packet collisions. Despite showing great performance in their function as demonstrated by their authors, none of these relevant protocols take into account the position accuracy at the application level, which is a relevant metric for most safety applications. In that sense, some protocols have been proposed which take into account tracking accuracies using a trajectory prediction approach, e.g., [Ban+13; Hua+10; NJ15; Sun+17]. Being [Hua+10] the one adopted as the official USA DCC, which correlates communication behavior with tracking error stochastically sending packets when the suspected error of neighbors grows above a defined threshold [SAE16b]. Sun et al. [Sun+17] used a trajectory prediction approach and a RSU to allocate channel resources according to tracking requirements from vehicles. Guan et al. [Gua+11] provided a congestion control method for road intersections using feedback from a RSU about optimal beacon rate and backoff slots previously computed offline. On another hand, Joerer et al. [Joe+14] proposed a situation-based rate adaptation scheme that allows temporary exceptions for endangered vehicles to use more than the equal fair share of the channel. Also, Joerer et al. [Joe+16] proposed another beacon rate adaptation algorithm relying on their intersection collision probability metric, stating that current state-of-the-art congestion control mechanisms are not able to support intersection assistance systems adequately. Nevertheless, neither of the aforementioned approaches take into account V2I application metrics and some are not optimal at application level since each vehicle has different needs to meet application's requirements at each instant of time [Sch+10].

## 2.2 IEEE 802.11p Overview

IEEE 802.11p is an evolution of IEEE 802.11a for vehicular communications. IEEE 802.11p uses an OFDM (Orthogonal Frequency Division Multiplexing)-based physical (PHY) layer with a channel bandwidth of 10 MHz. IEEE 802.11p uses the same modulation and coding schemes as IEEE 802.11a. It supports data rates ranging from 3 to 27 Mbps using coding rates 1/2, 2/3 or 3/4 (convolutional coding) and BPSK (binary phase shift keying), QPSK (quadrature phase shift keying), 16-QAM (16-quadrature amplitude modulation) or 64-QAM modulations. IEEE 802.11p uses the Outside the Context of a BSS (OCB) operation mode to avoid the latency associated with establishing association and authentication procedures in a BSS. The connection setup of IEEE 802.11 Basic Service Set (BSS) operations, like multiple handshakes before exchanging data, is reduced because of the demanding vehicular requirements of spontaneous and highly mobile ad-hoc communications. Likewise, the Request-To-Send/Clear-To-Send (RTS/CTS) handshake mechanism is not implemented as it increases latency in high mobile networks. The IEEE 802.11p basic access method is the Distributed Coordination Function (DCF) of IEEE 802.11, known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). In CSMA/CA, a node has to sense the radio channel before transmitting a packet, thus preventing it to transmit if another node is using the channel. If the channel is sensed as idle, the node can start its transmission. If the channel is sensed as busy, the node defers its transmission until the end of the current transmission. The radio channel is sensed as busy when the vehicle detects a signal with a received power strength higher than the Clear Channel Assessment (CCA) threshold, higher than the receiver's sensitivity level. At the end of the channel busy period, the node waits for a backoff time to minimize collisions during contention between multiple nodes that also deferred their transmission. This time is calculated for each packet by multiplying a specific slot time of the OFDM PHY layer and an integer number that is randomly selected in the interval from 0 to the size of the CW, where CW refers to Contention Window. The standard sets a CW of 15 time slots of 13  $\mu s$  each for transmitting broadcast packets in 10 MHz channels. The node decreases the backoff time when it senses idle the channel to finally start its transmission when its backoff time reaches zero.

### 2.2.1 Scalability Problem

One of the most critical weaknesses of IEEE 802.11p communication is data congestion suffered in high density scenarios. Periodic broadcasting of messages to build an accurate real-time image of the surrounding state of vehicles leads to serious redundancy, contention and collision probability as the number of vehicles on the network grows. Broadcast transmissions in the IEEE 802.11p CCH using CSMA/CA are not acknowledged, no ACKs are used, and therefore collisions cannot be detected. In high density scenarios the probability of two nodes choosing the same back-off time increases significantly. Recall that a node has to wait a random back-off time between the interval  $[0, CW]$  at the time of transmission if it finds the channel busy. In fact, under a high loaded channel the behavior of CSMA converges to an ALOHA process, where a node chooses a random transmission time without sensing the medium [Boq+18b]



## 2. Contributions in ITS Data Acquisition

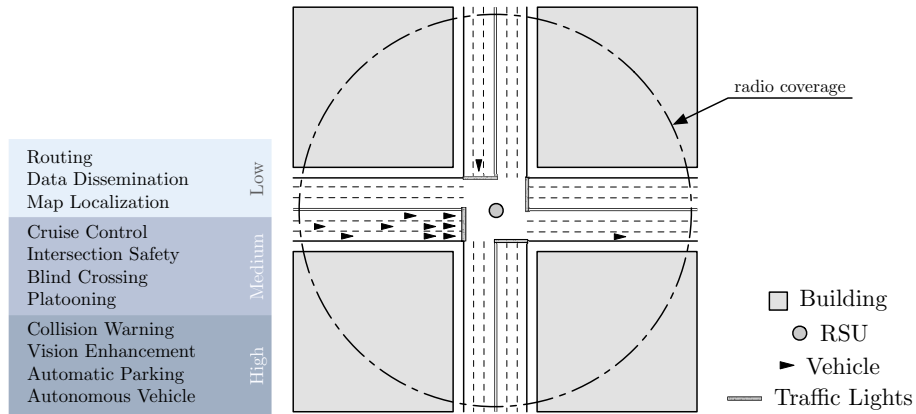


Figure 2.1: Required position accuracies of representative ITS applications grouped into low (10 to 30 meters), medium (1 to 5 meters) and high (0.5 to 1 meters) scales.

Figure 2.2: Urban traffic scenario in an intersection area obstructed by buildings. Vehicles continuously disseminate their position using direct V2I communication. Data is received at the RSU running a collision risk warning application. The vehicular network simulation was implemented using the framework Veins on top of OMNeT++ event-based network simulator and coupled with SUMO road traffic simulator.

(Appendix A). In addition, there is no RTS/CTS handshake mechanism, thus no way to detect a hidden terminal that its signal strength is received above the RSU's sensitivity level but below the receiver's sensitivity level. Random channel access schemes like IEEE 802.11 were designed for bursty data traffic patterns instead of periodic ones. Although the CAM transmit policy has been enhanced by additional triggers based on mobility changes, in numerous contexts vehicular mobility is little varying and highly correlated (e.g., platooning on a highway). As a consequence, the resulting broadcasts are likely to be correlated as well, which may cause correlated packet collisions. Whereas the loss of individual CAMs only has a minor impact on the current up-to-dateness of the cooperative awareness, several consecutive losses may quickly lead to outdated information, which is not viable for safety related applications anymore. IEEE 802.11p major challenges in heavy traffic scenarios are the broadcast storm problem, hidden terminal problem and the packet collision. Those are well-known and well-studied problems in static or quasi-static scenarios, however, VANETs consist of highly mobile nodes that require ubiquitous and reliable communications to sustain safety applications.

### 2.3 Vehicle-to-Infrastructure Communication in Intersection Areas

Intersection areas are characterized for unique and severe conditions where IAS are meant to detect hazardous situations and manage traffic to reduce vehicle related problems. Roadside ITS stations (IT-S) standardized by ISO and ETSI are designed to help in those situations. ETSI RSU-based intersection

## 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

services are many such as: collision warning, wrong way driving, traffic condition warning, signal violation warning, traffic light management and optimal speed advisory, traffic information and recommended itinerary. Among them, ICRW is its most relevant application [TS18]. In such a heterogeneous framework, applications' needs derive in very different required position accuracies that can be grouped into three scales: low (10-20 or even 30 *m*), medium (1 to 5 *m*), and high (a meter or sub-meter), Figure 2.1 [San+16]. Because safety applications have strict requirements in terms of awareness and latency, the main metric analyzed in this chapter is the position accuracy. It is defined as the error between the current vehicle's physical position and the last reported position to the RSU, which implicitly entails the requirement of latency. Error results shown in this chapter are compared to those scales.

### Intersection Scenario

An intersection simulation environment scenario was designed and built in order to test the performance of beaconing protocols and proposed methods. The scenario is an intersection area with a deployed RSU ITS-S running the ICRW application, see Figure 2.2. It requires real-time monitoring of all vehicles with a short end-to-end latency time in order to provide timely warning to drivers [ETS18]. The simulated scenario is described in detail in [Boq+18b] (Appendix A). Realistic vehicle mobility was simulated. IEEE 802.11p and ITS-G5 standard direct communication was assumed, where CAMs are periodically broadcasted by vehicles using default IEEE 802.11p PHY and MAC parameters. Radio signal attenuation was modeled as a function of path, shadowing and fading effects. Two scenarios were considered: (O) an intersection area fully obstructed by buildings (Figure 2.2) and (Ø) the same unobstructed area where LOS conditions exist between vehicles of different roads.

### 2.3.1 Network Behavior

An approximation of the behavior of the position error at the RSU is presented to understand why some protocols perform better than others and how to adjust the parameters or derive new protocols towards enhancing IAS. Hereafter, it is assumed that the positioning of the vehicle itself is error free, as we are only interested in the error contribution from the performance of the protocols.

#### Position Error Model

Assuming ideal channel conditions, a constant vehicle speed  $v$  and uniformly distributed events of looking up the position at the RSU during a fix beaconing interval  $t_b$ , the average position error at the receiver can be expressed as half the minimum plus the maximum position errors as:

$$\bar{e} = v t_{tx} + \frac{v(t_b - t_e)}{2}, \quad (2.1)$$

where  $t_{tx}$  is the transmission time of the beacon and  $t_e$  is the time between the position error computation at the RSU and the next beacon reception. The reception of the next beacon depends on several factors as a packet may not be received due to low SINR (i.e., collision) or low SNR (i.e., reception

## 2. Contributions in ITS Data Acquisition

---

power below receiver’s sensitivity). Contrary to (2.1), the average position error at the RSU increases as the average speed of vehicles drops while congestion occurs [Boq+17]. Hence, considering this and neglecting the error contribution of  $t_{tx}$  (cm-order), the maximum position error of a vehicle at time instant  $k$  can be estimated as:

$$\hat{e}_k = v_k \mathbb{E}[t'_b], \quad (2.2)$$

where  $t'_b$  is a random variable representing the actual time between two consecutive beacons. Its expectation can be expressed as the number of consecutive tries  $I$  needed to receive a beacon multiplied by the beacon interval,  $\mathbb{E}[t'_b] = \mathbb{E}[I] t_b$ . Assuming that packet loss is independent across time, the expected number of consecutive tries can be expressed as:

$$\mathbb{E}[I] = \sum_{i=1}^{\infty} i P_{\text{col}}^{(i-1)} (1 - P_{\text{col}}) \approx \frac{1}{\text{PDR}}. \quad (2.3)$$

Packet loss due to low SNR represent in average less than 0.6% of the total packet loss in the simulated scenario, thus the probability of a collision can be approximated as  $P_{\text{col}} \approx 1 - \text{PDR}$  [Boq+18a]. Unfortunately, (2.3) is not valid in complex vehicular situations. The probability of reception is based on a plurality of factors, to mention a few, IEEE 802.11p MAC contentions, the capture effect, the hidden terminal problem, fast varying density of traffic, correlated packet collisions due to quasi-periodic transmissions of beaconing protocols and relative mobility of vehicles towards the RSU. Due to these issues, the purely mathematical analysis of the position error becomes highly complex. Nevertheless, simulation results showed that (2.2) approximates the behavior of error, showing that error is a function of vehicle dynamics, which roughly depends on traffic conditions and scenario topology, and of probability of packet reception, which roughly depends on the number of vehicles and channel conditions [Boq+18a].

### Influence of Vehicle Dynamics

Using the default beaconing setting, the simulated vehicle traffic and the position error at the RSU are depicted in Figure 2.3 and Figure 2.4, respectively. Figure 2.4 illustrates that error follows two different patterns as a function of time as in (2.2). On the one hand, there are periodic fluctuations in the error similar to the evolution in time of the average speed. These are due to the behavior of vehicle traffic at the intersection. For example, at point A in Figure 2.4, one of the time instants in which the error is minimal, corresponds to when immediately traffic lights turn green: Vehicles in queue are stopped and those at the beginning start to accelerate, therefore the average speed of vehicles is much lower and the resulting error as well. Once the traffic light has turned green, the vehicles accelerate until reaching the maximum speed to leave the intersection, point B in Figure 2.4. In addition, vehicles that previously stood in the queue move towards the traffic light. All of these increases the average speed and consequently decreases the position accuracy. Finally, the same phenomenon can be perceived at point C for vehicles that were on turn lane. Although the phenomenon is on a smaller scale as there are fewer vehicles and the traffic light time is also shorter. The worst case scenario can be found at larger distances from the RSU where higher speeds are found. Also, the

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

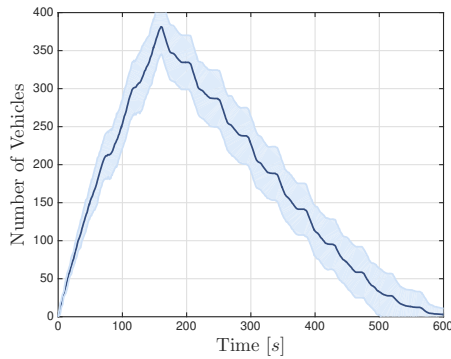


Figure 2.3: Average number of vehicles within the coverage area of the RSU with two the times standard deviation. The evolution of the vehicle number across time simulates a rush hour with high density traffic. Please note that vehicles stop appearing at  $t = 160$  s.

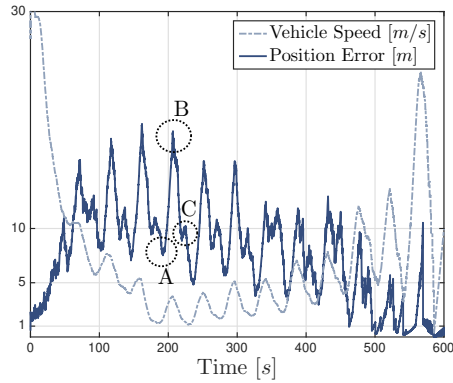


Figure 2.4: Average position error at the RSU compared to the average speed of vehicles across time in scenario  $\emptyset$ . Comparison of both allows to understand dependencies of the error on the scenario topology and traffic density. A fix-period beaconing of  $100$  ms was simulated, suggested by ETSI and SAE for cooperative awareness.

temporal analysis of lost packets showed that when vehicles stop appearing ( $t = 160$  s) packet loss due to SNR follow the same pattern as the speed. This is not due to the speed but to the distance in which vehicles are located because it coincides that they have the highest speed at further distances from the RSU.



#### Lessons learned [Boq+18b]:

- The error depends on the scenario topology and traffic characteristics, intersections are characterized by high mobile traffic.
- The fairness postulation of beaconing protocols is not optimal, each vehicle contributes differently to the error.
- Stopped vehicles near the RSU with low mobility capabilities saturate the channel with redundant information that does not improve accuracy, raise the probability of a collision and are the least likely to contribute to an accident.
- Beacon frequency should be adapted to the dynamics of the vehicles.

#### Influence of Packet Loss

The error grows similar to the evolution of the number of vehicles but attenuated by the decrease in the average speed when traffic congestion occurs, Figure 2.4.

## 2. Contributions in ITS Data Acquisition

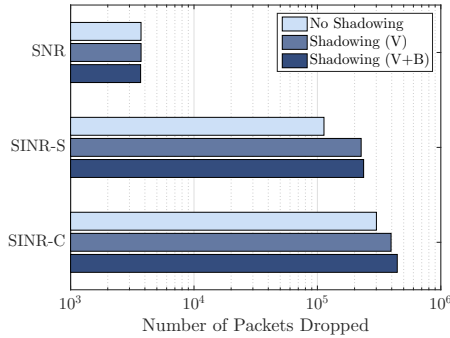


Figure 2.5: Simulated average number of packets dropped at the RSU due to low SNR (SNR), simultaneous transmission (SINR-S) and concurrent transmission (SINR-C) for different shadowing conditions.

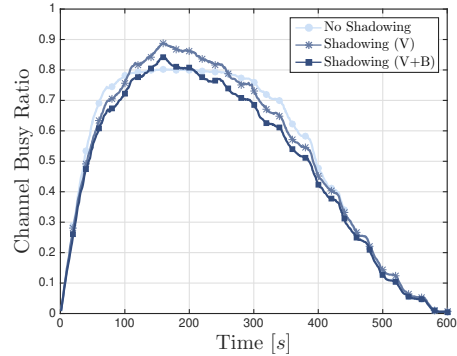


Figure 2.6: Impact of shadowing on the average CBR computed at the RSU. This figure illustrates the heavy influence of shadowing conditions on the behavior of CSMA/CA protocol.

This shows an inverse relationship between the influence of vehicle dynamics and channel congestion (proportional to traffic congestion) on the error. That is, when traffic congestion occurs the channel becomes more saturated increasing the error, however, the average speed of vehicles decreases at the same time, thus reducing the error. There are collisions in almost all instants of time due to the periodic transmission of beacons. The problem worsens as the number of vehicles increases, coinciding in  $t = 160$  s the greatest number of collisions with the maximum number of vehicles. Packets can not be received because they are considered as noise due to low SNR, discarded as collision due to low SINR during preamble reception (i.e., simultaneous transmission) or discarded due to bit errors caused by low SINR at some point during reception (i.e., concurrent transmission). Figure 2.5 illustrates the importance of shadowing on packet reception in an intersection area. Figure 2.5 shows the number of packets dropped at the PHY and MAC layer of the RSU for different kind of simulated shadowing conditions accounting for no shadowing at all, shadowing dynamics of vehicles (V) (i.e., Scenario  $\emptyset$ ) and the later one plus shadowing of buildings (V+B) (i.e., Scenario O). The worst case translates to minimum PDR values of 10%, corresponding to a 90% chance of collision. As the number of vehicles increases the effect of radio signal shadowing increases the probability that packets are not received due to a low SNR but, more importantly, hidden nodes are increased during which collision avoidance mechanism of CSMA is not involved. Thanks to shadowing of vehicles and buildings the range of distance at which vehicles sense each other is diminished increasing concurrent transmissions at the RSU. It is worth mentioning that Sommer et al. [Som+15] found out that shadowing diminishes collision probability from vehicles' POV, contrary, in this scenario collisions are increased from the static RSU's POV which is located at intersection's center.

High density traffic infer a high collision probability, arising the well-known scalability problem of the IEEE 802.11p MAC protocol, which precisely congestion control protocols try to avoid by monitoring the Channel Busy

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

Ratio (CBR). Hidden node and collision problems can also be observed on the evolution of the CBR over time in Figure 2.6. The CBR is computed at the RSU as the amount of time that the channel is sensed as busy during a second. The theoretical CBR limit in CSMA/CA without packet collisions can be computed as the total number of beacons that can be fitted in a second (deduced as the inverse of the packet duration plus the predetermined listening period, AIFS) multiplied by the duration of a beacon transmission. In this scenario the CBR theoretical limit equals 0.8 [Boq+18a]. However, as shown in Figure 2.6, CBR maximum value is close to 0.9 when shadowing conditions are considered, exceeding the limit value because of the hidden terminal problem. This result indicates that the behavior of CSMA/CA medium access protocol converges to an ALOHA process, where a node chooses a random transmission time without sensing the medium.



#### Lessons learned [Boq+18b]:

- Dense traffic with periodic or quasi-periodic beaconing increases correlated packet collisions, resulting in catastrophic position accuracies for infrastructure applications in intersections (Figure 2.7, Section 2.3.2).
- Congestion control protocols should manage collisions.
- From this point of view, the fairness postulation is neither optimal, since stopped vehicles with low contributions to the error interfere with further vehicles with higher speeds and prone to larger errors.
- Adaptation to vehicles dynamics of all communications parameters, not only beacon frequency, is mandatory to allow vehicles contributing more to the error to overcome the capturing effect of possible collisions.

### 2.3.2 Protocol Evaluation

Discussion on the performance of the different protocols under this section aims to extract value information of how different adaptation approaches perform in the scenario, if current protocols are able to sustain IAS and to conclude how protocols' design criteria influence information reception for IAS.

#### Evaluation Metrics

The following metrics were used to study the performance of each beaconing protocol. PDR was discarded since does not reflect consecutive packet collisions and can be misleading. Not receiving several consecutive beacons increases tracking error more than receiving packets alternatively, despite the ratio between the number of packets received and sent (i.e., the PDR) could be the same.

- Position error (PE): Defined as the Euclidean error between the current vehicle's position and the last reported position to the RSU. It is computed

## 2. Contributions in ITS Data Acquisition

---

at the RSU for each vehicle every 10 ms. This metric is used to evaluate the applicability of the protocols, as security applications are sustained on accurate and updated position information.

- Channel footprint (CF): Defined as the total channel resources consumed at the RSU in time and space. This metric provides information on the amount of channel bandwidth used and can be compared against tracking error reliability. In addition, a high channel footprint indicates worse conditions for dissemination of other types of messages on the same channel.
- The complementary cumulative distribution function (CCDF) of the PE: Defined as  $CCDF = 1 - CDF$ , provides the probability  $P_r(\text{Error} > n)$  of the position error to be greater than  $n$  at the RSU. It was used to evaluate PE reliability for safety applications, as against other approaches, e.g., average values and confident intervals, the distribution keeps all measured information.

### Summary of Protocols

We considered the three relevant beaconing protocols considered by standardization bodies that are described below. Default IEEE 802.11p values were used for the adaptation of specific parameters not considered by the protocols. The rest of parameters were adapted following the guidelines provided by their authors, the reader is referred to [Boq+20] (Appendix B) for specific details.

- Baseline. Beacon frequency fixed to a 100 ms period.
- LIMERIC combined with PULSAR [KBR11; Tie+11]. LIMERIC adapts the beacon frequency as a function of the CBR, such that all vehicles converge to the same beacon rate and to a desired channel load level that maximizes the throughput. PULSAR computes a global CBR for the vehicle as the maximum CBR between the one locally sensed and the one reported by neighbors during two hops. In this way CBR used by vehicles does not differ much from that measured at the RSU.
- ETSI DCC [ETS11]. Beacon frequency is defined by a periodic beacon interval given by a state machine that changes states according to the sensed CBR. Transmission power, data rate and the CCA threshold are also defined by the same state machine. Additionally, beacons are triggered when the difference between absolute values of current heading, position and speed compared to information disseminated in previous CAM exceeds 4 degree, 4 m or 0,5 m/s, respectively.
- USA DCC [Hua+10]. Beacon frequency is stochastically determined by each vehicle calculating the transmission probability based on suspected tracking error on neighboring vehicles towards its own position. The transmission probability is calculated as a function of user-defined sensitivity and error thresholds, inconsistency in sequence numbers of received packets and CBR measurements. It is assumed that each vehicle estimates the position of others using a predictor and the information disseminated in the shared channel. Accordingly, the RSU runs the same



### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

Table 2.1: Summary of statistical performance of the studied protocols w.r.t. position error and channel footprint for unobstructed ( $\emptyset$ ) and obstructed (O) scenarios. Results shown are the average of 10 simulation runs. *mean* values are computed averaging all values measured in the same time step for each vehicle and then averaging over all simulation time steps. 95% percentiles shown are the average of all percentiles computed in each time step.  $\max(d)$  are the maximum PE values calculated within  $d$  m from the RSU.  $\max(\text{CBR})$  are the maximum CBR values measured at the RSU. The best values are highlighted in bold.

		Baseline		LIM+PULS		ETSI DCC		USA DCC	
		$\emptyset$	O	$\emptyset$	O	$\emptyset$	O	$\emptyset$	O
PE [m]	mean	7.01	15.93	2.88	7.48	1.02	1.42	<b>0.44</b>	<b>0.65</b>
	95%	36.55	104.93	10.43	43.46	3.96	5.87	<b>1.09</b>	<b>1.17</b>
	$\max(50)$	436.9	561.8	266.2	410.9	<b>26.44</b>	<b>32.33</b>	295.2	403.4
	$\max(100)$	427.1	565.1	295.3	430.6	<b>40.9</b>	<b>154</b>	206.1	524.5
	$\max(400)$	466.5	565.1	446.4	432.9	<b>167.3</b>	<b>218.8</b>	542.6	742.6
CF	mean	0.53	0.5	0.49	0.49	0.25	0.26	<b>0.02</b>	<b>0.02</b>
	95%	0.55	0.52	0.51	0.52	0.27	0.28	<b>0.02</b>	<b>0.02</b>
	$\max(\text{CBR})$	0.9	0.87	0.82	0.84	0.51	0.58	<b>0.06</b>	<b>0.06</b>

prediction model for each vehicle in the scenario, a constant velocity model.

### Main Results

Table 2.1 summarizes the improvement of beaconing protocols w.r.t. the baseline for scenarios  $\emptyset$  and O. Figure 2.7 reveals the impact of shadowing on the CCDF of the position error computed at the RSU. Figure 2.8 illustrates the performance of each protocol in Scenario O. ETSI and USA DCCs are the ones performing better. It is clear that USA DCC achieves the best performance in overall despite still providing not negligible maximum values (Table 2.1). Contrary, ETSI DCC improves maximum PE values. There is a notable difference between the two scenarios  $\emptyset$  and O in PE. All the protocols decrease the CF, where USA DCC stands out for the almost null use of the channel. High values of CBR above  $\text{CBR}_{\max}$  are still measured by the RSU, meaning that there is a discrepancy between vehicle and RSU measures.

The distribution and maximum values of the error should be taken into account at the time of implementing an application based on a RSU that needs position information of vehicles approaching the intersection. In that sense, considering an average vehicle width of about 2 m, an overall accuracy of 1 m is needed in order to locate a vehicle in a particular driving lane. Therefore, considering values from Figure 2.8 and Table 2.1, it can be concluded that maximum errors and standard deviations are too large to consider implementing a critical safety application relying on each of the three protocols. This points out that further improvement is needed. However, regarding non-safety applications, 95% of error values would be under medium accuracy scale from up to 100 m using ETSI DCC. A similar performance would be obtained using USA DCC



## 2. Contributions in ITS Data Acquisition

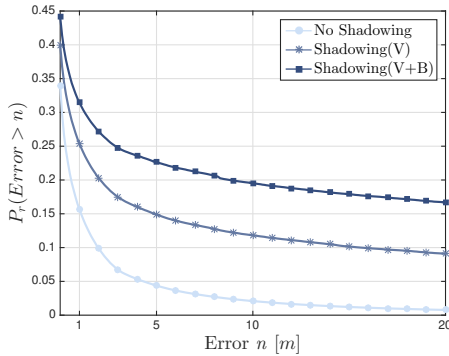


Figure 2.7: Impact of shadowing on the CCDF of the position error computed at the RSU. The hidden terminal problem increases the number of collisions, which translates into not negligible position errors and uncertainty for infrastructure applications.

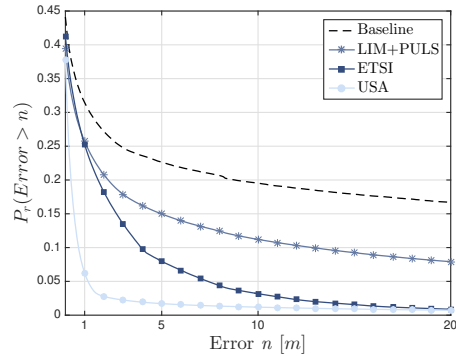


Figure 2.8: CCDF of the position error at the RSU in the obstructed scenario, Scenario O, for each protocol. Different adaptation criteria improve the accuracy of fix-period beaconing. Baseline results are the same results than Shadowing (V+B) in Figure 2.7.

from distances up to 175 m enabling, e.g., an efficient traffic light management from the RSU.

### Performance Discussion of Protocols

- LIM+PULS. The study of the temporal position error behavior showed an improvement when congestion occurred. PE improves w.r.t. the baseline when channel becomes saturated, although no application requirements are considered. LIMERIC aims only to achieve a target CBR, hence, its performance is explained by the correct adaptation to CBR disseminated by PULSAR. PULSAR dissemination of 2-hop maximum CBR of neighbors allows vehicles to react to similar CBR values measured at the RSU, mitigating shadowing effects. CBR information disseminated by the RSU would be more accurate, thus improve the accuracy. However, the major drawback limiting improvement is the fairness postulation: All vehicles transmit with the same beacon rate, but do not contribute equally to the error calculation. Error grows when dynamics are more relevant, as no adaptation to these is used and correlated collisions still occur because of periodic transmissions. Additionally, vehicles transmitting with same constant power limit the performance because packets sent by low speed vehicles are received with greater signal strength than vehicles with higher speeds, which are located far from the RSU.
- ETSI DCC. ETSI's protocol reacts to CBR to avoid channel congestion decreasing power transmission and beacon frequency and increasing data rate and sensitivity. Adaptation is based on CBR measures of vehicles, which differ significantly from the ones measured at the RSU. Using PULSAR approach or CBR disseminated by the RSU would improve performance. ETSI DCC does not only rely on the channel load to adapt

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

---

beacon frequency. In fact, the periodic component of the beacon frequency is adapted to the CBR but the other frequency component is derived from vehicles dynamics (CAM triggering conditions). Thus, as the channel becomes more saturated, the protocol decreases the beacon frequency and the later component acquire more relevance improving the position accuracy. On the other hand, a high data rate lowers the probability of collision, which enhances performance when congestion occurs. Decrease in power and increase in sensitivity objective are to avoid interfering with further vehicles and improve near communications, respectively. When congestion occurs, the carrier sense (CS) range is lowered thus further vehicles' signals are treated as noise enabling closer communications. This approach is not optimal in the specific scenario, because low speed vehicles are being prioritized as interferer vehicles get closer to the RSU.

