# Traffic Analysis and Prediction in Urban Areas

**Vasco Leal**

Thesis to obtain the Master of Science Degree in

**Information Systems and Computer Engineering**

Supervisors: Prof. Fernando Henrique Côrte-Real Mira da Silva
Eng. António Miguel de Abreu Ribeiro Henriques

## Examination Committee

Chairperson: Prof. Francisco João Duarte Cordeiro Correia dos Santos
Supervisor: Prof. Fernando Henrique Côrte-Real Mira da Silva
Member of the Committee: Prof. Rui Miguel Carrasqueiro Henriques

**October 2019**

# Acknowledgments

I would like to thank my parents for their friendship, encouragement and patience over all these years. This project would not be possible without them. I would also like to acknowledge my dissertation supervisors Prof. Fernando Mira da Silva and Eng. António Henriques, for their insight, support and sharing of knowledge that made this Thesis possible. Finally, I would also like to thank all my friends and colleagues that helped me grow and improve as a person throughout these past five years and without whom I probably would not have finished this course.

# Abstract

As the world's population continues to grow, cities get exponentially more crowded, which means new problems arise. One of the most dramatic impacts is on road traffic growth in most of Europe's capital cities, creating severe mobility constraints. One way to improve this issue is to deploy an Intelligent Transportation System (ITS) which includes the need to provide accurate traffic flow information to the users and enabling them to make a better and more informed use of transport networks. In this context, Traffic flow prediction is considered a critical element for the successful deployment of an ITS. In order to create an accurate traffic flow prediction model, a stable and consistent, traffic flow database is required. In order to enable the development of an optimized ITS, the Lisbon's City Council has been collecting mobility data. This thesis describes, analyses and applies different traffic predictive models to different traffic data sources available in Lisbon.

# Keywords

Short-term Traffic Flow Prediction. Machine Learning. Time-Series Forecasting. Spatio-Temporal Traffic Patterns.

# Resumo

À medida que a população mundial vai continuando a aumentar, os grandes centros urbanos estão cada vez mais densamente populados, o que faz com que surjam novos desafios. Este fenómeno resulta na agravação do congestionamento do fluxo do trânsito em grande parte das principais capitais Europeias, fazendo com que a mobilidade urbana dos cidadãos não seja eficiente. Uma possível solução para melhorar esta situação é a implementação de um sistema de transporte inteligente (ITS), que é um sistema que procura fornecer informação precisa aos utentes para que estes consigam fazer um uso mais informado e por isso mais eficiente da rede de transportes. Neste contexto, a predição do fluxo do trânsito é considerada um elemento crítico para o bom funcionamento de um sistema desta natureza. Para que seja possível desenvolver um modelo de predição de fluxo de tráfego estável e preciso é necessário uma base de dados consistente. Motivado por esta necessidade, a Câmara Municipal de Lisboa tem vindo a desenvolver esforços no sentido de recolher e fornecer dados de mobilidade urbana. No contexto deste projeto, a principal fonte de dados será proveniente de sensores contadores de trânsito colocados próximo de cruzamentos importantes no centro da cidade de Lisboa, sendo que também outras fontes de dados serão exploradas para testar a força preditiva dos modelos. O principal objetivo deste projeto é desenvolver métodos preditivos de fluxo do trânsito, utilizando as fontes de dados disponíveis, com o intuito de descobrir qual a abordagem que produz os melhores resultados.

# Palavras Chave

Predição de fluxo do trânsito. Aprendizagem Automática. Previsão de séries temporais. Padrões de trânsito.

# Contents

# List of Figures

# List of Tables

# 1

# Introduction

## Contents

As it is common knowledge, the human population has been increasing in the last few millennia. This tremendous growth has been, and still is, thoroughly studied by researchers throughout the world. These studies are motivated by multiple factors such as environmental, economic, sociological or even organizational. Although there are studies [7] claiming that the rate at which population is growing is slowing down, the world's population is still growing immensely. According to this study, the UN projections for the human population show that it may surpass the 10 billion mark by 2055.

The decrease in physical confrontations between civilizations, at least in the western world, the improvement of health-care and the implementation of human rights all led to an increase in the average life expectancy. However, given that our planet's resources are limited the population growth and increased life expectancy raise new issues. The abandonment of the rural areas in search of better opportunities and of increased quality of life leads to a substantial increase in population density in highly developed cities. Several articles and books have been published on this subject, in an attempt to analyze the causes and consequences of the phenomenon.

B. Bhatta [8] has carried out an extensive research in an attempt to understand what are the consequences that come from this and what can be done towards sustainable development of a city. Among the vast number of consequences presented in [8], one of the most relevant is the decreased urban mobility that comes from the uncontrolled growth of a city's population. The generalized increase in purchasing power is one of the factors that have led to the increase, in number, of personal vehicles in large urban centers. The problem of this increase, from an urbanistic point of view, is the fact that the number of cars that can flow in a city is finite. The analysis of this set of factors allows us to better understand the growth in traffic congestion in the last few decades.

## 1.1 Motivation

The topic of urban mobility has been extensively studied and it is central to the planning of a city since it affects all of its inhabitants. The decrease in urban mobility leads to more traffic jams and consequently a reduced quality of life for its occupants. This is a problem that concerns traffic operators of large cities around the world and because of this, numerous investments have been made to find and develop methods to mitigate these occurrences. From improving the city's urban transportation system to building and deploying an Intelligent Transportation System (ITS) there are multiple ways to address this issue. An Intelligent Transportation System aims to provide accurate information to the users and enabling them to make better and more informed use of transportation networks. Recently, it has been shown that Traffic Flow prediction is a crucial factor in the success of these systems.

As technology evolves, new traffic sensor technologies are emerging, making the amount of traffic data increase exponentially as we enter the era of big data in transportation systems. Traffic management and control are becoming increasingly data-driven [9, 10], which created a renewed interest in the field of traffic flow prediction.

The city of Lisbon shows many of the aforementioned problems. Namely, traffic mobility in the center of Lisbon has been significantly increasing in recent years. Therefore, there is a real need to develop strategies that will improve mobility and traffic flow.

## 1.2 Goals

The main objective of this thesis is to develop a traffic prediction system and applying it, as a case study, in some of the most congested locations in the city of Lisbon.

Traffic prediction systems have been proved to be an essential technique of traffic flow optimization. These systems serve as auxiliary methods to provide accurate information to traffic operators so that they can apply dynamic strategies in response to the predicted traffic conditions. Traffic flow data shows a considerable temporal dependency, given that the current traffic conditions heavily affect the immediate future. Consequently, since the imminent future is the most relevant, the prediction horizon of traffic flow prediction systems is usually small, i.e. **Short-Term traffic prediction**. This horizon usually varies from 15 minutes to an hour, depending on the context and the purpose of the system. Moreover, given that traffic flow is unstable and is prone to sudden changes due to external events, (e.g. vehicle collisions causing traffic jams) a larger prediction window would decrease the accuracy of the predictions.

In order to develop a system of this nature, traffic flow data is needed. Therefore, the City Council of Lisbon (CML) is developing an effort to gather and provide traffic flow data, namely, through traffic loop counters located strategically in multiple locations throughout the city of Lisbon. These sensors keep track of the number of cars passing through it in 15-minute intervals.

The results of this study could act as an indicator of the feasibility of implementing a system based on this model on a larger scale.

## 1.3 Organization

This thesis is structured as follows. Chapter 2 surveys literature on the most commonly used approaches for short-term traffic flow prediction, as well as an overview of the basic concepts needed to apply a traffic flow prediction model. In chapter 3, a thorough description and analysis of the data is presented. It contains descriptions about all the main data sources that were used in the work developed during the Master Thesis. Chapter 4 contains all the developed models for traffic flow prediction that were developed during the past semester, as well as the due results for each model. This chapter has two large sections, one for each dataset for which the models were developed. This chapter also contains a comparison of the developed models and some remarks on each model as well as some considerations about the available data. Finally, chapter 5 presents some concluding remarks about the work that was done during these past months as well as possible future lines of research.

# 2

# Related Work

Contents

This section will focus on describing some of the most commonly used and most successful approaches to predict short-term traffic flow as well as some of the key concepts on which these approaches are based on.

## 2.1 Background Research

Machine learning (ML) is the computational attempt on trying to teach machines how to learn from experience. ML algorithms that use computational methods to infer information directly from data are called supervised. The performance of supervised learning algorithms usually improves as the number of samples for learning increases. The set of learning instances is usually called **Training Set**. In order to assess the performance of the developed algorithm, a different, independent set of data is used, called **Testing Set**. In this section of the paper, the focus will be on reviewing some of the key concepts and Machine Learning algorithms that enable the implementation of traffic prediction systems.

### 2.1.1 Data Collection Techniques

The development of traffic prediction systems usually requires high-quality traffic flow information. For several years, with increasing pressure to improve urban traffic mobility, data collection techniques have been evolving considerably.

According to Guillaume Leduc [11], until recently, the most widely used data collection techniques relied on fixed road sensors (e.g loop counters). However, these are not sufficient due to the expensive costs of implementation and maintenance. In light of these limitations, alternative crowdsourced techniques based on vehicle location (Floating Car Data) have been emerging. One of the main advantages of Floating Car Data (FCD) is the reduced application costs, given that it does not require the implementation of road sensors.

FCD techniques rely on collecting real-time traffic information by locating the vehicle through mobile phones or GPS. In order for this to be a sustainable source of traffic information, a substantial number of vehicles have to be equipped with this technology. The data collected from the vehicles is anonymously sent to a central processing center. TomTom and Waze are relevant examples of the application of these techniques, where, with the consent of the users, traffic flow information is collected using GPS probe data. These sources are not meant to replace but to serve as a complement source of high-quality data to traditional methods, thus allowing the development of more robust traffic prediction systems.

### 2.1.2 Performance Metrics

In order to evaluate and compare the prediction accuracy of the models presented below three performance metrics will be considered. Namely, Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). These are determined by the following expressions:

$$MAPE = MRE(\%) = \frac{100\%}{n} \Sigma_{i=1}^{n} \frac{|pred_i - obs_i|}{obs_i} \tag{2.1}$$

$$RMSE = [\frac{1}{n} \Sigma_{i=1}^{n} \Big( |pred_i - obs_i| \Big)^2 ]^{\frac{1}{2}} \tag{2.2}$$

$$MAE = \frac{1}{n} \Sigma_{i=1}^{n} |pred_i - obs_i| \tag{2.3}$$

Where $pred_i$ represents the predicted value and $obs_i$ represents the observed value at time instant or interval $t_i$. In the context of traffic flow prediction, $pred_i$ and $obs_i$ represent traffic information such as speed or volume of vehicles per unit of time. Since traffic flow observations vary from a few hundred vehicles per hour in the off-peak to several thousand vehicles per hour during the peak periods, absolute percentage error (MAPE) provides the most useful basis for comparison. For this reason and for clarity the papers below will be evaluated according to their MAPE values except when the articles do not show these values. In this case, the RMSE will be used.

### 2.1.3 Neural Networks

Artificial Neural Networks (ANN) are among the most used and studied topics in Machine Learning. These Networks are vaguely inspired by the way that the information in our brains is processed and stored. The elementary units of an ANN are called **neurons** and looking at them separately, they are just small processing units that transform one input to one output with respect to a predefined activation function. Structurally, ANNs are very similar to the way the human brain processes information since they are composed by neurons organized in various layers with a varying degree of connections with the adjacent layers (**Weights**). When an ANN has multiple hidden layers it becomes a Deep Neural Network. These weights measure the relative influence between two neurons. In addition to the weights, the activation values of the previous layer also influence the next layer. The last layer of an ANN is the output layer, which, in the context of value prediction, presents the predicted value(s). The training of these networks is usually determined by minimizing an error function. In the training phase, the predicted values are compared to the real values and an error value is usually calculated based on the difference between these values (loss function). These errors are then sent back through the network, layer by layer, and, for each neuron, the derivative of this loss function is calculated. The weight is then changed according to the chosen optimization algorithm. Adam is one of the most widely used optimization algorithms and it shows many attractive benefits such as computational efficiency, has little memory requirements and the hyper-parameters require little to no tuning, which, in problems such as traffic flow prediction, could be a very time-consuming task. The main reason why this algorithm is so widely used and that it converges quicker than others is that, unlike the classical stochastic gradient descent, it adapts the learning rate of the network dynamically.

The training of a Neural Network is an iterative process and given that the adjustments in the weights are very small, it will need several iterations in order to converge (learn). The number of iterations that it takes to learn is not predictable and it depends on several factors, such as the quality of the training set and the chosen weight update rule. There are several variations of ANNs in terms of their architecture such as Stacked Autoencoders (SAEs) or Long Short-Term Memory Networks (LSTM) but they are all based on the aforementioned concepts.

### 2.1.4 Long Short-Term Memory Networks

Traffic flow shows a strong temporal dependency on the recent past of the state of traffic flow. In an attempt to better capture these dependencies an Long-Short Term Memory Network was developed, which is a type of Recurrent Neural Network (RNN). RNN's, unlike feed-forward neural networks (as is the one mentioned in Section 4.1.2), don't only take as its input the current input example but also the previous outputs of the network.

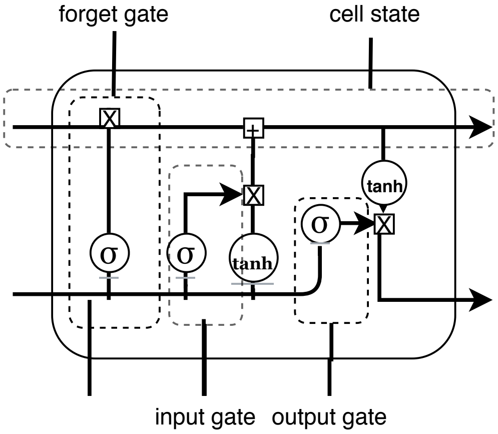


**Figure 2.1:** Diagram of a LSTM unit, adapted from [1]

This means that in an RNN, the output reached at t-1 affects the decision at timestep t. These networks have a feedback loop connected to their past decisions, which can be identified as the **Memory** of the network. The reason behind the addition of memory to neural networks is the belief that, in some problems, there is valuable information in the past occurrences.

The sequential information is stored in a hidden state, which manages to span many time steps as the simulation moves forward to affect the processing of each new input. The correlations between events separated by many timesteps are called **long-term dependencies**.

Each LSTM unit is composed by three main gates, namely, the forget gate, the input gate and the output gate as is shown in **Fig.** 2.1. The forget gate is used by the network to control whether to forget or keep old information. The current input and the previous hidden state are combined and passed through the sigmoid function. The output values come out between 0 and 1 where 0 means that the information is completely ”forgot” and 1 means that the totality of the information is kept. The input gate decides which new information is going to be stored in the cell state and the output gate determines what is going to be the next hidden state.

