Traffic Analysis and Prediction in Urban Areas

Vasco Leal
Instituto Superior Técnico
Lisbon, Portugal
vasco.leal@tecnico.ulisboa.pt

***Abstract*—As the world's population continues to increase, cities get exponentially more crowded, which means new problems arise. This phenomenon results in the aggravation of road traffic in most of Europe's capital cities, making the mobility of passengers not yet efficient. One way to improve this issue is to deploy an Intelligent Transportation System (ITS) which is an application that aims to provide accurate 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 database is required. Motivated by this necessity, the Lisbon City Council is developing an effort to collect and provide mobility data. In the context of this project, the main source of road data will be from loop counters placed at intersections throughout the center of Lisbon as well as other publicly available data from major traffic monitoring operators. The main goal of this project is to develop traffic predictive models, from the available data, in order to understand which produces the most accurate results.**

***Index Terms*—Short-term Traffic Flow Prediction. Machine Learning. Time-Series Forecasting. Spatio-Temporal Traffic Patterns.**

I. Introduction

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 [1] claiming that the rate at which population is growing is decreasing, the world's population is still growing immensely. According to this study, the UN projections for the human population show that it should surpass the 10 Billion mark by 2055. 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.

A. 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 [2], [3], 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 decreasing in recent years. Therefore, there is a real need to develop strategies that will improve traffic flow.

B. Goals

The main objective of this project 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 in 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. 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.

C. Organization

This document describes the work done during the school year of 2018/2019 in order to achieve the goals that were set for the master thesis in the first semester. Section II contains the basic concepts on which these methods are based on. Section III surveys literature on the most commonly used approaches for short-term traffic flow prediction. In Section IV, a thorough description and analysis of the data is presented. Section V 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. Lastly, Section VII contains an overview on the obtained results and a few final remarks about the work that was done during these past months as well as some considerations about future work that could be done to extend this project.

II. Background Research

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.

A. Data Collection Techniques

According to Guillaume Leduc [4], 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.

FCD techniques rely on collecting real-time traffic information by locating the vehicle through mobile phones or GPS. 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.

B. 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:

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

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

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}|pred_i - obs_i| \tag{3}$$

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.

C. 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**). These weights measure the relative influence between two neurons. 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 consists in minimizing an error function. 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. Each weight is then changed according to the derivative's rate. The main objective is to minimize the error function, therefore, the error will be at its lowest when the derivative is zero. If the derivative rate is positive then it means that an increase in the weight will increase the error, so this weight should be smaller. Inversely, if the derivative rate is negative, an increase in the weight should decrease the error. The training of a Neural Network is an iterative process and given that the adjustments in the weights are very small, it may need several iterations in order to converge. 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.

D. 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, don't only take as its input the current input example but also the previous outputs of the network.

This means that in an RNN, the output 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.** 1. The forget gate is used by the network to control

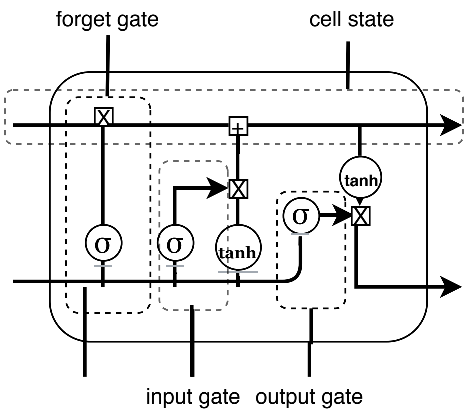

Fig. 1. Diagram of a LSTM unit, adapted from [5]

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.

E. 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.

F. Stochastic time-series models

According to [6], 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. 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.

The most commonly used linear time-series models are Autoregressive (AR) and Moving Average (MA), presented in [7], [8]. 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 [9] is given by:

$$w_t = \phi_1 w_{t-1} + \phi_2 w_{t-2} + ... + \phi_p w_{t-p} + a_t \tag{4}$$

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. 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. 6

$$x'_t = (x_{t-1} + x_{t-2} + ... + x_{t-p})/p \tag{5}$$

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

$$w_t = a_t + \theta_1 a_{t-1} + \theta_2 a_{t-2} + ... + \theta_q a_{t-q} \tag{6}$$

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 [10]). Although ARIMA methods are the most popular in time-series forecasting, exponential based methods such as the Holt-Winter models [11] have also shown to be useful prediction techniques.