- USA DCC. Vehicle dynamics and probability of collision are key components for tracking accuracy. USA DCC achieves high performance because of low collision probability conditions and fully adaptation to vehicles dynamics. Forcing all nodes to track vehicles has a computational cost disadvantage, however, it allows vehicles to estimate the error that others are having and to be able to react to it. Using an error threshold, it directly grants more priority to vehicles suspected of having more error. Besides, it reduces redundant information from the channel lowering collisions, which in turn leads to better opportunities to succeed for other kinds of messages. If a packet is lost, next one will be sent stochastically only when the predictor error exceeds the threshold, as there is no mechanism to avoid packet loss. This derives in large values of maximum error and uncertainty that do not cope with high accuracy position requirements. A major drawback is the significant reduction of awareness as new vehicles appearing inside the RSU range do not receive updated information about vehicles that already sent their beacon and that their model is predicting correctly. Adding periodicity to the beacon transmission will increase the error significantly, not scaling linearly with the traffic density (Figure 4 of [Boq+17]).

#### 2.3.3 PHY & MAC Adaptation Criteria

Results simulated and discussed in Section 2.3.2 showed that protocols designed using V2V metrics can barely support safety applications of IAS, despite being able to meet their own requirements as demonstrated by their authors. This shows a trade-off in the adaptation criteria between enhancing vehicle or RSU-based applications. Consequently, there is a need for new beaconing protocol criteria that yields to better performance on which RSU-based IAS can be sustained. Also, this shows the need to *adapt adaptation*, that is, to decide when and how to switch adaptations to comply with different kind of applications or scenarios [SG18]. Since the latter is out of the scope of this chapter, novel criteria for parameter adaptation to enhance beaconing performance towards IAS is proposed below. Lessons learned in Section 2.3.1 and Section 2.3.2 lead to the following infrastructure-oriented adaptation criteria with the aim of maximizing position accuracy for applications based on data collected from infrastructure in intersection areas.

## 2. Contributions in ITS Data Acquisition

---



Lessons learned about parameter adaptation [Boq+18b]:

- C1. Vehicles prone to larger errors must have higher priority. Fairness postulations are not derived from tracking accuracy and reliability needs.
- C2. All communication parameters must be adapted to the dynamics of the vehicles, not only to the state of the channel.
- C3. Effect of packet collisions has to be mitigated. In other words, protocols must aim for low probability of collision conditions while in the event of a collision the most relevant packet must be decodable to avoid correlated collisions.

### Parameter Adaptation Discussion

Parameter adaptation, limitations and their effects in an intersection area are discussed below. Please note that the discussion is from the POV of a RSU as a static node located in the middle of the intersection. Parameters selected are the most relevant in adaptive beaconing literature, aligned with ETSI's adaptation and limited by the current standardized MAC protocol.

- Power ( $P_t$ ). The maximum transmission power allowed in ITS-G5 CCH is 33 dBm.  $P_t$  determines communication range (CR) and carrier sense (CS) range. To obtain a lower collision probability (Criterion 3), it is interesting that the range in which vehicles are sensed is as large as possible to avoid the existence of hidden terminals. As power increases, so does CR and CS range. In that sense, the higher the power in which vehicles transmit the better. Additionally, high power implies greater robustness against signal attenuation. However, transmitting all with the same power does not solve the capturing effect (Section 2.3.2), nor does it to adapt the power to the distance towards the RSU because all packets will be received with similar power. Following Criterion 1 and Criterion 2, vehicles with higher speeds should transmit with higher power than vehicles with slower speeds, so that in case of interference the former vehicles achieve better SINR values. The difference between transmission powers is then subjected to the modulation being used which imposes the minimum SINR to correctly receive a packet.
- Data Rate (R). Available data rates in IEEE 802.11p with their corresponding modulation, coding rate, minimum sensitivity and SINR threshold needed to correctly decode are listed in Table 2.2. Sepulcre et al. [SGC17] discussed about optimum data rate for V2V beaconing. The use of higher transmission speeds implies a decrease in packet duration and thus a decrease in channel congestion but, on the other hand, implies a less robust modulation and a lower CR. Higher SNR and SINR values are required at the receiver in order to be correctly decoded. Vehicles using a more robust modulation and coding scheme will contribute more to the channel load because of a longer packet duration which, in turn, increases the probability of collision. In this context and following Criterion 1 and Criterion 2:

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

---

- For vehicles with higher speeds, prone to more error and usually found at larger distances from the RSU, lower data rates are preferred to achieve a higher priority and better PDR values accounting for interference from low speed vehicles (SINR threshold reduction) or severe signal attenuation at further distances (sensitivity reduction).
- Low speed vehicles require less priority, thus higher data rates should be used to contribute less to CBR and achieve lower  $P_{col}$  values (Criterion 3). Note that this also acts as a congestion control because vehicle speed is inversely proportional to the traffic density: The more traffic the lower the speed, thus the proportion of vehicles with low speed and high data rate will be higher. Unfortunately, packets colliding will not likely be decoded due to an increased SINR threshold required, but lost packets will have less impact on the overall error.
- Sensitivity ( $CCA_{th}$ ). It defines the threshold from which the IEEE 802.11p preamble and header can be detected and decoded or contrary considered as noise, so that the medium is sensed as occupied or idle respectively. Obviously,  $CCA_{th}$  is limited by receiver's sensitivity but ETSI DCC considers as minimum and maximum values,  $-95$  dBm and  $-65$  dBm respectively. Lowering the  $CCA_{th}$  (increasing CS range) of vehicles allows for the detection of transmissions from vehicles situated far away, reducing the number of hidden terminals. However, more contending neighbors result in nodes sensing the channel as busy for a longer period, thus it is more likely to occur that two or more nodes choose the same backoff time. On the other hand, reducing the CS range allows for more transmission opportunities because of lower local CBR values and reduces the number of simultaneous transmissions at the cost of getting interferer closer (high SIR values), increasing concurrent transmission. Section 2.3.1 showed that in this scenario concurrent transmissions are more influential than simultaneous transmissions. Therefore, we advocate for the use of the minimum receiver sensitivity as  $CCA_{th}$  to minimize  $P_{col}$  at the RSU, that is,  $-95$  dBm. Note that all the criteria can not be met at the same time: High speed vehicles can not be prioritized while  $P_{col}$  is minimized.
- Priority (AC). IEEE 802.11p EDCA mechanism allows prioritizing between data traffic using four different queues with different AIFS listening periods and CW settings.  $CW_{max}$  is omitted as it is never used on broadcast mode. AC\_BE ( $CW_{min} = 15$ , AIFS =  $11 \mu s$ ) category is intended to be used for CAMs which turns out to make use of the largest CW available. A large CW is preferred for both, high and low speed vehicles, to lower the probability of a simultaneous transmission (Criterion 3). Regarding Criterion 1 and Criterion 1, AC\_BE is preferred for high speed vehicles and AC\_BK ( $CW_{min} = 15$ , AIFS =  $149 \mu s$ ) for low speed vehicles. Vehicles with higher speeds will listen to the medium for shorter periods of time before transmitting, thus obtaining a higher priority. In this way, packets that have the most influence on the error are going to be transmitted first.
- Rate ( $1/t_b$ ). Beacon frequency is the most influential and versatile parameter and the one where more effort has been put into by researchers.

## 2. Contributions in ITS Data Acquisition

---

Table 2.2: IEEE 802.11p 10 MHz channel data rates [SGC17].

Data Rate [Mbps]	Modulation	Coding Rate	Minimum Sensitivity [dBm]	SINR Threshold [dB]
3	BPSK	1/2	-85	5
4.5	BPSK	3/4	-84	6
6	QPSK	1/2	-82	8
9	QPSK	3/4	-80	11
12	16-QAM	1/2	-77	15
18	16-QAM	3/4	-73	20
24	64-QAM	2/3	-69	25
27	64-QAM	3/4	-68	30

Adapting beacon rate following current fairness postulations does not cope with required position accuracies for IAS, neither it does aiming to achieve maximum throughput relying on CBR measurements of vehicles as shown in results of Section 2.3.2. With high accuracies in mind, beacon frequency must be adapted to vehicle dynamics while randomization is needed to avoid correlated packet collisions. The best approach that fits the criteria is the use of a prediction approach based on the position error. This approach decreases the uncertainty between beacon intervals allowing the opportunity to relax some adaptation criteria and improve the performance of the vehicle network. In this way, lower rates complying with maximum beacon intervals of standards can be achieved, generating low collision probability conditions (Criterion 3), while providing reliable awareness. Besides, a position error threshold condition implicitly considers vehicle dynamics (Criterion 1 and Criterion 2). Therefore, using a predictor at the RSU can benefit all existing protocols with only a minimum computational cost disadvantage, compared to force all vehicles to run a predictor. In fact, most intersection safety applications envisaged require monitoring position of vehicles. In that sense, previous evaluation of USA DCC revealed the potential of using a position predictor, despite showing high error values. Rate adaptation should use randomized redundant transmissions and feedback provided by the RSU about channel metrics and position tracking information.

### Criteria Implementation

The discussion provided is synthesized in the intersection assistance state machine (IASM) of Figure 2.9, based on vehicle dynamics. Beacon frequency adaptation is intentionally left out, so IASM can be implemented over existing protocols validating the proposed criteria. Two different states (LOOSE and RAISE) specify the corresponding parameters to be used and are selected according to the speed of the vehicle. LOOSE and RAISE states correspond to vehicles with low speed, which are intentionally prioritized less, and high speed, respectively. As discussed,  $CCA_{th}$  and  $CW_{min}$  values always remain the same for each vehicle. The minimum data rate (3 Mbps) and the maximum transmission power (33 dBm) have been selected for vehicles in RAISE state. For vehicles in LOOSE, a data rate of 18 Mbps has been chosen based on the

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

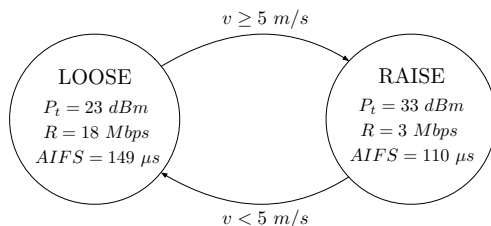


Figure 2.9: IASM state machine for an application that requires a maximum error of 0.5 meters and a minimum delay information interval of 0.1 seconds. Transmission power, data rate and listening period are adapted conditioned to the speed of the vehicle and the application requirements. Other parameters like the clear channel assessment threshold and the contention window are fixed.

work of Sepulcre et al. [SGC17]. A  $P_t$  of 23 dBm has been chosen to meet the SINR threshold (5 dB) required to decode a packet sent in RAISE state in case of a collision, plus a margin to account for signal attenuation due to large distances and shadowing dynamics because most of the vehicles with higher speeds are located further. This translates in a 10 dB ratio between both transmission powers at the senders which is two times the SINR threshold. Finally, conditions to distinguish between both states are:

$$\text{State} := \begin{cases} \text{LOOSE} & v t_{b,\min} < e_{th} \\ \text{RAISE} & \text{otherwise} \end{cases} \quad (2.4)$$

The rationale behind (2.4) is that high speeds are those whose contribution to the error is above the error requirement imposed by the application  $e_{th}$  during the period of time defined by the latency required. For example, in case of a required error threshold of 0.5 meters and a minimum delay information interval of 0.1 seconds, the LOOSE state is determined by the condition  $v < 5$  m/s, Figure 2.9.

#### Validation of Criteria

IASM was implemented over the evaluated protocols using the same simulation conditions and parameters of Section 2.3.2. Beacon frequency was adapted using the corresponding protocol's technique, while all other communication parameters were adapted using IASM. Figure 2.11 clearly illustrates the improvement on the CCDF of PE for the baseline protocol in both scenarios. Improvement on the CCDF is explained by IASM influence on the decrease in  $P_{\text{col}}$ , increased SINR values and MAC priority for high speed vehicles' packets. However, using baseline protocol the improvement is limited because of the over saturated channel. The protocol uses no adaptation of the beacon rate and the MAC protocol of the standard can not handle the large volume of beacons by itself. The improvement on maximum PE values is also limited because of the number of packets sent. IASM improves the mean and 95% percentile of PE by over 30%.

Figure 2.12 summarizes the improvement of IASM over all protocols sorted in ascending order of number of packets sent. One can clearly observe that error (PE mean and 95% percentile) improvement becomes larger when the

## 2. Contributions in ITS Data Acquisition

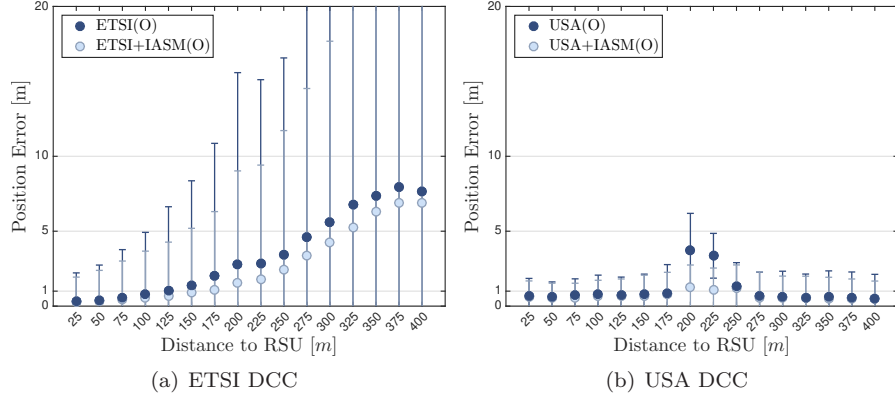


Figure 2.10: Comparison of the spatial distribution of PE using IASM on (a) ETSI DCC and (b) USA DCC in Scenario O. Mean values are represented by dots while 95% percentiles define the lengths of each bar. Only results obtained of two best performing protocols in worst case scenario conditions are shown.

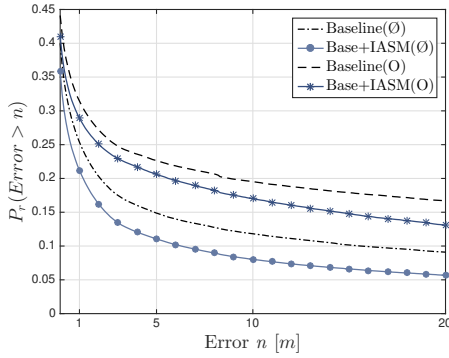


Figure 2.11: IASM improvement on the CCDF of PE using the baseline protocol for both scenarios.

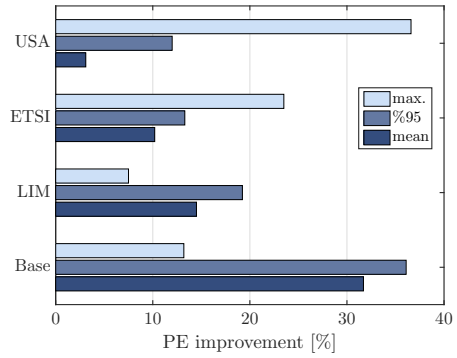


Figure 2.12: IASM improvement on PE over default studied protocols in Scenario O.

number of packets is increased. This becomes clear for USA DCC in Figure 2.10 and 2.12, where the limited improvement on the average PE is due to the low number of collisions. USA DCC approach limits IASM improvement as packets send at low speeds can become relevant in some circumstances. For instance, it only takes a few packets to predict a vehicle's trajectory traveling at constant low speed. Therefore, in this case, improving the reception of packets sent at high speeds over the aforementioned ones could not be optimal. However, IASM reduces uncertainty of USA DCC with a 10% improvement on the 95% percentile. On the other hand, changing the adaptation criteria of ETSI DCC to IASM, improves PE mean and 95% percentile over 10% while maximum values are improved over 23%. Besides, adding IASM criteria to LIMERIC improves the mean and 95% percentile by 14% and 20%, respectively.

Figure 2.10 shows the improvement on the spatial distribution of PE



### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

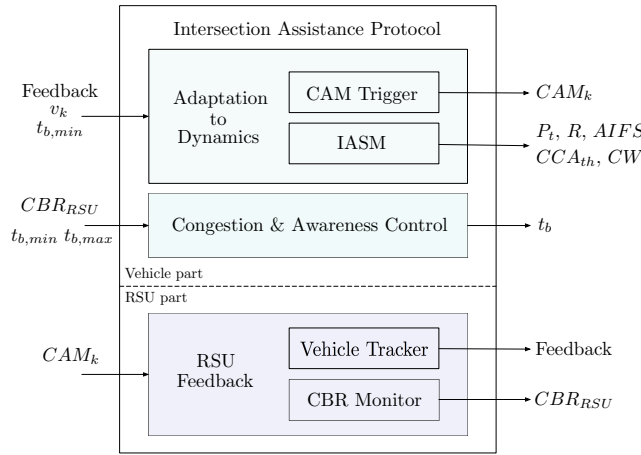


Figure 2.13: Design framework of the proposed IAP to enhance VANET performance towards RSU-based IAS requirements.

compared to results obtained using the default protocols. Now, medium accuracies are found up to almost 150 m using ETSI+IASM, which is a 50% range improvement w.r.t. default ETSI DCC. In Figure 2.10, maximum PE values can be found above the low precision scale for each distance interval. Therefore, IASM can not provide high precision on its own despite decreasing the uncertainty and the average PE. This indicates that the adaptation of the beacon rate is the one that most influences PE. In this sense, the protocols studied need to address directly the problem of correlated collisions to be able to achieve acceptable levels of accuracy.

#### 2.3.4 Intersection Assistance Protocol

Results of Section 2.3.3 validated that using IASM for adapting other communication parameters rather than beacon rate improved the performance of studied protocols. Section 2.3.3 discussed optimal beacon rate. This section proposes the Intersection Assistance Protocol (IAP), with a full adaptation of all parameters based on learning from Section 2.3.1, Section 2.3.2 and Section 2.3.3.

##### Design Framework

IAP is designed with the aim to enable RSU-based safety intersection assistance systems and it is aligned with standardization bodies guidelines. IAP design approach can be divided into three main blocks, illustrated in Figure 2.13:

- Adaptation to dynamics. This block uses IAS requirements of position accuracy and latency plus vehicle dynamics as input to: (i) set transmission power, data rate, sensitivity and beacon priority using IASM and (ii) send specific timed CAMs when needed, using ETSI's CAM triggering conditions, a trajectory prediction approach like USA DCC, feedback provided by the RSU or an imminent vehicle collision.



## 2. Contributions in ITS Data Acquisition

---

- Congestion and awareness control. This block controls congestion caused in the same intersection or by nearby interfering areas and provides up-to-date awareness required by applications and standards, using CBR information disseminated by the RSU in combination with maximum latency requirements. This solves some of the situations where trajectory prediction approaches does not comply with ETSI's standard minimum CAM frequency requirements, providing not enough awareness.
- RSU feedback. This block is intended to be integrated with the specific IAS operating in the intersection area. It uses received beacons to track each vehicle under its coverage area and compute the CBR. It periodically disseminates CBR information ( $CBR_{RSU}$ ) within ITS RSU standardized messages, e.g., intersection traffic light status (SPATEM), road topology (MAPEM) or infrastructure to vehicle Information (IVIM) [ETS16; SAE16a]. In addition, provides feedback about vehicles position, allowing vehicles to react to correlated or relevant packet loss. The standard contemplates that the ITS RSU may influence beacon rate of vehicles to increase safety [ETS18]. For example, the following information included in periodic messages can be exploited by the vehicles to avoid correlated collisions and maximum error values: a vector containing vehicle Ids from which a message has been received between consecutive messages or the Id of the vehicle which information has not been updated for the longest period of time.

### Implementation

A proof-of-concept implementation of IAP is implemented as follows. Algorithm 1 and 2, shown in pseudo-code, summarize the main procedures of the RSU and vehicles, respectively. It is assumed that the only feedback disseminated by the RSU is  $CBR_{RSU}$  and  $List_{RSU}$ , a list of 250 vehicle Ids whose packet has been received during last second. This information is encapsulated in a 1300 B packet broadcasted in CCH every second using default values of IEEE 802.11p.

1. Because of USA DCC approach proved a great potential: CAM triggering conditions are implemented using a deterministic trajectory prediction approach. In that sense, a beacon is sent only when the difference between the predicted position  $\hat{\mathbf{p}}$ , computed using a constant velocity model and the last velocity information sent in a CAM  $\mathbf{v}_j$ , and the actual position known by the vehicle  $\mathbf{p}$  exceeds  $e_{th} = 0.5$  m. On the other side, the RSU runs the same model to track each vehicle implementing the aforementioned *RSU Feedback: Vehicle Tracker* block. The use of a trajectory approach allows relaxing periodic beacon rate and aim for a controlled awareness under low probability of collision conditions.
2. Because LIMERIC proved a great adaptation to CBR and to overcome USA DCC approach limitations: a periodic beacon rate  $1/t_b$  is derived from LIMERIC implementation using  $CBR_{RSU}$  and a relaxed  $CBR_{max} = 0.25$  to meet Criterion 3, which is approximate derived from a  $P_{col} \leq 0.05$  [Som+15].  $CBR_{RSU}$  is used to overcome the discrepancy found in Section 2.3.2 between vehicle and RSU measures of CBR due to shadowing

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

---

effects. Also, the beacon period is limited to  $t_{b,\min} = 0.1$  s and  $t_{b,\max} = 1$  s as defined in the standard [ETS14].

3. Because protocols suffered from correlated packet loss causing not negligible maximum error values: if the vehicle Id is not present within two consecutive RSU packets, the next time scheduled beacon is randomized multiplying it by a uniform random variable in the range (0.001,1). Therefore, every vehicle increments an integer variable *notInListCounter* every time a RSU messages is received and does not contain the vehicle's Id.
4. Finally, IASM was used to adapt other communication parameters rather than beacon rate.

---

**Algorithm 1:** RSU procedure

---

```
tracker ← vehicleTracker()
while time step do
  %% Feedback and CBR monitor block
  CBRRSU ← computeCBR()
  ListRSU ← createListOfVehicles(CAMs)
  if 1 second elapsed then
    | disseminateCBR(CBRRSU, CCH)
  end
  %% Vehicle tracker block
  tracker(CAMs)
end
```

---

### Performance Evaluation

To validate the proposed implementation, we used the same simulation conditions and parameters of Section 2.3.2, despite the inclusion of the aforementioned RSU messages. Simulation results are illustrated in Figure 2.14 and 2.15, which clearly show a significant improvement on the CCDF and the spatial distribution of PE compared to Figure 2.8 and 2.10 against all studied protocols. IAP grants a probability of 99.56 % of the PE to be within medium accuracy scale for all its coverage area, which enables safety IAS contrary to the other protocols. Using IAP, 95 % percentile PE values under one meter accuracy are found from distances within 250 meters from the RSU. This is a great improvement on the PE uncertainty against the other protocols that were not able to provide sub-meter 95 % percentile values at any distance. If the aim is to implement critical safety applications requiring high accuracies, high maximum values are still found despite obtaining the lowest values of all simulated protocols. A maximum PE value of 7.12 m is found for distances within 25 m. To solve this, a more aggressive feedback from the RSU or a more elaborated protocol is required. In that sense, no feedback of vehicles position was used on the proposed protocols as this would require a detailed study and modifications. However, regarding non-safety applications, Figure 2.15 and Table 2.3 show that IAP provides in average PE values within high accuracy scale

## 2. Contributions in ITS Data Acquisition

---



---

### Algorithm 2: Vehicle procedure

---

```

%% Adaptation to dynamics block
while time step do
    %% CAM trigger block
     $\hat{\mathbf{p}}_k \leftarrow \hat{\mathbf{p}}_{k-1} + \mathbf{v}_j \Delta t$ 
    if  $\|\hat{\mathbf{p}}_k - \mathbf{p}_k\| \geq e_{th}$  then
        sendCAM()
         $\mathbf{v}_j \leftarrow \mathbf{v}_k$ 
         $\hat{\mathbf{p}}_k \leftarrow \mathbf{p}_k$ 
    end
    %% IASM block
    if  $v_k t_{b,min} < e_{th}$  then
         $P_t \leftarrow 23$  dBm
         $R \leftarrow 18$  Mbps
        AIFS  $\leftarrow 149$   $\mu s$ 
    else
         $P_t \leftarrow 33$  dBm
         $R \leftarrow 3$  Mbps
        AIFS  $\leftarrow 110$   $\mu s$ 
    end
    %% Congestion and awareness control block
    if  $t_b^k$  then
        sendCAM()
         $t_b^k \leftarrow \text{LIMERIC}(t_b^{k-1}, t_{b,min}, t_{b,max}, \text{CBR}_{\text{RSU}}, \text{CBR}_{\text{max}})$ 
        if  $\text{notInListCounter} > 1$  then
            |  $value \leftarrow \text{getUniformVariable}(0.001, 1)$ 
        else
            |  $value \leftarrow 1$ 
        end
        scheduleNextCAM( $value \times t_b$ )
    end
end
end

```

---

and great uncertainty values for all its coverage area. Therefore, information disseminated by IAP is reliable enough to enable non-safety applications.

Regarding CF, CBR values were found to oscillate near 0.25 with a maximum value found of 0.38 when congestion occurs and an overall average value of 0.2. This points out that IAP is able to keep the channel non-saturated increasing the probability of success of other messages with higher priority. Here, there is also room for improvement as LIMERIC linear parameters were set to constant values without any adaptation. Finally, Table 2.3 summarizes IAP performance and improvement over best results obtained in previous sections with significant 72.3 % to 87.7 % improvement values over the evaluated metrics of PE. In conclusion and loosely speaking, improvement comes from implementing a trajectory prediction approach with added redundant transmissions, randomized to avoid correlated packet collisions, derived from reliable V2I metrics under acceptable channel usage and awareness values.

### 2.3. Vehicle-to-Infrastructure Communication in Intersection Areas

Table 2.3: Summary of statistical performance of the proposed intersection assistance protocol. Improvement is calculated against best values of Table 2.1 among all protocols in the worst case scenario, the obstructed scenario O.

	IAP [m]	Improvement [%]
mean	0.08	87.7
95%	0.28	76.1
PE max.(50)	8.97	72.3
max.(100)	20.35	86.8
max.(400)	26	88.1

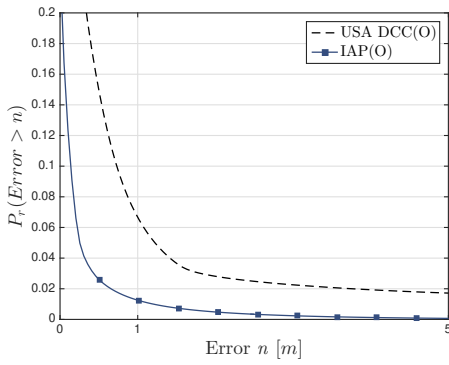


Figure 2.14: CCDF of PE of the proposed IAP in Scenario O compared to the best performing protocol of Section 2.3.2, USA DCC. Please note that axis scales changed to get more resolution when  $P_r \rightarrow 0$ .

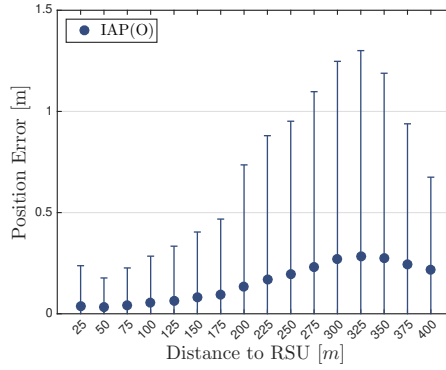


Figure 2.15: Spatial distribution of PE using IAP in Scenario O. Please note that y-axis scale changed compared to results of Figure 2.10.



## CHAPTER 3

---

# Deep Learning Solution for Exploiting Traffic Data in the ITS Data Analytics Layer

---

This chapter covers Objective II of the thesis by introducing a versatile framework to solve some of the major challenges of ITS and future traffic forecast. The model in [Boq+20] (Appendix B) is described in this chapter from the beginning, providing the necessary background to justify the adaptation to the field of ITS and the achievement of the objective. Section 3.3 goes over the key concepts that underpin the basis of the model: density estimation, latent variable models and variational inference. These are needed to understand the motivation, assumptions and derivations of the proposed solution, which uses deep neural networks for learning latent representations via stochastic parameter optimization. In case the reader is familiarized with the aforementioned, the solution is presented in Section 3.4. The model was evaluated in several ITS use cases with real world traffic data from California and UK. Section 3.5.1, Section 3.5.2 and Section 3.5.3 present the main results obtained in missing data imputation, dimension reduction and model selection, respectively. Section 3.5.4 explores the potential use in anomalous traffic detection.



“Develop a unique model for ITS to extract knowledge from traffic data to enhance traffic forecast, missing value imputation, model and data selection and anomaly detection.”