### 2.1.5 K-Nearest Neighbors

This algorithm is a non-parametric method mostly used in classification and regression. The basis of this algorithm lies in finding, for each sample, the *k-nearest* samples that are closest to each other according to a distance metric. The most commonly used distance metric for numerical values is the Euclidean distance. In the context of traffic flow prediction, the main idea behind the application of this algorithm is that traffic flow is periodic and therefore it is very likely that patterns in the past are going to be similar to future ones. Therefore, for this algorithm to predict what’s going to happen at a given day (Subject Profile) it identifies the *k* most similar patterns (Candidate Profiles) and then combines those patterns, providing future, unobserved patterns. Even though this is a much simpler algorithm than Neural Networks, it is considerably easier to implement and depending on the quality of the data, it can perform very well in the context of traffic flow prediction, especially for small prediction horizons, where its performance is similar to the performance of some deep learning methods [4].

### 2.1.6 Stochastic time-series models

According to [12], a time-series is a sequential set of data points, typically measured successively. Time-series forecasting techniques are a very important field of machine learning since prediction problems inherently depend on a time component. It is central in the field of traffic flow prediction since traffic flow shows a high temporal dependency. For some machine learning problems, such as the one studied in this project, the time dimension in the dataset provides a new source of information. A time-series is usually affected by three main components, namely, Seasonal, Trend and Random components. Trend dictates the general tendency of a time-series to increase, decrease or stagnate over time. In a time-series, the fluctuations within a year that follow similar patterns year after year are called seasonal variations. In the context of traffic flow, one of these variations is the increased traffic affluence during holiday seasons like Christmas. Through the analysis of time-series, it is often observed that it is affected by some unpredictable factors that are not regular and do not repeat in a particular pattern. These are called irregular or random variations. According to these for components, there are two different types of time-series models, the additive and the multiplicative model. The additive model is based on the assumption that the main components of a time-series are independent. Inversely, in the multiplicative model, it is assumed that the aforementioned components are not necessarily independent of each other.

There are many prediction models for time-series data. The most commonly used linear time-series models are Autoregressive (AR) and Moving Average (MA), presented in [13, 14]. In an AR model AR(p), the future value of a variable is assumed to be correlated with the past $t$ observations of the same variable, which, according to [15] is given by:

$$w_t = \phi_1 w_{t-1} + \phi_2 w_{t-2} + ... + \phi_p w_{t-p} + a_t \tag{2.4}$$

Where $w_t$ represents the current observation of the time series and $a_t$ represents the error between the forecast and the real value. The regression coefficients, $\phi_i, i = 1, ..., p$ are parameters to be estimated from the data. A Moving Average model is used as a filter to smooth the data. The basis of moving average models is to simply average time points. A simple example of a moving average is illustrated by Eq. 2.5 where p represents the size of the window of time-points. Predicting time-points based only on past observations has some problems. For example, if the time-series being modeled is increasing, averaging the past observations will produce estimations that are always smaller than the real values. In light of this limitation, these models are usually used over the error component $a_t$, as shown in Eq. 2.6

$$x'_t = (x_{t-1} + x_{t-2} + ... + x_{t-p})/p \tag{2.5}$$

An MA(q) models the averages of past and present noise terms and can be defined by:

$$w_t = a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + ... + \theta_q a_{t-q} \tag{2.6}$$

Where $w_t$ represents the current observation of the time series and $a_t$ represents the error term. The regression coefficients, $\theta_i, i = 1, ..., q$ are parameters to be estimated from the data. This removes noise in the data and facilitates the forecasting. The combination of these two models originated the ARMA model. These models can only be used to describe stationary time-series data. A Stationary time-series is one whose properties don't depend on the time point at which the series is observed. Usually, a stationary time-series has no long-term predictable patterns and its time plot is roughly horizontal. However, many time-series, such as traffic data, show non-stationary behavior. The ARIMA model was developed precisely to solve this problem, as it can deal with non-stationary data. This model is defined by three parameters, where each of them refers to the Autoregressive, integrated and moving average part of the model, respectively. When the seasonality of the data is known, an extension of ARIMA that models seasonal data can be developed (SARIMA [3]). Although ARIMA methods are the most popular in time-series forecasting, exponential based methods such as the Holt-Winter models [16] have also shown to be useful prediction techniques.

## 2.2 Traffic prediction with Neural Networks

The study presented in [6] was conducted by Yisheng Lv Et al. in 2015 in the United States and proposes a Deep Learning approach to address the issue of Traffic Flow Prediction.

The basis behind the application of this family of algorithms lies in the ability that the architectures used by Deep Learning algorithms have to extract inherent features in the data, discovering hidden structure in the data. In the context of Traffic flow, these features could be the influence that a periodic event has on the regular traffic flow or the periodicity of traffic flow, e.g., peak and off-peak hours.

The Deep Learning model chosen for the purposes of this paper was the Stacked Autoencoder (SAE). This model uses Autoencoders as building blocks to create a deep network. Autoencoders are a type of Artificial Neural Network generally used for dimensionality reduction and are almost never used in forecasting models. In this work, a SAE is used to extract traffic flow features and a logistic regression layer is applied for the prediction.

The proposed method was applied to data collected from freeways in the United States as a performance benchmark for this algorithm. The data was collected every 30 seconds from over 15000 individual sensors and then aggregated in 5-minute intervals for each sensor. For the experimental part of the paper, three months worth of data was collected. The first two months were used as a training set and the remaining month as a validation set. This model was tested for multiple windows of prediction, namely, 15, 30, 45 and 60 minutes. As it was previously said, one of the reasons that enable traffic flow prediction is the periodicity of traffic flow. Interestingly, after running the system multiple times, it was found that, for any of the aforementioned prediction horizons, the performance peaked when, at most, the previous hour of data was used as an input for the network. After the analysis of the results, it was evident to conclude that the system had a better performance in heavy traffic over low traffic conditions. One possible reason for this phenomenon is that, in low traffic conditions, small deviations can cause a larger relative error.

The performance metrics used to test the effectiveness of the system were the MAE, MRE and RMSE. This algorithm achieved very impressive results, given that the MRE remained relatively constant with the increase in prediction horizons (contrarily to the work developed by Filmon G. Habtemichael and

Mecit Cetin [4]).

From the observation of Table 2.1 it is possible to conclude that the MRE(%) (= MAPE) values are in the range between [6.2%,6.75%] which shows the aforementioned stability of the proposed algorithm. It is also interesting to point out that the values of the MRE have shown a tendency to be smaller with larger prediction horizons, which is unusual since in larger prediction spans its more likely to see an increase in prediction errors.

| | SAE | | |
|---|---|---|---|
| | MAE | MRE (%) | RMSE |
| 15-Minute Traffic Flow Prediction | 34.1 | 6.75 | 50.0 |
| 30-Minute Traffic Flow Prediction | 64.1 | 6.48 | 95.2 |
| 45-Minute Traffic Flow Prediction | 92.0 | 6.17 | 138.1 |
| 60-Minute Traffic Flow Prediction | 122.8 | 6.21 | 183.9 |

**Table 2.1:** Performance of the traffic prediction model proposed in [6]

In conclusion, this unique approach seems to have paid off, since the average accuracy is over 93% (100-MRE(%)). Although, similarly to other studies in traffic flow prediction, these methods were only tested on highways, that are much more controlled and predictable environments than urban areas.

Corrado de Fabritiis Et al. [17] introduced a working application of a Floating-Car Data system, delivering accurate traffic speed information. Unlike regular traffic data collection techniques (loop counters, sensors), cars equipped with these GPS receivers work as moving sensors, not requiring any type of instrumentation to be set on the roadway. More than 600,000 of these specific GPS devices were distributed in Italy and this number is said to be increasing, with an average increase of 30,000 per month. The main purpose of this paper is to present a short-term traffic flow prediction model based on historical and present Floating-Car data. These sensors, referred to as On-Board Units (**OBU**) transmit every 12 minutes when the equipped car travels along one of the predefined locations.

For the purposes of the prediction, the data was aggregated in intervals of 3 minutes and then used to predict traffic flow in temporal windows of 15 and 30-minutes. As a case study, data from a motorway in Rome was used, where important locations (e.g motorway exits) were identified as the nodes relevant for the prediction. Two different algorithms were developed to perform short-term traffic flow prediction.

A simple *Pattern Matching Algorithm* was developed and applied over data that was already classified into speed categories. This approach didn't produce stellar results, achieving a misclassification error of

19% for 15-min prediction.

For more complete, numerical, speed traffic data, an Artificial Neural Network (ANN) was used for the prediction. Two feed-forward ANNs were developed, one was trained for 15-Minute prediction while the other was trained for the 30-Minute prediction. Both networks were trained using the Levenberg-Marquardt algorithm [18]. In order to test the effectiveness of these methods, two metrics were used, namely MAPE and RMSE.

With regard to the 15-Minute prediction, the values of MAPE ranged from 2% to 8% and the RMSE varied from 2 to 7 km/h. As for the 30-Minute prediction the MAPE ranged from 3% to 16% while the RMSE varied from 3.5 to 9.5 km/h.

The following article was published in 2016, in India, which is one of the largest populations in the world with over 1.3 Billion people. Consequently, traffic congestion is a major challenge in traffic management, therefore multiple studies have been published that address and try to improve this situation. In this context, [19] approaches this problem by exploring the use of automated sensor data and traffic prediction techniques.

The data used as a use case of this research was collected using a location-based sensor that uses infra-red technology to detect vehicles passing through. Only one of these sensors was used to collect data. This article aims to predict, instead of the total number of cars passing through a given location, the traffic density. Traffic density is estimated through the speed and volume values provided by the sensor. Traffic density measures the average amount of vehicles that occupy one kilometer of road space and it is expressed in vehicles/km. The speed and volume data were aggregated at intervals of 5 minutes. Data from 4 weekdays was used as the training set and the same number of days of a different week were used as a testing set.

Along with the stand-alone application of k-NN and ANN, a model that combines (Fig. 2.2) these two algorithms was also developed. In this model, k-NN is used to determine the first k nearest neighbors of the target record, which are later fed to the ANN that uses them to predict the traffic density.

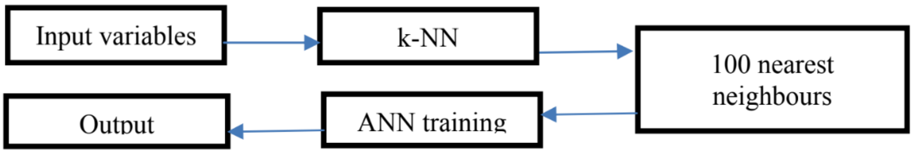
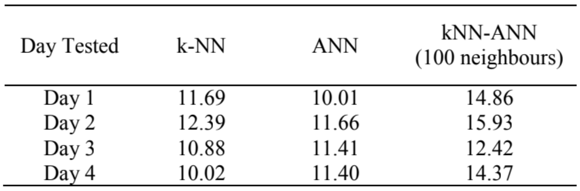


**Figure 2.2:** Model for combining ANN with k-NN (from [19])

The metric used to measure the performance of the models was MAPE. After testing the three models, it was found that the combination of k-NN with ANN didn't improve the results of the prediction as it is shown in Fig. 2.3. The results presented in Fig. 2.3 show that the MAPE values are between 10-12% for the k-NN. Regarding the ANN the results weren't much better with the values for the MAPE ranging



| Day Tested | k-NN | ANN | kNN-ANN<br>(100 neighbours) |
|---|---|---|---|
| Day 1 | 11.69 | 10.01 | 14.86 |
| Day 2 | 12.39 | 11.66 | 15.93 |
| Day 3 | 10.88 | 11.41 | 12.42 |
| Day 4 | 10.02 | 11.40 | 14.37 |

**Figure 2.3:** MAPE obtained for the prediction

between 10-11%. One of the possible reasons for the low performance is the very small amount of training data used. The authors concluded that combining k-NN with ANN did not show any improvement in the overall performance of the system and hence it is not recommended.

Istanbul, being one of the most crowded cities in Europe, with more than 15 Million (10 Million at the time of the study) inhabitants and having the highest percentage of newly registered vehicles in Turkey (around 47% in 2017 [20]), has accentuated traffic congestion problems. As such, Istanbul can be considered an optimal use case to test the effectiveness of traffic-flow prediction methods. Accordingly, B. Gültekin Çetiner Et al. [21] developed and tested a short-term traffic flow prediction model, in order to measure the feasibility of the application of a wider, real-world system based on this model.

The government has invested a lot in traffic surveillance, providing live monitoring of 180 major locations using cameras and other sensor technologies [22]. The system provides valuable information to traffic operators that can act and inform drivers of the expected traffic conditions for the next hour. The data used for the prediction in this research was provided by ISBAK, which is the semi-government organization responsible for maintaining roads, including traffic junctions in Istanbul. The data was aggregated in intervals of 5-minutes and each interval describes the total number of cars that passed through that sensor in that period of time. This model was only tested in one location, using data of only one remote microwave traffic sensor (RTMS).

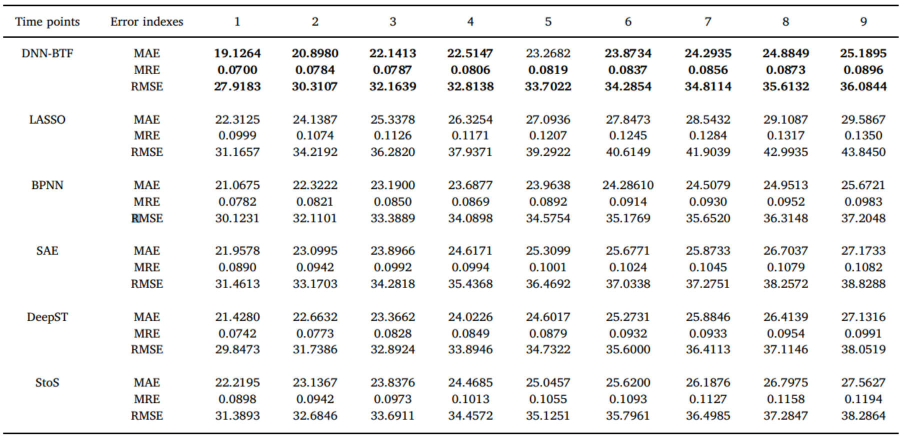
The data considered in the development of this model covers a one-year period, from January 2006 to the last day of the same year. Two ANNs were tested, the first predicts traffic flow with a prediction window of 5 minutes, which is not ideal in a real-world scenario since that amount of time is not sufficient in order for traffic controllers to address the issue. The second uses a preprocessed version of the dataset, where the data was aggregated in intervals of one hour. Therefore, only the performance of the second network will be presented here. The data was split equally into two different sets, namely, a training set and a testing set. Unlike other papers, the metric used to evaluate this system was the correlation coefficient. The correlation coefficient for the second network using the testing set as an input was 0.88.

The study presented above seems to have a few unclear aspects. However, it shows some relevant features, namely, the usage of microwave sensors for the data collection and the usage of an Artificial Neural Network to predict future traffic flow.