III. Related Work

There are two main approaches on traffic flow prediction. The most classical methods, such as time-series forecasting, that tried to accomplish this were based on statistical methods. ARIMA is the most commonly used approach within this family of methods. The other methodology is the development of Deep Learning based models such as Neural Networks in order to predict the future of traffic flow. This section contains a summary of a few of the main contributions in the field of traffic flow prediction.

A. Time-Series Forecasting

The work developed by S. Vasantha Kumar and Lelitha Vanajakshi [10] addresses the problem of limited input data in Traffic Flow prediction by proposing a traffic prediction system using a seasonal ARIMA model (SARIMA). 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) 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 parameters necessary for the application of the SARIMA were found using the maximum likelihood method [12]. After the developing part, the model was validated by performing 24h ahead forecast and comparing the actual values with the predicted ones. Fig. 2 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.

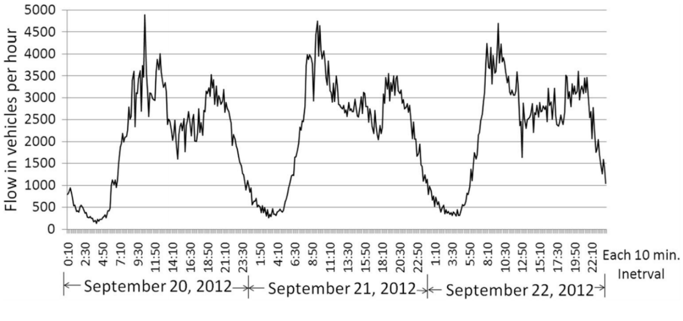

Fig. 2. Time-series data of observed traffic flow in three consecutive days (from [10]). ScenarioID represents the number of days that were used as historical data for the prediction.

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. 3 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. 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.

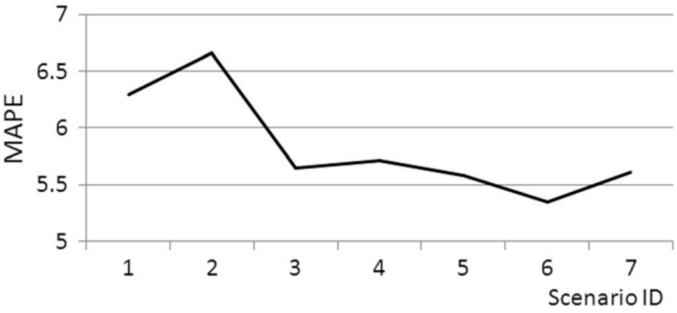

Fig. 3. Variation of the MAPE across the different scenarios (From [10])

B. Traffic prediction with Neural Networks

The study presented in [13] 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.

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 usually not 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. 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. 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 [14]).

From the observation of Table I 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. There are many other deep learning based methods that also achieved promising such as Corrado de Fabritiis Et al. [15], [16], B. Gültekin Çetiner Et al. [17], Yuankai Wu Et al. [18] and Yaguang Li Et al. [19].

	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 I
PERFORMANCE OF THE TRAFFIC PREDICTION MODEL PROPOSED IN [13]

IV. Data Sources

The focus of this section will be on the description of the available data sources and their structure.

A. 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 the city. Back in the 80's, traffic flow optimization was already a pressing concern for Lisbon's city council and 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 in order 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 road sensors.

These sensors are installed in the pavement, usually near stoplights or important junctions. Each and 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, meaning that, 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. In order 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 an entire day of traffic data, following the structure shown in Fig. II. 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 $\times$ 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
1/09/2018	2	1_CTs	ct1	117	320	103	95

TABLE II
STRUCTURE OF THE LOOP SENSOR DATA

The raw data consists in 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, so that a more thorough analysis of the data could be performed.