### 3.1 Related Work

Laña et al. [Lañ+18b] reviewed in 2018 the future major challenges of traffic forecasting. All models previously proposed in literature aim to solve only one of those challenges at the time. Trying to overcome that, we proposed a generative model based on the variational autoencoder (VAE) to solve missing data imputation, dimension reduction, model selection and anomaly detection. This is clearly a differential feature and novelty compared to state-of-the-art proposals (generative or non). [Boq+20] (Appendix B) is the first work that implements a VAE-based model for solving ITS related tasks. Compared to other generative models applied to the transportation field, Xen et. al [CHS19] proposed a Bayesian imputation model to characterize the data generation

### 3. Contributions in ITS Data Analytics

---

process and learn underlying statistical patterns in traffic data. The model can only be used to impute lost values, therefore, if the ITS requires addressing other traffic problems, such as how to forecast traffic or compress data, it is forced to implement other models, resources, practitioners, etc., contrary to the solution presented in Section 3.4 or [Boq+20] (Appendix B). More generally speaking, if we compare the proposal to generative models such as Restricted Boltzmann machine (RBM), Deep Belief Network (DBN) or the trendy Generative Adversarial Network (GAN)-based approaches, the VAE-based approach possesses certain desirable properties for ITS that are advantageous: stable training, interpretable encoder-decoder with a continuous latent space and outlier-robustness [Bro+16; Dai+18; Tol+17]. Additionally, GAN are known to be difficult to tune and train and RBM/DBN require careful model design to maintain tractability [Goo+16]. In conclusion, using the contributions in this chapter, the ITS traffic modeler can implement a unique model to compress the traffic data and efficiently forecast, impute missing values, select the best data and model for an specific problem and detect anomalous traffic data at the same time with no additional knowledge required and no labeled data. Related work in each ITS field is separately discussed below.

#### Missing Data Imputation

ITS are deployed in scenarios where sensor and system failure are common. Missing values are known to negatively affect the precision of the forecast, although they are often underestimated in current forecast models [Lañ+18b; Pam18; VKG14]. The current strategy is to preprocess the data by inferring the missing values from the known part of the data. Three well-known imputation methods in traffic forecasting are ARIMA, KNN and PCA-based methods. Among them, the probabilistic PCA is the most effective in terms of performance and implementation [LLL14] but, recently, more complex models have been proposed. A spatial context sensing model is proposed by Laña et. al [Lañ+18a], which is based on an automated clustering analysis tool and the information provided by surrounding sensors. Li et. al [Li+18] proposed a model that combines long-short term memory (LSTM), SVR and collaborative filtering. With a similar approach to the one presented in this chapter, Chen et. al [CHS19] proposed a Bayesian imputation model to characterize the data generation process and learn underlying statistical patterns in traffic data.

On the other hand, state-of-the-art imputation methods from other research fields that could potentially be ported to the transportation field can be classified as either discriminative, such as multiple imputation by chained equations (MICE) [BG10] and matrix completion [YRD16], or generative methods based on DNN. For example, Gondara et al. [GW17] proposed an overcomplete denoising autoencoder (DAE) to be able to reconstruct data by stochastically corrupting it. Closer to our work, Bowman et al. [Bow+15] and Jang et al. [JSK19] proposed a RNN-based VAE which succeed at imputing missing words from sentences. Fortuin et al. [FRM19] applied a deep sequential VAE with a Gaussian process prior in the latent space to capture temporal dynamics to impute real-world medical data. Similarly, Yoon et al. [YJS18] and Shang et al. [Sha+17] proposed also a generative model imputation method but using generative adversarial networks (GAN).

### Dimension Reduction

The number of features available from data sources jointly with the number of available data points in road networks are excessive. Forecasting with all those features can be computational inefficient and undertakes the risk of over-fitting. Therefore, it is essential to reduce the dimension of the feature space before applying a prediction model [YQ19]. Reduction of the data is done by learning the principal components or independent factors of a given data manifold, i.e., feature extraction. Recently, a systematic literature review of feature selection and extraction in spatiotemporal traffic forecasting was presented by Pavlyuk et al. [Pav19]. Note that feature extraction does not necessarily mean reducing the dimension of the data space, that is, dimension reduction is a subclass of feature extraction methods. The low-dimensional representation is traditionally obtained by PCA approaches that had been widely used to extract the linear correlations between the variables [LJY07; YQ19]. In DNN data-driven approaches, RNN and CNN are used to extract temporal and spatial characteristic within the regression model. Liu et. al [LWZ18] used a LSTM and CNN mixed with an attention layer, but can not be used as an independent layer to the regression task. Similarly to our work, features learned from a stack of autoencoders (SAE) have been previously used in literature to improve traffic forecasting [Lv+15; YDC17]. Contrary to the autoencoder, the VAE encourages the model to generalize features and reconstruct samples as an aggregation of those, forces the latent space to be continuous and is a generative model. Thus, other VAE approaches have been used successfully for dimensionality reduction within other research fields, such as fault diagnosis and towards sequencing the RNA of individual cells [San+18; WG18].

### Model & Data Selection

There is no best method that suits all situations in traffic forecast, which implies an applicability at a higher level of the method to choose the most suitable model given the characteristics of the forecasting problem [Lañ+18b; VV12]. Traffic modelers frequently face several optimization challenges related to model selection, while there are no clear baselines to find the best method and its configuration [ATM18]. According to the best of author's knowledge, few works are related to the traffic forecast context. Vlahogianni et al. [Vla15] proposed a metamodeling technique to optimize both algorithm selection and hyperparameter setting. Angarita et al. [ATM18] explored the use of AutoWEKA, an automatic algorithm selection method. On the contrary, the proposal of this chapter approaches the problem from a data perspective. The solution provides a tool based on the clustering of data in the learned latent space to select the data from which the best forecasting model will be built to solve a specific problem. Similar to this approach, Van et al. [VDW96] proposed a hybrid method of short-term traffic forecasting using a self-organized Kohonen map as an initial classifier, where each class had an individually associated tuned ARIMA model. The explanatory and representative power of models is also valuable for traffic modelers to obtain information on how transportation networks behave and evolve. Some efforts have been devoted to explain the behavior of the models in ITS literature as a second derivative of traffic forecast. For example, Polson et al. [PS17] discussed how the input variables



### 3. Contributions in ITS Data Analytics

---

relate to the predicted output using the coefficients of the fitted linear model. Wu et al. [Wu+18] analyzed the spatial features captured by CNN through characterizing the information that retained layer by layer. The proposal of this chapter is able to classify traffic in a continuous latent space with interpretable dimensions that can be used as a tool to perform model and data selection.

#### Anomaly Detection

One of the main applications of urban traffic analysis lies in detecting anomalies from traffic data [Dje+19]. Djenouri et al. [Dje+19] reviewed on existing outlier detection techniques in traffic data in three main categories: statistical, similarity-based and based on pattern analysis. Among them, some find outliers in subspaces, which is exactly what the VAE solution can provide. In [DNL15], dimensionality reduction is performed by PCA and a kNN-based outlier detection is applied in the derived subspace. On the contrary, this chapter proposal is based on DNN that has greater modeling capabilities. It is able to learn a latent space, where traffic samples are clustered and projections close to each are forced to have similar reconstructions helping in the detection of outliers. Moreover, when it is trained with much more normal samples than the anomalous ones, the reconstruction errors of normal data are relatively higher than those of anomalous data. Therefore, the model loss function provides an anomaly score function, which can be exploited as an anomaly detection technique. In that regard, VAE-based outlier detection methods had been used successfully in other research fields: Kawachi et al. [KKH18] added a supervised method to the VAE approach to enhance detection of seen anomalies without degrading the performance for unseen anomalies on real industrial data, Solch et al. [Söl+16] proposed a RNN-based VAE to detect anomalies on robot time series data and An et al. [AC15] proposed an anomaly detection method based on a reconstruction probability derived from the VAE loss function.

### 3.2 Road Traffic Forecast

Consider an ITS that collected an historical road traffic dataset composed of  $N \geq 1$  datapoints or *traffic samples* from a concrete road traffic network:

$$\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\} \equiv \{\mathbf{x}^{(i)}\}_{i=1}^N \equiv \mathbf{x}^{(1:N)}.$$

Let each element of a traffic sample represent a value of a traffic variable associated to an specific time and space:  $\mathbf{x}^{(i)} \in \mathbb{R}^{n \times d}$ , where  $n \geq 1$  is the number of past traffic variables and  $d \geq 1$  the number of traffic sensors deployed into the road network. Notice that we defined  $\mathbf{x}$  as a real-valued vector, which is intended to represent traffic variables such as speed, flow, density, etc. We assume a big data paradigm, that is, the data set  $\mathcal{D}$  is too large to fit in memory, so we can only work with small sub-sampled batches of  $\mathcal{D}$ . Henceforth, the superscript <sup>(i)</sup> denoting the  $i$ -th sample will be omitted to avoid clutter, except in cases where some ambiguity may exist.

**Real-world data:** For model evaluation purposes, three different kind of traffic data sets were gathered and cleaned. They are briefly described in Section 3.5, but more details can be found in [Boq+20] (Appendix B).

### 3.2.1 Forecasting Problem

Let  $\mathbf{y} \in \mathbb{R}^m$  denote the future state of the subset of  $m \leq d$  sensors in the time horizon of  $h \geq 1$  samples. The traffic forecast problem is usually modeled as  $\mathbf{y} = f^*(\mathbf{x})$ , where forecast systems aim to make an accurate estimate of  $\mathbf{y}$  from  $\mathbf{x}$ . The major challenge of the problem remains on deriving a function that closely resembles  $f^*$ . However, the widely-used regression-based road traffic forecasts tend to be noisy, uncertain and challenging in estimating confidence.

**Intuition:** Suppose we want to infer the traffic behavior during the next two hours or that we have a partially occluded traffic sample due to a sensor or system failure. Missing data could be anything if there is no underlying structure from which the data are generated. In that sense, we know that strong spatiotemporal relationships exist between road network's points [Lañ+18b]. For instances, due to seasonality, it is possible to discern between a work day or not just by observing how morning traffic develops through time and space, Figure 3.1.

### 3.2.2 Probabilistic Approach

In order to easy things to the subsequent supervised learning algorithm, another approach to enhance the forecast is to extract knowledge from the data set  $\mathcal{D}$  to preprocess the data or adjust the subsequent model accordingly. If we view the problem under the lens of probability, we can think of the observed data  $\mathcal{D}$  as a finite set of samples drawn from an unknown true probability distribution. Using lower case for notation simplicity, we will consider  $\mathbf{x}$  as a vector of observed continuous random variables whose true distribution we would like to approximate. Furthermore, we will assume that  $\{\mathbf{x}^{(i)}\}_{i=1}^N$  are i.i.d., i.e., that the data generating process is stationary. To account for this, we will mention later a mechanism to detect model drift as road networks and mobility patterns may change over time.

Although we do not have access to the true distribution, we denote uniform sampling from the finite data set as the empirical distribution  $p_{\text{data}}$ . Learning  $p_{\text{data}}(\mathcal{D})$  means to learn the underlying structure directly from data. Therefore, the natural question that follows is how we can learn a model to approximate  $p_{\text{data}}$  given access to the data set  $\mathcal{D}$ . In other words, the probability density must be approximated using the process known as probability density estimation (or learning a generative model).

## 3.3 Background on Probability

### 3.3.1 Learning from Data

Let  $p_{\text{model}}$  be a distribution from a family of candidate distributions  $\mathcal{P}_x$  that define a density over  $\mathbf{x}$ . The task of learning the generative model consists of picking the distribution  $p_{\text{model}} \in \mathcal{P}_x$  such that  $\mathbf{x} \sim p_{\text{model}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$ , that is, that the model distribution  $p_{\text{model}}$  approximates the empirical data distribution  $p_{\text{data}}$ . Mathematically, this can be written as the optimization problem

$$\min_{p_{\text{model}} \in \mathcal{P}_x} d(p_{\text{data}}, p_{\text{model}}), \quad (3.1)$$

### 3. Contributions in ITS Data Analytics

---

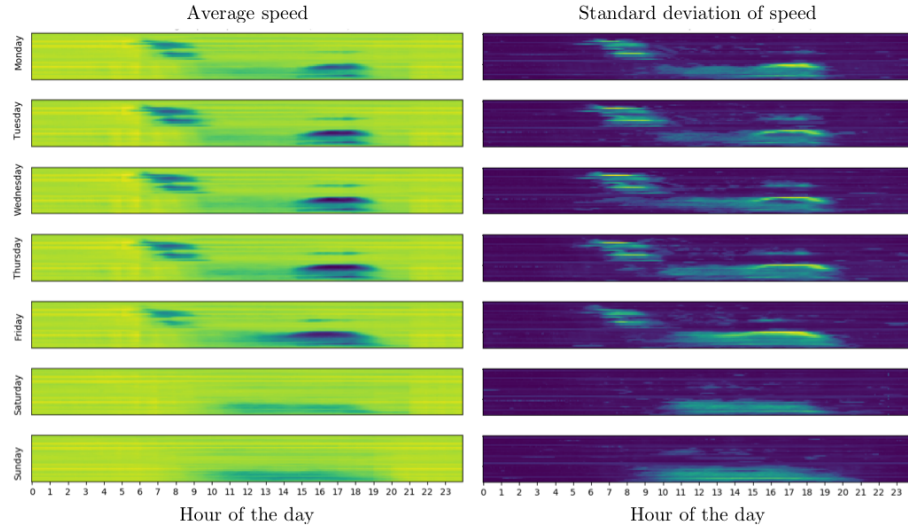


Figure 3.1: The average and standard deviation of speed measures for different days of the week in a two-year period of a road traffic network in California. The intensity of the color is proportional to the variable represented, for example, a darker color means lower speed in the figures on the left. The y-axis of each image represents 31 consecutive and equally spaced traffic sensors. Peak hours are clearly distributed differently for work days and weekends. The probabilistic approach is motivated by the fact that different traffic patterns exist, which suggests that traffic data are not randomly generated.

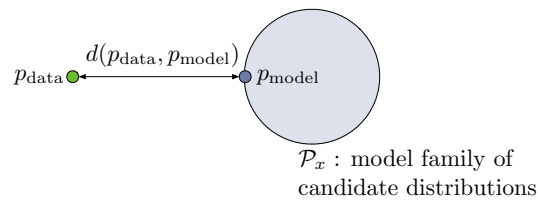


Figure 3.2: Density estimation or the process of learning a generative model finds the distribution  $p_{\text{model}}$  that best approximates  $p_{\text{data}}$  under some measure of similarity or *closeness*.

where  $d(\cdot)$  is a notion of *distance* between probability distributions that we would like to minimize, see Figure 3.2. At the time of solving (3.1) we would like to learn a generative model that can perform the following three inference tasks for ITS: **density estimation**, **sampling** and **unsupervised representation learning**, so that we can detect anomalous traffic samples, impute missing values and extract the best possible features to enhance the traffic forecast and compress the data. That is, given a data point  $\mathbf{x}$ :

- The model must compute its probability, i.e.,  $p_{\text{model}}(\mathbf{x})$ .
- The model must generate novel data from the model distribution, i.e., be capable of producing  $M$  new traffic samples  $\{\mathbf{x}^{(j)}\}_{j=1}^M$ ,  $\mathbf{x}^{(j)} \sim p_{\text{model}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$ .
- The model must learn meaningful feature representations with lower dimensions than the given data point  $\mathbf{x}$ .

To find the adequate  $p_{\text{model}}$  for the aforementioned inference tasks, next, we will define the objective function  $d(\cdot)$ , the representation for the family of distributions  $\mathcal{P}_x$  and the optimization procedure for minimizing it.

### 3.3.2 A Maximum Likelihood Problem

Let's start by first defining the objective function, so that we know the implications of choosing  $p_{\text{model}}$  to the computation and optimization complexity. A common approach is to use maximum likelihood estimation (MLE), because its estimator attains the Cramer-Rao lower bound on the variance when  $N \rightarrow \infty$ .

In MLE, the Kullback-Leibler (KL) divergence is used to measure the *distance* between probability distributions. From information theory, the KL divergence between  $p_{\text{data}}$  and  $p_{\text{model}}$  is defined as the expectation over  $p_{\text{data}}$  of the logarithmic difference, i.e.,

$$\begin{aligned} d_{\text{KL}}(p_{\text{data}} \parallel p_{\text{model}}) &= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[ \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x})} \right] \\ &\geq 0 \\ &\neq d_{\text{KL}}(p_{\text{model}} \parallel p_{\text{data}}). \end{aligned} \tag{3.2}$$

Note that (3.2) is non-negative and asymmetric, because the expectation is taken only from one distribution. As a note, when minimizing (3.2), it heavily penalizes model distribution  $p_{\text{model}}$  which place little mass on any data point that has a non-zero probability under  $p_{\text{data}}$ . For instance, if the density  $p_{\text{model}}(\mathbf{x})$  evaluates to zero for a data point sampled from  $p_{\text{data}}$ , the objective evaluates to  $+\infty$ . Using (3.2) as a measure of *distance* or similarity, the optimization problem (3.1) can be rewritten as

$$\min_{p_{\text{model}} \in \mathcal{P}_x} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[ \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x})} \right] \equiv \max_{p_{\text{model}} \in \mathcal{P}_x} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log p_{\text{model}}(\mathbf{x})]. \tag{3.3}$$

In practice,  $N \rightarrow \infty$  is not reasonable, but we can approximate the expectation over the unknown  $p_{\text{data}}$  of the right side of (3.3) with an unbiased Monte Carlo

### 3. Contributions in ITS Data Analytics

---

estimate. Recall that we assumed a number of finite  $N$  data points in the data set  $\mathcal{D}$  sampled *i.i.d.* from  $p_{\text{data}}$ , therefore,

$$\max_{p_{\text{model}} \in \mathcal{P}_x} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log p_{\text{model}}(\mathbf{x})] \simeq \max_{p_{\text{model}} \in \mathcal{P}_x} \frac{1}{N} \sum_{i=1}^N \log p_{\text{model}}(\mathbf{x}^{(i)}). \quad (3.4)$$

Interestingly, to see the relation of  $d_{\text{KL}}$  with MLE, we can derive the MLE criterion starting from (3.3) as

$$\begin{aligned} \min_{p_{\text{model}} \in \mathcal{P}_x} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \left[ \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x})} \right] &\iff \min_{p_{\text{model}} \in \mathcal{P}_x} -\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log p_{\text{model}}(\mathbf{x})] \\ &\iff \max_{p_{\text{model}} \in \mathcal{P}_x} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log p_{\text{model}}(\mathbf{x})] \\ &\stackrel{N \rightarrow \infty}{\iff} \max_{p_{\text{model}} \in \mathcal{P}_x} \frac{1}{N} \sum_{i=1}^N \log p_{\text{model}}(\mathbf{x}^{(i)}) \\ &\iff \max_{p_{\text{model}} \in \mathcal{P}_x} \frac{1}{N} \prod_{i=1}^N p_{\text{model}}(\mathbf{x}^{(i)}) \\ &\iff \max_{p_{\text{model}} \in \mathcal{P}_x} p_{\text{model}}(\mathcal{D}), \end{aligned}$$

to proof that minimizing the KL divergence is equivalent to maximizing the likelihood. Furthermore, to see why (3.2) is a measure of *distance* or similarity, we assume that both distributions are continuous and expand the corresponding definition of KL divergence as

$$\begin{aligned} d_{\text{KL}}(p_{\text{data}} || p_{\text{model}}) &= \int p_{\text{data}}(\mathbf{x}) \log \left( \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{model}}(\mathbf{x})} \right) d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\text{model}}(\mathbf{x}) d\mathbf{x} \\ &= -\underbrace{\text{entropy}(p_{\text{data}})}_{\text{Information in } \mathcal{D}} + \underbrace{\text{crossentropy}(p_{\text{data}}, p_{\text{model}})}_{\text{Information in model}}, \end{aligned}$$

where we see that the KL divergence is measuring the difference between the negative entropy (a quantity of information) of the data and the cross-entropy between the data and the model. Since the entropy of the data remains constant, this also proofs that maximizing the likelihood is equivalent to minimizing the cross-entropy, a matching information problem.

Notice that in (3.4) the true distribution might not actually be inside of the family of likelihoods we perform MLE over. Despite that, MLE will pick the distribution  $p_{\text{model}} \in \mathcal{P}_x$  that maximizes the average of the log-probability of the observed data points in  $\mathcal{D}$ . Thus, there is an inherent bias-variance trade off when selecting the hypothesis family of distributions: We want a model that is sufficiently rich to be useful, yet not so complex as to overfit the training set. This restricts us to only a set of relative simple distributions (e.g., the known exponential family) because

- $p_{\text{model}} \in \mathcal{P}_x$  needs to have a computationally tractable density, since the objective (3.4) requires to compute  $p_{\text{model}}$ , and
- $\mathcal{P}_x$  needs to be flexible but not so expressive to include  $p_{\text{data}}$  (to avoid memorization).

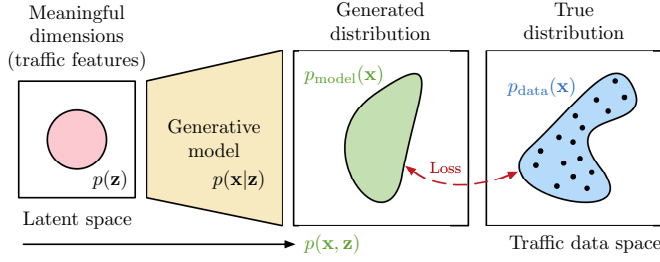


Figure 3.3: Graphical representation of the latent variable model. If  $\mathbf{z}$  captures meaningful information about  $\mathbf{x}$ , the generative process can be viewed as generating the *high-level* information about the traffic data,  $\mathbf{z} \sim p(\mathbf{z})$ , before fully generating the traffic data,  $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$ . Black dots represent each observed traffic samples  $\mathbf{x}^{(i)}$ .

### 3.3.3 Latent Space of Traffic Data

Consequently, we now introduce a direct latent variable model (LVM) to infer the hidden structure in the underlying data, which also narrows the value space of candidate distributions. Let  $\mathcal{Z} = \{\mathbf{z}^{(i)}\}_{i=1}^N$  be a set of vectors composed of random variables defining a representation of the significant factors of variation in  $\mathcal{D}$ . Then, a complex joint distribution can be defined as a product of two relative simple distributions:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}), \quad (3.5)$$

where  $\mathbf{x}$  and  $\mathbf{z}$  denote the observed and latent variables, respectively. This generative model with latent space, Figure 3.3, will allow us to transform the traffic data into a simpler space, explore it in that space and understand it better, i.e., learn meaningful feature representations of traffic data. Moreover, we would be able to perform dimension reduction if we constrain the latent space to have fewer dimensions than the data:  $\mathbf{z} \in \mathbb{R}^K, K \ll n \times d$ .

Hence, let  $\mathcal{P}_z$  be a family of relative simple distributions where  $p(\mathbf{z}) \in \mathcal{P}_z$  defines a density over  $\mathbf{z}$ . Let  $\mathcal{P}_{x|z}$  be a family of relative simple conditional distributions where  $p(\mathbf{x}|\mathbf{z}) \in \mathcal{P}_{x|z}$  describes a conditional distribution over  $\mathbf{x}$  given  $\mathbf{z}$ . Then, we construct the family of candidate distributions as

$$\mathcal{P}_{x,z} = \{p(\mathbf{x}, \mathbf{z}) | p(\mathbf{z}) \in \mathcal{P}_z, p(\mathbf{x}|\mathbf{z}) \in \mathcal{P}_{x|z}\} = \mathcal{P}_z \times \mathcal{P}_{x|z},$$

where each  $(p(\mathbf{x}|\mathbf{z}), p(\mathbf{z})) \in \mathcal{P}_{x,z}$  defines the joint distribution (3.5) over the observed variable  $\mathbf{x}$  and the latent variable  $\mathbf{z}$ . Notice that we have deliberately chosen  $\mathcal{P}_{x,z}$  such that (3.5) has a computationally tractable density.

Our goal is still to fit the marginal distribution over the visible variables to that observed in our data set. Hence our previous discussion about KL divergences applies here as well and by the same argument, we should be maximizing the marginal log-likelihood of the data. However, the computation of  $p_{\text{model}}(\mathbf{x})$  in order to maximize the log-likelihood (3.4) requires marginalization of the latent variable  $\mathbf{z}$ , which given (3.5) takes the form of

$$\log p_{\text{model}}(\mathbf{x}) = \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[p(\mathbf{x}|\mathbf{z})]. \quad (3.6)$$

### 3. Contributions in ITS Data Analytics

---

This integral is unavailable in closed form for many models and becomes intractable for high-dimensional  $\mathbf{z}$  as it involves the integration over all of its dimensions, which requires exponential time to compute. One solution is to approximate the integral using Monte-Carlo sampling methods such as

$$\log p_{\text{model}}(\mathbf{x}) \approx \log \frac{1}{K} \sum_{i=1}^K p(\mathbf{x}|\mathbf{z}^{(i)}), \mathbf{z}^{(i)} \sim p(\mathbf{z}),$$

which are unbiased as  $N \rightarrow \infty$  but suffer from high variance, i.e., poor generalization. In practice, traffic data is limited and we want our model to generalize to unseen traffic samples. Another solution is to apply variational inference, which derives a low variance but biased solution as we will explain next. For notation simplicity and to avoid clutter, we will omit the subscript  $\text{model}$  in the remaining text.

#### 3.3.4 The Variational Inference Way

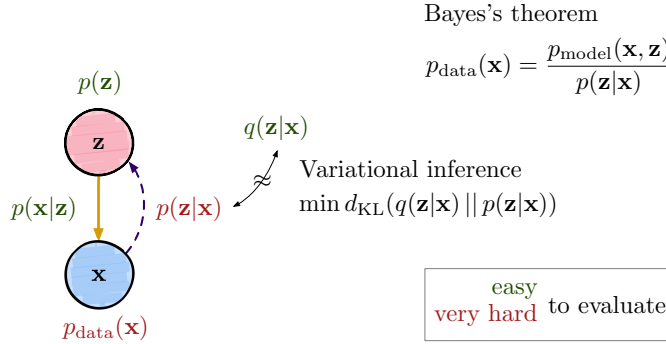
We know by definition from Bayes' theorem that

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})}, \quad (3.7)$$

where  $p(\mathbf{z}|\mathbf{x})$  is the posterior over the latent variables  $\mathbf{z}$ . Instead of evaluating the marginal likelihood  $p(\mathbf{x})$ , variational inference tries to approximate the intractable posterior  $p(\mathbf{z}|\mathbf{x})$  with a tractable distribution  $q$  converting a difficult computation problem to an optimization problem, see Figure 3.4. Thus, we introduce a family of conditional distributions  $\mathcal{Q}$ , where  $q(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$  defines a distribution over  $\mathbf{z}$  conditioned on  $\mathbf{x}$ . Variational inference consists of minimizing the KL divergence between  $q(\mathbf{z}|\mathbf{x})$  and  $p(\mathbf{z}|\mathbf{x})$  (in that order) to construct a lower bound on the likelihood (the bias). The divergence between both distributions is computed as

$$\begin{aligned} d_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}|\mathbf{x})) &= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x})} \right] \\ &= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{q(\mathbf{z}|\mathbf{x})}{\frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})}} \right] \\ &= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right] \\ &= \log p(\mathbf{x}) - \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right]. \end{aligned} \quad (3.8)$$

Despite choosing a proper family of variational distributions, the  $\log p(\mathbf{x})$  itself cannot easily be computed, i.e., we cannot actually minimize the KL divergence directly. Instead, what variational inference does is to maximize a lower bound on that quantity.



Bayes's theorem

$$p_{\text{data}}(\mathbf{x}) = \frac{p_{\text{model}}(\mathbf{x}, \mathbf{z})}{p(\mathbf{z}|\mathbf{x})}$$

Figure 3.4: A graph of the latent variable model  $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$  and the variational inference approach, which gives us ways to perform both inference (evaluate  $p(\mathbf{z}|\mathbf{x})$ ) and maximum likelihood parameter learning (learn  $p(\mathbf{x})$ ), by approximating the intractable posterior  $p(\mathbf{z}|\mathbf{x})$  with a simpler distribution  $q(\mathbf{z}|\mathbf{x})$  called the approximate posterior or inference model.

### 3.3.5 Evidence Lower Bound Objective

To derive the lower bound, we rearrange the terms in (3.8) for clarity as

$$\log p(\mathbf{x}) = d_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}|\mathbf{x})) + \underbrace{\mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right]}_{\mathcal{L}}, \quad (3.9)$$

where the first term of the right-hand side of the equation is the quantity we aim to minimize and that we denoted the second term as  $\mathcal{L}$ . Minimizing the divergence between both distributions is the same as maximizing  $\mathcal{L}$ , because the left hand side of (3.9) is fixed for a given  $\mathbf{x}$  as does not depend on  $q$  (recall that KL divergence (3.2) is non-negative) Hence,  $\mathcal{L}$  is less than or equal to the log marginal probability of the observations (or evidence), which is known as the evidence (or variational) lower bound (ELBO). This is the same bound used in deriving the expectation-maximization (EM) algorithm. All together, the ELBO for our probability latent variable model  $p(\mathbf{x}, \mathbf{z})$  (i.e.,  $p_{\text{model}}$ ) and the approximation  $q(\mathbf{z}|\mathbf{x})$  to the intractable posterior is

$$\begin{aligned} \mathcal{L}_{p(\mathbf{x}, \mathbf{z}), q(\mathbf{z}|\mathbf{x})}(\mathbf{x}) &= \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log q(\mathbf{z}|\mathbf{x})] \\ &= \log p(\mathbf{x}) - d_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}|\mathbf{x})) \\ &\leq \log p(\mathbf{x}). \end{aligned} \quad (3.10)$$



### 3. Contributions in ITS Data Analytics

---

As a side note,  $\mathcal{L}$  is often derived in literature using the Jensen's inequality on the log probability of the observation as

$$\begin{aligned} \log p(\mathbf{x}) &= \log \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} \\ &= \log \int p(\mathbf{x}, \mathbf{z}) \frac{q(\mathbf{z}|\mathbf{x})}{q(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &= \log \left( \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right] \right) \\ &\geq \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right], \end{aligned}$$

like in the original literature on variational inference [Jor+99].