The study performed by Yuankai Wu Et al. [2] proposes a deep neural network based traffic flow prediction model (DNN-BTF). This architecture uses a convolutional neural network in order to capture the spatial features inherent to traffic flow as well as a recurrent neural network to capture the temporal dependency of traffic flow. It is stated that the architecture chosen by the authors performs better than other Neural Network architectures in capturing the spatial-temporal characteristics of traffic flow. A similar architecture was developed in [23] with the same objective of capturing these traffic flow features. It is well known that traffic flow is periodic and that historical data is important in the prediction of future traffic flow. In most cases, like in [4], the recent past is scored more heavily since it is natural to assume that recent events have stronger correlations with the immediate future than older ones. However, the temporal correlation of traffic flow is influenced by many factors such as weather events, time of day or road accidents. The model presented can learn the relative importance of each data point, with the objective of maximizing the prediction results.

Another characteristic of traffic flow is its spatial correlation, the traffic conditions of adjacent locations strongly influence each other [24, 25]. In light of this property, this model was designed to be able to capture these spatial features.

To test all of these hypotheses, data from an open-access database was collected. The traffic flow data used to test this model was from a stretch of a freeway in California and it covers 15 months. This database contained traffic flow information from 33 sensors placed along the freeway. The data was aggregated into intervals of 5 minutes. The prediction horizon was set to 45 minutes, and 105 minutes of historical data were used for the prediction. It is worthy to note that this model predicts traffic flow for all the 33 sensors.



| Time points | Error indexes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| DNN-BTF | MAE | **19.1264** | **20.8980** | **22.1413** | **22.5147** | 23.2682 | **23.8734** | **24.2935** | **24.8849** | **25.1895** |
| | MRE | **0.0700** | **0.0784** | **0.0787** | **0.0806** | **0.0819** | **0.0837** | **0.0856** | **0.0873** | **0.0896** |
| | RMSE | **27.9183** | **30.3107** | **32.1639** | **32.8138** | **33.7022** | **34.2854** | **34.8114** | **35.6132** | **36.0844** |
| LASSO | MAE | 22.3125 | 24.1387 | 25.3378 | 26.3254 | 27.0936 | 27.8473 | 28.5432 | 29.1087 | 29.5867 |
| | MRE | 0.0999 | 0.1074 | 0.1126 | 0.1171 | 0.1207 | 0.1245 | 0.1284 | 0.1317 | 0.1350 |
| | RMSE | 31.1657 | 34.2192 | 36.2820 | 37.9371 | 39.2922 | 40.6149 | 41.9039 | 42.9935 | 43.8450 |
| BPNN | MAE | 21.0675 | 22.3222 | 23.1900 | 23.6877 | 23.9638 | 24.28610 | 24.5079 | 24.9513 | 25.6721 |
| | MRE | 0.0782 | 0.0821 | 0.0850 | 0.0869 | 0.0892 | 0.0914 | 0.0930 | 0.0952 | 0.0983 |
| | RMSE | 30.1231 | 32.1101 | 33.3889 | 34.0898 | 34.5754 | 35.1769 | 35.6520 | 36.3148 | 37.2048 |
| SAE | MAE | 21.9578 | 23.0995 | 23.8966 | 24.6171 | 25.3099 | 25.6771 | 25.8733 | 26.7037 | 27.1733 |
| | MRE | 0.0890 | 0.0942 | 0.0992 | 0.0994 | 0.1001 | 0.1024 | 0.1045 | 0.1079 | 0.1082 |
| | RMSE | 31.4613 | 33.1703 | 34.2818 | 35.4368 | 36.4692 | 37.0338 | 37.2751 | 38.2572 | 38.8288 |
| DeepST | MAE | 21.4280 | 22.6632 | 23.3662 | 24.0226 | 24.6017 | 25.2731 | 25.8846 | 26.4139 | 27.1316 |
| | MRE | 0.0742 | 0.0773 | 0.0828 | 0.0849 | 0.0879 | 0.0932 | 0.0933 | 0.0954 | 0.0991 |
| | RMSE | 29.8473 | 31.7386 | 32.8924 | 33.8946 | 34.7322 | 35.6000 | 36.4113 | 37.1146 | 38.0519 |
| StoS | MAE | 22.2195 | 23.1367 | 23.8376 | 24.4685 | 25.0457 | 25.6200 | 26.1876 | 26.7975 | 27.5627 |
| | MRE | 0.0898 | 0.0942 | 0.0973 | 0.1013 | 0.1055 | 0.1093 | 0.1127 | 0.1158 | 0.1194 |
| | RMSE | 31.3893 | 32.6846 | 33.6911 | 34.4572 | 35.1251 | 35.7961 | 36.4985 | 37.2847 | 38.2864 |

**Figure 2.4:** The MAEs, MREs, and RMSEs on different prediction horizons (h = 9). (from [2])

The proposed model was compared with several state-of-the-art methods and the results are presented in Fig. 2.4. By analyzing Fig. 2.4 it is possible to conclude that the proposed model (DNN-BTF) outperforms all of the other methods, averaging an MRE of 0.082. The MRE increased slightly with each prediction horizon but the maximum error value was 0.0896 for 45-minute prediction, which is a very impressive result. An error analysis was also conducted to examine the prediction capability of the proposed model to understand how multiple conditions influence the prediction performance. Interestingly, it was found that the proposed model had better performance during the daytime and that it showed little variation on the MRE of rush-hour and non rush-hours. Fig. 2.5 shows that, between 1h00 and 3h00, the traffic volume is really low, which means that it is more likely that traffic flow is more likely to show free flow state at that period, thus making these hours less predictable (which translates into an increase in the MRE, as is shown in Fig. 2.6).

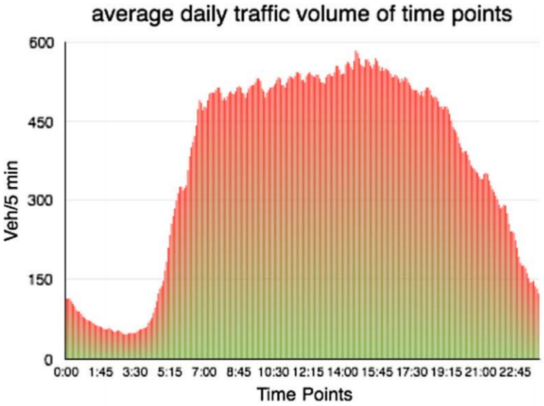


**Figure 2.5:** Average traffic flow per hour (from [2])

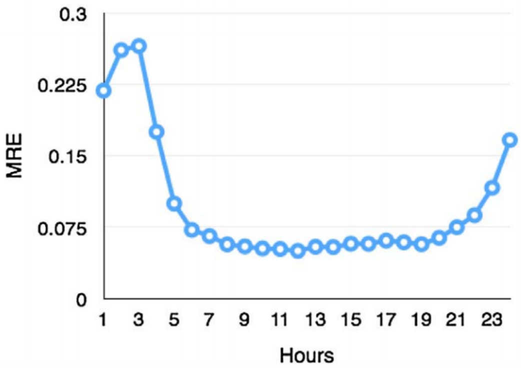


**Figure 2.6:** Average variation of MRE in 24h (from [2])

In conclusion, the model developed in this study achieved some interesting results, showing its robustness by having a small range of values for the MRE(%) (7% - 8.96%) regardless of the prediction horizon.

From some of the previously presented papers, it is possible to conclude that the most recent research in this field aims at capturing the temporal and spatial dependencies of traffic flow. The model proposed in [26] tries to accomplish this by developing a deep learning approach. Accurate traffic flow prediction is challenging due to the existence of these complex dependencies.

One thing that differentiates the work carried out by Yaguang Li Et al. is that, in this model, traffic flow at time-step t is represented as a directed graph where each node represents a sensor and has its respective traffic information (velocity, volume). The weights of the edges connecting each node (sensor) are determined by taking into consideration the nodes proximity (e.g as a function of their road network distance). Then these are fed as input to the Neural Network.

This study was tested in two real-world large-scale datasets. The first dataset was **METR-LA**, which contains traffic information collected from loop detectors in the highway of Los Angeles County. In the context of this project, 207 sensors were selected and 4 months of data were collected. The second one was **PEMS-BAY** which contains traffic data from the San Francisco Bay Area region. In this case, 6 months of data were collected and 325 sensors were chosen. In both datasets, the data was aggregated into 5-minute intervals. Also, in both cases, 70% of the dataset was used as a Training Set, 10% was used as a Testing Set and the remaining 20% was used as the Validation Set. This model was tested for 3 different forecasting horizons: 15, 30, 60 minutes. In the case of the Los Angeles dataset, the mean MAPE for the three different horizons was, respectively, 7.3%, 8.8%, and 10%. The error results for the Bay Area dataset were much smaller, 2.9%, 3.9%, and 4.9%. One reason for this discrepancy could be related to the adverse traffic conditions that LA is known for. This model was compared to other baseline models and it was found that Deep Neural Network based methods tend to perform better than other algorithms for long-term forecasts (e.g. 1 hour ahead). One explanation for this phenomenon is that the temporal dependency becomes increasingly non-linear with the growth of the horizon.

## 2.3 Time-Series Forecasting

As it was previously mentioned, one of the main obstacles for a real-world application of these data-driven algorithms is precisely the lack of consistent usable data sources. The work developed by S. Vasantha Kumar and Lelitha Vanajakshi [3] addresses the problem of limited input data in Traffic Flow prediction. This study tries to overcome the issue of limited available data by proposing a traffic prediction system using a seasonal autoregressive integrated moving average (SARIMA) model. Firstly, this model is only applicable if the span of seasonality is known. Seasonality can be defined by a pattern in the data that repeats over S time periods, where S defines the number of time periods until it repeats again. The difference of SARIMA in respect to ARIMA is that the former models the seasonality of the data. In a SARIMA model the predictions are calculated using data values at times with lags that are multiple of S. Through the observation of the plot of the data (Fig. 2.7) from the three consecutive days, it is clear that there is a seasonality of 24h in the data. Thus, the seasonal period S is 144 (24h × 6 points/hr). As a case study for the effectiveness of the proposed methodology, a very busy 3-lane arterial roadway in

Chennai, India was selected and only three consecutive days were used as the input data for the model development. The data was collected with a single automated traffic sensor placed in the selected road.

The parameters necessary for the application of the SARIMA were found using the maximum likelihood method [27]. After the developing part, the model was validated by performing 24h ahead forecast and comparing the actual values with the predicted ones. For the prediction, the total number of vehicles aggregated into 10-minute time intervals were considered as input. Fig. 2.7 also shows that the morning and evening peak hours were clearly repetitive and showed similar variation across the days. This piece of information is crucial since it shows that traffic flow data is periodic and therefore can be modeled using SARIMA. Initially, the model was tested with the aforementioned input data and the results were encouraging, with a MAPE of 9,22%. After the analysis of the results, it was found that the model performed worse in off-peak hours since their patterns are more random thus making it harder for a time-series model to perform at its best.

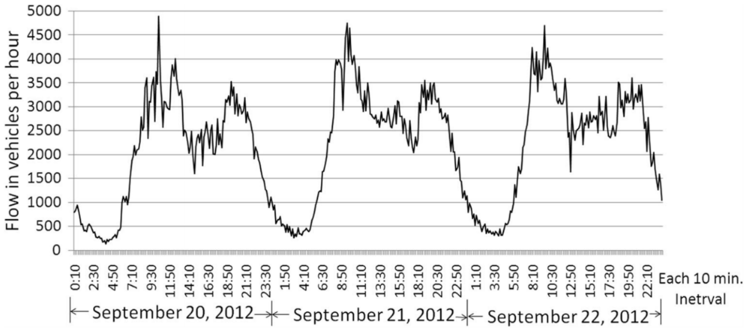


**Figure 2.7:** Time-series data of observed traffic flow in three consecutive days (from [3]). ScenarioID represents the number of days that were used as historical data for the prediction.

In order to check whether the results would improve with an increase in the input data, 6 new scenarios were tested. Instead of the initial 3 days, the model was tested with up to 9 previous days as input.

Through the analysis of Fig. 2.8 it can be seen that, initially, the MAPE increases slightly, however, when the past week, including the same weekday as the target day, is considered as the input the MAPE suffers a sudden drop. This reinforces the idea that same weekdays follow similar patterns. In this paper, short-term traffic prediction was also attempted, with a maximum prediction horizon of 1h ahead. In this case, the historical data (five previous days of traffic flow data) was used along with the real-time data until the time of prediction. The MAPE values between the predicted and the real traffic flow for morning and evening peak hours were found to be 4.37 and 3.83, respectively. These values are much lower when compared to other traffic prediction models. However, these only refer to the peak hours, which are, usually, the most easily predictable hours because these tend to repeat the same patterns every day. Non-rush hours tend to be more stochastic, making them less predictable. It is important to keep

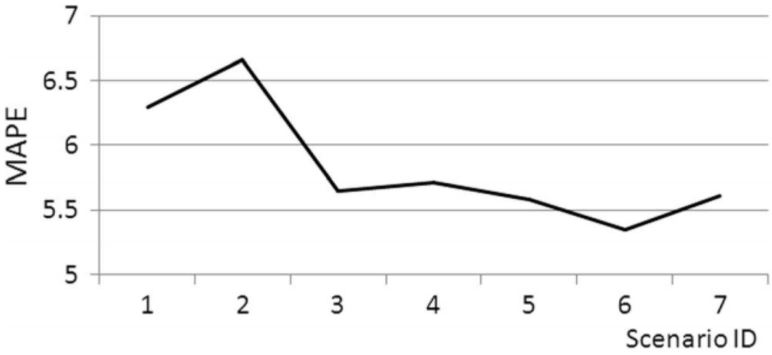


**Figure 2.8:** Variation of the MAPE across the different scenarios (From [3])

this in mind when comparing these MAPE results with similar articles where the MAPE considers any time of day.

In conclusion, the overall results were encouraging and the proposed model for traffic prediction could be an effective solution in situations where only limited observations are available.

## 2.4 Other Traffic prediction methods

Filmon G. Habtemichael and Mecit Cetin [4] proposed a traffic prediction approach based on the k-Nearest Neighbors algorithm, similar to the study carried out by Lun Zhang Et al. [28]. The main objective of [4] was to develop an accurate traffic flow prediction system for non-urban areas, i.e freeways. The data used in this study was collected through multiple sensors placed throughout specific freeways in the US and the UK. An enhanced version of ***k-NN*** was used in order to create the prediction model. In the context of traffic flow prediction, the main premise behind the application of this algorithm is that traffic flow is periodic. Therefore it is very likely that patterns in the past are going to be similar to future ones. Consequently, for this algorithm to predict what's going to happen at a given day (Subject Profile) it identifies the *k* most similar patterns(Candidate Profiles) and then combines those patterns, providing future, unobserved patterns. For the purposes of this study, the data was aggregated in 15-minute intervals with a prediction horizon ranging from 15 minutes to an hour.