Each zone has multiple sensors and they can be identified by a sensor id which is unique in the context of the zone but not in the global scope of the city. In order to unquestionably identify the sensors an unique id was created using the pair (zone, sensor id). These zones and the corresponding sensors are mapped in an Excel file that was made available by the city council and it contains the coordinates of each sensor. However, given that there is no indication or additional information on the orientation of the sensor it can be 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.

In order 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.

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. Furthermore, some of these extreme events happen in the middle of the night, which strongly suggests that they are in fact erroneous.

The plot shown in Fig. 4 is an example of one of these extreme scenarios. The blue line 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. Just through the analysis of the plot it is clear that the data is very noisy and that there are some instants where the 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.

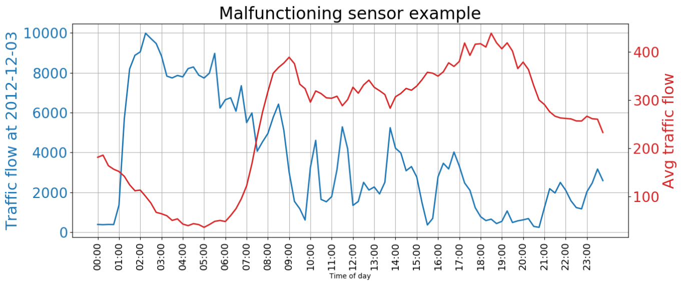

Fig. 4. Example of a malfunctioning sensor.

When working with real data such as traffic data, it is extremely common to have sporadic missing values or errors in some data records. Several possible reasons 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 numerous articles such as [20], [21]. 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, concretely if a value (X) at a specific time of day (T) is greater than the average plus two times the standard deviation at the same time of day then X is considered an outlier. As shown in **Equation 7** where μ(T) represents the average flow at time of day T and σ(T) represents the standard deviation of the flow at time of day T.

$$X > \mu(T) + 2 \times \sigma(T) \tag{7}$$

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 used consists in averaging the value of the current time step (T) with values of the previous (T-1) and next time step (T+1).

B. Freeway Data

Most of the work performed in traffic flow prediction uses freeway data as the primary data source and the results are encouraging. In order to be able to validate the developed models, a more stable and tested dataset is needed.

Accordingly, I asked Drs. Guo and Williams if they could provide the data that was used in Guo et al. [22] which they promptly and kindly agreed. This data is a collection of multiple datasets from different regions in the United States and in the United Kingdom. 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.

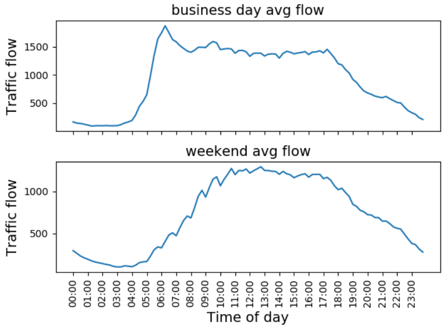

Fig. 5. Average flow on weekends versus average flow on business days.

Naturally, this data source also had some missing values that had to be addressed. A 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 on the quality of the collected data, given that the missing values percentage sits in the range between 0% and 6%.

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 on the data was performed and, as it is shown in Fig. 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 III and it is a good indicator on the robustness of the data at hand.

Through the analysis of Fig. 5 it can be seen that on the first plot there is a clear peak of traffic flow in the morning, at about 6 am, that can be what is usually called rush hour, where as 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. This is relevant because it shows that traffic flow follows the patterns that were described in Section III and that that traffic flow is not a completely stochastic phenomenon, making it feasible to be predicted.

V. Predictive Analysis

In this section the focus will be on describing the developed models throughout the past months and on discussing the prediction results in order to draw conclusions on which methods obtain the best performance for the task at hand.

A. 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.

1) Time-series forecasting: As it was previously discussed in Section III-A 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. The data was split into two sets, a training and a test set with an 80%-20% ratio, respectively.