Notice that (3.10) imposes to choose a family of variational distributions such that the two expectations can be computed and, interestingly, that the second expectation is the entropy. Now, instead of computing  $\log p(\mathbf{x})$  by marginalization, we optimize the ELBO by maximizing (3.10) over  $q \in \mathcal{Q}$ :

$$\max_{q \in \mathcal{Q}} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right] = \log p(\mathbf{x}) - \min_{q \in \mathcal{Q}} d_{\text{KL}}(q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}|\mathbf{x})),$$

which is equivalent to finding  $q(\mathbf{z}|\mathbf{x})$  that best approximates the intractable posterior  $p(\mathbf{z}|\mathbf{x})$ . Moreover, we see that the KL divergence between both distributions is the gap (or bias) between the original objective (the marginal log-likelihood) and the ELBO. The gap equals to zero when the variational distribution  $q(\mathbf{z}|\mathbf{x})$  matches the posterior  $p(\mathbf{z}|\mathbf{x})$ , thus the *tightness* of the lower bound depends on the choice of  $q \in \mathcal{Q}$  and the optimization procedure.

Finally, we can learn a latent variable model by maximizing  $\mathcal{L}$  for any given data point  $\mathbf{x}$  with respect to the model and the variational distributions by

$$\max_{p \in \mathcal{P}_{x,z}, q \in \mathcal{Q}} \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right],$$

which learns to approximate  $p_{\text{data}}$  while simultaneously approximating the posterior of our latent variable model. Furthermore, as we assumed a data set  $\mathcal{D}$  with i.i.d. data, the optimization problem can be written as maximization of the sum of individual data point ELBO's:

$$\max_{p \in \mathcal{P}_{x,z}, q \in \mathcal{Q}} \sum_{\mathbf{x} \in \mathcal{D}} \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z}|\mathbf{x})} \right]. \quad (3.11)$$

In summary, we just defined the objective (3.11) to optimize and the requirements for  $p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$  and  $q(\mathbf{z}|\mathbf{x})$ , but the open issues that have yet to be resolved are the exact choice of distributions and how to optimize the target within the transportation field.

### 3.4 ITS Big Data Analytics Solution

There already exists standard approaches that try solve the main inference tasks of the model. The EM algorithm can be used to learn latent variable models,

however, performing the expectation step requires computing the approximate posterior, which we have assumed to be intractable. To perform approximate inference, we may use mean field, but one step of mean field requires us to compute an expectation whose time complexity scales exponentially. Another approach would be to use sampling-based methods that do not scale well to large data sets [KW13]. Instead, parametric generative models scale more efficiently with large data sets like  $\mathcal{D}$ .

### 3.4.1 Neural Network Parametrization

We will assume that  $\mathbf{x}$  comes from a population that can be adequately modeled by a probability distribution that has a fixed set of parameters: Large data sets with varying dimensions are commonly found in the transportation field. Henceforth, we will assume a parametric setting where the distribution  $p(\mathbf{x}, \mathbf{z}) \in \mathcal{P}_{x,z}$  is specified via the set of parameters  $\boldsymbol{\theta}$ , thus we will denote it as  $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ . Likewise, we assume the variational distribution  $q_{\phi}(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$  is specified via the set of parameters  $\phi$ . The ELBO (3.10) and the objective to optimize (3.11) are rewritten as

$$\mathcal{L}_{\boldsymbol{\theta}, \phi}(\mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\log q_{\phi}(\mathbf{z}|\mathbf{x})] \quad (3.12)$$

and

$$\max_{\boldsymbol{\theta}, \phi} \sum_{\mathbf{x} \in \mathcal{D}} \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right], \quad (3.13)$$

respectively. This allow us to parametrize the model using deep neural networks, taking advantage of their flexible modeling capabilities and efficient optimization. That is,  $\{\boldsymbol{\theta}, \phi\}$  are the weights and biases of two deep neural networks, which will be learned relying on stochastic gradient descent (SGD) or similar optimization methods.

### 3.4.2 Variational Autoencoder

A solution to this framework was proposed by Kingma et al. [KW13] and Rezende et al. [RMW14] and it is known as VAE, an efficient deep learning technique for learning latent representations in case of intractable true posterior and large data sets. In VAE, the whole data model may be viewed as consisting of two parts that form an autoencoder architecture. An autoencoder is a pair of neural networks encoder and decoder trained to minimize the reconstruction error  $\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ , such that  $\hat{\mathbf{x}} = \text{decoder}(\text{encoder}(\mathbf{x}))$ . In practice,  $\text{encoder}(\mathbf{x})$  learns an embedding representation of  $\mathbf{x}$  in a latent space that often has an intuitive interpretation. For example, it is known that linear autoencoders learn to span the same subspace than PCA. Analogously, we have

- the encoder network  $g_{\phi}$  as the approximate posterior (also known as inference model)  $q_{\phi}(\mathbf{z}|\mathbf{x})$ , that given a data point  $\mathbf{x}$  it produces a distribution over the possible values of  $\mathbf{z}$  from which the data point  $\mathbf{x}$  could have been generated, and
- the decoder network as the generative model  $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ , which given  $\mathbf{z}$  it produces a distribution over the possible corresponding values of  $\mathbf{x}$ .

### 3. Contributions in ITS Data Analytics

---

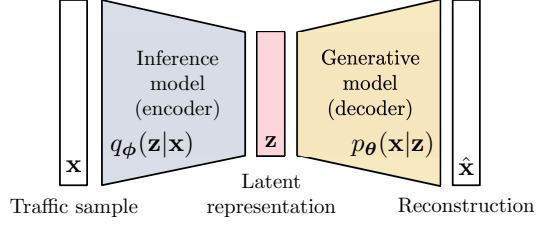


Figure 3.5: The VAE framework adds an inference model  $q_\phi(\mathbf{z}|\mathbf{x})$  parameterized with a neural network to approximate the intractable true posterior distribution  $p_\phi(\mathbf{z}|\mathbf{x})$ . This may be viewed as a *stochastic* autoencoder as the input and output of the model should be the same. The model does not learn the exact data distribution  $p_{\text{data}}$  but an approximation (3.10), which can be optimized via stochastic gradient descent.

The autoencoder interpretation comes from an straightforward reparametrization of the ELBO (3.12) as

$$\mathcal{L}_{\theta, \phi}(\mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - d_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z})), \quad (3.14)$$

using both Bayes and KL divergence definitions. Notice that the right-hand side consists of two terms that involve taking a sample  $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ , which can be interpreted as a code describing the observed data point  $\mathbf{x}$ . The first term  $\log p_\theta(\mathbf{x}|\mathbf{z})$  is the log-likelihood of the observed  $\mathbf{x}$  given the sampled code  $\mathbf{z}$ . This term is maximized when  $p_\theta(\mathbf{x}|\mathbf{z})$  assigns high probability to the original  $\mathbf{x}$ , thus it is trying to reconstruct  $\mathbf{x}$  given the code  $\mathbf{z}$ . For that reason  $p_\theta(\mathbf{x}|\mathbf{z})$  is called the decoder network and the term is called the negative reconstruction error. The second term (or regularization term) is the divergence between  $q_\phi(\mathbf{z}|\mathbf{x})$  and the prior  $p(\mathbf{z})$ . It encourages the codes  $\mathbf{z}$  to look like the defined  $p(\mathbf{z})$ , thus it prevents  $q_\phi(\mathbf{z}|\mathbf{x})$  from simply encoding an identity mapping (i.e., copy the input). Instead, forces it to learn some more interesting representation, hopefully traffic features. In summary, our optimization objective is trying to fit a  $q_\phi(\mathbf{z}|\mathbf{x})$  that will map  $\mathbf{x}$  into a useful latent space  $\mathbf{z}$  from which the model is able to reconstruct  $\mathbf{x}$  via  $p_\theta(\mathbf{x}|\mathbf{z})$  to  $\hat{\mathbf{x}}$ , see Figure 3.5.

#### Stochastic Gradient Descent on the ELBO

The ELBO allows joint optimization with respect to  $\theta$  and  $\phi$  using mini-batch SGD. This technique allows us to sub-sample the data set during optimization but requires the objective (3.13) to be differentiable with respect to  $\theta$  and  $\phi$  at the same time. On one hand, we can easily compute unbiased gradients of the ELBO (3.12) with respect to the generative model parameters  $\theta$  via a Monte Carlo estimator as the expectation does not depend on  $\theta$ :

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\theta} \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}, \mathbf{z})] \\ &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\nabla_{\theta} \log p_\theta(\mathbf{x}, \mathbf{z})] \\ &\simeq \frac{1}{K} \sum_{i=1}^K \nabla_{\theta} \left( \log p_\theta(\mathbf{x}, \mathbf{z}^{(i)}) \right), \mathbf{z}^{(i)} \sim p(\mathbf{z}). \end{aligned}$$

On the other hand, when trying to do the same with respect to  $\phi$ , we cannot push the gradient through the expectation since the expectation depends on  $\phi$ :

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\phi} \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x})] \\ &\neq \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\nabla_{\phi} (\log p_{\theta}(\mathbf{x}, \mathbf{z}) - \log q_{\phi}(\mathbf{z}|\mathbf{x}))].\end{aligned}$$

To resolve this issue, we would like to sample from  $q_{\phi}(\mathbf{z}|\mathbf{x})$  at the same time that the expectation does not depend on it. We can obtain a naive unbiased estimator using the log-derivative trick to estimate the gradient of an expectation before approximating via Monte-Carlo sampling as

$$\begin{aligned}\nabla_{\phi} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\phi} \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] \\ &= \int \nabla_{\phi} q_{\phi}(\mathbf{z}|\mathbf{x}) \left( \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\ &= \int \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \nabla_{\phi} q_{\phi}(\mathbf{z}|\mathbf{x}) \left( \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\ &= \int q_{\phi}(\mathbf{z}|\mathbf{x}) \nabla_{\phi} \log q_{\phi}(\mathbf{z}|\mathbf{x}) \left( \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) d\mathbf{z} \\ &= \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \nabla_{\phi} \log q_{\phi}(\mathbf{z}|\mathbf{x}) \left( \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right) \right] \\ &\simeq \frac{1}{K} \sum_{i=1}^K \left[ \nabla_{\phi} \log q_{\phi}(\mathbf{z}^{(i)}|\mathbf{x}) \left( \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z}^{(i)})}{q_{\phi}(\mathbf{z}^{(i)}|\mathbf{x})} \right) \right], \mathbf{z}^{(i)} \sim p(\mathbf{z}),\end{aligned}$$

which is often noted to suffer from high variance [DW19].

### Reparametrization Trick

Instead, a change of variables called the *reparametrization trick* was noted empirically to have lower variance [KW13; RMW14]. The trick consists of expressing  $\mathbf{z}$  as some deterministic, differentiable and invertible transformation  $T$  of another random variable  $\epsilon$ , such that the procedure

$$\begin{aligned}\epsilon &\sim p(\epsilon) \\ \mathbf{z} &= T_{\phi}(\epsilon, \mathbf{x})\end{aligned}$$

is equivalent to sampling from  $q_{\phi}(\mathbf{z}|\mathbf{x})$  and holds

$$\nabla_{\phi} \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})} \right] = \nabla_{\phi} \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[ \log \frac{p_{\theta}(\mathbf{x}, T_{\phi}(\epsilon, \mathbf{x}))}{q_{\phi}(T_{\phi}(\epsilon, \mathbf{x})|\mathbf{x})} \right].$$

Thus,  $q_{\phi}(\mathbf{z}|\mathbf{x})$  needs to be chosen judiciously so that the trick is possible. Under the reparametrization, the expectation and gradient operators become commutative by the law of the unconscious statistician, so we can form a one-sample Monte Carlo unbiased estimator for the gradients of the ELBO with

### 3. Contributions in ITS Data Analytics

---

respect to  $\phi$  as

$$\begin{aligned}
 \nabla_{\phi} \mathcal{L}_{\theta, \phi}(\mathbf{x}) &= \nabla_{\phi} \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[ \log \frac{p_{\theta}(\mathbf{x}, T_{\phi}(\epsilon, \mathbf{x}))}{q_{\phi}(T_{\phi}(\epsilon, \mathbf{x})|\mathbf{x})} \right] \\
 &= \mathbb{E}_{\epsilon \sim p(\epsilon)} \left[ \nabla_{\phi} \log \frac{p_{\theta}(\mathbf{x}, T_{\phi}(\epsilon, \mathbf{x}))}{q_{\phi}(T_{\phi}(\epsilon, \mathbf{x})|\mathbf{x})} \right] \\
 &\simeq \nabla_{\phi} \log \frac{p_{\theta}(\mathbf{x}, T_{\phi}(\epsilon, \mathbf{x}))}{q_{\phi}(T_{\phi}(\epsilon, \mathbf{x})|\mathbf{x})}.
 \end{aligned} \tag{3.15}$$

Notice that the complexity of the computation of  $\log q_{\phi}(\mathbf{z}|\mathbf{x})$  required by the ELBO estimator in (3.15) depends on the right choice of  $p(\epsilon)$  and  $T_{\phi}(\epsilon, \mathbf{x})$ . The resulting gradient estimator is used to update  $\{\theta, \phi\}$  at each mini-batch sample while SGD has not converged and can be directly implemented in machine learning software platforms like the well-known Tensorflow. During the procedure, the ELBO is optimized stochastically since noise originates from both the mini-batch random sampling and  $\epsilon \sim p(\epsilon)$ .

#### 3.4.3 Exact Model Definition

Now, we have a differentiable ELBO that can be optimized via SGD. Next, we need to specify the exact model distributions to be able to implement it. First, we let the encoder network model a multivariate Gaussian with a diagonal co-variance structure with means  $\mu_{\phi}(\mathbf{x})$  and standard deviations  $\sigma_{\phi}(\mathbf{x})$ ,

$$\begin{aligned}
 q_{\phi}(\mathbf{z}|\mathbf{x}) &= \mathcal{N}(\mathbf{z}|\mu_{\phi}, \text{diag}(\sigma_{\phi}^2)) \\
 &= \prod_i q_{\phi}(z_i|\mathbf{x}),
 \end{aligned} \tag{3.16}$$

where the dependency on  $\mathbf{x}$  of the moments was omitted to avoid clutter. The encoder is implemented using a 1 hidden layer multi-layer perceptron (MLP) with weights and biases  $\phi = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ . Thus, the outputs of the encoder network are the moments of the approximate posterior  $q_{\phi}(\mathbf{z}|\mathbf{x})$ , which are computed given a traffic sample  $\mathbf{x}$  and the encoder hidden layer  $\mathbf{h}^{\text{encoder}}$  as

$$\begin{aligned}
 \mathbf{h}^{\text{encoder}} &= \text{LReLU}(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) \\
 \mu_{\phi} &= \mathbf{W}_2 \mathbf{h}^{\text{encoder}} + \mathbf{b}_2 \\
 \sigma_{\phi} &= \mathbf{W}_3 \mathbf{h}^{\text{encoder}} + \mathbf{b}_3,
 \end{aligned} \tag{3.17}$$

where REctified Linear activation Unit (ReLU) and Leaky ReLU (LReLU) are nonlinear activation functions that proved to work. The factorized Gaussian model allows a simple choice of  $p(\epsilon)$  and  $T_{\phi}(\epsilon, \mathbf{x})$  and the computation of an unbiased estimator for the gradients of our objective. After reparametrization and equivalently to sample  $\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})$ , we can obtain during training a latent representation (or code) following the procedure

$$\begin{aligned}
 \epsilon &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
 \mathbf{z} &= \mu_{\phi} + \epsilon \odot \sigma_{\phi},
 \end{aligned}$$

where  $\odot$  is the element-wise product. Secondly, to specify the generative model  $p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ , we let the prior over the latent variables be the centered

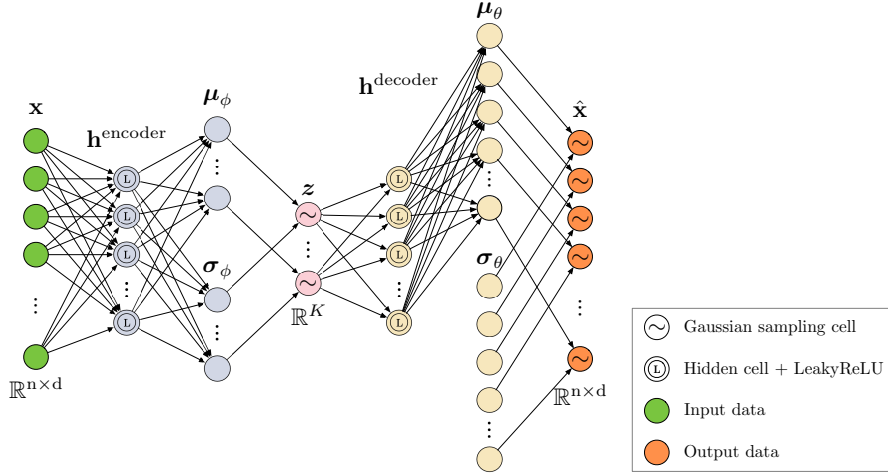


Figure 3.6: Deep neural network graph of the model implemented in Section 3.4.3.

isotropic multivariate Gaussian,

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}).$$

Note that the zero mean and unit variance introduced jointly with the KL term in (3.14) allows for a continuous latent (or code) space. Without the KL term, the encoder could learn to give each traffic sample a representation in a different region of the latent space. Rather, we are interested in having the samples closest to each other have similar meanings. Furthermore, we let the decoder network model a multivariate Gaussian with means  $\boldsymbol{\mu}_\theta(\mathbf{z})$  and unit variances  $\boldsymbol{\sigma}_\theta = \mathbf{I}$ ,

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_\theta(\mathbf{z}), \mathbf{I}).$$

The decoder is implemented using a 1 hidden layer MLP with weights and biases  $\boldsymbol{\theta} = \{\mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_4, \mathbf{b}_5\}$ . As we deliberately chose not to learn the variance parameter of our observation, the output is computed given a latent representation  $\mathbf{z}$  and the decoder hidden layer  $\mathbf{h}^{\text{decoder}}$  as

$$\begin{aligned} \mathbf{h}^{\text{decoder}} &= \text{LReLU}(\mathbf{W}_4 \mathbf{z} + \mathbf{b}_4) \\ \hat{\mathbf{x}} &= \boldsymbol{\mu}_\theta(\mathbf{z}) = \mathbf{W}_5 \mathbf{h}^{\text{decoder}} + \mathbf{b}_5, \end{aligned} \quad (3.18)$$

where the reconstruction  $\hat{\mathbf{x}}$  is directly  $\boldsymbol{\mu}_\theta(\mathbf{z})$ . See Figure 3.6 for the full model implementation graph and Section 3.4.4 to know about the motivation of these choices.

### Computation of the Loss Function

To compute the first term of the ELBO given the defined model and a traffic sample, we can estimate the expectation of the reconstruction error in (3.14) using  $L$  samples of  $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$  per data point  $\mathbf{x}^{(i)}$  as

$$\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] \simeq \frac{1}{L} \sum_{i=1}^L \log p_\theta(\mathbf{x}|\mathbf{z}^{(i)}).$$

### 3. Contributions in ITS Data Analytics

---

When optimizing via SGD or similar, in practice it is enough to set  $L = 1$  as long as the mini-batch size is large enough. Since the variance for the inference model  $\sigma_\theta$  was fixed to 1, we can minimize the  $l_2$  norm between  $\mathbf{x}$  and  $\hat{\mathbf{x}} = \mu_\theta(\mathbf{z})$  analogously to maximize  $\log p_\theta(\mathbf{x}|\mathbf{z})$ , as we know that for a multivariate Gaussian:

$$\log p_\theta(\mathbf{x}|\mathbf{z}) = \frac{1}{2\sigma_\theta} \|\mathbf{x} - \mu_\theta(\mathbf{z})\|_2^2 - \frac{1}{2} \log(2\pi\sigma_\theta^2). \quad (3.19)$$

To compute the second term in (3.14), we do not need to estimate by sampling. As both the prior and the approximated posterior are Gaussian, the KL divergence in (3.14) can be analytically derived as

$$\begin{aligned} d_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) || p(\mathbf{z})) &= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left( \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \right) d\mathbf{z} \\ &= \int q_\phi(\mathbf{z}|\mathbf{x}) \log q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z} - \int q_\phi(\mathbf{z}|\mathbf{x}) \log p(\mathbf{z}) d\mathbf{z} \\ &= \int \mathcal{N}(\mathbf{z}|\mu_\phi, \sigma_\phi) \log \mathcal{N}(\mathbf{z}|\mu_\phi, \sigma_\phi) d\mathbf{z} - \int \mathcal{N}(\mathbf{z}|\mu_\phi, \sigma_\phi) \log \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) d\mathbf{z} \\ &= -\frac{K}{2} \log(2\pi) - \frac{1}{2} \sum_{k=1}^K (1 + \log \sigma_k^2) - \left( -\frac{K}{2} \log(2\pi) - \frac{1}{2} \sum_{k=1}^K (\mu_k^2 + \sigma_k^2) \right) \\ &= -\frac{1}{2} \sum_{k=1}^K (1 + \log \sigma_k^2 - \mu_k^2 - \sigma_k^2), \end{aligned}$$

where  $K$  is the dimension of  $\mathbf{z}$  and  $k$  indicates each component of the encoder moments  $\mu_\phi$  and  $\sigma_\phi$  evaluated at  $i$ -th traffic sample.

All together, the weights and biases  $\{\theta, \phi\}$  of the neural networks can be estimated using the estimator of the marginal likelihood lower bound of the full data set based on mini-batches of data. Assume that during training we are given mini-batches of  $M$  randomly drawn samples of  $\mathcal{D}$ . In SGD,  $\{\theta, \phi\}$  are initialized to random values and updated until convergence based on the gradient estimators computed from the ELBO for each mini-batch. Because the loss function is simply the negation of the objective function ELBO we want to maximize, we end with the following function to minimize

$$\text{Loss}_{\theta, \phi}(\mathbf{x}) = \frac{N}{M} \sum_{i=1}^M \left( -\|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|_2^2 + \frac{1}{2} \sum_{k=1}^K \left( 1 + \log \sigma_\phi^{(i,k)^2} - \mu_\phi^{(i,k)^2} - \sigma_\phi^{(i,k)^2} \right) \right), \quad (3.20)$$

where  $\hat{\mathbf{x}}^{(i)} = \mu_\theta(\mathbf{z}^{(i)})$  with  $\mathbf{z}^{(i)} = \mu_\phi(\mathbf{x}^{(i)}) + \epsilon^{(i)} \odot \sigma_\phi(\mathbf{x}^{(i)})$  and  $\epsilon^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ . Notice that the first term is the reconstruction error and the second term is the regularizer (or penalty term) in an autoencoder sense. Finally, a forward pass of the network is summarized in Figure 3.7.

#### 3.4.4 Model Implementation

There is the trade-off between model capacity and computational cost that we have to decide. Even that the architecture constructed in Section 3.4.3 is not that complex, we showed in the experimentation section of [Boq+20] (Appendix B) that is enough to solve general road ITS use cases with the

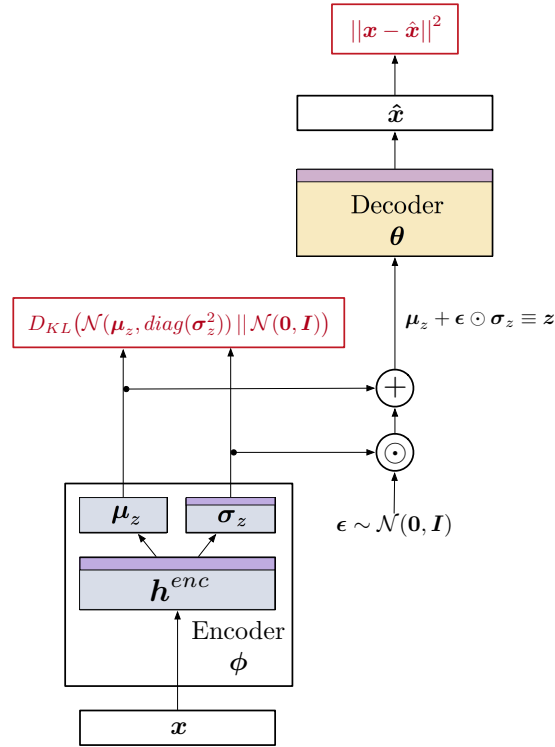


Figure 3.7: Feed forward pass of the network using the *reparametrization trick* for training via SGD. Red and purple colors show the loss and non linear activation layers, respectively.  $\theta$  and  $\phi$  are updated on the backward pass with the back-propagation of the error.

real-world traffic data. This is aligned with recent findings suggesting that Gaussian assumptions do not reduce the effectiveness of VAEs [DW19].

The model should match the complexity of the problem and data because this substantially increases the complexity of model implementation, optimization, training and tuning. Increasing the complexity of the VAE model leads to several issues identified in literature [Zha+18]. For example, Sonderby et al. [Søn+16] proposed the Ladder Variational Autoencoder (LVAE) to train deeper architectures for more representational power, but tend to ignore latent space and use only the decoding distribution to represent the entire data set because of the powerful decoding capabilities [Ale+18]. A common practice in DL is to start easy and move to a more complex model if the learned model does not perform well. Next, we explain the decisions and motivation of Section 3.4.3 proposal, plus we provide some guidance to increase the complexity of the model.

### Parametrization Complexity

We constructed the encoder and decoder networks using a MLP, considering that during this work we will be experimenting with speed and flow data separately. However, thanks to the versatility and continuous development of neural



### 3. Contributions in ITS Data Analytics

---

networks, almost any architecture could be used as part of the encoder-decoder to parametrize  $q_\phi(\mathbf{z}|\mathbf{x})$  and  $p_\theta(\mathbf{x}|\mathbf{z})$ . Even stack several traffic variables as input to the network. For example, Ma et al. [Ma+17] treated  $\mathbf{x}$  as an image using CNN to exploit the spatial and temporal correlation information between road network points to enhance the forecast. To increase the complexity, autoregressive models (e.g., PixelRNN [OKK16] or PixelCNN [Oor+16]) may be integrated with VAE and used as encoder-decoder like in the Variational Lossy Autoencoder [Che+16], or use the Channel-Recurrent Variational Autoencoder [SST18] that uses recurrent connections across CNN channels to circumvent the simplification of VAE’s latent space. These models may be a powerful tool for traffic forecasting as they are good at capturing local statistics [ODM18; Pav17].

#### Lower Bound of the Inference Model Variance

In practice, we computed the variance  $\sigma_z$  filtering the layer activation by a ReLU (or softplus) function, since the co-variance is positive definite. We also added a fudge factor  $1e^{-5}$  to help for numerical stability during training. The factor imposes a lower bound on the variance achievable by  $q_\phi(\mathbf{z}|\mathbf{x})$ , otherwise the density could tend towards infinity if we allow  $q_\phi(\mathbf{z}|\mathbf{x})$  to have arbitrarily small variance. Therefore, we implemented the last row of (3.17) as:

$$\sigma_z = \text{ReLU}(\mathbf{W}_3 \mathbf{h}^{\text{encoder}} + \mathbf{b}_3) + 1e^{-5}.$$

A similar approach motivated by the same fact would be to construct the encoder to output the logarithm of the variance  $\log \sigma_z$ .

#### Lower Bound of the Generative Model Variance

In (3.18) we omitted the actual sampling of  $p_\theta(\mathbf{x}|\mathbf{z})$  during training and fixed the variance  $\sigma_\theta$  to 1. We considered it as a global hyper-parameter. In addition to reducing the parameters that can be learned, there is a mathematical explanation for why to return only the mean  $\mu_\theta(\mathbf{z})$  as the sampled reconstruction. Notice in (3.19) that if the variance was learnable,  $\sigma_\theta(\mathbf{z})$  directly governs the *weighting* of the  $l_2$  reconstruction loss. Intuitively, in that case, the objective will shrink the variance towards zero if there exists  $\{\theta, \phi\}$  such that  $\mu_\theta(\mathbf{z})$  provides a sufficiently good reconstruction of  $\mathbf{x}$ :  $-\log \sigma_\theta^2(\mathbf{z})$  will encourage the variance to go close to zero first before  $1/\sigma_\theta(\mathbf{z})$  catches up.

#### Plain ELBO Optimization

In practice, when training VAE, it’s possible that that the system converges to a local minimum in which the latent variable is completely ignored and the encoder always predicts the prior. This phenomenon is known as posterior collapse [Bow+15]. We found that in most cases a straightforward optimization of the ELBO ignored the latent space, that is,  $q_\phi(\mathbf{z}|\mathbf{x})$  was learned by setting  $q_\phi(\mathbf{z}|\mathbf{x}) \sim p(\mathbf{z})$  thus bringing the KL term close to zero. Notice that a model that encodes useful information in the latent variable  $\mathbf{z}$  will have a non-zero KL divergence term. To prevent that, we modified the training objective (3.14) by weighting the KL term with  $\beta \in [0, 1]$ , starting at zero and increasing its value on each training step during training. This *annealing strategy* yielded to better results, despite not optimizing the proper lower bound during the

early stages of training. Other alternative approaches apply modifications to the ELBO [Che+16].