A weighted version of the Euclidean Distance (Eq. 2.7)was used as a distance metric in order to emphasize more recent data over older records where the weights were distributed linearly according to the recentness of the values.

$$d(x,y) = \sqrt{\sum_i w_i \times (X_i - Y_i)^2} \tag{2.7}$$

In Eq. 2.7, $X$, and $Y$ represent two different traffic profiles and $w_i$ represents the weights assigned to the traffic profiles. Given that real data is prone to have outliers, the process of Winsorization was applied to the data to dampen the effects of extreme candidate values. Winsorization [29, 30] can be very useful when working with traffic flow data that, as its known, can easily be affected by multiple factors such as adverse weather conditions and special events like concerts or football matches.

After the $k$ candidates are found, it would be unwise to assign each of them the same weight (uniform weighting scheme) for the forecasting, since not all of them are at the same (Euclidean) distance from the Subject Profile. As a consequence, each candidate has an attributed weight value according to its similarity to the subject profile.

As it was previously mentioned, this algorithm provides forecasts ranging from 15 minutes to 1h30, with steps of 15 minutes (Step $\in \{1, 6\}$). Unsurprisingly, with an increase in the prediction steps the forecasting accuracy tends to decrease, which is supported by the increase in MAE, MAPE and RMSE. According to the authors, the average increases of MAPE, MAE, and RMSE by prediction step were 7%, 9%, and 9%, respectively.

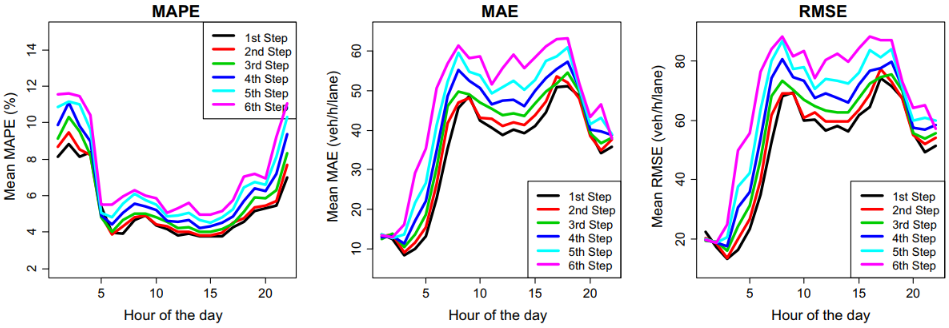


**Figure 2.9:** Forecast errors for multiple forecast steps by hour of day (from [4])

Performance wise this paper achieved very interesting results, namely, an average **MAPE** of 5.3% for a **15-minute prediction horizon**. Even though the error is low, it is wise to remember that it refers to a 15-minute prediction which is not an adequate prediction horizon if the objective is to provide real-time information to prevent or reduce traffic congestion. Thus, realistically, if this paper was applied in a real-world situation, traffic operators would need a larger prediction horizon in order to apply effective dampening strategies.

Finally, it is also relevant to note that the data was collected from **freeways** which are much more stable and predictable environments than **urban** areas.

Lun Zhang Et al. proposed an improved version of the k-NN algorithm to perform traffic flow prediction [28]. This study was done in Shanghai, China, where traffic congestion is a constant concern since its a very densely populated city with over 24 Million inhabitants. This study aims at predicting traffic flow in a span of 5 minutes. This interval is unusually small when compared with other papers presented in this section. However, the authors claim that a cycle of traffic control (e.g control of stoplights) regularly takes less than 3 minutes, and, therefore, traffic flow prediction within 5 minutes is considered reasonable. The main data source of this project comes from the Traffic Information Center of Shanghai. In this study, an elevated expressway called North-South Elevated Road was selected to test the performance of the developed model. The data is collected through loop sensors that are placed in the road pavement. The flow data from October 1st to November 17th was used for the training. In order to test the accuracy of the model, the following two days of November were used as a testing set. This paper uses, as evaluation metrics, MAPE and the Mean Absolute Difference (MAD). However, MAPE was used as the standard metric. This model was tested with several values for K and for the hysteresis (q). Hysteresis is generally defined as the dependence of the state of a system on its history. In the context of this project, it refers to the amount of historical data that is used.

It was found that the proposed model got the best results with k = 18 and q = 4. This was the chosen setup for the result analysis. This paper achieved a mean MAPE value of 9,44%, translates to an average accuracy of 90,56%.

# 3

# Data Sources

## Contents

This chapter describes the available traffic data sources, and discusses their structure and limitations.

There are two main data sources, namely, an urban dataset, specifically, a dataset that contains traffic flow information of the city of Lisbon and a freeway dataset that will be used to validate the predictive ability of the developed models.

## 3.1 Lisbon Sensor Data

The interest in urban mobility is not new. There has always been a need to improve and optimize the flow of traffic in city centers, since they are usually the most densely populated areas in urban areas. Lisbon is no exception. Back in the '80s, traffic flow optimization was already a pressing concern for the city council and because of that, a considerable investment in cutting edge intelligent stoplight control was made, culminating with the implementation of the GERTRUDE system. This system depended on, among other things, loop sensors to estimate the traffic intensity in specific locations so that certain stoplights could have their loops optimized dynamically according to traffic flow conditions.

The Lisbon City Council (CML) has been collecting urban flow data in multiple arterial junctions throughout the city for several years using the aforementioned GERTRUDE road sensors. In the context of this project, and since urban traffic mobility problems are increasing, a partnership between research institutes and CML was established where this data was made available.

### 3.1.1 Data Structure

The data being collected by GERTRUDE sensors is the absolute count of cars that passed through each of the sensors. These counts are captured by a single channel inductive loop sensor. See, [31] for an example of a similar system. These sensors are installed in the pavement, usually near stoplights or important junctions. An inductive loop measures the change in the magnetic field when objects, in this case, vehicles, pass over them. When a vehicle drives over one of these sensors, the loop field changes, which allows the device to identify its presence (explained in more detail in [32]). Every one of these sensors keeps track of the total number of cars that pass over them and stores this value every 15 minutes, which adds up to 4 data points per hour.Therefore, every day, 96 new counts are recorded.

These sensors are labeled by zones, the identifier of the junction and the identifier of the sensor itself. There are over 120 sensors placed throughout the center of Lisbon. The geographical placement of the sensors is displayed in Fig. 3.1. The dataset contains historical data dating from 2011 to April 2019. However, there are some periods between those dates for which we have no data records, specifically, we received three chunks of data as depicted in **Table 3.1**. All the data is being provided directly by the city council and at the present, there is no automated manner to get real time traffic data. In the context of this project, only historical data will be used to train, validate and test the prediction model.

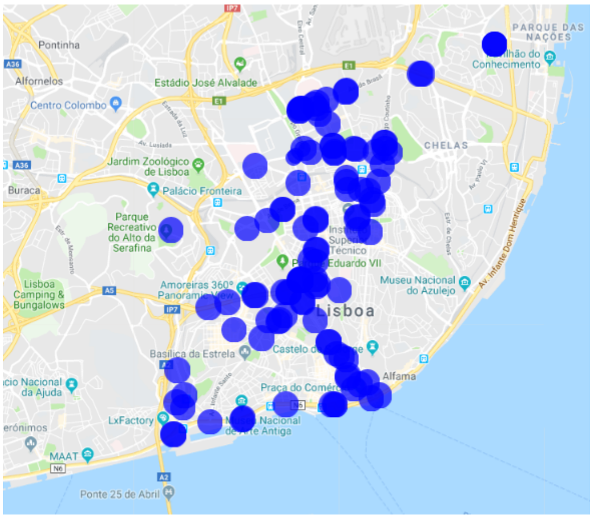


**Figure 3.1:** Map of the sensors placed in Lisbon

| Folder name | Start Date<br>(dd-mm-yyyy) | End Date<br>(dd-mm-yyyy) |
|---|---|---|
| InfoCentral_20180119_1803 | 21-02-2011 | 16-01-2018 |
| CT15Mn-150818_101018 | 15-08-2018 | 10-10-2018 |
| CT15Mn-111018-150419 | 11-10-2018 | 15-04-2019 |

**Table 3.1:** Description of the available sensor data provided by Lisbon's City Council.

However, to extend this model to a real-world application, a real-time database would be required, given that future traffic flow conditions heavily depend on the current status of traffic flow.

To understand what kind of information was available, all the data points were integrated into a single dataset. Each row of the data set corresponds to a full day of traffic data, following the structure shown in Fig. 3.2. The first column contains the date of the data point, the second shows the zone in which the sensor is located, the third specifies the identifier of the sensor. Each of the following 96 (24h × 4 data points/hour) columns contains the number of cars that passed through this sensor in intervals of 15 minutes.

| Date | Zone | Counter | SensorID | 0h00 | 0h15 | 0h30 | 0h45 | 1h00 |
|---|---|---|---|---|---|---|---|---|
| 1/09/2018 | 2 | 1_CTs | ct1 | 117 | 320 | 103 | 95 | 75 |

**Table 3.2:** Structure of the loop sensor data

The raw data consists of a collection of text files where each file contains the daily traffic flow data, aggregated in intervals of 15 minutes, of one of the predefined zones. The first task that had to be done was the integration of these data points in a single dataset in order to facilitate data access and enable a detailed analysis of the available data.

### 3.1.2 Data Analysis

Through a simple python script, the raw data is transformed in a single Comma-separated values (CSV) file where each row contains the date, zone, sensor id, and the corresponding daily traffic flow counts. This script also identifies which files are broken or empty. A file is considered broken if any information is not present, such as date, zone or sensor id. To achieve this, the script creates two logs in the form of plain text files.

Each zone has multiple sensors and they can be identified by a sensor id that is unique in the context of the zone but not in the global scope of the city. To unquestionably identify the sensors, a unique id was created using the pair (zone, sensor id). For example "4_ct3" refers to the loop sensor with id 3 in zone 4. These zones and the corresponding sensors are mapped in an data sheet that was made available by the city council and which contains the coordinates of each sensor. This data sheet allowed us to create the map of sensors presented in **Fig.**3.1. However, given that there is no indication or additional information on the orientation of the sensor it is somewhat difficult to accurately identify the direction of traffic flow that each sensor is measuring. This makes it more challenging to find confluent sensors that might be correlated and could improve the prediction results.

Through the comparison of the data points with the mapping of the sensors, it was possible to conclude that there are no counts for some of the sensors that were mapped initially. This could mean that the sensors are malfunctioning or that they were deactivated and it is an indicator that the mapping that was made available to us might not be up to date. To develop a deeper understanding of the data, a statistical analysis was performed. The output of this analysis allowed us to identify which are the non-mapped sensors and which were mapped initially but do not exist anymore in more recent data as well as identify the average counts for each sensor per time of day.

Globally, analyzing the three chunks of data as a whole, it was possible to find out that only two of the selected zones had all of the mapped sensors working at least once. All the remaining zones had more sensors than they were supposed to, according to the aforementioned map of the sensors as shown in **Fig.**3.2. Currently, there are 47 sensors for which we do not know the exact location since they are not on the map. With the help of this new information it was found that some of the data seemed inconsistent, some sensors showed counts of over five thousand cars in the span of 15 minutes which amounts to over 5 cars per second. Keeping in mind that these sensors are placed in an urban context and are mainly near stoplights, which for a considerable amount of time remain closed, not allowing any cars through, it seems unlikely. There is no information regarding the number of lanes that are affected by the sensors. However, using the example described above, even if the sensor was placed in a road with 5 lanes, it still amounts to 1 car per second, which does not seem feasible. Furthermore, some of these extreme events happen in the middle of the night, which strongly suggests that they are erroneous.

The plot being shown in Fig.3.3 is an example of one of these extreme scenarios. The blue line

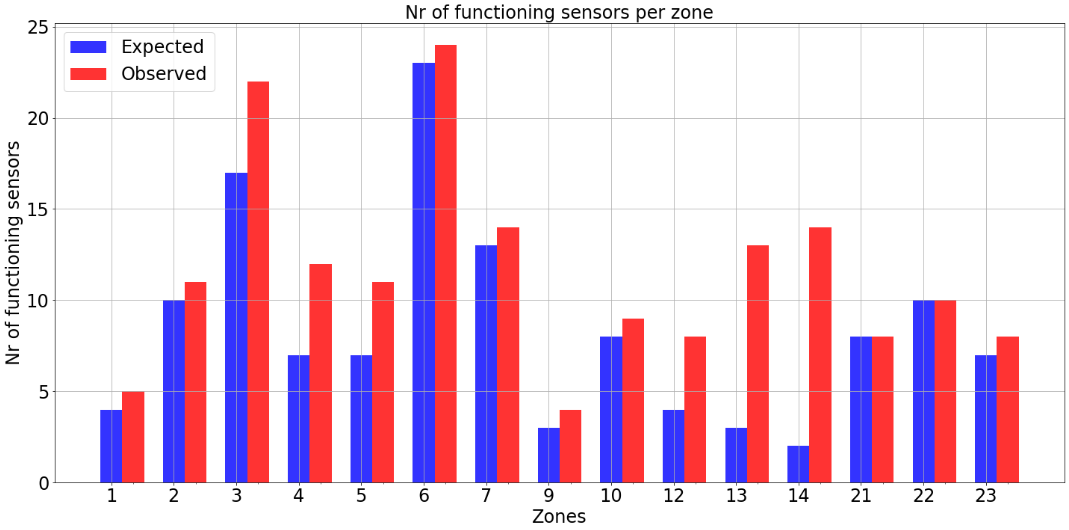


**Figure 3.2:** Number of sensors that have at least one day worth of data per zone.

shows the traffic flow counts measured by the sensor throughout Monday, 3$^{rd}$ of December 2012 while the red line shows the average traffic flow recorded by the same sensor on Mondays. We opted to present the average flow of the same business day since, as it was mentioned in **Chapter** 2, although traffic tends to have a seasonality of 24h, it also shows similar patterns for the same business days. Just through the analysis of the plot, it is clear that the data is very noisy and that there are some samples in which traffic flow is much larger than the average flow. This particular sensor is located at a very important street in the city center that has 3 lanes. Even if the sensor is measuring traffic flow for the 3 lanes, these values are disproportional according to the average flow in this same location.

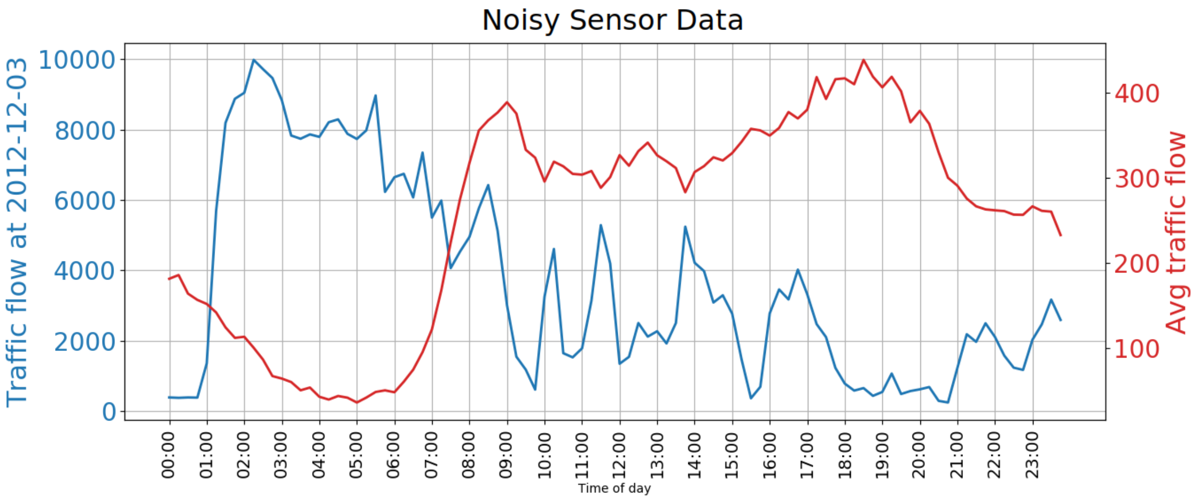


**Figure 3.3:** Example of a malfunctioning sensor.

### 3.1.3 Outlier Identification

When working with real data such as traffic data, it is very common to have sporadic missing values or errors in some data records. Through a quick run of a simple python script, a few of these missing or broken files were found. Several possible reasons could explain this phenomenon, such as, the occurrence of an error in the transmission of the data from the sensor to the central database or even random objects that cause interference with the sensors, causing it to malfunction temporarily. Consequently, these occurrences have to be dealt with. Imputation is a very commonly used method in machine learning and consists of, basically, filling in the missing data according to a predefined technique.

There are multiple imputation techniques for dealing with missing or erroneous data, studied in several papers such as [33, 34]. Even though there are some missing values, the main challenge on the data is the detection of outliers since if they exist in abundance they can hurt badly our predictions. In the scope of this project, outliers are identified with a heuristic based on the average and standard deviation. More specifically, if $\mu$(T) represents the average flow at the time of day T and $\sigma$(T) represents the standard deviation of the flow at the time of day, then X is an outlier if at a specific time of day (T) it is greater than the average plus two times the standard deviation at the same time of day (see Equation 3.1).

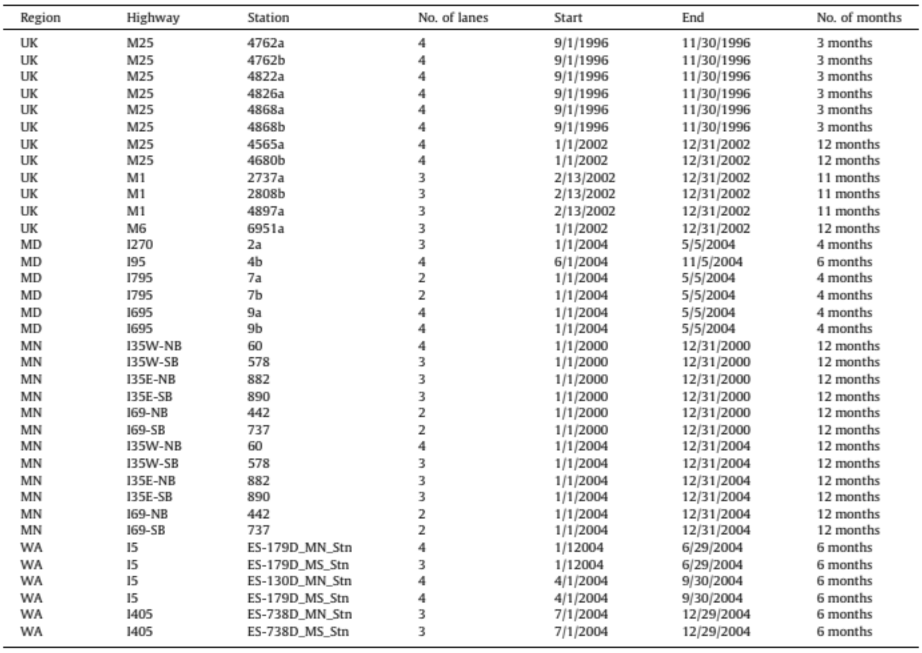
$$X > \mu(T) + 2 \times \sigma(T) \tag{3.1}$$

When an outlier is detected, the next step is to decide how to replace it for a more reasonable value. In this case, the heuristic that was adopted replaces the value at time T with the average values of the previous (T-1) and next time step (T+1).

## 3.2 Freeway Data

As it was previously mentioned in **Chapter 2**, most of the published work on traffic flow prediction uses freeway data as the primary data source and the results are encouraging. However, in our case, a decrease in the performance of the models is expected since our main data source is obtained via urban sensors and since an urban environment is considerably less stable than a freeway environment. Moreover, our sensors are placed in stoplights near important junctions, which make them especially prone to unexpected behavior. Nevertheless, it would be important to have a more stable and tested dataset to test the developed models and to compare their performance with the results from published papers.

To achieve this, we considered as reference the data that was used in Guo et al. [35]. In order to have access to the source data we contacted the authors, who kindly agreed to send us the available data.



| Region | Highway | Station | No. of lanes | Start | End | No. of months |
|---|---|---|---|---|---|---|
| UK | M25 | 4762a | 4 | 9/1/1996 | 11/30/1996 | 3 months |
| UK | M25 | 4762b | 4 | 9/1/1996 | 11/30/1996 | 3 months |
| UK | M25 | 4822a | 4 | 9/1/1996 | 11/30/1996 | 3 months |
| UK | M25 | 4826a | 4 | 9/1/1996 | 11/30/1996 | 3 months |
| UK | M25 | 4868a | 4 | 9/1/1996 | 11/30/1996 | 3 months |
| UK | M25 | 4868b | 4 | 9/1/1996 | 11/30/1996 | 3 months |
| UK | M25 | 4565a | 4 | 1/1/2002 | 12/31/2002 | 12 months |
| UK | M25 | 4680b | 4 | 1/1/2002 | 12/31/2002 | 12 months |
| UK | M1 | 2737a | 3 | 2/13/2002 | 12/31/2002 | 11 months |
| UK | M1 | 2808b | 3 | 2/13/2002 | 12/31/2002 | 11 months |
| UK | M1 | 4897a | 3 | 2/13/2002 | 12/31/2002 | 11 months |
| UK | M6 | 6951a | 3 | 1/1/2002 | 12/31/2002 | 12 months |
| MD | I270 | 2a | 3 | 1/1/2004 | 5/5/2004 | 4 months |
| MD | I95 | 4b | 4 | 6/1/2004 | 11/5/2004 | 6 months |
| MD | I795 | 7a | 2 | 1/1/2004 | 5/5/2004 | 4 months |
| MD | I795 | 7b | 2 | 1/1/2004 | 5/5/2004 | 4 months |
| MD | I695 | 9a | 4 | 1/1/2004 | 5/5/2004 | 4 months |
| MD | I695 | 9b | 4 | 1/1/2004 | 5/5/2004 | 4 months |
| MN | I35W-NB | 60 | 4 | 1/1/2000 | 12/31/2000 | 12 months |
| MN | I35W-SB | 578 | 3 | 1/1/2000 | 12/31/2000 | 12 months |
| MN | I35E-NB | 882 | 3 | 1/1/2000 | 12/31/2000 | 12 months |
| MN | I35E-SB | 890 | 3 | 1/1/2000 | 12/31/2000 | 12 months |
| MN | I69-NB | 442 | 2 | 1/1/2000 | 12/31/2000 | 12 months |
| MN | I69-SB | 737 | 2 | 1/1/2000 | 12/31/2000 | 12 months |
| MN | I35W-NB | 60 | 4 | 1/1/2004 | 12/31/2004 | 12 months |
| MN | I35W-SB | 578 | 3 | 1/1/2004 | 12/31/2004 | 12 months |
| MN | I35E-NB | 882 | 3 | 1/1/2004 | 12/31/2004 | 12 months |
| MN | I35E-SB | 890 | 3 | 1/1/2004 | 12/31/2004 | 12 months |
| MN | I69-NB | 442 | 2 | 1/1/2004 | 12/31/2004 | 12 months |
| MN | I69-SB | 737 | 2 | 1/1/2004 | 12/31/2004 | 12 months |
| WA | I5 | ES-179D_MN_Stn | 4 | 1/12004 | 6/29/2004 | 6 months |
| WA | I5 | ES-179D_MS_Stn | 3 | 1/12004 | 6/29/2004 | 6 months |
| WA | I5 | ES-130D_MN_Stn | 4 | 4/1/2004 | 9/30/2004 | 6 months |
| WA | I5 | ES-179D_MS_Stn | 4 | 4/1/2004 | 9/30/2004 | 6 months |
| WA | I405 | ES-738D_MN_Stn | 3 | 7/1/2004 | 12/29/2004 | 6 months |
| WA | I405 | ES-738D_MS_Stn | 3 | 7/1/2004 | 12/29/2004 | 6 months |

**Figure 3.4:** Brief description of the freeway datasets (from [4]).

This data is a collection of multiple datasets from different regions in the United States and in the United Kingdom as displayed in **Fig** 3.4. Each of these datasets contains traffic flow information collected by a single sensor aggregated in intervals of 15 minutes. The sensors measure the number of cars that passed through them and each traffic flow record reflects the average flow by lane.

### 3.2.1 Data Analysis

In order to have a better understanding of the data and to begin the pre-processing needed to use it in the prediction models, the data was integrated into 36 datasets, one for each freeway. Naturally, this data source also had some missing values that had to be addressed. Statistical analysis was performed in order to identify the percentage of missing values in each dataset and, more importantly, to deal with these occurrences. It was found that the missing values in each dataset were not statistically significant, which is a good indicator of the quality of the collected data, given that the percentage of the missing values is in the range of 0% to 6%.

In order to deal with these occurrences, whenever a missing value is found, it is replaced by an average between the previous and the next value. After the imputation process, a more thorough analysis of the data was performed. As it is shown in Fig. 3.5 there seems to be a clear difference between the flow on business days and traffic flow on weekends. This is a phenomenon that was mentioned in Chapter 2 and it is a good indicator on the robustness of the data at hand.

Through the analysis of Fig. 3.5 it can be seen that on the first plot there is a clear peak of traffic

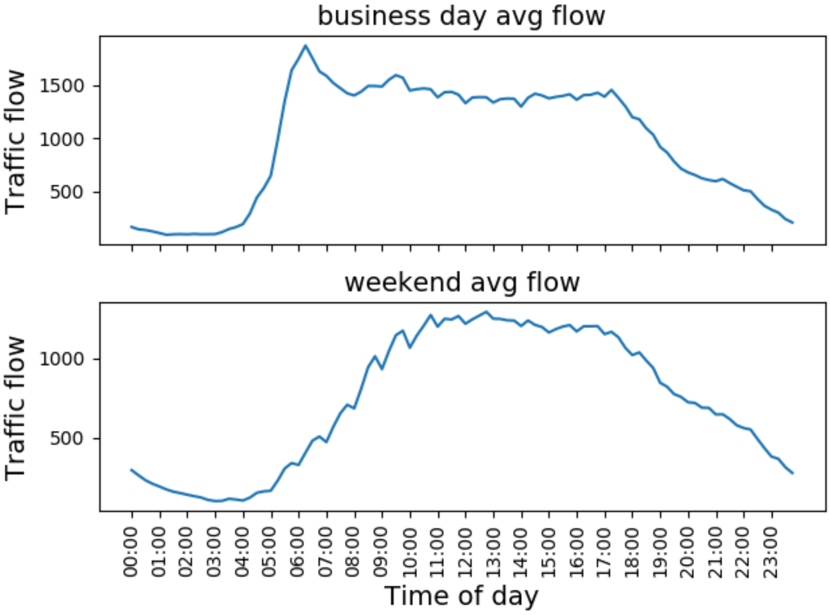


**Figure 3.5:** Average flow on weekends versus average flow on business days.

flow in the morning, at about 6 am, that can be what is usually called rush hour, whereas on the second plot the traffic flow follows a much more subtle curve throughout the day and it only peaks at 12 pm. Moreover, there is another, smaller, peak on the first plot at 6 pm, that can be identified as the evening rush hour, when people leave their jobs and return home.

## 3.3 Waze

CML has also developed a partnership with Waze where the latter provides access to real-time traffic flow information and the former provides information regarding city-mandated road activities such as strangulated roads, blocked streets or construction work. This could be an interesting secondary source to improve the effectiveness of our model by correlating the data from the road sensors with Waze traffic data.

The typology of this data is rather different from the sensor data. Each data point indicates that in a given location there was a disturbance in the normal traffic flow, meaning that there is no scheduled updating window for this data. Each of these occurrences is called a snap, and each of them includes information about the date and time of the occurrence, its location, the current average speed and the level of congestion.

There are some challenges in using the Waze data:

- The locations of the occurrences do not match exactly with the sensors' locations, given that these locations are determined by a few pairs of coordinates to indicate not only the location but also the direction of the traffic it refers to. Therefore, a mapping of these occurrences with the location of the nearest sensor has to be implemented.

- Since the data is not continuous, it is challenging to use it in a prediction model.

Considering that our main goal is to supply valuable information to the city to prevent or, at least, reduce traffic jams in the most critical areas, we decided not to use the Waze data. This decision was due mainly to the nature of the data given that, as it was previously mentioned, it does not measure traffic flow regularly, it measures occurrences. Given that Waze is one of the most widely used navigation applications and that it works considerably well on route selection using prediction methods to avoid congestion, they may possess data with a different granularity, which would be useful for this project.

However, this data source should not be discarded since it can be used as an additional source in future projects, to try and improve existing, stable, models.

## 3.4 TomTom Traffic API

In addition to the aforementioned available data, a new possible source of data was discovered. Through some research, it was found that TomTom supplies traffic flow data through an API. TomTom is a Dutch company that provides mapping, traffic, and navigation products. TomTom navigation systems can be found in several car brands, is one of the most popular companies for this type of system. According to [36], TomTom relies on large volumes of anonymous probe data to generate traffic flow data.

GPS probe data is collected from users that accepted to share their travel information anonymously. Incident data from journalistic data and other sources are also used to complement the probe data to increase the accuracy of the traffic information. Real-Time Traffic information can be requested by supplying the coordinates of the selected road. As a response to this request, estimates of the current/expected average travel speed, the current/expected average travel time in that stretch of road are going to be provided. Several coordinates are also returned in order to describe the shape of the road segment.