### Mean Field Variational Inference

The motivation of using a mean field, i.e., the assumption that the variational distribution over the latent variables factorizes as

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \prod_i q_{\phi}(z_i|\mathbf{x}),$$

is that there are fewer variational parameters to learn. Instead of a full co-variance matrix, the model only learns the diagonal values of the co-variance matrix because it is a diagonal square matrix. The parameter learning process is easier with fewer model parameters, which means better optimization. Furthermore, the KL divergence with respect to the prior can be computed in closed-form. The downside of this approach is that it can only model a much smaller subset of distributions, which limit the latent variables  $\mathbf{z}$  to be independent of each other. In reality, random variables in a true posterior may correlate with each other.

### Gaussian Prior

Regarding the definition of  $p(\mathbf{z})$ , we let  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$  for our purpose, which has the following computational and implementation benefits: The samples of  $\mathbf{z}$  can be drawn from a simple distribution. It forces a continuous latent space. The KL divergence is given in closed form. However, by doing so, we assumed that latent representations of samples are i.i.d., which for many data sets, such as time-series of images, can be a strong assumption [Cas+18].

### Model Drift

When time dimension joins the game, adaptation must be considered as an iterative stage of the data pipeline, aimed at maintaining learned models updated and adapted to eventual changes in the data distribution. This adaptation is crucial for real-life ITS scenarios, where changes can happen in all stages, from variations of the input data sources, to interpretation adjustments and other sources of non-stationarity. The whole chapter assumed a stationary data generating process. Since road traffic networks evolve with time, the reconstruction error of new traffic samples can be used as an indicator of when to adjust the model to new data. A high reconstruction error would mean that samples reconstructed conform to a different data distribution than the already learned by the model. This is a mechanism to detect model drift when road networks and mobility patterns change over time.

## 3.5 Applicability in ITS Use Cases

Three different real-world data sets were gathered and cleaned to validate the proposed model, Figure 3.8. It should be noted that there is a lack of benchmark data sets in traffic forecasting literature that has been identified as a problem to compare different proposals [Lañ+18b]. The three data sets are briefly described

### 3. Contributions in ITS Data Analytics

---

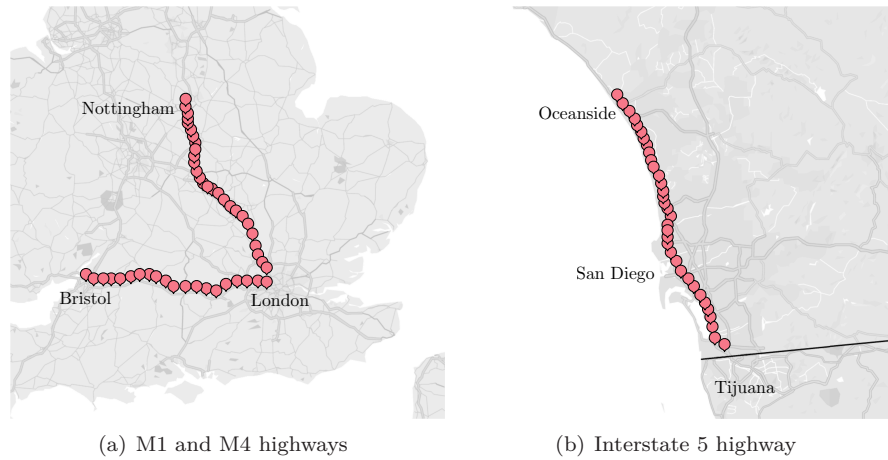


Figure 3.8: Approximate location of the traffic sensors in (a) England and (b) California.

below. The three come from highways, despite that the method can be applied to any road traffic network or urban areas.

**PeMS** (<http://pems.dot.ca.gov>): The data set consists of data from 31 loop detectors installed on a south-bound section of Interstate 5 (I-5), available from the freeway Performance Measurement System (PeMS) of the California Department of Transportation (Caltrans). Data collected covers the two-year period from 2015 until 2017. Detectors used span spaced equally apart 82 km of the highway in San Diego County, concretely from post mile (PM) 1.1 to 52.3. Each detector reports the speed, occupancy and flow, which are aggregated into 5-minute intervals including a reliable measure of data quality showing the percent of observed samples. Incorrect values are filtered out, while missing samples are imputed using linear regression [CKV02].

**UKM1** (<https://data.gov.uk>): Traffic speed and flow data from 19 junctions (J27 to J1) of the English M1 section from Nottingham to London, covering a four-year period from 2011 until 2015. The M1 is a major motorway of the Strategic Road Network (SRN), which runs between London to Leeds in the United Kingdom. The data are averaged between junctions and aggregated into 15-minute intervals. Junctions span 210 km and consist of different road lengths. Speeds are estimated using a combination of sources, including automatic number plate recognition (ANPR) cameras, in-vehicle global positioning systems and inductive loops built into the road surface.

**UKM4** (<https://data.gov.uk>): Traffic speed and flow data from 19 junctions (J22 to J2) of the English M4 section from Bristol to London, from 2011 until 2015. The SRN's M4 motorway connects London to South Wales. Similar to UKM1.

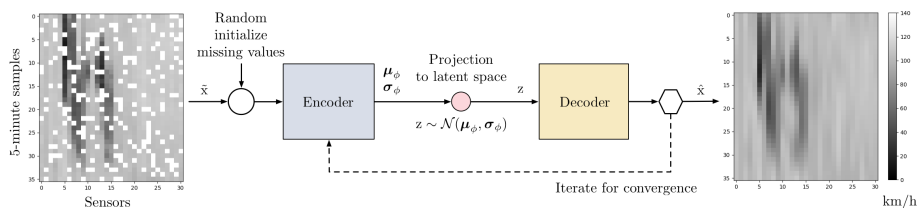


Figure 3.9: The imputation procedure to reconstruct a corrupted sample with missing values.  $y$ -axis shows 3 hours divided into 5-minute samples and  $x$ -axis represents the data from 31 sensors. The colored variable is the traffic speed in km/h from the PeMS data set (darker means congestion).

### 3.5.1 Missing Data Imputation

The first implication of the model defined in Section 3.4 is that new unobserved traffic samples with missing values can be reconstructed from the learned  $p_{\theta}(\mathbf{x}|\mathbf{z})$ . A corrupted data sample  $\tilde{\mathbf{x}}$  can be reconstructed once the whole network is trained on historical data minimizing (3.20). The imputation procedure depicted in Figure 3.9 consists of:

1. Random initialize the missing values.
2. Sample from the inference model, i.e., encode  $\tilde{\mathbf{x}}$  sampling from  $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{\phi}, \boldsymbol{\sigma}_{\phi})$  where  $\boldsymbol{\mu}_{\phi}$  and  $\boldsymbol{\sigma}_{\phi}$  are given by the encoder (3.17)
3. Sample from the generative model, i.e., map back the resulting  $\mathbf{z}$  to the data space using decoder (3.18) to obtain a reconstructed data sample  $\hat{\mathbf{x}}$ .

This procedure can be iterated until convergence, simulating a Markov chain that can be shown that converges to the true marginal distribution of missing values given observed values [RMW14]. In practice, a more straightforward method is to sample only using the mean, i.e.,  $\mathbf{z} = \boldsymbol{\mu}_{\phi}$ , which leads to similar results. Recall that the KL term of the objective forces the model to be able to decode plausible traffic samples from every point in the latent space that has a reasonable probability under the prior. On the contrary, an autoencoder without the latent variable model would have learned a latent space which may not be continuous or allow interpolation.

### Evaluation Details

The model was validated as an imputation method using a defined set of synthetically generated missing data, while determining to what extent an improvement on the imputed values yields an enhanced accuracy of the subsequent traffic forecast model. The final performance of the whole ITS traffic forecasting system was evaluated instead of measuring the distance between the real data and its reconstruction, as imputation requirements may vary depending on the final application [Lañ+18a]. There are cases in which improving the data imputation does not necessarily mean that forecast will improve, e.g., when there is sufficient information in the observed data for the traffic forecasting system to estimate; the reader may think on the increase in root mean squared error (RMSE) when the reconstruction is the same as

### 3. Contributions in ITS Data Analytics

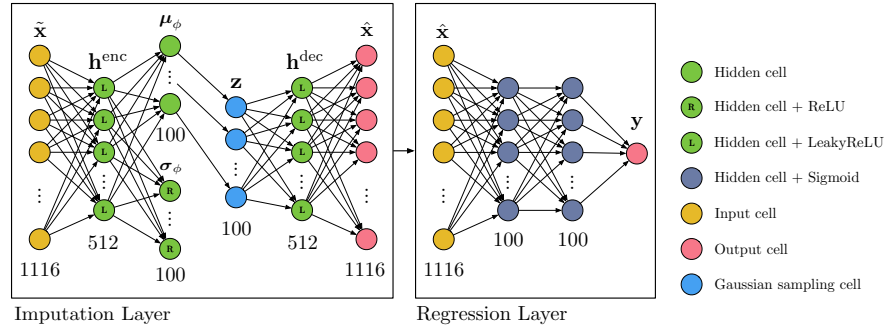


Figure 3.10: The production ITS traffic forecasting system considered for evaluation. First, the imputation layer imputes the missing values of the 1116 dimension input, 36 time samples per 31 sensors. The VAE approach follows the imputation procedure using  $z = \mu_z$ . Then, an independent regression layer estimates the future traffic speed of one sensor using the reconstructed sample. Three different imputation layers based on VAE, AE and PCA were evaluated computing the RMSE and MAPE of the regression layer prediction task.

the original but shifted by one value. Two types of missing value patterns were generated, which are consistent with real-world the types analyzed in literature [GW17; LR14].

**Not Missing at Random (NMAR):** The data set was obtained from PeMS test data, considering all 5-minute samples that did not meet a 75% quality measure as missing values. The PeMS quality measure shows the percentage of valid samples within the aggregated 5-minute samples. This data set shows a pattern where consecutive missing values are found in not so random time instants and sensors, with a 11.28% of missing values.

**Missing Completely at Random (MCAR):** To investigate the robustness of the system against higher shares of missing values, additional observations from the data were removed. A 10%, 20% and 40% missing data proportion on the PeMS test data were generated for evaluation.

Figure 3.10 shows the scenario considered divided into two parts: An imputation layer (referred as the IL layer) that preprocesses corrupt speed traffic samples that are then fed separately to a regression layer (referred as the RL layer) to estimate future traffic speed. The RL was set to estimate 1 hour ahead traffic speed of sensor number 15, the one presenting less corrupted data (0.07%). The last 3 hours of speed samples were used as input. Evaluation was done on all possible 3-hour speed traffic samples of PeMS-NMAR and PeMS-MCAR from 2016 (105360 samples each), while the rest were used for training (105072 samples each).

### Main Results & Discussion

The proposal (VAE) was compared against a non-linear autoencoder (AE) and PCA. Implementation and training details are found in [Boq+20]

### 3.5. Applicability in ITS Use Cases

Table 3.1:  $\overline{\text{RMSE}}$  [km/h] average results on the estimation of one hour ahead traffic speed of sensor number 15 of PeMS test data. The first row shows the performance of the RL alone and should be compared to the results when an IL is added, denoted as IL + RL. The compression factor, computed as the ratio between the dimensions of the data space and the latent space, value is shown between parenthesis near each imputation method. MCAR-(%) indicates the proportion of generated missing data. In bold are the results closer to the performance of the RL on the *Original* data containing no missing values (the closer the better).  $\overline{\text{MAPE}}$  [%] results showed the same behavior, thus they are omitted here. The gray shaded rows highlight the performance of the proposal.

	Original	NMAR	MCAR-10	MCAR-20	MCAR-40
RL	5.53	19.37	27.24	30.07	33.28
PCA (11.16) + RL	N/A	12.42	10.68	14.35	18.46
AE (11.16) + RL	N/A	9.74	10.69	14.02	18.16
VAE (11.16) + RL	N/A	<b>5.89</b>	8.98	11.79	15.01
VAE (22.32) + RL	N/A	8.70	8.58	10.61	11.98
VAE (111.6) + RL	N/A	7.71	<b>7.86</b>	<b>8.57</b>	<b>9.18</b>

(Appendix B). Performance metrics are reported in Table 3.1. The proposed VAE implementation showed an RMSE improvement of 69.6%, 52.6% and 39.5% over RL, PCA and AE on NMAR test data, respectively. Likewise, VAE showed superior performance for each different missing value proportion on MCAR. For example, on MCAR-40, VAE showed an RMSE improvement of 54.9%, 18.7% and 17.3% over RL, PCA and AE, respectively. The main difference between VAE and AE is that a regularizing term on the objective function is imposed on the former to force the model to learn a continuous latent space. Results indicate that learning the data distribution helps to infer missing data as the model is able to decode plausible unseen data samples from every point in the latent space that has a reasonable probability under the prior, which validates our initial assumption. We also found that non-linearity helps to impute missing values when larger gaps of missing data are found (NMAR pattern). Looking at the VAE and AE performance against PCA in Table 3.1 on NMAR data, the linear model performs poorly. However, no relevant differences were found between PCA and AE on MCAR. In this case, the PCA performs similarly to AE because of the MCAR pattern, which implies less consecutive missing values and thus the linear model is able perform better. Another interesting finding is that VAE performed better in NMAR than MCAR-10 even when the missing data proportion of the former is greater, which makes the proposed method more suitable for real-world data set where mostly NMAR patterns are found. The latent space dimension was varied and Table 3.1 provides some of the results, where the compression factors applied on the data are shown between parenthesis near each IL method. A compression factor of 11.16 means to extract 100 features from the 1116 dimension data. Results showed that accuracy increased jointly with the compression factor but to a certain extent. Constraining the latent space dimensions forces the network to learn better features until the space becomes small enough. Same thing happened while increasing the dimensions, suggesting the existence of a lower and higher bound

where only an insignificant improvement can be observed. Thus, the optimal latent space dimension should be empirically defined as a hyper-parameter or by means of new theoretical approaches like in Chapter 4.

#### 3.5.2 Dimension Reduction

The second implication is that the learned latent space can be exploited in several different ways that are of interest to traffic forecasting systems. The latent space defined by  $\mathbf{z}$  is forced to capture useful information about the data because  $\mathbf{z}$  is limited to having fewer dimensions than  $\mathbf{x}$ . Therefore, VAE is learning the principal components or independent factors of the highly non-linear latent manifold of the given traffic data set. Recall that a linear autoencoder minimizing the mean squared error learns to span the same subspace as PCA. This can be exploited as an unsupervised dimension reduction or feature extraction independent layer for traffic forecasting systems. On the one hand, the data can be compressed using the encoder to store and reconstruct them when necessary using the decoder. To aim for the lowest possible dimension of  $\mathbf{z}$  (i.e., maximum compression) that does not degrade the performance, one may find it empirically using trial-and-error methods or rely on the algorithm presented in Chapter 4, which inspects the mutual information evolution between layers. On the other hand, features learned may be used by a regression layer to improve traffic estimation as the compressed information filters out useless information and allows data-driven models to easily learn. In this case, the performance is less conditioned to the dimensions of  $\mathbf{z}$  since in practice we have obtained similar results for different latent space dimensions, except with very small or very large dimensions. The whole procedure consists of the following steps:

- Pre-train the model to reconstruct its input in an unsupervised manner.
- Use the pre-trained encoder as input to a regression model for supervised traffic estimation.
- Fine-tune the entire network if the regression model is a DNN, if not, supervise train with the latent representations.

Fine-tuning yields slightly better results than fixing the weights and biases of the encoder, but modifying the encoder derives in a useless decoder.

#### Evaluation Details

The model was validated as a data compression tool to explore if the learned subspace results in representative and powerful features of the traffic data that enhance traffic forecast. A more complex problem was set which aimed to estimate 1 hour ahead speed of all the network sensors, using the last 12 hours of data of PeMS and the last 18 hours of UKM1 and UKM4. A feature extraction layer and a regression layer were considered but, in this case, models were evaluated on PeMS, UKM1 and UKM4 test data. The latent space dimension was set to 100, thus models were forced to extract 100 features from a 4464 and 1368 input data space depending on the data set. Implementation and training details are found in [Boq+20] (Appendix B).



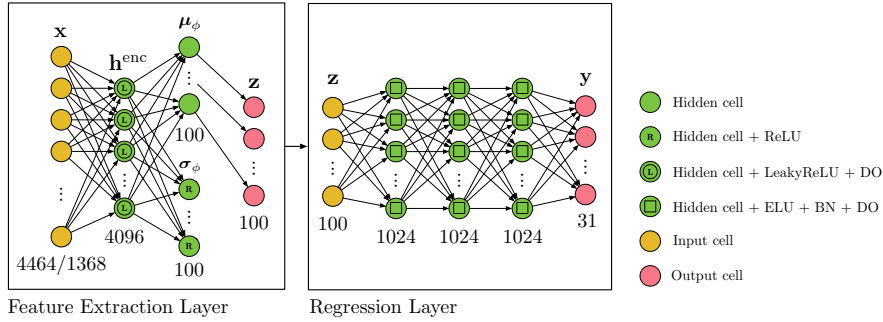


Figure 3.11: The production forecasting system under evaluation for dimension reduction. The encoder of an independently pre-trained VAE is used as a feature extraction layer prior to long-term wide-network traffic forecast. For comparison, the plain AE and PCA approaches were also evaluated as feature extraction layers. Note that the 4464 or 1368 input dimension is reduced down to 100 features to perform the forecast. DO stands for a Dropout layer with a drop probability of 0.5. BN stands for a Batch Normalization layer.

## Main Results

The proposal (VAE) was compared against a non-linear autoencoder (AE) and PCA. Main results are reported in Table 3.2. The first two rows show that the tuned RL improved accuracy for all data sets compared to a naive approach, where the last input sample is used as the estimation. The RL performed the forecast from 4464 samples input for PeMS and 1368 for the other data sets, which equals to 12 and 18 hours of data respectively. The rest of the rows show the accuracy of the models under evaluation. These models first project the data to a 100-dimensional subspace, which is then used as input to train another RL, always maintaining the same architecture. The data compression factor was 44,64 on PeMS and 13,68 on UKM1 and UKM4. In Table 3.2, VAE outperforms all the compared models. It even exceeds the performance of the original RL for all the data sets despite having significantly reduced the space dimension of the input. Although the improvement is slight, below 5% on the RMSE. The introduction of non-linearities and the latent variable model of VAE is well suited to extract useful features to perform traffic forecasting while at the same time for cloud computing and storage as significant compression factors are achieved.

## Discussion

The proposal is intended to be a tool independent from the model used in the prediction part. However, care must be taken in choosing the forecasting model because compressed data can degrade the performance of models that exploit the spatial or temporal structure of the data. As  $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$ , the latent components were set to be orthogonal. Using a CNN or LSTM approach to forecast using the compressed data will not lead to improvement over MLP, since the latent representations of the data do not keep the temporal or spatial



### 3. Contributions in ITS Data Analytics

---

Table 3.2:  $\overline{\text{RMSE}}$  [km/h] average results of the forecast task, showing the dimension reduction and feature extraction impact on the accuracy of the test data of PeMS (4464 space dimension), UKM1 and UKM4 (1368 space dimension) traffic data. The gray shaded row highlights the performance of the proposal.

Model	RL input type	PeMS	UKM1	UKM4
Naive	$\mathbb{R}^{4464; 1368}$	10.83	13.06	15.24
RL	$\mathbb{R}^{4464; 1368}$	7.51	11.11	10.73
PCA + RL	$\mathbb{R}^{100}$	7.61	11.38	10.56
AE + RL	$\mathbb{R}^{100}$	7.63	11.28	10.37
VAE + RL	$\mathbb{R}^{100}$	<b>7.49</b>	<b>10.89</b>	<b>10.23</b>

structure. Some testing was done using LSTM and CNN prediction models that confirmed a degradation on the accuracy performance.

During training, increasing the number of hidden layers derived in VAE ignoring most of the latent space. Instead, using one layer with a higher number of neurons led to learning better features for the traffic forecast task. In that sense, the KL cost annealing and the dropout layer also proved to be useful. The former helped to avoid the posterior collapse problem and the later to prevent overfitting of the model. In this section, the same time was spend tuning each FL independently of the RL. Results were considered enough to validate the model as a prominent solution for dimension reduction of traffic data. However, there exists room for improvement on optimizing the architecture of the model for this specific data set, which is the scope of Chapter 4.

#### 3.5.3 Model & Data Selection

The latent space can be exploited as a tool for the selection of models and data, since similar data is encoded closer in the latent space. Traffic samples are clustered in an unsupervised manner in the latent space learned by VAE. This can be used to distinguish between work days, weekends, holidays, anomalous days, etc. or to compare the traffic from different road traffic networks and time periods. This explanatory power makes the model adaptable and responsive to dynamic traffic and road environment changes over time. Traffic modelers may use the tool as an indicator of model performance against new data, thus, the need to train a new model, or to gain deeper knowledge of the traffic behavior by exploring the latent space. In that way, accuracy of traffic forecasting systems can be enhanced by splitting the data into the classes learned by the model and fitting a separate model to each class [VDW96]. This can be done by projecting the new data into the learned subspace and comparing it with new data using clustering algorithms [DNL15]. Further, modelers can visually search for correlations and seasonality by using visualization techniques of high-dimension data sets such as PCA or t-SNE [MH08].

#### Data Visualization

To experiment with the representational power of the VAE model and its learned latent space, the same model of Section 3.5.2 was trained, but only using unique

day samples of traffic flow and speed for all of the three data sets. Then data was projected to the learned subspace and analyzed it from the point of view of traffic modelers with the goal to improve the prediction accuracy of a traffic forecasting system. Note that the model can be used as an unsupervised tool to learn insights about traffic data without previous knowledge of the road traffic network.

Figure 3.12 shows the two principal components (PC) of the latent space. The pattern of flow and speed differs between weekends and weekdays, even a separate cluster for Fridays can be clearly distinguished from the flow. The flow is classified similarly for the three data sets, instead, the model classifies speed differently for PeMS rather than for the rest of data sets. Only in UKM1, the model can cluster between speed samples from Saturday and Sunday as speed has more complex behavior than flow. In PeMS, the weekend cluster is more separated from the weekdays cluster suggesting a greater difference between both and the possibility that two specific models for each cluster perform better than a global one. In UKM4, the model also clearly identified two clusters which are distinguished by different instants of time, Figure 3.13. A similar trend can be slightly appreciated for UKM1. The data from 2012 and 2013 are classified in the upper cluster, while the data for the years 2011 and 2014 are classified in the lower one. Those differences at the time of fitting the forecast model can influence its performance since the 2012 data may not be beneficial for predicting 2014 traffic, as pointed out by the model.

#### Evaluation of Model Selection

Two new data sets types were created from the main PeMS, UKM1 and UKM4 data sets. The first type consists of just weekdays, referred as WD. The second type consists of just weekends, referred as WE. The main data set containing the whole week is referred in this section as WW. A MLP speed forecasting model was fitted to each one of the data sets. The forecast model goal was to predict 1 hour ahead of all network sensors using the last 3 hours of data. This resulted in three different models: MLP-WW, MLP-WD and MLP-WE, respectively. The goal was to see if the overall performance increased using the two separated models and data compared to using a single model trained on all data.

#### Results

The RMSE performance of the MLP-WW model is used as a benchmark. The RMSE improvement in % with respect to the benchmark of the rest of the models MLP-WD and MLP-WE is shown in Table 3.3. From these results, it can be concluded that predicting the speed by using two separate models for weekdays and weekends in UKM1 and UKM4 shows little improvement over the results of the models for the whole week. On the other hand, in the case of PeMS, training a separated model only on weekend data improves the RMSE by 17.7% on weekend test data. However, no improvement on WD data was obtained by the MLP-WD, meaning that the performance resembles to the MLP-WW model. The latter model, which was trained on WW data, mainly learns how traffic behaves on weekdays because weekend samples are imbalanced w.r.t weekday samples. Those results are related to the cluster

### 3. Contributions in ITS Data Analytics

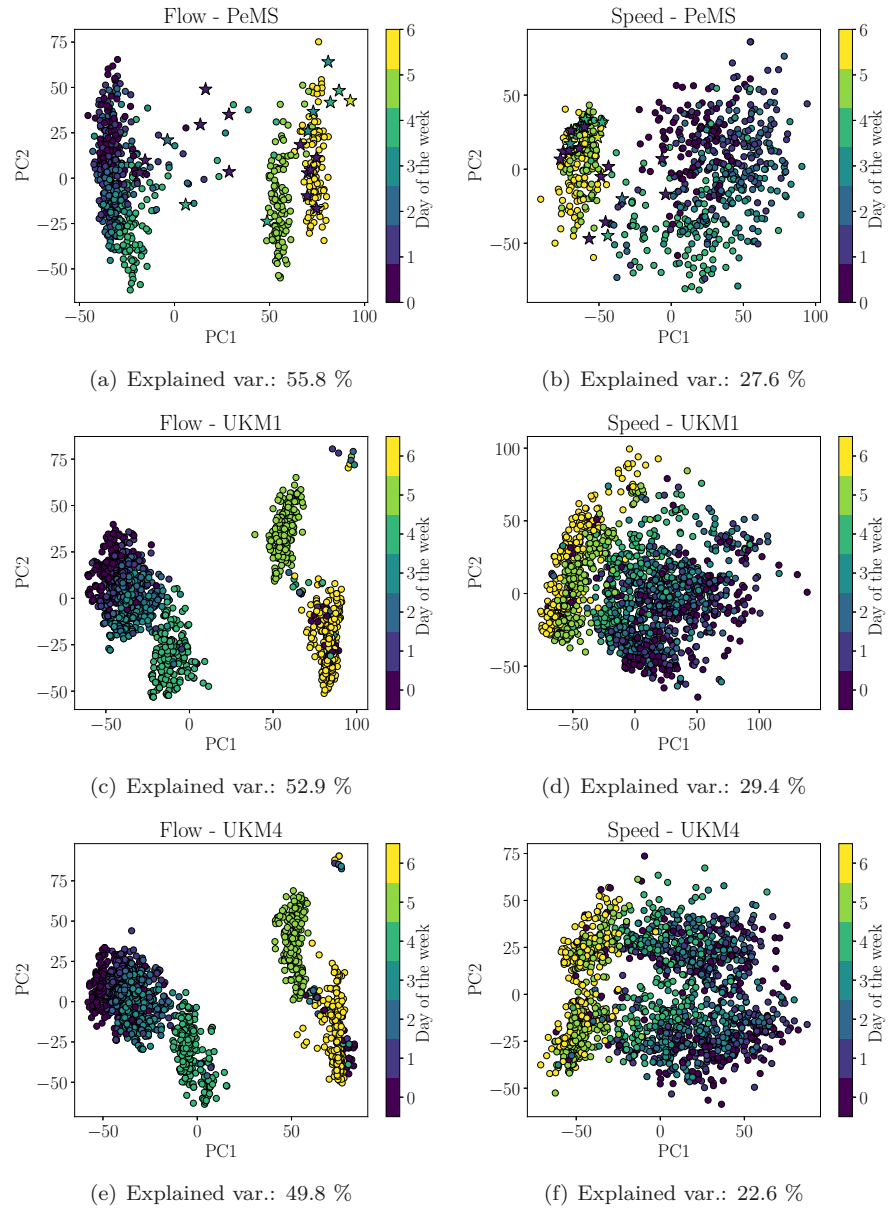


Figure 3.12: Traffic flow and speed samples projected to a latent space and colored by day of the week (0-6: Monday to Sunday). Day samples were projected to a 100-dimension latent space learned by the VAE model in an unsupervised manner. The two principal components (PC) of the projected data were plotted with the help of PCA. The cumulative explained variance of PC1 and PC2 is shown below each figure, which means that the other dimensions that are not seen still capture more traffic characteristics. In PeMS, holidays samples are plotted with a star marker.

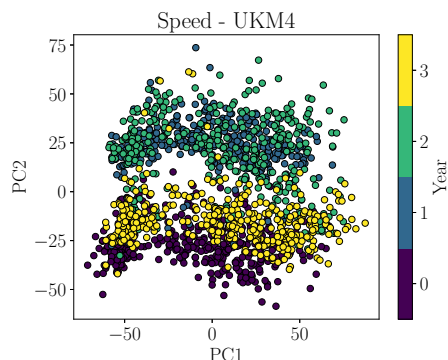


Figure 3.13: 2D PCA visualization of one-day UKM4 speed samples projected to the learned latent space ( $\mathbb{R}^{100}$ ) and colored by year (0-3: 2011 to 2014). The clustering shows that traffic behavior changed between years, cluster  $\{0,3\}$  and cluster  $\{1,2\}$ .

Table 3.3: RMSE improvement [%] of MLP-WD and MLP-WE models w.r.t. MLP-WW for the three different split types of the test data. The text in bold indicate the type of test data split on which the model was intended to improve the performance.

Data set	Split type	MLP-WD	MLP-WE
PeMS	WW	-7.0	-45.0
	WD	<b>0.1</b>	-52.0
	WE	-26.0	<b>17.7</b>
UKM1	WW	-1.6	-17.6
	WD	<b>1.0</b>	-21.3
	WE	-7.1	<b>2.2</b>
UKM4	WW	0.5	-20.9
	WD	<b>3.8</b>	-26.0
	WE	-6.9	<b>3.0</b>

separation that exhibit the two classes in the latent space, which can be seen in the two-dimensional visualization of Figure 3.12. More precisely, the euclidean distances in the latent space ( $\mathbb{R}^{100}$ ) between weekday and weekend cluster centroids of PeMS, UKM1 and UKM4 are 87.3, 65.6 and 62.7, respectively. PeMS' clusters are the ones that the VAE model projected more separated apart, that is, that were considered more dissimilar. This validates the latent space as an indicator of the performance of separated models for different classes of data. Therefore, the VAE model can be used by traffic modelers as a tool to decide when it is best to make use of different models instead of one unique model to predict traffic.

#### 3.5.4 Anomaly Detection

The anomaly detection with VAE can be done online and offline. A simple but powerful approach is to visually compare projected samples in latent space, which may be useful for traffic modelers. For example, by projecting the samples in the latent space and displaying them colored by type of day, the modeler can see if a Tuesday sample deviates significantly from his cluster, which may mean that an anomaly is occurring or that it is a holiday if it's closer to Sunday's cluster. Figure 3.12 is a clear example of that.

On the other hand, a more interesting scenario for ITS is to detect anomalies automatically. For statistical methods, key statistics are used when anomalies are detected if the statistic exceeds a certain threshold value. If anomalies are labeled, one may project a sample to the latent space, compute the Euclidean distance of the sample to its class centroid and then establish the threshold by means of the AUROC, for example. If anomalies are not labeled, one may assume that clusters are normal distributed and set the threshold proportional to the s.d., or even use kernel density estimation setting a minimum probability threshold. Nevertheless, VAE inherently provides the two typically steps of statistical anomaly detection techniques: dimension reduction and a statistical anomaly criterion. Similarly, Dang et al. [DNL15] performed dimensionality reduction by PCA and then they applied kNN outlier detection. VAE provides a probability measure with the KL divergence term in (3.14) rather than a reconstruction error as an anomaly score function. Probabilities are more objective than reconstruction errors and do not require model specific thresholds for judging anomalies [AC15]. When the VAE is trained with far more normal samples than anomalous ones, the VAE learns to model the distribution of normal traffic data, thus a traffic sample can be detected as anomalous if it statistically deviates from what the model has learned [KKH18]. This particularly suits the traffic domain because traffic data sets are usually imbalanced, samples are only labeled by days and most of the anomalies are still unseen.

#### Model Exploratory Power & Discussion

Training a VAE model with unique day samples leads to Figure 3.12-like images that can be used to detect anomalies. Samples corresponding to holidays with a star marker on PeMS are shown in Figure 3.12-a and 3.12-b. Although holidays can be considered non-anomalies, it is more likely that during these the behavior of the traffic will deviate from the usual. First thing that Figure 3.12-a shows is that the majority of the holiday days behave like Sundays, which confirms a common and known fact of most road networks. Figure 3.14 shows the same samples of Figure 3.12-e, but colored proportionally to the Euclidean distance to their respective class centroids. A quick visual comparison shows that all holidays and anomalies are distinguished in darker color without previous knowledge or labeled data, validating the viability of the approach.

In Figure 3.12-a, a few of the samples are projected in the middle between the workday and weekend clusters, thus those samples were inspected more closely as anomalous traffic was not labeled for the data sets under consideration. We visually compared the Monday and Sunday samples closer to their centroid against the holiday sample (Monday) placed between both clusters in Figure 3.12-a. This simplifies the analysis because the three data points compared vary

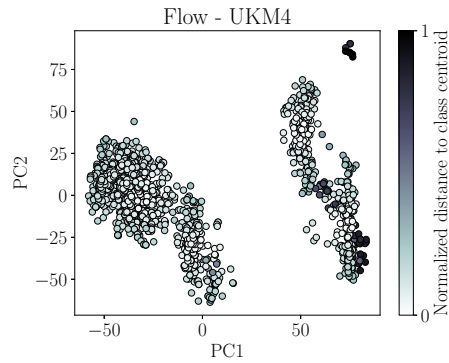


Figure 3.14: 2D PCA visualization of one-day UKM4 flow samples projected to the latent space ( $\mathbb{R}^{100}$ ). Samples are colored proportionally to the distance (normalized to the maximum distance found) to their corresponding cluster centroid, which highlights the anomalous traffic samples (darker). See Figure 3.12-e for comparison.

greatly along PC1 (the x-axis), while the variability in PC2 (y-axis) is much smaller. In this case, PC1 and PC2 represented the 55.8% of the variance of the 100 features learned by VAE. The anomaly is that the targeted sample does not behave like a Monday or Sunday, which should be expected because the sample is labeled as a holiday Monday. To understand what caused the anomaly, the latent space was investigated by not varying PC2 and comparing the three mentioned traffic samples which produced a variation only on PC1. Upon investigation, an increase of traffic flow around sensor 9 for all three samples was found, which means that PC2 is modeling where the traffic intensity is located in the road traffic network. Contrary, the main difference was the intensity of traffic flow and the peak hours. The intensity decreased proportionally from the Monday sample to Sunday while a light peak hour moved from morning to the afternoon, meaning that PC1 component is modeling those traffic features. Therefore, the conclusion is that the anomaly was the intensity of the flow and when happened, not where it was located. There is no way to justify this behavior as the data is not labeled. However, this anomaly may be explained by the effect of non-traffic features (e.g., weather conditions, unusual events, etc.). That said, traffic modelers may consider the holiday sample as an anomaly and plan accordingly to absorb the specific traffic intensity at noon on that holiday. Additionally, since it is a generative model with a continuous latent space plus learned meaningful dimensions, a traffic modeler exploring the rest of the latent space could answer questions like: *What the holiday would have been like on a Wednesday?*



## CHAPTER 4

---

# Theoretical-Tuning Deep Learning Architectures of ITS Solutions

---

This chapter covers Objective III of the thesis. Some concepts of information bottleneck theory in deep learning are applied to derive an algorithm that efficiently finds the sufficient architecture of the autoencoder for accurate traffic forecast with maximal compressed traffic data. Section 4.2 reviews information-theoretic concepts to describe the information plane of an autoencoder. Section 4.3 presents the main results: Section 4.3.2 analyzes why the entropy of the compressed representations can be used as a key performance indicator (KPI), Section 4.3.3 presents the proposed algorithm, Section 4.3.4 selects a mutual information estimator and Section 4.4 experimentally validates the claims.



“Develop an efficient methodology that automatically defines the minimum-expression architecture of ITS solution of Objective 2 that can provide maximum data compression without diminishing the accuracy of the subsequent forecasting system. ”

### 4.1 Related Work

The number of features available from ITS data sources along with the number of available data points in road traffic networks are growing excessively, leading to several critical problems. For example, forecasting with all those features can be computationally inefficient and undertakes the risk of over-fitting. Therefore, we can safely assume that in the era of big data it will be essential for ITS to reduce the dimension of the feature space before applying a prediction model [YQ19].

Reduction of the dimensions is done by learning the principal components or independent factors of a given data manifold, commonly known as feature extraction [Pav19]. Dimension reduction is a subclass of feature extraction methods, as the latter does not necessarily imply reducing the dimension of the data space. Low-dimensional representations of the data are traditionally obtained in the transportation field via PCA approaches or the least absolute shrinkage and selection operator (LASSO), a well-known technique used for feature selection in short-term traffic flow prediction [PS17]. Recently, data-based approaches such as DNN have become increasingly relevant as current technologies facilitate access to dynamic and big data with great computational power. Within this field, features learned by an autoencoder (AE) have been



#### 4. Theoretical-Tuning Deep Learning Architectures

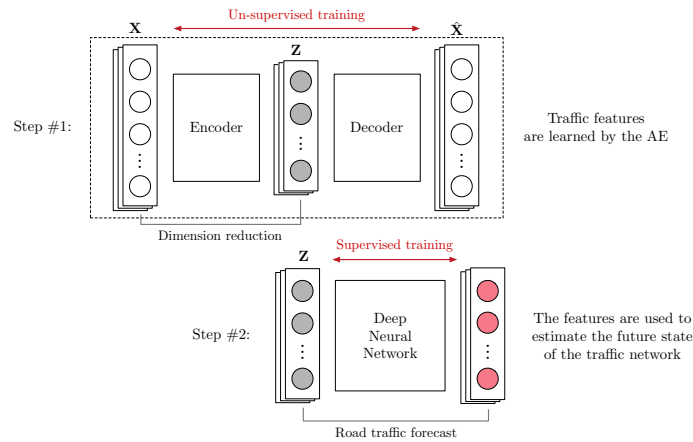


Figure 4.1: AE-based traffic forecast consists of two steps. First, the AE seeks to map the traffic data  $\mathbf{X}$  into a lower-dimensional space (step #1), prior to running the supervised learning algorithm. Second, the compressed data are used to train a regression model to forecast the future state of the road traffic network (step #2).

proved in the literature to improve the traffic forecast Under AE-based traffic forecast, the encoder’s output is used as input for a regression network, which is then trained in a supervised manner, Figure 4.1. Sometimes, the weights and biases of the encoder are not updated during the supervised training, while other times the entire network is fine-tuned. The latter may be useful for slightly increasing the accuracy of the forecast, but will no longer serve as a data compression tool as the updated encoder will not match the decoder. Anyway, the dimension of the bottleneck layer that we will denote as  $K$  is crucial in both cases and affects the performance of the forecasting system:

- A *large* size may lead to redundant dimensions and high computational cost.
- A *small* size might lead to high information loss.

Despite that, all of the authors reviewed in the traffic forecasting domain estimate said number of dimensions arbitrarily or by trial-and-error methods, leaving aside the importance of the data compression feature of the AE for ITS.

Lv et al. [Lv+15] used the AE to process big data and forecast traffic, its architecture was determined via exhaustive search. Yang et al. [YDC16] proposed the SAE-ML, where the optimized architecture was found using a Taguchi method with  $K$  as a design factor. Zhoua et al. [Zho+17] presented the AdaBoost SAE for short-term traffic flow forecasting, showing RMSE results for  $K$  varying from 10 to 100. Yu et al. [Yu+17] merged in parallel the SAE with a LSTM to forecast extreme conditions events, presenting results with the best architecture found by trial and error. Wei et al. [WWM19] sequentially combined the AE with a LSTM for traffic flow prediction, but they arbitrarily set  $K$  without further discussion. Zhang et al. [Zha+19] used the AE for short-term traffic congestion prediction, presenting the results for different

---

## 4.2. Information Perspective of the Autoencoder

values of  $K$ . Although AE is currently very relevant in many fields for being a powerful unsupervised method with many applications, the adequacy of the bottleneck layer dimension has only been addressed in the literature by Gupta et al. [GBR16], hence there is no standard way for automatic selection of the dimension. They verified experimentally that the reconstruction error is not a reliable indicator of the performance of the end application, so they proposed an automatic method to find the critical bottleneck dimension for text language representation. Their proposal was based on the percentage difference between the slopes connecting consecutive bottleneck sizes performances. The metric monitored was the structure preservation index (SPI), a language-related metric that captures the structural distortion incurred by the encoding and posterior decoding process of the data. This method is not directly portable to the transportation field and it is not efficient as does not save a practitioner from having to train all the AE completely to derive the critical dimension, contrary to the algorithm proposed in this chapter.

## 4.2 Information Perspective of the Autoencoder

The information bottleneck (IB) theory of deep learning, initially proposed by Tishby et al. [TZ15], suggests that a learned latent representation in a neural network (NN) should contain all information from the input required for estimating the target, but not more than this required information. The interaction of IB and DNN in the literature can be divided in two main categories: The first is to use the IB theories in order to analyze DNNs. The other is to use the ideas from IB to improve the DNN-based learning algorithms [HK19]. The approach to understand deep neural networks using information-theoretic concepts was further developed by Shwartz-Ziv et al. [ST17] and it has been a topic of ongoing research [Gei20].

### 4.2.1 Autoencoder Role in ITS

Consider an ITS that collected an historical road traffic data set composed of  $N \geq 1$  data points from a concrete road traffic network,

$$\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\} \equiv \{\mathbf{x}^{(i)}\}_{i=1}^N.$$

Let each element in  $\mathbf{x}^{(i)}$  represent a value of a traffic variable such as speed, flow, density, etc. associated to an specific time and space, gathered from a sensor deployed into the road network. The purpose of the AE is to enforce the output  $\hat{\mathbf{X}}$  equal to  $\mathbf{X}$  with high fidelity by minimizing the squared reconstruction error  $\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2$ . Usually, the AE is trained to reconstruct its input through a layer with fewer dimensions than the data space, called the bottleneck layer since it restricts the amount of information that can pass through it. That is, the AE first encodes the input into a hidden representation (or codes) with fewer dimensions,  $\mathbf{Z} = \text{encoder}(\mathbf{X})$ , and then decodes it back into a reconstruction,  $\hat{\mathbf{X}} = \text{decoder}(\mathbf{Z})$ . Along the way, the data is processed by intermediate layers. So, let

$$\{\mathbf{Y}_l^{e;d}\}_{l=1}^L$$

## 4. Theoretical-Tuning Deep Learning Architectures

---

denote the output data or activation of the  $l$ -th layer of  $L$  hidden layers of the encoder or decoder, according to the superscript  $e$  or  $d$ . Hence,  $\mathbf{Y}_L^e = \mathbf{Z}$  and  $\mathbf{Y}_L^d = \hat{\mathbf{X}}$ .

### 4.2.2 Mutual Information

The key quantity in the IB framework is mutual information (MI), derived from the concept of entropy. Consider  $\mathbf{X}$  and  $\mathbf{Y}$  as two continuous variables with a joint probability density function (PDF)  $p(\mathbf{x}, \mathbf{y})$  and marginals  $p(\mathbf{x})$  and  $p(\mathbf{y})$ . The Rényi's entropy of order  $\alpha$  of  $\mathbf{X}$  is defined as

$$H_\alpha(\mathbf{X}) = \frac{1}{1-\alpha} \log \int p^\alpha(\mathbf{x}) d\mathbf{x}, \quad (4.1)$$

where  $\alpha \geq 0$  and  $\alpha \neq 1$ . When  $\alpha \rightarrow 1$ , (4.1) is defined in the limit and yields Shannon's differential entropy  $h := H_{\alpha=1}$  and the Kullback-Leibler divergence, a special case where the chain rule of conditional probability holds exactly. Then, the mutual information  $I(\mathbf{X}; \mathbf{Y})$  is defined as the relative entropy between the joint distribution and the product distribution  $p(\mathbf{x})p(\mathbf{y})$ , which can be expressed as

$$I(\mathbf{X}; \mathbf{Y}) = h(\mathbf{X}) + h(\mathbf{Y}) - h(\mathbf{X}, \mathbf{Y}), \quad (4.2)$$

where  $h(\mathbf{X})$  and  $h(\mathbf{Y})$  are the marginal differential entropies of  $\mathbf{X}$  and  $\mathbf{Y}$  and  $h(\mathbf{X}, \mathbf{Y})$  is their joint differential entropy. The MI in (4.2) measures the dependency between  $\mathbf{X}$  and  $\mathbf{Y}$ , and attains its minimum, equal to zero, if they are independent. Specifically, if  $\mathbf{X}$  is continuous and  $\mathbf{Y}$  is discrete, then  $I(\mathbf{X}; \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}) = h(\mathbf{X}) - h(\mathbf{X}|\mathbf{Y})$ , where  $H$  denotes entropy and all terms can be assumed to be finite. In contrast,  $H(\mathbf{X}) = \infty$  whenever  $\mathbf{X}$  is not discrete and  $h(\mathbf{Y}) = -\infty$  whenever  $\mathbf{Y}$  is not continuous [AG19].

### 4.2.3 Data Processing Inequalities

Another crucial concept of the IB framework is the data processing inequality (DPI). Let  $A$ ,  $B$  and  $C$  form a Markov chain such that  $A \rightarrow B \rightarrow C$ , meaning that  $C$  is conditionally independent of  $A$  given  $B$ . Then, they satisfy

$$I(A; B) \geq I(A; C). \quad (4.3)$$

Essentially, the DPI in (4.3) means that the information that  $B$  contains about  $A$  cannot be increased through any transformation of  $B$ .

Analogously, the same concept can be applied to deep neural networks, which are usually trained via back-propagation and stochastic gradient descent. Both feedforward and backward passes are unidirectional and only depend upon the previous variables, forming a Markov chain. In the special case of the autoencoder, the decoder undoes what the encoder does, so it makes sense to divide the chain into two dual Markov processes:

$$\begin{aligned} \mathbf{X} &\rightarrow \mathbf{Y}_1^e \rightarrow \dots \rightarrow \mathbf{Y}_{L-1}^e \rightarrow \mathbf{Z}, \\ \mathbf{Z} &\rightarrow \mathbf{Y}_1^d \rightarrow \dots \rightarrow \mathbf{Y}_{L-1}^d \rightarrow \hat{\mathbf{X}}. \end{aligned}$$

The rationale behind that duality is that the encoder maps the data to a lower-dimension space and the decoder always maps it back to the data space,

### 4.3. Information-Theoretic Tuning of the Architecture

that is, the decoder undoes what the encoder does. Under this assumption, Schwartz-Ziv et al. [YP19] stated that the following DPIs are satisfied:

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}_1^e) &\geq \dots \geq I(\mathbf{X}; \mathbf{Y}_{L-1}^e) \geq I(\mathbf{X}; \mathbf{Z}), \\ I(\mathbf{Y}_{L-1}^d; \hat{\mathbf{X}}) &\geq \dots \geq I(\mathbf{Y}_1^d; \hat{\mathbf{X}}) \geq I(\mathbf{Z}; \hat{\mathbf{X}}), \\ I(\mathbf{X}; \hat{\mathbf{X}}) &\geq I(\mathbf{Y}_1^e; \mathbf{Y}_{L-1}^d) \geq \dots \geq I(\mathbf{Y}_{L-1}^e; \mathbf{Y}_1^d) \geq I(\mathbf{Z}; \mathbf{Z}), \end{aligned} \quad (4.4)$$

which are known as forward, backward and symmetric DPI, respectively. Interestingly, note that if the forward DPI is extended until the output layer as

$$\begin{aligned} I(\mathbf{X}; \mathbf{Y}_1^e) &\geq \dots \geq I(\mathbf{X}; \mathbf{Y}_{L-1}^e) \geq I(\mathbf{X}; \mathbf{Z}) \geq I(\mathbf{X}; \mathbf{Y}_1^d) \geq \dots \\ &\geq I(\mathbf{X}; \mathbf{Y}_{L-1}^d) \geq I(\mathbf{X}; \hat{\mathbf{X}}) \end{aligned} \quad (4.5)$$

implies that the information is decreased or at most preserved from input to output of the autoencoder.

### 4.3 Information-Theoretic Tuning of the Architecture

Hereafter, it is assumed the existence of a *sufficient* architecture with bottleneck layer size  $D$  such that when  $K < D$  the compressed data will not provide the subsequent regression network of enough information to achieve a reliable forecast (see Figure 4.1). In other words,  $D$  is the dimensionality of the latent space that does not degrade the subsequent system performance while achieving maximum data compression.

#### 4.3.1 Information Plane

The information plane (IP), initially proposed by Tishby et al. [TZ15], depicts how information quantities flow during the training of a DNN with the aim to unveil interesting dynamics. Later, its definition was adapted and extended by Yu et al. [YP19] to the special case of the AE that hold (4.4). The IP of the AE is the space composed of coordinate axes  $I(\mathbf{X}; \mathbf{Y})$  and  $I(\mathbf{Y}; \hat{\mathbf{X}})$  at which the hidden data  $\mathbf{Y}$  in a given training iteration is mapped onto a single point, describing a trajectory during training. For readability,  $I(\mathbf{X}; \mathbf{Y})$  and  $I(\mathbf{Y}; \hat{\mathbf{X}})$  are called input MI and output MI, respectively. In this section, we assume that the MI is not infinite and can be computed or estimated, which is not always true (Section 4.2.2). If the IB theory is not wrong, the information flow during training of each layer of the autoencoder must obey the DPIs in (4.4), so we should expect the same behavior of the information quantities to be reflected in the IP. The specific trajectory followed from initialization to convergence for each layer depends on the optimization process used for training [Gei20]. But, its behavior can be sketched from (4.4), resulting into the three stages described below and Figure 4.2.

**Early stage of training:** A significant information loss through the layers is expected when initializing the AE with random weight values. During the first epochs of training, it can be anticipated a strict inequality in (4.4). Extending the forward DPI until the last layer (4.5), one can clearly see that the input MI will be greater than the output MI and  $I(\mathbf{X}; \hat{\mathbf{X}})$  is

#### 4. Theoretical-Tuning Deep Learning Architectures

---


going to have a small value, possibly equal to zero for the last layers since the input is recursively multiplied by noise at each layer. For the same reason, the initial input MI of the encoder layers  $I(\mathbf{X}; \mathbf{Y}_l^e)$  will be greater than the initial input MI of the decoder layers  $I(\mathbf{X}; \mathbf{Y}_l^d)$ , Figure 4.2.

**Evolution during training:** A direct consequence of the forward DPI in (4.4) is the feasible region of the IP. The trajectories that the input MI and output MI follow are restricted to the region below the bisector  $I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{Y}; \hat{\mathbf{X}})$ , where the optimal solution  $\mathbf{X} = \hat{\mathbf{X}}$  resides when  $\|\mathbf{X} - \hat{\mathbf{X}}\|_2^2$  is minimized. The feasible region is depicted as the shaded area jointly with arbitrary trajectories for each layer in Figure 4.2.

If the layer  $\mathbf{Y}$  is a deterministic function of  $\mathbf{X}$  then all information conveyed by  $\mathbf{Y}$  is shared with  $\hat{\mathbf{X}}$ , thus

$$I(\mathbf{Y}; \hat{\mathbf{X}}) = h(\hat{\mathbf{X}}) - h(\hat{\mathbf{X}}|\mathbf{Y}) = h(\hat{\mathbf{X}}). \quad (4.6)$$

That is, knowing  $\mathbf{Y}$  determines the value of  $\hat{\mathbf{X}}$  and vice versa by symmetry. The following lemma that describes the behavior of the output MI, at any given training iteration, derives from (4.6):

	<p style="margin: 0;">Output MI lemma [TE20]</p> <hr style="width: 100%; border: 0.5px solid black;"/> $I(\mathbf{Y}; \hat{\mathbf{X}}) = I(\mathbf{X}; \hat{\mathbf{X}}). \quad (4.7)$
--	---

Therefore, the output MI is the same for any hidden layer  $\mathbf{Y}$  at any given training iteration. Unfortunately, the specific trajectory of the input MI followed from initialization to convergence depends on the optimization process used for training. But, for example, a common behavior for the first layer of the encoder is that the input MI  $I(\mathbf{X}; \mathbf{Y}_1^e)$  begins to grow much faster than the output MI  $I(\mathbf{Y}_1^e; \hat{\mathbf{X}})$ . This makes sense, since the information in the first layer is maximized at first when the input reconstruction is still poor. At some point the behavior is reversed, the information in  $\mathbf{Y}_1^e$  saturates and the information in  $\hat{\mathbf{X}}$  is maximized by improving the subsequent layers.

**Convergence:** The ideal AE achieves perfect reconstruction at the end of training, thus requiring that all input information is contained at the output layer. The upper bound of  $I(\mathbf{X}; \hat{\mathbf{X}})$  is achieved in the particular case where  $\hat{\mathbf{X}} = \mathbf{X}$ . The feasible region is constrained to the maximum value of  $I(\mathbf{X}; \mathbf{X}) = H(\mathbf{X})$ , following (4.5). This theoretical limit provides an ideal convergence for each IP, the total information available at the input denoted as  $M$  in Figure 4.2.

A well-trained AE of enough capacity approximates the ideal AE at the end of training as much as allowed by its bottleneck layer size  $K$ . Let  $\lambda_K = \lambda(K)$  be an increasing function of the bottleneck layer size  $K$  that represents the maximum amount of information that can be transferred from the encoder to the decoder, that is, through the bottleneck layer. An ideal autoencoder satisfies:

### 4.3. Information-Theoretic Tuning of the Architecture

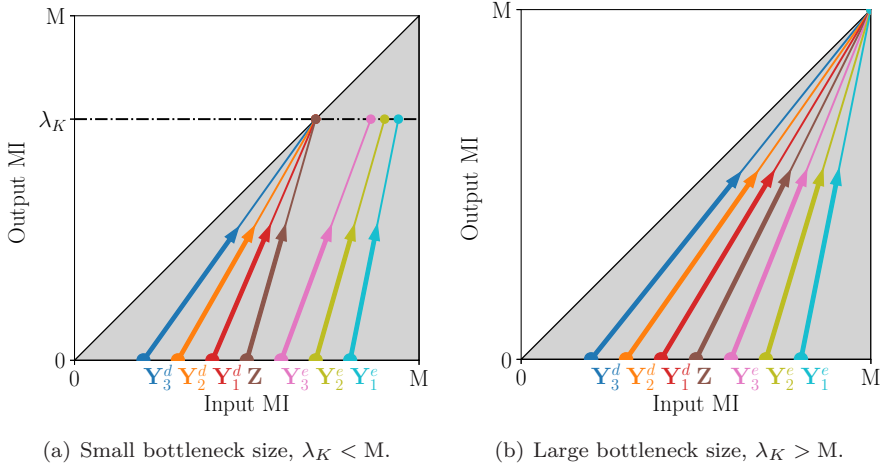


Figure 4.2: Theoretical IP of the autoencoder, when the bottleneck layer size is (a) small and (b) large [TE20]. The information flow during training of each layer is depicted in a different color. The feasible region corresponds to the shaded area, limited to  $M = I(\mathbf{X}; \mathbf{X})$ . Note that the output MIs are always equal. The evolution of the input MIs depend on the optimization algorithm, so they were arbitrarily drawn as straight lines.

$$\begin{array}{|l} \text{Input MI lemma [TE20]} \\ \hline I(\mathbf{X}; \hat{\mathbf{X}}) = I(\mathbf{X}; \mathbf{Z}) = \min(\lambda_K, I(\mathbf{X}; \mathbf{X})) . \end{array} \quad (4.8)$$

The forward DPI implies

$$I(\mathbf{X}; \mathbf{X}) \geq I(\mathbf{X}; \mathbf{Z}) \geq I(\mathbf{X}; \hat{\mathbf{X}}) . \quad (4.9)$$

The ideal autoencoder maximizes  $I(\mathbf{X}; \hat{\mathbf{X}})$  to minimize the reconstruction error. Since the transfer of information is restricted only on the bottleneck layer, the decoder contributes to the maximization of  $I(\mathbf{X}; \hat{\mathbf{X}})$  by achieving  $I(\mathbf{X}; \mathbf{Z}) = I(\mathbf{X}; \hat{\mathbf{X}})$  in (4.9). Additionally, the encoder contributes to the maximization of  $I(\mathbf{X}; \hat{\mathbf{X}})$  by maximizing  $I(\mathbf{X}; \mathbf{Z})$ , which is bounded by  $I(\mathbf{X}; \hat{\mathbf{X}})$  in (4.9). Due to the bottleneck restriction,  $I(\mathbf{X}; \mathbf{Z})$  is also bounded by  $\lambda_K$ , implying that the achievable maximum is  $\min(\lambda_K, I(\mathbf{X}; \mathbf{X}))$ , Figure 4.2.

#### 4.3.2 Entropy of the Subspace as KPI

Interestingly the information plane of the autoencoder shows different behavior when  $K$  is smaller or larger, which was first seen on MNIST data [YP19]. This behavior is of particular interest to ITS, as  $K$  governs how much traffic data is compressed and its quality for the subsequent forecasting system. From the

#### 4. Theoretical-Tuning Deep Learning Architectures

---

lemma (4.7) and (4.8), we can derive what exactly smaller and larger means. Assuming an ideal AE restricted by the bottleneck layer, two different cases can be given:

- C1. If  $\lambda_K > I(\mathbf{X}; \mathbf{X})$ , then the output MI and the input MI are equal to  $I(\mathbf{X}; \mathbf{X})$  for every hidden layer  $\mathbf{Y}$  (Figure 4.2-b). All layers converge together on the bisector because they contain all the input information. No input information is compressed, which relates to perfect reconstruction.
- C2. If  $\lambda_K < I(\mathbf{X}; \mathbf{X})$ , then the output MI is equal to  $\lambda_K$  for every hidden layer  $\mathbf{Y}$  (Figure 4.2-a). The encoder has input MIs satisfying

$$I(\mathbf{X}; \mathbf{X}) \geq I(\mathbf{X}; \mathbf{Y}_1^e) \geq \dots \geq I(\mathbf{X}; \mathbf{Z}) = \lambda_K$$

and the decoder has input MIs satisfying

$$I(\mathbf{X}; \mathbf{Z}) = I(\mathbf{X}; \mathbf{Y}_1^d) = \dots = I(\mathbf{X}; \hat{\mathbf{X}}) = \lambda_K .$$

Some information is compressed through the encoder to achieve the allowed information at the bottleneck. Then, it is transferred through the decoder preserving as much as possible, without further compression, to minimize the reconstruction error.