Given that this data is real-time, the data has to be collected through a simple script in order for it to be useful for this study. However, there were some challenges in matching the data from TomTom and the Lisbon Sensor data for two main reasons. Firstly, since TomTom limits the data requests per day, we could not request data for all of the locations of the sensors. Therefore, a few test locations had to be chosen in order to not exceed the daily request limits. In order to maximize the results, research had to be done to establish which sensors were the best candidates for this test. Secondly, after these locations were found, data with matching dates were still needed. This was proven to be a difficult task due to the limited timetable of this project and since we rely on Lisbon's City Council to provide the data.

Even though this source was not used in this project, it must not be discarded as a useful data source for future projects. Due to the large amount of traffic data that TomTom possesses, this could be a truly important complement to validate the main data sources and to improve the performance of the predictions.

# 4

# Predictive Analysis

## Contents

This chapter describes and discusses the prediction results. A comparative analysis of the different models is presented.

## 4.1 Freeway Data

The freeway traffic data source was the first one to be tested in order to act as a result benchmark for this project, given that the results can be compared to other published papers on the subject and, perhaps more importantly because freeways are much more stable environments than city centers and therefore more predictable. As it was described in Section 3.2, specifically in Fig. 3.4, the freeway dataset includes information from multiple freeways with varying duration. However, these datasets contain, at most, one year worth of traffic data, which means that we will not be able to fully capture the underlying yearly seasonality of this dataset, such as yearly events like the increased traffic flow around Christmas time or the decrease on traffic flow on Summer. In spite of this, the available data enables us to capture the daily seasonality of traffic flow, which should allow us to predict, with a fair degree of certainty, the immediate future of traffic flow.

### 4.1.1 Time-series forecasting

As it was previously discussed in Section 2.3 the first attempts on traffic flow prediction were based on statistical methods such as auto-regression and moving average. In the context of this project, a Seasonal ARIMA model (SARIMA) was developed in order to predict traffic flow. This model was created using as software the R language and, more specifically, the forecast package available on the R 3.4.4 distribution.

Firstly, the data is loaded into the program and it is transformed into a time-series format, which is a requirement to build the model.

The data already had the necessary structure for the development of the SARIMA model, the only pre-processing needed was the conversion of the date into a DateTime format. The dataset follows the structure described in Table 4.1.

| Date | Count |
|---|---|
| 2002-01-01 00:00:00 | 35.99 |
| 2002-01-01 00:15:00 | 37.99 |
| 2002-01-01 00:30:00 | 60.99 |
| 2002-01-01 00:45:00 | 99.99 |
| 2002-01-01 01:00:00 | 133.99 |

**Table 4.1:** Structure of data for ARIMA model

Accordingly, to develop the first SARIMA model, one of the freeways with the largest amount of data

(12 months) was chosen. The data was then split into two sets, training, and a test set. The train-test split was roughly 80%-20% meaning that the first 10 months of data were used as the training set and the remaining 2 months as the testing set. To find, in an automatic manner, the best parameters for the model, the function **auto.arima** from the Forecast package in R was used. This is a function that searches through multiple parameter settings and selects the one that has the best results.

The SARIMA setting that was found by the algorithm to be the fittest was (3,0,2)x(0,1,0)96. Which means that the model has a auto correlation coefficient (p) equal to 3. This means that the model uses the 3 most recent past observations to produce each prediction. The moving average coefficient (d) is equal to 2. The seasonal period (S) of the SARIMA is 96, in order to capture the daily seasonality, where 96 represents the amount of timesteps that it takes for the patterns to repeat themselves.

After the optimal setting was found, the next step was to predict and test the accuracy of the predictions, measured by the Mean Average Precision Error (MAPE).

However, comparing with other data-driven approaches on short-term prediction such as Neural Networks, these methods have a few limitations. In other methods, each entry of the test set contains actual real data to calculate the prediction. On time-series forecasting, the scenario is rather different since, if one wanted to predict the full test set based only on the training set, the predictions after a few timesteps would be based on other predictions, increasing the error. Therefore, to avoid this issue, walk-forward validation was applied. This method was applied using two different approaches.

The first approach consists in predicting values one step at a time, which translates to 15 minutes in this case and re-feeding the model after each evaluation, the next row of the test set. The diagram showed in Fig. 4.1 illustrates how walk-forward validation works. In our case, the rectangles named

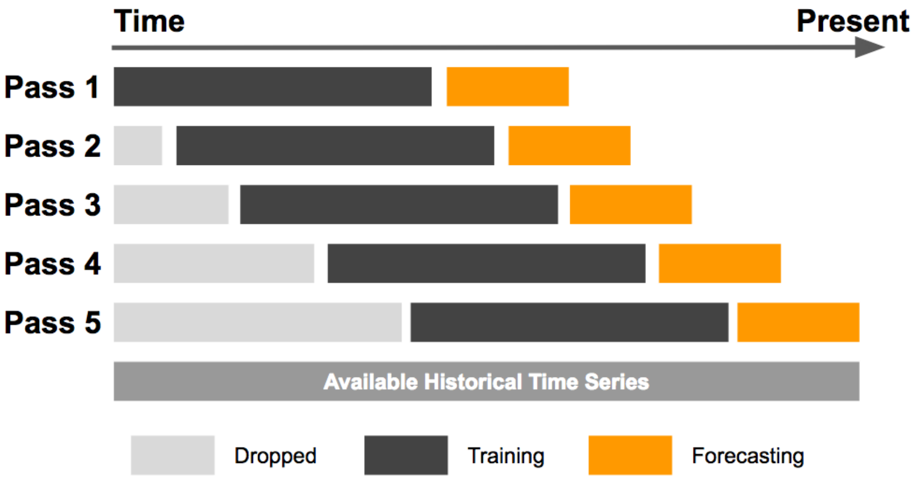


**Figure 4.1:** Illustrative diagram of the walk-forward validation. Taken from [5]

”Forecasting” are 15-minute predictions and this process is repeated 96 times to have an entire day of predictions, given that there are 4 data-points per hour and 24h*4 = 96 data-points.

This method is, however, extremely costly, given that to do 96 passes of walk-forward validation it took over 7 hours on an Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz with 8Gb of RAM. Due to the high computational cost of this methodology, it may be currently too costly to use it on a real-world application. The main advantage of this method is that it produces promising results, with an average MAPE of 2.72% when tested with a full day's worth of data. Plus, due to the long computation time, this methodology was not tested for more days, which means that these results may not be a good indicator of the overall performance of this method if it was tested for the entire test set.

The other approach for evaluating these models consists of fitting the model with a set of training data and re-feeding it with new observations, within a wider period of time. The model could predict an entire day's worth of traffic flow and re-fed with new observations once every 24h. With this method, the performance of the model performed slightly worse, with an average MAPE of 5.39% when tested with the same test set of one day that was used on the walk-forward validation. On the other hand, this process took less than 4 minutes. However, to test the stability of this methodology, this process was repeated for 14 days, meaning that the model was re-fed 14 times. The average error that resulted from this experiment was much higher, achieving an average MAPE of 19.79%.

### 4.1.2 Neural Network

After thorough research on related work it became clear that (Section 4.1.2) the state-the-art models were based on deep learning the next model to be developed was a feed-forward neural network. Roughly 80% of the dataset was used as a training set while the remaining 20% were used as the testing set. The model was tested using the Python language and, more specifically, using the Keras library, which is a powerful machine learning library implemented in Python.

The model is composed of one input layer, 2 hidden layers, each with 64 neurons, and an output layer. The activation function of the input and hidden layers is a Rectified Linear Function Unit (RELU) which is one of the most widely used activation functions in machine learning at the moment. The optimization algorithm Adam was used to update the network's weights according to the training data.

In the first runs of the model, the validation loss was considerably higher than the training loss, suggesting that the model was overfitted so, in order to address this problem, weight regularization was applied to avoid having large weight values. In addition to this, early stopping was also used in the training. Early stopping is a method to fight overfitting that consists of monitoring the evolution of the validation loss and stopping the training if there is a clear degradation of the performance of the model on the validation set. The number of epochs used in the model was found empirically by testing multiple scenarios and deciding, taking into consideration the ratio between computational time and the training loss value. After several test runs to get the best number of epochs, it was found that 600 epochs was a reasonable value given that, according to Fig 4.2, at that point the training loss seems to have reached a stable state. Showing only a slight decrease in the mean squared error (loss function) until it reaches

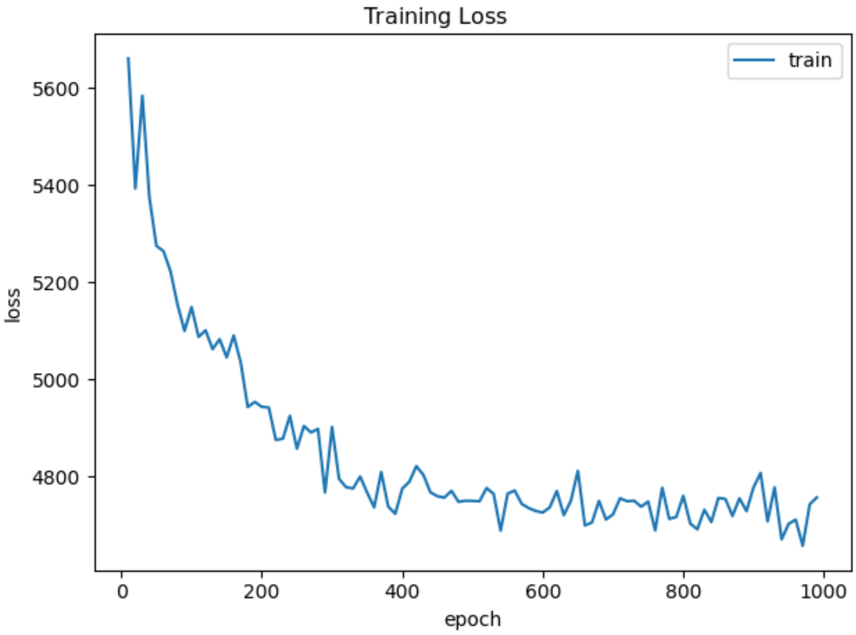


**Figure 4.2:** Evolution of training loss by number of epochs

1000 epochs and given that it takes a considerable amount of computation time it does not seem worth it to increase the number of epochs any further.

As was mentioned before, traffic flow usually shows a strong dependency on the recent past and most researchers in this field agree that this dependency starts to fade after one hour. Therefore, to test this hypothesis, the network was tested with multiple time windows, where tw represents the number of past traffic flow measurements considered in the prediction. For example, if the past hour is used in the prediction then tw = 4 (15 min aggregation). The plot being shown in Fig. 4.3 shows the variation of the mean average percentage error with the increase in the input size. In this test we started with tw= 1, meaning that only the last 15 minutes were considered in the model, and ended with tw= 10, which means that the past 2h30 of traffic flow measurements were used. It is also visible that the error decreases almost in a linear fashion from tw=1 to tw=3, which was expected according to the literature. This plot is relevant because it demonstrates empirically that the threshold of temporal dependency of traffic flow starts to fade after 1h, which is visible since from tw=4 to tw=9 the variation in the prediction error is close to zero (approximately 0.5%). This suggests that if a larger window were to be used then it would mainly add irrelevant data that may work as a source of noise for prediction purposes.

In order to evaluate the performance of the developed model three metrics were used as described in Section 2.1.2, with emphasis on MAPE since it is relative and it enables a more effective comparison between different datasets.

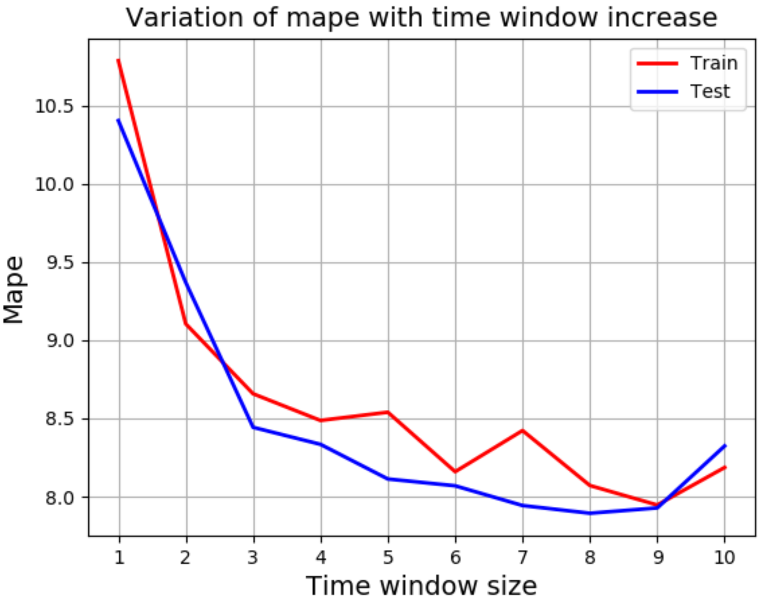


**Figure 4.3:** Mape variation with the increase in the time window

To evaluate the performance of the model, one freeway was selected as the target. The mean MAPE for 15-minute prediction was approximately 8%, which seems consistent with the results of some of the methods presented in Section 4.1.2.

Even though the raw data in our possession shows a 15-minute granularity and most of the literature on traffic flow prediction suggests that this is the most commonly used prediction horizon, our model is prepared to predict with different time windows. To test the changes in the model's performance, we ran our network model with prediction horizons of 15, 30, 45 and 60 minutes.

| | MAPE | MAE | RMSE |
|---|---|---|---|
| 15-min prediction | 8.13% | 46.51 | 74.11 |
| 30-min prediction | 8.96% | 106.99 | 169.31 |
| 45-min prediction | 11.16% | 184.64 | 314.17 |
| 60-min prediction | 12.60% | 285.37 | 461.20 |

**Table 4.2:** Metric results for every prediction horizon

As it was expected, the error increased slightly as the prediction horizon was larger, as it is shown in Table 4.2. It is also visible that there is not a large difference in the error between 15-min prediction and 30-min prediction, showing an increase of only 0.8%. The same can not be said for the 45-min and 60-min prediction, which showed a steeper growth. This can be explained by the fact that traffic flow is somewhat unstable and although it is possible to predict accurately the future of traffic flow in a short-term fashion (15/30 minutes), if you expand the prediction horizon, the error will increase rapidly.

### 4.1.3 Long Short-Term Memory Network

As it was previously mentioned, traffic flow shows a strong temporal dependency on the recent past of the state of traffic flow. In an attempt to better capture these dependencies traffic prediction was tested using a LSTM.

As before, the model was tested using Python 3.7.3 and, more specifically, using the Keras library. To adequately compare the performance of this method versus the previous ones, the data was divided into two different sets, in the same way as the others. The training set was composed of 80% of the data and the testing set by the remaining 20%. In an attempt to avoid overfitting of the network, 20% of training set was used as a validation set. This network is composed by one input layer, one hidden layer with 32 units, and one output layer.