The role of the AE is to maximize the entropy in the hidden layers in order to pass the input information towards the output layer to achieve perfect reconstruction. When the AE is well-trained in terms of generalization to reconstruct its input through the low-dimension subspace,  $\mathbf{Z}$  must describe useful traffic characteristics that sufficiently describe  $\mathbf{X}$ . Otherwise, the AE would not be able to faithfully reconstruct the input data traffic  $\mathbf{X}$ . Thus, the MSE minimization guarantees useful traffic characteristics in  $\mathbf{Z}$  for traffic forecasting while, at the same time, the maximization of  $I(\mathbf{Z}; \mathbf{Z})$  guarantees that  $\mathbf{Z}$  has the same capacity in terms of information than  $\mathbf{X}$ . In other words, given that the regression network in the second stage of AE-based forecasting (see Figure 4.1) is usually trained minimizing the MSE using  $\mathbf{X}$ ,  $I(\mathbf{X}; \mathbf{X}) > 0$ . The same network will be able to learn and have similar performance if we provide it with  $\mathbf{Z}$ ,  $K < \dim(\mathbf{X})$  and  $\lambda_K \geq I(\mathbf{X}; \mathbf{X})$  to learn from. This leads us to the definition of the sufficient dimensionality  $D$ , which is the bottleneck layer size  $K$  that verifies  $\lambda_K = I(\mathbf{X}; \mathbf{X})$ . As said, if  $\lambda_K > I(\mathbf{X}; \mathbf{X})$ , then the output MI and the input MI are equal to  $I(\mathbf{X}; \mathbf{X})$  for every hidden layer  $\mathbf{Y}$ . Therefore, if a deterministic linear activation function is used at the bottleneck layer, from lemma (4.8) we have that  $I(\mathbf{X}; \hat{\mathbf{X}}) = I(\mathbf{X}; \mathbf{X}) = I(\mathbf{Z}; \mathbf{Z}) = H(\mathbf{Z})$ , and the entropy of the subspace representations  $H(\mathbf{Z})$  acts as a key performance indicator of the subsequent traffic forecasting system. These claims are experimentally validated in Section 4.4, but the theoretical proof is left as future work.

#### 4.3.3 The Sufficient Autoencoder Algorithm

One thing to take into account when designing an autoencoder is that giving too much capacity to its layers can be counterproductive to the learning task. This means that when given too much capacity to work with, they will tend to learn to avoid extracting information and rather to just copy the information, which is an undesirable outcome. On the other hand, trying to set an encoder to

code the input signal into a single dimension could result in the loss of valuable information. Even with a very powerful decoder a very optimized autoencoder will struggle to perform this task, specially when introducing very big sets of data as the input. Taking into account this issue, the general rule in literature to design them is just by using trial and error.

Currently, there is only one algorithm to find the intrinsic dimensionality of the data proposed by Yu et al. [YP19], but requires human observation which is not applicable in practice. To solve that, we propose an efficient and automatic algorithm based only on the evolution of the estimation of  $\hat{I}(\mathbf{Z}; \mathbf{Z})$  or  $\hat{H}(\mathbf{Z})$  from mini-batches of data (proposed in Section 4.3.2). Algorithm 3 proposes to look for  $D$  from an array containing an effective searching range of dimensions  $K$ , where the lower bound is limited to 1 and the upper bound is limited by the dimensionality of the layer before the bottleneck layer. The whole algorithm proposed consists of searching the sorted array by repeatedly dividing the search interval in half (Algorithm 3) and check if  $\hat{I}(\mathbf{Z}; \mathbf{Z}) \geq \hat{I}(\mathbf{X}; \mathbf{X})$  holds at each array division when training the AE with the corresponding  $K$  (Algorithm 4). The output of Algorithm 3 is the sufficient dimensionality  $D$  of the bottleneck layer that verifies  $\lambda_K = I(\mathbf{X}; \mathbf{X})$ . Thus, an autoencoder trained with a bottleneck layer size of  $D$  would encode  $\mathbf{X}$  to the representation  $\mathbf{Z}$  that:

1. Inform about  $\mathbf{X}$ . This means that the representation contains as much information about the data  $\mathbf{X}$ , i.e.,  $\mathbf{Z}$  should be a sufficient statistic for  $\mathbf{X}$ .
2. Be maximally compressed. The representation  $\mathbf{Z}$  does not tell more about  $\mathbf{X}$  than is necessary to correctly perform traffic forecast, i.e., it should attain invariance to nuisance factors which are not relevant to the forecasting system.

### Practical Implementation

The algorithm starts to mini-batch train a new AE with bottleneck layer size given by the dimension  $K$  of the middle position of the searching array. At each training update, the updated AE is used to project the data through each layer and estimate the MI of the input mini-batch and  $\hat{I}(\mathbf{X}; \mathbf{X})$  and its low-dimension representation  $\hat{I}(\mathbf{Z}; \mathbf{Z})$ . For a practical implementation,  $\hat{I}(\mathbf{X}; \mathbf{X})$  can be trimmed to two decimals and, thus, check for the condition  $\hat{I}(\mathbf{Z}; \mathbf{Z}) \simeq \hat{I}(\mathbf{X}; \mathbf{X})$  at the end of the epoch. Then, the interval is narrowed to the lower half if the condition held. Otherwise, it is narrowed to the upper half. Finally, the algorithm repeatedly checks the condition until the interval is empty to find  $D$ . Since noise is present and metrics are computed at mini-batch level, to smooth information quantities and speed up computation it is enough estimated the MI and entropy only every 10 mini-batches and average it accordingly at the end of each epoch. Note that both procedures are omitted in Algorithm 4 to avoid clutter.

Algorithm 3 runs in logarithmic time in the worst case making  $O(\log(\text{len}(\mathbf{sr})))$  runs of Algorithm 4, where  $\mathbf{sr}$  is a vector containing the effective searching range of  $K$ . This is much more efficient than trial and error methods as there is no need to fully train the AE and information quantities converge in few epochs at the early stages of training. In practice, PCA can be applied to the training data to trim the searching range. The upper bound of



#### 4. Theoretical-Tuning Deep Learning Architectures

---

the searching range can be set to the number of dimensions that explain the 90% of the variance, as it is known that non linear AE have higher modeling capabilities than PCA. Furthermore, evenly spaced values within an interval of 2 can be used, as we found no critical differences on the prediction accuracy using consecutive dimensions. To compensate for that, Algorithm 3 must return  $D + 1$ .

---

**Algorithm 3:** Estimation of the sufficient bottleneck layer dimension.

---

```
input : Traffic data set  $\mathbf{X}$ , dimension searching range  $sr$ , batch size  $N$ 
output : Sufficient dimension  $D$ 
 $L \leftarrow 0$ 
 $R \leftarrow \text{getLength}(sr)$ 
while  $L < R$  do
   $D \leftarrow \text{ceil}((L + R)/2)$ 
   $C \leftarrow \text{Algorithm4}(\mathbf{X}, sr[D], N)$ 
  if  $C$  then
     $R \leftarrow D - 1$ 
     $L \leftarrow D + 1$ 
  end
end
```

---

---

**Algorithm 4:** Monitoring  $H(\mathbf{Z})$  during training of the AE.

---

```
input : Data  $\mathbf{X}$ , bottleneck dimension  $K$ , batch size  $N$ 
output : Condition  $C$ 
 $\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{valid}} \leftarrow \mathbf{X}$ 
 $nbatches \leftarrow \text{getLength}(\mathbf{X}_{\text{train}})/N$ 
 $\text{autoencoder} \leftarrow \text{createAutoEncoder}(K)$ 
while  $epoch$  do
   $\mathbf{X}_{\text{train}} \leftarrow \text{randomShuffle}(\mathbf{X}_{\text{train}})$ 
  for  $batch \leftarrow 0$  to  $nbatches$  do
     $\mathbf{X}_b \leftarrow \text{getBatch}(\mathbf{X}_{\text{train}})$ 
     $\hat{\mathbf{X}}_b, \mathbf{Z}_b \leftarrow \text{autoencoder.train}(\mathbf{X}_b)$ 
     $\hat{I}(\mathbf{X}_b; \mathbf{X}_b), \hat{I}(\mathbf{Z}_b; \mathbf{Z}_b) \leftarrow \text{getMetrics}(\mathbf{X}_b, \mathbf{Z}_b, \hat{\mathbf{X}}_b)$ 
     $batch \leftarrow batch + 1$ 
  end
   $\bar{I}_X, \bar{I}_Z \leftarrow \text{getAverage}(\hat{I}(\mathbf{X}_b; \mathbf{X}_b), \hat{I}(\mathbf{Z}_b; \mathbf{Z}_b), nbatches)$ 
  if  $\bar{I}_Z \geq \bar{I}_X$  then
     $C \leftarrow 1$ 
  end
  if  $\text{earlyStopping}(\bar{I}_X, \bar{I}_Z, \mathbf{X}_{\text{train}}, \mathbf{X}_{\text{valid}})$  then
     $C \leftarrow 0$ 
  end
   $epoch \leftarrow epoch + 1$ 
end
```

---

#### 4.3.4 Mutual Information Estimation

The estimation of mutual information or entropy in neural networks is not trivial. Since the algorithm is based on  $I(\mathbf{X}; \mathbf{Y})$  and  $I(\mathbf{Y}; \hat{\mathbf{X}})$  variations during training, these quantities need to be estimated from the activations of DNN layers. This is at least theoretically possible if the quantities are finite, thus, one can reasonably assume that  $\hat{I}(\mathbf{X}; \mathbf{Y}) \approx I(\mathbf{X}; \mathbf{Y})$  if the estimator is adequately parameterized. To yield a finite mutual information, some noise in the mapping is required. A common choice is to analyze a new variable with additive noise, which allows the overall information to remain finite. This noise assumption is not present in the actual neural networks either during training or testing, and is made solely for the purpose of calculating the mutual information. Another strategy is to partition the continuous variable into a discrete variable, for instance by binning the values. This allows use of the discrete entropy, which remains finite. There exist several mutual information estimators in literature such as binning estimators [ST17], kernel density estimation (KDE) with the addition of Gaussian noise [KT17], variational and neural network-based estimators [JCE20] with and without noise addition, kernel-based estimators [YP19] and the well-known estimator based on k-nearest neighbor distances [KSG04]. Additionally, note that the analytical evaluation of (4.2) in DNN traffic forecasting is not possible, because (4.1) requires precise PDF estimation of  $\mathbf{X}$  and  $\mathbf{Y}$  in high-dimensional space. Therefore, one is forced to efficiently estimate its value from a limited number of samples, that is, the mini-batches of data used at each training iteration. For that purpose, we use the matrix-based Rényi's mutual information estimator derived by Giraldo et al. [GRP14].

Given the batch  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$ , an i.i.d. sample of  $N$  realizations of  $\mathbf{X}$ . The Gram matrix  $\mathbf{K}$  is obtained from evaluating a real valued positive definite kernel  $\kappa : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$  on all pairs of data points such that  $(\mathbf{K})_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ . Then, a matrix-based analogue to Rényi's  $\alpha$ -entropy (4.1) can be defined for a matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$  that holds  $tr(\mathbf{A}) = 1$  as

$$S_\alpha(\mathbf{A}) = \frac{1}{1-\alpha} \log \left[ \sum_{i=1}^N \lambda_i(\mathbf{A})^\alpha \right], \quad (4.10)$$

where  $\lambda_i(\mathbf{A})$  denotes the  $i^{th}$  eigenvalue of  $\mathbf{A}$  with

$$\mathbf{A}_{ij} = \frac{1}{N} \frac{\mathbf{K}_{ij}}{\sqrt{\mathbf{K}_{ii}\mathbf{K}_{jj}}},$$

the normalized version of  $\mathbf{K}$ . Furthermore, a matrix-based estimation of the joint entropy can be defined as

$$\hat{H}_\alpha(\mathbf{X}) = S_\alpha(\mathbf{A}, \mathbf{B}) = S_\alpha \left( \frac{\mathbf{A} \odot \mathbf{B}}{tr(\mathbf{A} \odot \mathbf{B})} \right), \quad (4.11)$$

where  $\odot$  denotes the Hadamard product and the matrix  $\mathbf{B}$  is obtained analogously to  $\mathbf{A}$ , but given  $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N$  samples of the targeted layer from the same realizations of  $\mathbf{X}$ . Notice that matrices  $\mathbf{A}$  and  $\mathbf{B}$  simplifies the estimation of the joint distribution to pairwise element multiplication, making it suitable for the efficiency of the proposed algorithm. Finally, the matrix-based Rényi's mutual information is defined as

$$\hat{I}_\alpha(\mathbf{A}; \mathbf{B}) = S_\alpha(\mathbf{A}) + S_\alpha(\mathbf{B}) - S_\alpha(\mathbf{A}, \mathbf{B}), \quad (4.12)$$

## 4. Theoretical-Tuning Deep Learning Architectures

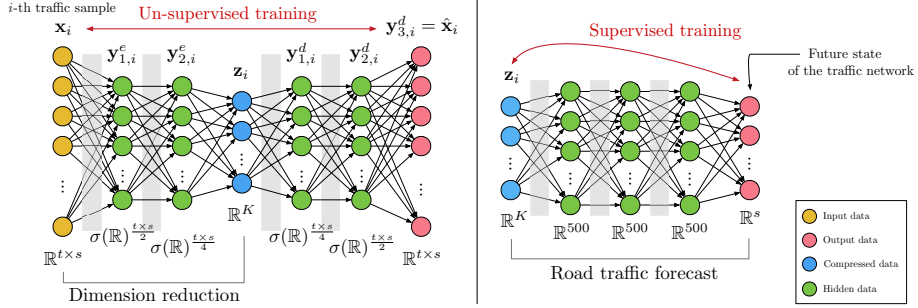


Figure 4.3: AE-based traffic forecast model. Given the traffic sample  $\mathbf{x}_i$ , the AE with a funnel architecture (left) seeks to map it into a lower-dimensional space prior to running the supervised learning algorithm. Once the AE is trained, all traffic samples are compressed by the encoder and used to train a regression network (right) to forecast the future state of the road traffic network. Weights and biases are represented by arrows. Sigmoid activation functions  $\sigma(a) = \frac{1}{1+e^{-a}}$  are represented with gray filled rectangles between layers. Data space dimensions are shown below each layer.

in analogy with Shannon’s definition (4.2).

Finally, since the information estimates depend on the choice of the estimator, the performance of the algorithm would be subject to them and the IPs will only be interpreted when the details of the estimate are taken into account (the IPs obtained by different estimators are not directly comparable).

## 4.4 Experimentation

In this section, real-world traffic data is used to validate the theoretical information plane behavior, the mutual estimation approach and the algorithm proposed. The exact same PeMS and UKM4 data sets described in Section 3.5 of Chapter 3 were used, jointly to the well-known MNIST data set.

### 4.4.1 Evaluation Model

Figure 4.3 shows the traffic forecast model under consideration, which is based on the autoencoder. Several AE with different  $K$  values were trained to reconstruct its normalized input by minimizing the MSE. Once trained, the compressed data  $\mathbf{Z}_K$  given by the encoder part was used as the training data set for the subsequent forecasting network. The forecasting network was set to estimate 1 hour ahead of traffic speed of the whole network ( $s = 31$  for PeMS and  $s = 19$  for UKM4) using the last three hours of data ( $t = 36$  for PeMS and  $t = 18$  for UKM4), Figure 4.3. Note that for the supervised training part:  $\mathbf{z}_i = \mathbf{x}_i \in \mathbb{R}^{t \times s}$  when raw data is used and  $\mathbf{z}_i \in \mathbb{R}^K$  when the compressed data from the AE is used. Two *dropout* layers were added after the first two hidden layers with a drop rate of 0.5 to avoid over-fitting, not shown in Figure 4.3. The reader is referred to [Boq+21b] for more precise details of the training.

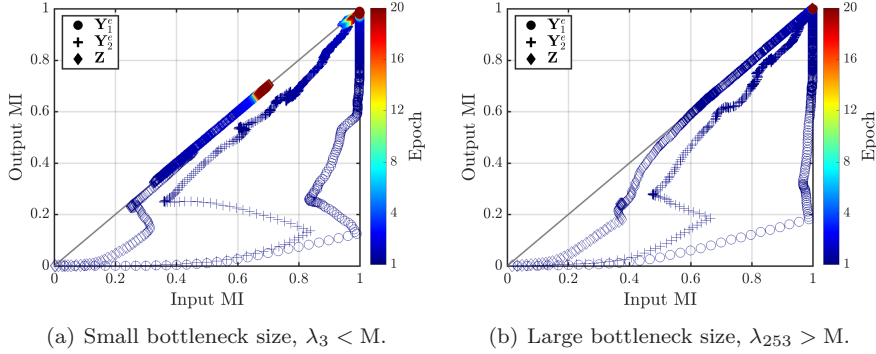


Figure 4.4: Estimated information planes of the encoder. Input and output MI were normalized to  $I(\mathbf{X}; \mathbf{X})$ . The plots show different behavior when the bottleneck layer size is (a) small and (b) large.

#### 4.4.2 Information Plane Validation

In this section, the theoretical DPI and information plane are discussed and compared to the experimental results. Figure 4.4 shows the evolution of the normalized information quantities in the encoder computed at each network training update (the decoder showed similar behavior, not shown to avoid clutter). The AE was trained to reconstruct the traffic samples of PeMS data set. All metrics were computed at each training iteration and were averaged with a moving average of window size 20 to smooth the results and compensate for noise in the measurements. The information quantities converged before epoch 20, while the MSE converged at around epoch 300. This difference is interesting because conclusions can be drawn without having to finish the AE training, making the proposed algorithm more efficient. The training procedure in Figure 4.4 begins by first maximizing the amounts of information in the encoder layers rather than in the decoder. All quantities are maximized as the AE trains on more epochs. The IP shows that all metrics converge to the line  $x = y$ , where the optimal solution resides when MSE is minimized because  $\mathbf{X} = \hat{\mathbf{X}}$ . As expected, the IP shows a different behavior for a small and large value of  $K$ . For a small value of  $K = 3$ , Figure 4.4 shows that the bottleneck layer restricts the amount of information that flows through the network. The input and output MI are limited to approximate  $\lambda_3 = 0.7 I(\mathbf{X}; \mathbf{X})$  at epoch 20 until the end of training, which was stopped by an earlystopping policy monitoring only the MSE metric. Similarly, the MI of symmetric layers increase to converge to  $I(\mathbf{X}; \hat{\mathbf{X}})$  for a large value of  $K = 253$ , Figure 4.5. On the contrary, the entropy of the bottleneck layer data is limited to  $\lambda_3$  for a small value of  $K$ , validating that the AE tries to maximize  $H(\mathbf{Z})$  to achieve a good reconstruction.

Interestingly, some DPIs are violated. The reader may visually compare Figure 4.4 to the theoretical IP sketch of Figure 4.2 to note that the output MI when  $K = 3$  of the hidden layer 1 and 2 of the encoder are greater than the

#### 4. Theoretical-Tuning Deep Learning Architectures

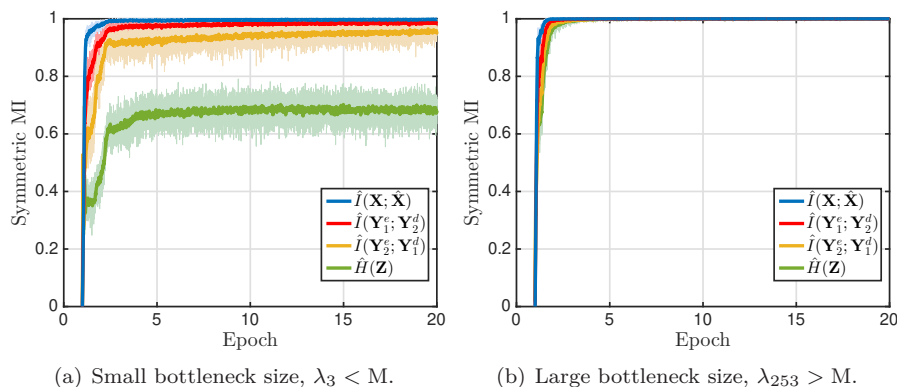


Figure 4.5: Estimated symmetric MI normalized to  $I(\mathbf{X}; \mathbf{X})$ . The plots show different behavior when the bottleneck layer size is (a) small and (b) large.

output MI of the bottleneck layer. Specifically, Figure 4.4-a shows that:

$$I(\mathbf{Y}_1^e; \hat{\mathbf{X}}) > I(\mathbf{Y}_2^e; \hat{\mathbf{X}}) > I(\mathbf{Z}; \hat{\mathbf{X}}) = \lambda_3,$$

against the theoretical analysis that predicted that all layers have the same output MI at every iteration. There are two main reasons this can happen. Either the theory behind the data processing inequalities is wrong, which is not the case, or the information estimate is wrong:

**Number of mini-batch samples:** Like with all estimators, more samples can improve the results, but the memory constraints did not allow to use more than 256 samples at a time. No significant differences were found between using a batch size  $N$  of 100 or 256.

**Rényi's order:** The choice of the order  $\alpha$  in (4.10) is associated with the task goal. If the application requires emphasis on tails of the distribution (rare events) or multiple modalities,  $\alpha$  should be less than 2 and possibly approach to 1 from above. If the goal is to characterize modal behavior,  $\alpha$  should be greater than 2. Finally,  $\alpha = 2$  provides neutral weighting [Yu+19].  $\alpha = 1.01$  would approximate Shannon's entropy, satisfying the data processing inequality. This is not clear for Rényi's entropies of order different from 1, as the concept of mutual information is not unique anymore. The Rényi's divergence of arbitrary order satisfies a data processing inequality, but it is not clear how that carries over to the definition of mutual information [VH14]. Despite that, no differences were found in the results for  $\alpha = 2$ .

**Kernel choice:** We used the already normalized radial basis function (RBF) kernel

$$\kappa_G(\mathbf{x}_i, \mathbf{x}_j; \sigma) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}\right), \quad (4.13)$$

also known as Gaussian kernel. The *bandwidth*  $\sigma$  of the kernel is a free parameter that needs to be defined because it strongly influences the

estimate obtained. The following properties hold for the Gaussian kernel:

$$\begin{aligned}\lim_{\sigma \rightarrow 0} S_\alpha(\mathbf{A}) &= \log(N), \\ \lim_{\sigma \rightarrow 0} \hat{I}_\alpha(\mathbf{A}; \mathbf{B}) &= \log(N), \\ \lim_{\sigma \rightarrow \infty} S_\alpha(\mathbf{A}) &= 0, \\ \lim_{\sigma \rightarrow \infty} \hat{I}_\alpha(\mathbf{A}; \mathbf{B}) &= 0.\end{aligned}$$

They imply that the value of  $\sigma$  controls the operating point of the estimator relative to the bounds because a value too large or too small saturates the estimated information quantities. This is not a critical problem for the proposed algorithm unless  $\hat{I}(\mathbf{Z}_K; \mathbf{Z}_K)$  saturates to  $\log(N)$  for  $K < D$ , thus this saturation has to be avoided to have discriminative estimates.

**Kernel width:** We followed Silverman’s rule of thumb for Gaussian density estimation as  $\hat{\sigma} = h N^{-1/(4+d)}$ , where  $N$  is the sample size,  $d$  is the number of dimensions of the data sample and we defined  $h$  as the mean of the empirical standard deviations of each dimension of the data. Contrary to a constant  $h$ , calculating  $h$  as said adapts the width of the kernel to the different layers in different iterations, since in neural networks the layers change during training. However, recently was noted that higher dimensions decrease the effective kernel width on average, increasing the estimated MI value [Gei20; TE20]. This explains why DPIs are violated only for the layers with higher number of dimensions in Figure 4.4-a. To see this, assume for a moment that  $\mathbf{X}$  has zero mean and unit variance dimension-wise and  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are two i.i.d. samples. Then,  $\mathbb{E}[\|\mathbf{x}_1 - \mathbf{x}_2\|] = 2d$  and, therefore, (4.13) is proportional to the number of dimensions, implying that neural layers with more units will tend to show an overestimated MI.

### 4.4.3 Algorithm Validation

In this section, the entropy of the compressed representations is validated as a KPI of the forecasting network and results are discussed.

#### Evaluation Metrics

The RMSE of the forecasting network trained with raw data,  $\text{RMSE}_{\text{bm}}$ , is used as the benchmark. The  $\text{RMSE}_{\text{increase}}$  is defined as the increment with respect to  $\text{RMSE}_{\text{bm}}$  that the RMSE of the estimation suffers when the network is trained using compressed data, that is,

$$\text{RMSE}_{\text{increase}} = \frac{\text{RMSE} - \text{RMSE}_{\text{bm}}}{\text{RMSE}_{\text{bm}}}.$$

The data compression ratio (DC) is defined as the ratio of reduction that the uncompressed data suffers while using a concrete bottleneck dimension  $K$ ,

$$\text{DC} = \frac{(t \times s) - K}{t \times s}.$$

Both metrics allow to compare more efficiently the obtained results against different data sets with different number of sensors available.

#### 4. Theoretical-Tuning Deep Learning Architectures

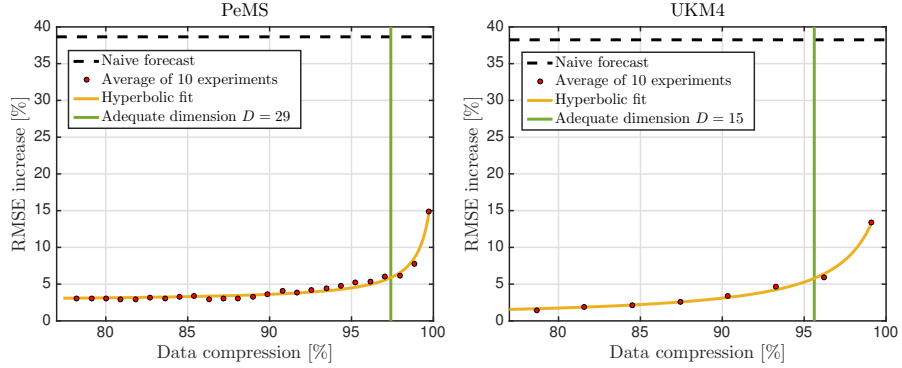


Figure 4.6: RMSE increase of traffic forecast vs. the data compression ratio, i.e., the amount of compressed data obtained using the encoder with different values of the bottleneck layer size  $K$ . The sufficient dimension  $D$  coinciding in the elbow region is efficiently obtained applying the algorithm proposed.

### Results

Several autoencoders were trained varying down the size of the bottleneck layer from  $K = \dim(\mathbf{X})/4$  to  $K = 3$  with step-sizes of 10. Note that the AE architecture considered in Figure 4.3 has a funnel architecture. The whole process took 2 days to compute for each of the data sets. The code was written in Python using the Tensorflow library, which was executed in a Ubuntu server mounting an Intel Xeon W-2123 and three NVIDIA GeForce RTX 2080 Ti. Input dimensions were 1116 and 684 for PeMS and UKM4 data sets, respectively. Figure 4.6 shows the accuracy of the forecast for the aforementioned sizes of bottleneck layer. One can clearly see that when the data compression is increased, that is,  $K$  is decreased, the forecast performance decreases. In Figure 4.6, the naive forecast consisted on setting the last speed values measured as the estimation. The results of  $\text{RMSE}_{\text{bm}}$  for PeMS and UKM4 data were 7.81 km/h and 11.02 km/h, respectively. The adequate dimensions given by our algorithm were  $D = 29$  and  $D = 15$ , estimated in 64 and 11 minutes for PeMS and UKM4, respectively. The little time needed to find  $D$  is clearly an advantage over current trial-and-error methods. Because the first phase of the training is dedicated to bring  $I(\mathbf{X}; \hat{\mathbf{X}})$  closer as possible to  $H(\mathbf{X})$ , one can check the condition in the first stages of the training without waiting for the AE to be fully trained, that is, the MSE to converge, making this criterion more efficient in practice. Overall, results showed that the entropy of  $\mathbf{Z}$  always increased if the bottleneck layer dimension was increased, with the upper bound being the entropy of the input or the MI between the input and output when the AE is well-trained. Furthermore, the performance of the subsequent traffic forecast suffered when the entropy of codes was lower than  $\hat{I}(\mathbf{X}; \hat{\mathbf{X}})$  or  $\hat{H}(\mathbf{X})$  because of the reduction of information. For MNIST,  $D = 19$  was estimated in 6 minutes and similar results were found on the classification accuracy metric. This results validate our assumptions, showing that the entropy of codes is a KPI of the forecasting system. As it becomes evident from Figure 4.6, the RMSE exhibited a clear elbow region pattern. The adequate dimensions coincided in the elbow

region below the point where the RMSE of the forecast changes its behavior. For  $K > D$ , that is, higher data compression ratios, the forecast error showed a quasi-linear behavior with almost constant slope. This explains why AE road traffic forecasting literature choose  $K$  arbitrarily, as long as  $K$  is not very small, achieving good forecasting performance.

For both traffic data sets,  $D$  show a similar  $\text{RMSE}_{\text{increase}}$  around 5% despite compressing the input data more than 95%. The data compression ratios achieved depend on the complexity of the road traffic network and data. The 5% degradation on performance can be explained because either the information estimates were trimmed to two decimals or because information estimates are overestimate proportionally to the layer size  $K$ , as discussed in Section 4.4.2. To compensate for that, Tapia et. al [TE20] recently proposed to normalize  $\mathbf{X}$  dimension-wise using the standard dimension of each dimension and compute the Gaussian kernel width as  $\hat{\sigma} = \gamma \sqrt{d} N^{-1/(4+d)}$ , where  $\gamma > 0$  is an empirically determined constant.





## CHAPTER 5

---

# Main Results of the Dissertation

---

### 5.1 Conclusions

Chapter 1 of this thesis has introduced three high-level objectives to systematically identify needs, develop and analyze solutions to improve the performance of ITS. The material exposed throughout chapters 2 to 4 covers the objectives stated in Chapter 1 and allows to safely state that the objectives of the thesis have been achieved. The work presented is supported by two journal publications annexed to the thesis and 4 conference papers, whose relevance is analyzed in Chapter 1. More specifically, the contributions of this thesis against each of the objectives can be summarized as follows.

#### Contributions of Objective I:



“Develop an IEEE 802.11p V2I beaconing protocol to support RSU mission-critical applications that require low position error with high reliability in road intersections.”