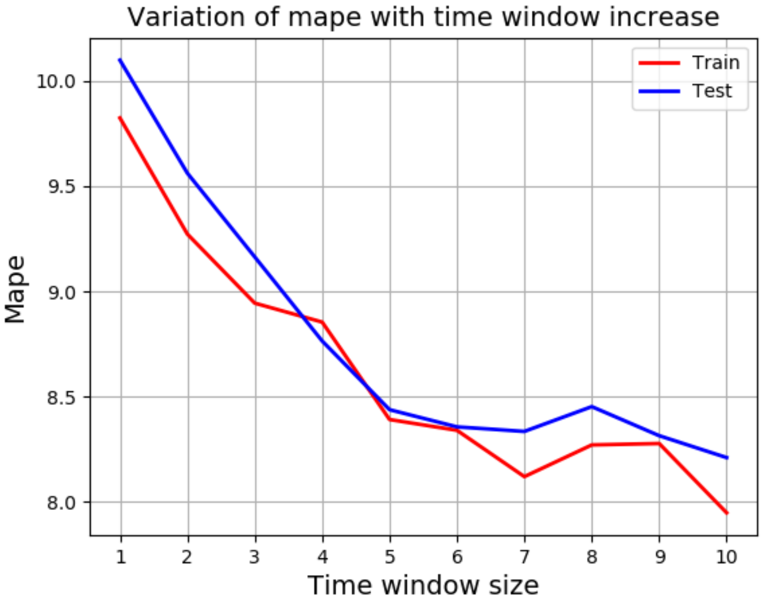


**Figure 4.4:** Mape variation with the increase in the time window

The first parameter of the network that was tweaked was the size of the input layer or, in other terms, the number of past timesteps that were being considered in the predictions. Much like in the feed-forward network, due to the characteristics of traffic data, it is expected an improvement in the model's performance as the number of timesteps that are fed as input increases. Moreover, this phenomenon should be even more noticeable due to the ability of this network to capture temporal dependencies. The plot being shown in Fig. 4.4 shows the variation of the error as the input size increases. The results represent an average of 5 runs. There is a clear, and somewhat steep, decrease in the prediction error from tw=1 to tw=5, which is a similar behavior as the one shown by the feed-forward network (see Fig. 4.3). However, from tw=8 to tw=10, the error, that was relatively stable before, diminished even further.

Taking into consideration that the literature suggests that using a time window larger than 1h30 is unstable and could add more noise to the models, even though the error, in this case, decreased slightly from tw=8 to tw=10 (2h/2h30), the input size that will be used for the next results will be 5 (1h15 of past observations).

Although the default setting of the model is 15-minute predictions, it is also prepared to predict traffic flow with different horizons. In this case, we tested the network with four different prediction horizons, namely, 15, 30, 45 and 60 minutes. The values being shown in Table. 4.3 show decrease in prediction power as the prediction horizon increases. These values were obtained by averaging 5 runs of the model, using the same train and test sets. Similarly to what happened on the feed-forward network, the prediction power of the LSTM decreases as the prediction horizon is larger. The average MAPE for 15 and 30-min prediction is very similar, with a difference of less than 1%. There is a more clear increase in error when predicting the next 45 minutes, showing an increase of 2% relatively to the 30-minute prediction.

| | MAPE | MAE | RMSE |
|---|---|---|---|
| 15-min prediction | 7.94% | 46.18 | 73.49 |
| 30-min prediction | 8.87% | 106.62 | 172.54 |
| 45-min prediction | 11.54% | 206.20 | 327.21 |
| 60-min prediction | 14.35% | 337.51 | 519.57 |

**Table 4.3:** Metric results for every prediction horizon

## 4.2 Lisbon Sensor Data

As it was previously mentioned, one of the main goals of this project was to test the predictability of traffic flow in urban areas. Urban areas are more unstable environments meaning that accurately predicting the future of traffic flow should be somewhat more challenging than in a freeway environment. Accordingly, in this section, all the previously developed models are going to be applied to the Lisbon Sensor Data with the needed adaptations.

### 4.2.1 Time-Series Forecasting

The first challenge of this task was altering the structure of the data to transform it into a time-series format. As it was described in Section 3.1.1, each row of this data set corresponded to an entire day of traffic flow for a given sensor and, to be able to develop an ARIMA model for this data, it had to be converted in the format described in table 4.1.

As it was previously mentioned, we possess loop sensor data dating back from 2011. For example, for sensor (4_ct4) there are over 2000 days of traffic flow information and, given that each day contains

96 data points this amounts to over 190000 (96 data points x 2000 days=190000) data points. Keeping in mind that the freeway dataset used in the previous section had approximately 35000 data points and that the process of testing the ARIMA was computationally costly it would be unreasonable to simply split the data into a training and testing set, so one year of data was used to train the model and test the model. The train-test split was 80%-20%, respectively.

Another challenge that we encountered was the amount of missing data on this data set. There are entire months absent from the data set, which makes it unreasonable to simply substitute these missing values according to a heuristic. Furthermore, in time-series forecasting methods such as the SARIMA model, the sequence of the data is important, so that the model can capture the underlying temporal dependencies. Therefore, the first year of data was used since it is the one that seems to have fewer missing days. This increase in missing data throughout the years could indicate decay in the condition of the sensors.

After all these issues were sorted, the next step was to find the best SARIMA configuration for the training data. This was done using the **auto.arima** function in R. The SARIMA setting that was found by the algorithm to be the fittest was ARIMA(5,0,1)x(0,1,0)96. This means that we have an ARIMA model with an auto-regression coefficient (p) equal to 5. In practical terms, this means that to predict one timestep, the past 5 observations are considered, each with a given weight. This value is also higher than the one found by the same method in the freeway data section (see Section 4.1.1), which could indicate that in an urban environment, the repercussions of traffic flow events affect future traffic flow on a wider period of time than in freeway environments.

The testing procedure was the same as the one applied to the freeway data and it is composed of two scenarios, both based on walk-forward validation.

The first test scenario consisted of re-feeding a new observation to the model at each timestep and repeating this process for a duration of a full day, which means that this process was performed 96 times. Table. 4.4 demonstrates the performance metric values achieved by this experiment.

|  | MAPE | MAE | RMSE |
|---|---|---|---|
| 1-step prediction | **27.19%** | 7.09 | 9.60 |
| 96-step prediction | **28.7%** | 8.20 | 11.20 |

**Table 4.4:** Metric results obtained by the SARIMA model.

The MAPE values presented in the table were calculated using only the observations different than 0. The first row of the table corresponds to the application of the walk-forward validation for 96 iterations, with one-step predictions (15-min) per iteration. The last entry on the table corresponds to the application of a similar methodology but instead of 15-min predictions, the model predicted an entire day of traffic flow. This process was repeated for test 14 days and the values are the average of the values obtained for each of these days.

The error was higher in the scenario where the model is only updated once every day, as it was expected. However, to be able to accurately conclude that this model performs well in the long term, it should be tested more extensively.

### 4.2.2 Neural Network

Similarly to what happened in Section 4.1.2, the first step in order to develop a neural network model for the urban data was to import the data and to transform it into a format more suitable for the neural network. Unlike the freeway data, the urban data showed a rather different structure, as it was described in Section 3.1. Also, as was previously mentioned, the data seemed somewhat noisy, showing multiple inexplicably high traffic flow volumes. Given that these peaks are not going to be ”learned” by the network and are going to hurt the performance of the model, a smoothing function was applied to the data. The model was also developed in Python 3.7.3, with the Keras library.

The base model is composed of one input layer, 2 hidden layers, and one output layer. The activation function chosen for the hidden layers was the Rectified Linear Function Unit (RELU).

Adam was the chosen algorithm for updating the network's weights, since, currently, it is the most widely used in problems such as the one at hand.

In order to avoid overfitting, weight regularization and early stopping methods were applied. The process of choosing the most suitable number of epochs consisted of training the network for a varying number of epochs and evaluating the loss values and the time it took to compute it. After several runs, it was found that 450 was a reasonable number of epochs given that, according to Fig. 4.5, after that point there seems to be no meaningful improvement on the loss value. Furthermore, in this case, it looks like the network struggled to learn the underlying dependencies in the data, given that the loss drastically decreased at first but then it remained almost constant in the following epochs.

In order to try to capture the temporal dependencies that are present in traffic flow data, the size of the time lag was tweaked and the network was trained with multiple input sizes. A time window (tw) in this scenario corresponds to the number of past observations that are used to predict future traffic flow. Given that our data has a 15-minute granularity, each time lag corresponds to 15 additional minutes of past observations. For example, tw = 1 would mean that 15 minutes of past observations were used and tw= 4 would mean 60 minutes of past observations (4 x 15 min = 60).

The plot being shown in Fig. 4.6 shows the variation of the mean average percentage error in the increase in the size of the time window. In this test we started with tw = 1, meaning the only the past 15 minutes were used and it ended with tw = 10. By analyzing the plot in Figure. 4.6 it is visible that the number of past observations that minimizes the relative error is 9 (2h15 of past observations). As it was previously mentioned, literature in traffic flow prediction suggests that the temporal dependencies

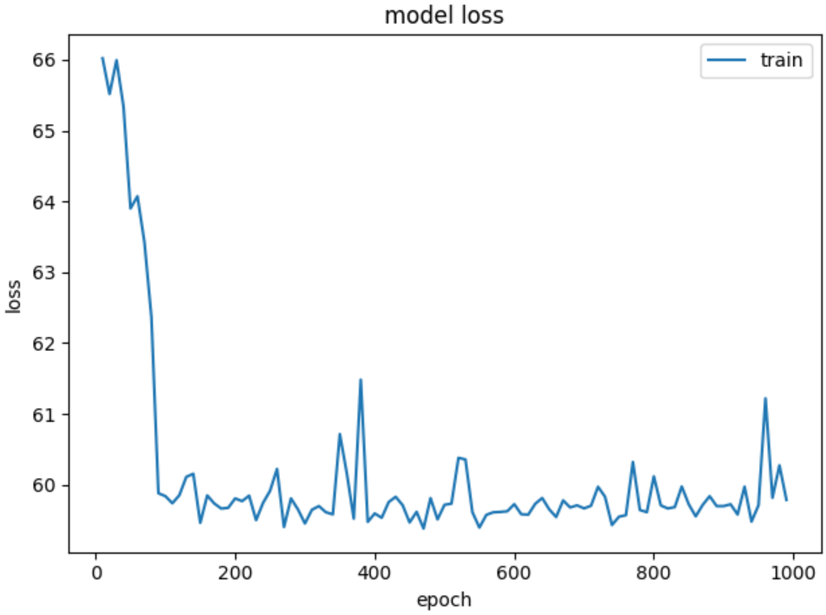


**Figure 4.5:** Evolution of training loss by number of epochs.

start to fade after 1 hour. However, these studies mainly refer to freeway traffic patterns and since this is an urban environment, this may not apply directly. Moreover, since the data is rather noisy and full of peaks, a larger number of past observations may help the network in the predictions in cases where one or more of the past observations are peaks. The values present in this plot are an average of 5 test runs, performed in the same conditions.

After the optimal value for the number of time lags was found, the next, and final task was to test the performance of the model for multiple prediction horizons. The network was tested for four different prediction horizons, specifically, 15, 30, 45 and 60 minutes. Table. 4.5 contains the prediction results for all the previously mentioned prediction horizons. The MAPE values were calculated only for the non zero observations, given that the domain of MAPE is $D = \{\mathbb{R}, observation \neq 0\}$.

To be able to produce predictions for larger time intervals, the granularity of the data was altered. These results were achieved by averaging 5 runs of the model for each prediction horizon. As it was expected, there is a clear difference in the accuracy of the predictions when predicting for shorter periods (15 and 30 minutes) versus predicting longer periods. However, this difference in prediction accuracy was not exactly as expected. Usually, as the prediction horizon increases, so does the error, however, since the data shows a large variance, the increase in prediction horizon, as it was implemented, may have acted as a smoothing method. This might be the reason why the relative error decreased as the prediction horizon increased.

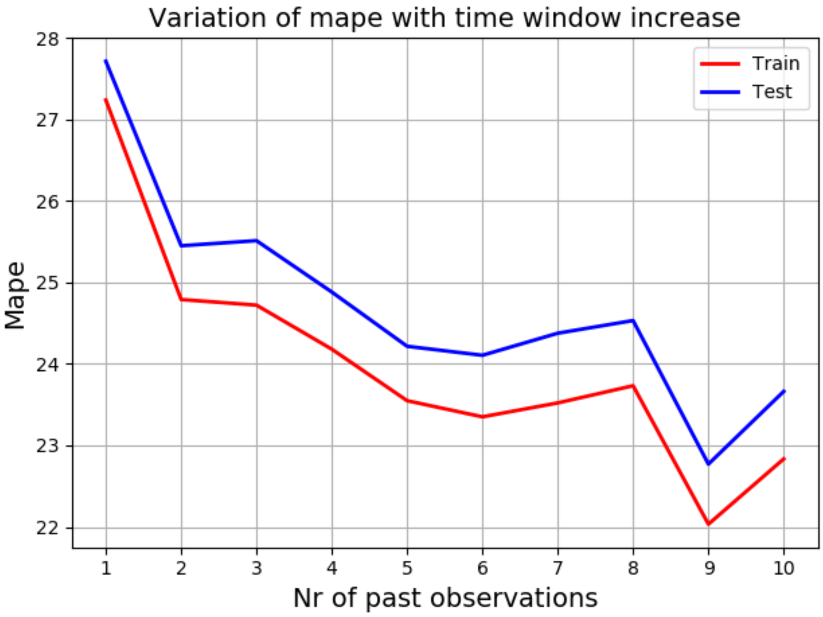


**Figure 4.6:** Variation of the relative error according to the number of past observations used in the predictions.

| | MAE | MAPE | RMSE |
|---|---|---|---|
| 15-min prediction | 5.60 | **23.33%** | 7.72 |
| 30-min prediction | 10.10 | **18.28%** | 14.56 |
| 45-min prediction | 15.70 | **15.70%** | 23.77 |
| 60-min prediction | 20.92 | **16.24%** | 31.35 |

**Table 4.5:** Metric results through different prediction horizons

### 4.2.3 Long Short-Term Memory Network

To try to capture the temporal dependencies present in the data, a LSTM network was implemented following a procedure similar to the one described in Section 4.1.3. The train-test split was 80%-20%, respectively. 20% of the training set was used as a validation set during training. In order to avoid overfitting early stopping was applied, monitoring the validation loss. The network has the same structure as the one developed for the freeway data, with one input layer, one hidden layer with 32 units, and one output layer with a single neuron.

The first test that was performed with this model was the variation of the size of the input layer or, in other terms, the number of past observations being considered in the predictions. The plot is shown in Fig. 4.7 shows the variation of the relative error as the number of past occurrences used in the predictions increases. Through the analysis of the graph, it is visible that the error suffers from a steep decrease from tw=1 to tw=2 and then it decreases, in a more gradual manner from tw=4 to tw=10. Similarly to what happened on the same test performed on the feed-forward neural network, the number of past occurrences that produce the most accurate predictions is considerably higher than the one used

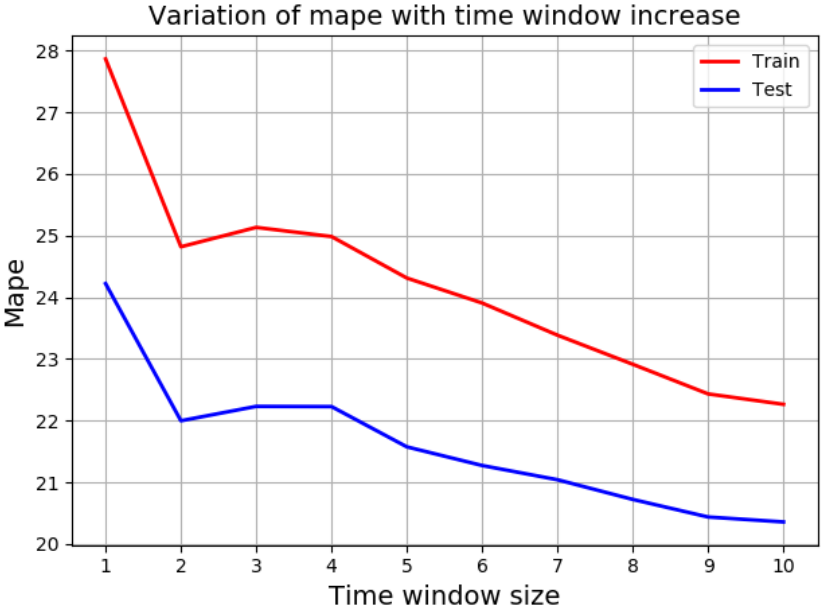


**Figure 4.7:** Variation of the relative error according to the number of past observations used in the predictions.

in the freeway dataset.

After the optimal input size was found, the next step was to test the network for multiple prediction horizons. The tested prediction horizons were the same as in the previous tests, 15, 30, 45 and 60 minutes. Table. 4.6 contains the results of the performance metrics all the prediction horizons.

| | MAE | MAPE | RMSE |
|---|---|---|---|
| 15-min prediction | 6.84 | **20.90%** | 9.81 |
| 30-min prediction | 14.11 | **16.81%** | 23.11 |
| 45-min prediction | 24.02 | **16.09%** | 40.30 |
| 60-min prediction | 35.64 | **17.10%** | 58.76 |

**Table 4.6:** Metric results through different prediction horizons

The MAPE results that are highlighted in the table reflect the prediction error for the observations that are different from 0. Similarly to what happened on the feed-forward network, there was a decrease in the prediction error as the prediction horizon increased. Again, this could be due to the topology of the data, given that it has a large number of peaks. It is also noticeable that the MAE and RMSE increase as the prediction horizon is larger, which is expected given that the scale of the data increases with each prediction horizon.

## 4.3 Result Analysis

As it was expected, all the models performed much better on the freeway data than on the Lisbon dataset. This validates the application of these methods to the prediction of future traffic flow. In the following paragraphs, the emphasis will be on discussing the results of the models that were developed for the Lisbon Sensor Data, given that it was the main motivation behind this work.

As it was previously mentioned, the more classical models, in this case, the SARIMA model, performed relatively well, achieving a mean absolute error of 7.09, which is slightly worse than the performance of the deep learning based methods. However, this was achieved by performing just 96 iterations of walk-forward validation, to safely assume that this would be a stable method it should be tested for a longer period. This was tested just for a single day due to the amount of time that it takes to run.

The deep learning approach in this project consisted of the development of two different networks, a feed-forward neural network, and a long short-term memory network. The performance of both methods was similar, however, the LSTM performed slightly worse in all of the prediction horizons. Table. 4.7 shows the prediction results for the Lisbon Sensor Data achieved by the deep learning based methods. After a thorough analysis of the results, I realized that MAPE, although it is a useful tool to compare the performance of the different models it shows some limitations. In the urban dataset, at night, the traffic flow counts are extremely small, usually lower than 10 cars per 15 minutes. Let's say that, for example, the real count is 2 and the network predicted 6, this would mean a MAE of 4 and a MAPE of **200%**. Therefore, this could be another reason why the MAPE values are considerably higher in the urban data versus the freeway data.

| | LSTM | | | Feed-Forward NN | | |
|---|---|---|---|---|---|---|
| | MAE | MAPE | RMSE | MAE | MAPE | RMSE |
| 15-min prediction | 6.84 | **20.90%** | 9.81 | 5.60 | **23.33%** | 7.72 |
| 30-min prediction | 14.11 | **16.81%** | 23.11 | 10.11 | **18.28%** | 14.56 |
| 45-min prediction | 24.02 | **16.09%** | 40.30 | 15.70 | **15.70%** | 23.77 |
| 60-min prediction | 35.64 | **17.10%** | 58.76 | 20.92 | **16.24%** | 31.35 |

**Table 4.7:** Performance metrics results of the deep learning methods for sensor 4_ct4.

Keeping in mind the main goal of this project, which is to develop a prediction mechanism that would supply crucial information about future possible traffic flow congestion in order to try to avoid them, it is relevant to emphasize the importance of the MAE and RMSE.

To test the stability of the developed models and, given that the aforementioned results use only one sensor, two additional sensors were tested. Keeping in mind the limitations of the MAPE metric and that these sensors might have different scales of traffic volume, the average flow of the each test set is also shown. This will help evaluate the performance of the models. Tables 4.8 and 4.9 show the metrics results of both networks when tested with new, unseen data for each prediction horizon.

|  |  | Feed-Forward NN |  |  | LSTM NN |  |  |
|---|---|---|---|---|---|---|---|
| Sensor ID : 4_ct2 | AVG Flow | MAPE | MAE | RMSE | MAPE | MAE | RMSE |
| 15-min prediction | 50.53 | 22.41% | 7.59 | 10.38 | 21.79% | 7.58 | 10.36 |
| 30-min prediction | 103.65 | 18.63% | 13.63 | 18.92 | 20.06% | 13.77 | 18.87 |
| 45-min prediction | 160.33 | 19.01% | 20.55 | 28.70 | 18.81% | 20.97 | 28.77 |
| 60-min prediction | 221.62 | 18.08% | 27.98 | 39.57 | 16.82% | 29.23 | 40.25 |

**Table 4.8:** Prediction results for sensor 4_ct2.

|  |  | Feed-Forward NN |  |  | LSTM NN |  |  |
|---|---|---|---|---|---|---|---|
| Sensor ID : 21_ct23 | AVG Flow | MAPE | MAE | RMSE | MAPE | MAE | RMSE |
| 15-min prediction | 130.11 | 17.70% | 14.02 | 19.76 | 19.14% | 14.62 | 20.37 |
| 30-min prediction | 269.48 | 15.10% | 28.22 | 40.21 | 20.92% | 30.78 | 42.57 |
| 45-min prediction | 421.15 | 14.57% | 43.36 | 61.35 | 22.41% | 45.39 | 63.64 |
| 60-min prediction | 587.25 | 18.89% | 62.78 | 90.39 | 22.21% | 77.96 | 106.72 |

**Table 4.9:** Prediction results for sensor 21_ct23.

It is clear that sensor 21_ct23 measures a much higher volume of traffic flow, which is the main reason behind the inclusion of the average flow information in the table. For this reason, we cannot establish a direct relation for the accuracy of the models for each sensor. However, considering the average flow as the scale of each sensor, it seems that the MAE in both cases is reasonable, given that it represents approximately 10% of the average flow on the feed-forward NN and 11% on the LSTM NN. Table 4.8 and Table 4.9 show that there was virtually no difference in the performance of both models.

# 5

# Conclusion

## Contents

This chapter discusses the main results and contains a comparative analysis between all of them as well as some conclusions that we were able to draw during the development of the models. Finally, it presents possible lines of future work on this topic.

## 5.1 Discussion

The main purpose of this project was to test the feasibility of the application of a short-term prediction mechanism in an urban context. In order to achieve this, a data set of Lisbon Sensor Data was used. To benchmark the results that would validate the models, a freeway dataset was also used. As it would be expected, all the developed models achieved better results on the freeway data. This can be explained by two main reasons.

First, as it was previously mentioned, an urban environment is much more unstable than a freeway environment, which as itself alone could explain the decrease in the performance of the models. The second reason that could explain the large error in the predictions is the large variance and noisy values that are present in almost all data sensors. There are so many noisy values that some of them are incredibly difficult to be identified as outliers.

Therefore, one of the conclusions that can be drawn from this work is that, in order to implement a working traffic flow prediction system, there should be an investment towards more accurate traffic flow data collection.

## 5.2 Future Work

An interesting experiment that could be an extension of this project could be the inclusion of other urban data sources such as Waze and TomTom Data to test the difference in the models' performance. Another extension that could be implemented is the development of other types of algorithm, such as K-NN based methods, just as the one developed by Filmon G. Habtemichael and Mecit Cetin [19] since it seemed to achieve promising results when tested with freeway data. Finally, it would be interesting to explore the spatial dependencies of traffic data such as the influence that traffic conditions on arterial roads might have on other roads int the city. This might be an important addition to improve the performance of the developed algorithms on urban traffic prediction.

As the world's population keeps growing and cities become more crowded, the topic of urban mobility is becoming increasingly relevant. Through technological advances, new traffic sensor technologies are emerging, making the amount of traffic data increase exponentially as we enter the era of big data in transportation systems. Traffic management and control are becoming increasingly data-driven, creating a renewed interest in the field of traffic flow prediction.

The main purpose of this work was the development and application of state-of-the-art methods in traffic flow prediction to establish the practical foundations that will allow the implementation of a working real-world prediction model in an urban environment.

# Bibliography

[1] M. Nguyen, "Illustrated guide to lstm's and gru's: A step by step explanation," accessed: 2019-10-10. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

[2] W. Yuankai, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, 05 2018.

[3] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal arima model with limited input data," *European Transport Research Review*, vol. 7, no. 3, p. 21, 2015.

[4] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61 – 78, 2016.

[5] R. Yang, "Omphalos, uber's parallel and language-extensible time series backtesting tool," accessed: 2019-10-14. [Online]. Available: https://eng.uber.com/omphalos

[6] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[7] M. Roser, "Future population growth," 2017. [Online]. Available: https://ourworldindata.org/future-population-growth

[8] B. Bhatta, "Causes and consequences of urban growth and sprawl," in *Analysis of urban growth and sprawl from remote sensing data*. Springer, 2010, pp. 17–36.

[9] J. Zhang, F. Wang, K. Wang, W. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[10] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information Sciences*, vol. 275, pp. 314 – 347, 2014.

[11] G. Leduc, "Road traffic data: Collection methods and applications," *Working Papers on Energy, Transport and Climate Change*, vol. 1, no. 55, 2008.

[12] R. Adhikari and R. K. Agrawal, "An introductory study on time series modeling and forecasting," *arXiv preprint arXiv:1302.6613*, 2013.

[13] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[14] K. W. Hipel and A. I. McLeod, *Time series modelling of water resources and environmental systems*. Elsevier, 1994, vol. 45.

[15] S. Bisgaard and M. Kulahci, *Time series analysis and forecasting by example*. John Wiley & Sons, 2011.

[16] P. S. Kalekar, "Time series forecasting using holt-winters exponential smoothing," *Kanwal Rekhi School of Information Technology*, vol. 4329008, pp. 1–13, 2004.

[17] C. de Fabritiis, R. Ragona, and G. Valenti, "Traffic estimation and prediction based on real time floating car data," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 197 – 203, 11 2008.

[18] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[19] J. Raj, H. Bahuleyan, and L. D. Vanajakshi, "Application of data mining techniques for traffic density estimation and prediction," *Transportation Research Procedia*, vol. 17, pp. 321–330, 2016.

[20] T. S. Institute, "Number of road motor vehicles registered to the traffic during the year by classification of statistical region units level 1," accessed: 2018-12-15. [Online]. Available: http://www.turkstat.gov.tr

[21] B. G. Çetiner, M. Sari, and O. Borat, "A neural network based traffic-flow prediction model," *Mathematical and Computational Applications*, vol. 15, no. 2, pp. 269–278, 2010.

[22] C. Harlow and S. Peng, "Automatic vehicle classification system with range sensors," *Transportation Research Part C: Emerging Technologies*, vol. 9, pp. 231–247, 08 2001.

[23] S. Du, T. Li, X. Gong, Z. Yu, and S.-J. Horng, "A hybrid method for traffic flow forecasting using multimodal deep learning," *arXiv preprint arXiv:1803.02099*, 2018.

[24] S. Yang, S. Shi, X. Hu, and M. Wang, “Spatiotemporal context awareness for urban traffic modeling and prediction: Sparse representation based variable selection,” *PLOS ONE*, vol. 10, no. 10, pp. 1–22, 10 2015.

[25] A. Ermagun, S. Chatterjee, and D. Levinson, “Using temporal detrending to observe the spatial correlation of traffic,” *PLOS ONE*, vol. 12, no. 5, pp. 1–21, 05 2017.

[26] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” *CoRR*, 2018.

[27] P. J. Brockwell, R. A. Davis, and M. V. Calder, *Introduction to time series and forecasting*. Springer, 2002, vol. 2.

[28] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, “An improved k-nearest neighbor model for short-term traffic flow prediction,” *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[29] D. Lien and N. Balakrishnan, “On regression analysis with data cleaning via trimming, winsorization, and dichotomization,” *Communications in Statistics - Simulation and Computation*, vol. 34, no. 4, pp. 839–849, 2005.

[30] R. Chambers, P. Kokic, P. Smith, and M. Cruddas, “Winsorization for identifying and treating outliers in business surveys,” in *Proceedings of the Second International Conference on Establishment Surveys*. American Statistical Association Alexandria, Virginia, 2000, pp. 717–726.

[31] Nortech Detection, “Single channel inductive loop traffic detector - TD136,” accessed: 2018-12-18. [Online]. Available: https://nortechdetection.com.au/wp-content/uploads/2014/12/TD136_ds.pdf

[32] I. Teknik, “Inductive loops,” accessed: 2018-12-18. [Online]. Available: https://www.its-teknik.dk//CustomerData/Files/Folders/13-datablade/183_datablad-inductive-loops-cs6.pdf

[33] J. Scheffer, “Dealing with missing data,” 2002.

[34] F. M. Shrive, H. Stuart, H. Quan, and W. A. Ghali, “Dealing with missing data in a multi-question depression scale: a comparison of imputation methods,” *BMC Medical Research Methodology*, vol. 6, no. 1, p. 57, Dec 2006.

[35] J. Guo, W. Huang, and B. M. Williams, “Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50 – 64, 2014, special Issue on Short-term Traffic Flow Forecasting. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X14000382

[36] TomTom, “Tomtom real time traffic information,” accessed: 2018-12-27. [Online]. Available: https://www.tomtom.com/lib/img/REAL_TIME_TRAFFIC_WHITEPAPER.pdf