Chapter 2 has introduced an adaptive beaconing IEEE 802.11p communication protocol for intersection assistance systems. The work in this chapter is covered by papers [Boq+17] and [Boq+18b] (Appendix A). With respect to the state-of-the-art, the analysis performed in the intersection area found that standardized beacon protocols were not capable of sustaining envisaged safety applications running on IEEE 802.11p roadside units. PHY and MAC parameter adaptation criteria were optimized for ITS applications running in intersection areas to provide low position error with higher reliability than the protocols analyzed. The intersection assistance protocol designed in compliance with the standards improved the data acquisition layer of ITS: The introduced novelties finally allow to achieve vehicle tracking accuracies that safety applications can rely on.

#### Contributions of Objective II:



“Develop a unique model for ITS to extract knowledge from traffic data to enhance traffic forecast, missing value imputation, model and data selection and anomaly detection.”

## 5. Main Results of the Dissertation

---

Chapter 3 has introduced a unified model for big data analytics in ITS. The work in this chapter is covered by papers [Boq+19] and [Boq+20] (Appendix B). The models previously proposed in the literature aimed to solve only one of the future challenges of ITS traffic forecasting. As a novelty, we proposed a generative deep learning model based on the variational autoencoder that can be used in an unsupervised manner to solve multiple challenges. An ITS traffic modeler can implement the model to efficiently compress traffic data and forecast, impute missing values, select the best data and models for a specific problem and detect anomalous traffic data at the same time, without the need for additional knowledge nor labeled data. This provides a way to exploit the data constantly collected by ITS, making it valuable for safety applications and decision making.

### Contributions of Objective III:



“Develop an efficient methodology that automatically defines the minimum-expression architecture of ITS solution of Objective 2 that can provide maximum data compression without diminishing the accuracy of the subsequent forecasting system. ”

Chapter 4 has introduced an efficient algorithm to derive a sufficient architecture for autoencoder solutions of ITS. The work in this chapter is covered by paper [Boq+21b]. The introduced novelties allow practitioners to automatically select the minimum expression architecture that provides maximal compressed representations that inform about the original traffic data. In this way, the performance of the subsequent traffic forecasting system is not adversely affected, but benefits from data being represented with fewer dimensions, which is vitally important in the age of big data. Regarding the state-of-the-art, the basis of the algorithm are taken from theoretical concepts of Information Theory applied to neural networks, allowing to highly improve the current methods that are based on trial and error.

## 5.2 Future Lines of Research

This thesis has covered soundly the objectives listed in Chapter 1. Observing the current trends in ITS, the work presented in each chapter could be immediately expanded as listed below (ordered by complexity).

### Chapter 2:

- Validate the protocol in different scenarios where ITS are deployed.
- Explore the concept of *adapt adaptation*, that is, decide when and how to switch adaptations to comply with different kind of applications or scenarios.
- Further improve the protocol to achieve the requirements of critical safety applications.

**Chapter 3:**

- Validate results in more complex traffic road networks like urban scenarios.
- Extend the solution with an online and robust outlier detection mechanism.
- Explore transfer knowledge from an already trained model towards a new one operating at a different traffic road network.

**Chapter 4:**

- Validate results with other non-funnel autoencoder architectures.
- Explore the algorithm sensitivity to the MI estimator.
- Provide a bandwidth estimator for the RBF kernel that does not over estimate the MI *proportionally* to the dimensionality of the layer.
- Theoretically proof that  $H(\mathbf{Z})$  is a KPI of the subsequent forecasting network. Proof that the same network will have similar performance if it is trained with  $\mathbf{X}$  or with  $\mathbf{Z}_D$ ,  $D < \dim(\mathbf{X})$  and  $I(\mathbf{Z}_D; \mathbf{Z}_D) \geq I(\mathbf{X}; \mathbf{X})$ .



PART II

---

**Journal Publications (Appendix)**

---



## APPENDIX A

---

# Adaptive Beacons for RSU-based Intersection Assistance Systems: Protocols Analysis and Enhancement

---

Due to a signed copyright transfer agreement with the journal, only the bibliographic reference is attached below.

**Reference:** Guillem Boquet, Ivan Pisa, Jose Lopez Vicario, Antoni Morell, Javier Serrano, Adaptive beaconing for RSU-based intersection assistance systems: Protocols analysis and enhancement, Vehicular Communications, Volume 14, 2018, Pages 1-14, ISSN 2214-2096, <https://doi.org/10.1016/j.vehcom.2018.08.003>.

**Abstract:** Current envisaged cooperative vehicular applications require moderate to severe requirements of reliability and latency according to their purpose. Dedicated Short Range Communications (DSRC)-based applications mainly rely on the periodic exchange of information that under certain circumstances may cause congestion problems on the communication channel obtaining unreliable and outdated information at application level. Adaptive beaconing protocols adapt transmission parameters to different criteria such as the channel load and application requirements to improve the overall performance of the vehicle network. Nevertheless, it has not been determined yet if the information disseminated by these protocols is suitable enough for the implementation of specific applications, e.g., Road Side Unit (RSU)-based Intersection Assistance Systems (IAS) like Intersection Collision Risk Warning (ICRW). In this context, we first analyze the network behavior in a realistic simulated intersection area where probability of packet reception becomes difficult to predict and models become highly complex. In that scenario, we present a critical analysis on the performance of current EU and US decentralized congestion control protocols while their performance is evaluated with respect to tracking accuracies required by Intelligent Transportation System (ITS) applications. Results obtained lead us to conclude that adaptation criteria of beaconing protocols is not able to support different safety applications at the same



## A. Adaptive Beaconing for RSU-based Intersection Assistance Systems

time, that is, there is a tradeoff in the selection of such criteria between enhancing applications supporting vehicles or infrastructure. In that sense, we discuss and provide novel adaptation criteria (Intersection Assistance State Machine, IASM) to improve the performance of beaconing protocols towards assisting safety RSU-based IAS. Finally, we propose and validate through simulations a novel beaconing protocol (Intersection Assistance Protocol, IAP) that improves performance over studied protocols.

## APPENDIX B

---

# A Variational Autoencoder Solution for Road Traffic Forecasting Systems: Missing Data Imputation, Dimension Reduction, Model Selection and Anomaly Detection

---

Due to a signed copyright transfer agreement with the journal, only the bibliographic reference is attached below.

**Reference:** Guillem Boquet, Antoni Morell, Javier Serrano, Jose Lopez 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*, Volume 115, 2020, 102622, ISSN 0968-090X, <https://doi.org/10.1016/j.trc.2020.102622>.

**Abstract:** Efforts devoted to mitigate the effects of road traffic congestion have been conducted since 1970s. Nowadays, there is a need for prominent solutions capable of mining information from messy and multidimensional road traffic data sets with few modeling constraints. In that sense, we propose a unique and versatile model to address different major challenges of traffic forecasting in an unsupervised manner. We formulate the road traffic forecasting problem as a latent variable model, assuming that traffic data is not generated randomly but from a latent space with fewer dimensions containing the underlying characteristics of traffic. We solve the problem by proposing a variational autoencoder (VAE) model to learn how traffic data are generated and inferred, while validating it against three different real-world traffic data sets. Under this framework, we propose an online unsupervised imputation method for unobserved traffic data with missing values. Additionally, taking advantage of the low dimension latent space learned, we compress the traffic data before applying a prediction model obtaining improvements in the forecasting

## B. Variational Autoencoder Solution for Road Traffic Forecasting Systems

accuracy. Finally, given that the model not only learns useful forecasting features but also meaningful characteristics, we explore the latent space as a tool for model and data selection and traffic anomaly detection from the point of view of traffic modelers.

---

## Bibliography

---

- [5GA19] 5GAA. *White Paper on CV2-X Use Cases: Methodology, Examples and Service Level Requirements*. Tech. rep. 5G Automotive Association (5GAA), June 2019.
- [AC15] An, J. and Cho, S. “Variational autoencoder based anomaly detection using reconstruction probability”. In: *Special Lecture on IE* vol. 2 (2015), pp. 1–18.
- [AG19] Amjad, R. A. and Geiger, B. C. “Learning representations for neural network-based classification using the information bottleneck principle”. In: *IEEE transactions on pattern analysis and machine intelligence* vol. 42, no. 9 (2019), pp. 2225–2239.
- [Ale+18] Alemi, A. et al. “Fixing a broken ELBO”. In: *International Conference on Machine Learning*. 2018, pp. 159–168.
- [ATM18] Angarita-Zapata, J. S., Triguero, I., and Masegosa, A. D. “A Preliminary Study on Automatic Algorithm Selection for Short-Term Traffic Forecasting”. In: *International Symposium on Intelligent and Distributed Computing*. Springer. 2018, pp. 204–214.
- [Ban+13] Bansal, G. et al. “EMBARC: Error model based adaptive rate control for vehicle-to-vehicle communications”. In: *Proceeding of the tenth ACM international workshop*. ACM. 2013, pp. 41–50.
- [BG10] Buuren, S. v. and Groothuis-Oudshoorn, K. “mice: Multivariate imputation by chained equations in R”. In: *Journal of statistical software* (2010), pp. 1–68.
- [Boq+17] Boquet, G. et al. “Trajectory prediction to avoid channel congestion in V2I communications”. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–6.
- [Boq+18a] Boquet, G. et al. “Analysis of adaptive beaconing protocols for intersection assistance systems”. In: *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2018, pp. 67–74.
- [Boq+18b] Boquet, G. et al. “Adaptive beaconing for RSU-based intersection assistance systems: Protocols analysis and enhancement”. In: *Vehicular Communications* vol. 14 (2018), pp. 1–14.

## Bibliography

---

- [Boq+19] Boquet, G. et al. “Missing Data in Traffic Estimation: A Variational Autoencoder Imputation Method”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 2882–2886.
- [Boq+20] Boquet, G. et al. “A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection”. In: *Transportation Research Part C: Emerging Technologies* vol. 115 (2020), p. 102622.
- [Boq+21a] Boquet, G. et al. “Offline Training for Memristor-based Neural Networks”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 1547–1551.
- [Boq+21b] Boquet, G. et al. “Theoretical Tuning of the Autoencoder Bottleneck Layer Dimension: A Mutual Information-based Algorithm”. In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021, pp. 1512–1516.
- [Bow+15] Bowman, S. R. et al. “Generating sentences from a continuous space”. In: *arXiv preprint arXiv:1511.06349* (2015).
- [Bro+16] Brock, A. et al. “Neural photo editing with introspective adversarial networks”. In: *arXiv preprint arXiv:1609.07093* (2016).
- [Cas+18] Casale, F. P. et al. “Gaussian process prior variational autoencoders”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 10369–10380.
- [Che+16] Chen, X. et al. “Variational lossy autoencoder”. In: *arXiv preprint arXiv:1611.02731* (2016).
- [CHS19] Chen, X., He, Z., and Sun, L. “A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation”. In: *Transportation Research Part C: Emerging Technologies* vol. 98 (2019), pp. 73–84.
- [CKV02] Chen, C., Kwon, J., and Varaiya, P. “The quality of loop data and the health of California’s freeway loop detectors”. In: *PeMS Development Group* (2002).
- [Cor+17] Correa, A. et al. “Autonomous car parking system through a cooperative vehicular positioning network”. In: *Sensors* vol. 17, no. 4 (2017), p. 848.
- [Dai+18] Dai, B. et al. “Connections with robust PCA and the role of emergent sparsity in variational autoencoder models”. In: *The Journal of Machine Learning Research* vol. 19, no. 1 (2018), pp. 1573–1614.
- [Dje+19] Djenouri, Y. et al. “A Survey on Urban Traffic Anomalies Detection Algorithms”. In: *IEEE Access* (2019).
- [DNL15] Dang, T. T., Ngan, H. Y., and Liu, W. “Distance-based k-nearest neighbors outlier detection method in large-scale traffic data”. In: *2015 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE. 2015, pp. 507–510.
- [DW19] Dai, B. and Wipf, D. “Diagnosing and enhancing vae models”. In: *arXiv preprint arXiv:1903.05789* (2019).

- [ETS11] ETSI. *Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part*. TS 102 687 V1.1.1. 2011.
- [ETS14] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. EN 302 637-2 V1.3.2. 2014.
- [ETS16] ETSI. *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services*. TS 103 301 V1.1.1. 2016.
- [ETS18] ETSI. *Intelligent Transport Systems (ITS); V2X Applications; Part 2: Intersection Collision Risk Warning (ICRW) application requirements specification*. TS 101 539-2 V1.1.1. 2018.
- [FRM19] Fortuin, V., Rätsch, G., and Mandt, S. “Multivariate Time Series Imputation with Variational Autoencoders”. In: *arXiv preprint arXiv:1907.04155* (2019).
- [GBR16] Gupta, P., Banchs, R. E., and Rosso, P. “Squeezing bottlenecks: exploring the limits of autoencoder semantic representation capabilities”. In: *Neurocomputing* vol. 175 (2016), pp. 1001–1008.
- [Gei20] Geiger, B. C. “On Information Plane Analyses of Neural Network Classifiers—A Review”. In: *arXiv preprint arXiv:2003.09671* (2020).
- [Gha+13] Ghafoor, K. Z. et al. “Beaconing approaches in vehicular ad hoc networks: a survey”. In: *Wireless personal communications* vol. 73, no. 3 (2013), pp. 885–912.
- [Goo+16] Goodfellow, I. et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.
- [GRP14] Giraldo, L. G. S., Rao, M., and Principe, J. C. “Measures of entropy from data using infinitely divisible kernels”. In: *IEEE Transactions on Information Theory* vol. 61, no. 1 (2014), pp. 535–548.
- [Gua+11] Guan, W. et al. “Adaptive congestion control of DSRC vehicle networks for collaborative road safety applications”. In: *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*. IEEE, 2011, pp. 913–917.
- [GW17] Gondara, L. and Wang, K. “Multiple imputation using deep denoising autoencoders”. In: *arXiv preprint arXiv:1705.02737* (2017).
- [HK19] Hafez-Kolahi, H. and Kasaei, S. “Information bottleneck and its applications in deep learning”. In: *arXiv preprint arXiv:1904.03743* (2019).
- [Hua+10] Huang, C.-L. et al. “Adaptive intervehicle communication control for cooperative safety systems”. In: *IEEE network* vol. 24, no. 1 (2010).
- [JCE20] Jónsson, H., Cherubini, G., and Eleftheriou, E. “Convergence Behavior of DNNs with Mutual-Information-Based Regularization”. In: *Entropy* vol. 22, no. 7 (2020), p. 727.

## Bibliography

---

- [Joe+14] Joerer, S. et al. “Fairness kills safety: A comparative study for intersection assistance applications”. In: *PIMRC IEEE International Symposium*. IEEE. 2014, pp. 1442–1447.
- [Joe+16] Joerer, S. et al. “Enabling situation awareness at intersections for IVC congestion control mechanisms”. In: *IEEE Transactions on Mobile Computing* vol. 15, no. 7 (2016), pp. 1674–1685.
- [Jor+99] Jordan, M. I. et al. “An introduction to variational methods for graphical models”. In: *Machine learning* vol. 37, no. 2 (1999), pp. 183–233.
- [JSK19] Jang, M., Seo, S., and Kang, P. “Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning”. In: *Information Sciences* vol. 490 (2019), pp. 59–73.
- [KBR11] Kenney, J. B., Bansal, G., and Rohrs, C. E. “LIMERIC: a linear message rate control algorithm for vehicular DSRC systems”. In: *Proceedings of the Eighth ACM international workshop on Vehicular inter-networking*. ACM. 2011, pp. 21–30.
- [KKH18] Kawachi, Y., Koizumi, Y., and Harada, N. “Complementary set variational autoencoder for supervised anomaly detection”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 2366–2370.
- [Klo+16] Kloiber, B. et al. “Random transmit power control for DSRC and its application to cooperative safety”. In: *IEEE Transactions on Dependable and Secure Computing* vol. 13, no. 1 (2016), pp. 18–31.
- [KNZ21] Kaffash, S., Nguyen, A. T., and Zhu, J. “Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis”. In: *International Journal of Production Economics* vol. 231 (2021), p. 107868.
- [KSG04] Kraskov, A., Stögbauer, H., and Grassberger, P. “Estimating mutual information”. In: *Physical review E* vol. 69, no. 6 (2004), p. 066138.
- [KT17] Kolchinsky, A. and Tracey, B. D. “Estimating mixture entropy with pairwise distances”. In: *Entropy* vol. 19, no. 7 (2017), p. 361.
- [KW13] Kingma, D. P. and Welling, M. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).
- [Lañ+18a] Laña, I. et al. “On the imputation of missing data for road traffic forecasting: New insights and novel techniques”. In: *Transportation research part C: emerging technologies* vol. 90 (2018), pp. 18–33.
- [Lañ+18b] Laña, I. et al. “Road Traffic Forecasting: Recent Advances and New Challenges”. In: *IEEE Intelligent Transportation Systems Magazine* vol. 10, no. 2 (2018), pp. 93–109.
- [Lañ+21] Laña, I. et al. “From data to actions in intelligent transportation systems: A prescription of functional requirements for model actionability”. In: *Sensors* vol. 21, no. 4 (2021), p. 1121.
- [Li+18] Li, L. et al. “Missing Value Imputation for Traffic-Related Time Series Data Based on a Multi-View Learning Method”. In: *IEEE Transactions on Intelligent Transportation Systems* (2018).

- [LJY07] Li, Q., Jianming, H., and Yi, Z. “A flow volumes data compression approach for traffic network based on principal component analysis”. In: *2007 IEEE Intelligent Transportation Systems Conference*. IEEE. 2007, pp. 125–130.
- [LLL14] Li, Y., Li, Z., and Li, L. “Missing traffic data: comparison of imputation methods”. In: *IET Intelligent Transport Systems* vol. 8, no. 1 (2014), pp. 51–57.
- [LR14] Little, R. J. and Rubin, D. B. *Statistical analysis with missing data*. Vol. 333. John Wiley & Sons, 2014.
- [Lv+15] Lv, Y. et al. “Traffic flow prediction with big data: A deep learning approach.” In: *IEEE Trans. Intelligent Transportation Systems* vol. 16, no. 2 (2015), pp. 865–873.
- [LWZ18] Liu, Q., Wang, B., and Zhu, Y. “Short-Term Traffic Speed Forecasting Based on Attention Convolutional Neural Network for Arterials”. In: *Computer-Aided Civil and Infrastructure Engineering* vol. 33, no. 11 (2018), pp. 999–1016.
- [Ma+17] Ma, X. et al. “Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction”. In: *Sensors* vol. 17, no. 4 (2017), p. 818.
- [Mac+19] Macias, E. et al. “Novel imputing method and deep learning techniques for early prediction of sepsis in intensive care units”. In: *2019 Computing in Cardiology (CinC)*. IEEE. 2019, pp. 1–4.
- [MGS20] Molina-Masegosa, R., Gozalvez, J., and Sepulcre, M. “Comparison of IEEE 802.11 p and LTE-V2X: An evaluation with periodic and aperiodic messages of constant and variable size”. In: *IEEE Access* vol. 8 (2020), pp. 121526–121548.
- [MH08] Maaten, L. v. d. and Hinton, G. “Visualizing data using t-SNE”. In: *Journal of machine learning research* vol. 9, no. Nov (2008), pp. 2579–2605.
- [NJ15] Nguyen, H.-H. and Jeong, H.-Y. “Crosslayer beaconing design toward guaranteed cooperative awareness with contending traffic”. In: *Vehicular Networking Conference (VNC), 2015 IEEE*. IEEE. 2015, pp. 131–134.
- [ODM18] Ostrovski, G., Dabney, W., and Munos, R. “Autoregressive Quantile Networks for Generative Modeling”. In: *arXiv preprint arXiv:1806.05575* (2018).
- [OKK16] Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. “Pixel recurrent neural networks”. In: *arXiv preprint arXiv:1601.06759* (2016).
- [Oor+16] Oord, A. van den et al. “Conditional image generation with pixelcnn decoders”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 4790–4798.
- [Pam18] Pamula, T. “Impact of Data Loss for Prediction of Traffic Flow on an Urban Road Using Neural Networks”. In: *IEEE Transactions on Intelligent Transportation Systems* (2018).



## Bibliography

---

- [Pav17] Pavlyuk, D. “Short-term traffic forecasting using multivariate autoregressive models”. In: *Procedia Engineering* vol. 178 (2017), pp. 57–66.
- [Pav19] Pavlyuk, D. “Feature selection and extraction in spatiotemporal traffic forecasting: a systematic literature review”. In: *European Transport Research Review* vol. 11, no. 1 (2019), p. 6.
- [Pis+18] Pisa, I. et al. “Vaima: A v2v based intersection traffic management algorithm”. In: *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE. 2018, pp. 125–128.
- [PS17] Polson, N. G. and Sokolov, V. O. “Deep learning for short-term traffic flow prediction”. In: *Transportation Research Part C: Emerging Technologies* vol. 79 (2017), pp. 1–17.
- [RMW14] Rezende, D. J., Mohamed, S., and Wierstra, D. “Stochastic backpropagation and approximate inference in deep generative models”. In: *arXiv preprint arXiv:1401.4082* (2014).
- [SAE16a] SAE. *SAE J2735: Dedicated Short Range Communications (DSRC) Message Set Dictionary*. J2735. 2016.
- [SAE16b] SAE. *SAE J2945: On-Board System Requirements for V2V Safety Communications*. J2945. 2016.
- [San+16] Santos, F. A. et al. “A Roadside Unit-Based Localization Scheme to Improve Positioning for Vehicular Networks”. In: *Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th*. IEEE. 2016, pp. 1–5.
- [San+18] San Martin, G. et al. “Deep variational auto-encoders: A promising tool for dimensionality reduction and ball bearing elements fault diagnosis”. In: *Structural Health Monitoring* (2018), p. 1475921718788299.
- [Sch+10] Schmidt, R. K. et al. “Exploration of adaptive beaconing for efficient intervehicle safety communication”. In: *IEEE network* vol. 24, no. 1 (2010).
- [Sep+11] Sepulcre, M. et al. “Congestion and Awareness Control in Cooperative Vehicular Systems”. In: *Proceedings of the IEEE* vol. 99, no. 7 (2011), pp. 1260–1279.
- [Sep+16] Sepulcre, M. et al. “Integration of congestion and awareness control in vehicular networks”. In: *Ad Hoc Networks* vol. 37 (2016), pp. 29–43.
- [SG18] Sepulcre, M. and Gozalvez, J. “Context-Aware Heterogeneous V2X Communications for Connected Vehicles”. In: *Computer Networks* (2018).
- [SGC17] Sepulcre, M., Gozalvez, J., and Coll-Perales, B. “Why 6Mbps is not (always) the Optimum Data Rate for Beaconing in Vehicular Networks”. In: *IEEE Transactions on Mobile Computing* (2017).
- [Sha+16] Shah, S. A. A. et al. “Adaptive beaconing approaches for vehicular ad hoc networks: a survey”. In: *IEEE Systems Journal* (2016).

- [Sha+17] Shang, C. et al. “VIGAN: Missing view imputation with generative adversarial networks”. In: *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE. 2017, pp. 766–775.
- [Som+15] Sommer, C. et al. “How Shadowing Hurts Vehicular Communications and How Dynamic Beaconing Can Help”. In: *IEEE Transactions on Mobile Computing* vol. 14, no. 7 (2015), pp. 1411–1421.
- [SST18] Shang, W., Sohn, K., and Tian, Y. “Channel-recurrent autoencoding for image modeling”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 1195–1204.
- [ST17] Shwartz-Ziv, R. and Tishby, N. “Opening the black box of deep neural networks via information”. In: *arXiv preprint arXiv:1703.00810* (2017).
- [Sun+17] Sun, L. et al. “Adaptive beaconing for collision avoidance and tracking accuracy in vehicular networks”. In: *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*. IEEE. 2017, pp. 1–6.
- [Söl+16] Sölch, M. et al. “Variational inference for on-line anomaly detection in high-dimensional time series”. In: *arXiv preprint arXiv:1602.07109* (2016).
- [Søn+16] Sønderby, C. K. et al. “Ladder variational autoencoders”. In: *Advances in neural information processing systems*. 2016, pp. 3738–3746.
- [Tat+20] Tatara, H. et al. “6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities”. In: *Proceedings of the IEEE* (2020).
- [TE20] Tapia, N. I. and Estévez, P. A. “On the Information Plane of Autoencoders”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2020, pp. 1–8.
- [Tie+11] Tielert, T. et al. “Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications”. In: *Vehicular Networking Conference (VNC), 2011 IEEE*. IEEE. 2011, pp. 116–123.
- [Tol+17] Tolstikhin, I. et al. “Wasserstein auto-encoders”. In: *arXiv preprint arXiv:1711.01558* (2017).
- [TS18] TS, E. *101 539-2 V1. 1.1: Intelligent Transport Systems (ITS), V2X Applications, Part 2: Intersection Collision Risk Warning (ICRW) Application Requirements Specification 1*. 2018.
- [TZ15] Tishby, N. and Zaslavsky, N. “Deep learning and the information bottleneck principle”. In: *2015 IEEE Information Theory Workshop (ITW)*. IEEE. 2015, pp. 1–5.
- [Ull+20] Ullah, Z. et al. “Applications of artificial intelligence and machine learning in smart cities”. In: *Computer Communications* vol. 154 (2020), pp. 313–323.

## Bibliography

---

- [VDW96] Van Der Voort, M., Dougherty, M., and Watson, S. “Combining Kohonen maps with ARIMA time series models to forecast traffic flow”. In: *Transportation Research Part C: Emerging Technologies* vol. 4, no. 5 (1996), pp. 307–318.
- [VH14] Van Erven, T. and Harremos, P. “Rényi divergence and Kullback-Leibler divergence”. In: *IEEE Transactions on Information Theory* vol. 60, no. 7 (2014), pp. 3797–3820.
- [VKG14] Vlahogianni, E. I., Karlaftis, M. G., and Golias, J. C. “Short-term traffic forecasting: Where we are and where we’re going”. In: *Transportation Research Part C: Emerging Technologies* vol. 43 (2014), pp. 3–19.
- [Vla15] Vlahogianni, E. I. “Optimization of traffic forecasting: Intelligent surrogate modeling”. In: *Transportation Research Part C: Emerging Technologies* vol. 55 (2015), pp. 14–23.
- [VV12] Van Lint, J. and Van Hinsbergen, C. “Short-term traffic and travel time prediction models”. In: *Artificial Intelligence Applications to Critical Transportation Issues* vol. 22, no. 1 (2012), pp. 22–41.
- [WG18] Wang, D. and Gu, J. “VASC: dimension reduction and visualization of single-cell RNA-seq data by deep variational autoencoder”. In: *Genomics, proteomics & bioinformatics* vol. 16, no. 5 (2018), pp. 320–331.
- [WP00] Weiland, R. J. and Purser, L. B. “Intelligent transportation systems”. In: *Transportation in the new millennium* (2000).
- [Wu+18] Wu, Y. et al. “A hybrid deep learning based traffic flow prediction method and its understanding”. In: *Transportation Research Part C: Emerging Technologies* vol. 90 (2018), pp. 166–180.
- [WWM19] Wei, W., Wu, H., and Ma, H. “An autoencoder and LSTM-based traffic flow prediction method”. In: *Sensors* vol. 19, no. 13 (2019), p. 2946.
- [YDC16] Yang, H.-F., Dillon, T. S., and Chen, Y.-P. P. “Optimized structure of the traffic flow forecasting model with a deep learning approach”. In: *IEEE transactions on neural networks and learning systems* vol. 28, no. 10 (2016), pp. 2371–2381.
- [YDC17] Yang, H.-F., Dillon, T. S., and Chen, Y.-P. P. “Optimized structure of the traffic flow forecasting model with a deep learning approach”. In: *IEEE transactions on neural networks and learning systems* vol. 28, no. 10 (2017), pp. 2371–2381.
- [YJS18] Yoon, J., Jordon, J., and Schaar, M. van der. “GAIN: Missing Data Imputation using Generative Adversarial Nets”. In: *arXiv preprint arXiv:1806.02920* (2018).
- [YP19] Yu, S. and Principe, J. C. “Understanding autoencoders with information theoretic concepts”. In: *Neural Networks* (2019).
- [YQ19] Yang, S. and Qian, S. “Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents”. In: *arXiv preprint arXiv:1901.06766* (2019).

- 
- [YRD16] Yu, H.-F., Rao, N., and Dhillon, I. S. “Temporal regularized matrix factorization for high-dimensional time series prediction”. In: *Advances in neural information processing systems*. 2016, pp. 847–855.
- [Yu+17] Yu, R. et al. “Deep learning: A generic approach for extreme condition traffic forecasting”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM. 2017, pp. 777–785.
- [Yu+19] Yu, S. et al. “Multivariate Extension of Matrix-Based Rényi’s  $\alpha$ -Order Entropy Functional”. In: *IEEE transactions on pattern analysis and machine intelligence* vol. 42, no. 11 (2019), pp. 2960–2966.
- [Zha+18] Zhang, C. et al. “Advances in variational inference”. In: *IEEE transactions on pattern analysis and machine intelligence* (2018).
- [Zha+19] Zhang, S. et al. “Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks”. In: *Sensors* vol. 19, no. 10 (2019), p. 2229.
- [Zho+17] Zhou, T. et al. “ $\delta$ -agree AdaBoost stacked autoencoder for short-term traffic flow forecasting”. In: *Neurocomputing* vol. 247 (2017), pp. 31–38.
- [Zhu+18] Zhu, L. et al. “Big data analytics in intelligent transportation systems: A survey”. In: *IEEE Transactions on Intelligent Transportation Systems* vol. 20, no. 1 (2018), pp. 383–398.