In order to find, in an automatic manner, the best parameters for the model, the function **auto.arima** from the Forecast package in R was used. The SARIMA setting that was found by the algorithm to be the most fit was (3,0,2)x(0,1,0)96. 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).

This model was tested in two different scenarios using walk-forward validation. 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. Due to the high computational cost of this approach, another scenario was tested where the model was only re-fed once every 24h. The first approach achieved largely better results, achieving a MAPE of 2.72% for a test set of one day. The second approach achieved worse results, since it consisted in predicting one day at a time, with an average MAPE of 19.79%.

2) Neural Network: After the data was fully integrated and the missing values were dealt with, and given that after a thorough research on related work (Chapter III), it seemed that the state-the-art models were based on deep learning the next model to be developed was a feed-forward neural network.

The base model is composed by one input layer, 2 hidden layers 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 in order to update the networks 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 weight regularization was applied.

In addition to this, early stopping was also used in the training. Early stopping consists in 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 plot shown in Fig. 6 shows the variation of the mean average percentage error with the increase in the number of past observations used in the predictions (time window size). In this test we started with tw = 1, meaning that only the past 15 minutes were considered in the model, and ended with tl = 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 tl=1 to tl=3, which it was expected according to the literature on traffic prediction. In order to evaluate the performance of the developed model MAPE was used. With the objective of evaluating the performance of the model, one freeway was selected as the target. Concretely, in the case of this simple feed-forward neural network the results for 15 min prediction achieved an average MAPE of 10.81% on the training set and 8.18% on the testing set.

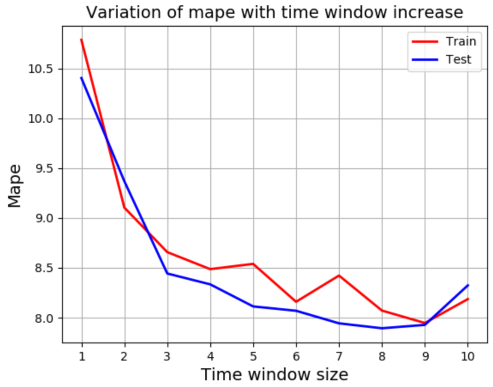

Fig. 6. Mape variation with the increase in the time window

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. As it was expected, the error increased slightly as the prediction horizon was larger, as is displayed in Table III. 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 0.45% in the training set and

	MAPE (Train)	MAPE (Test)
15-min prediction	10.81%	8.18%
30-min prediction	10.45%	8.58%
45-min prediction	12.00%	10.69%
60-min prediction	13.33%	12.99%

TABLE III
MAPE variation with the increase in the prediction horizon

0.35% in the testing set. 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 the prediction horizon is expanded, the error will increase rapidly.

3) Long Short-Term Memory Network: In an attempt to better capture the temporal dependencies of traffic flow, a Long Short-Term Memory Network was developed with the data in our possession.

In order to be able to adequately compare the performance of this method versus the previously mentioned methods, the data was divided into two different sets, in the same way as the others. Concretely, the training set is composed by 80% of the data and the testing set by the remaining 20%. In an attempt to avoid overfitting of the network, there is also a validation set composed by 20% of the training set. This network is composed by one input layer, one hidden layer and one output layer.

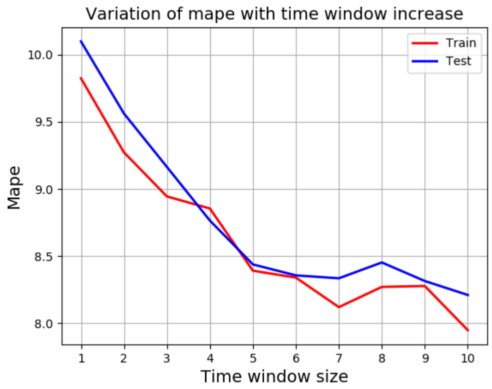

Fig. 7. Mape variation with the increase in the time window

The plot shown in Fig. 7 shows the variation of the error as the input size increases and it represents the average of 5 runs in order to improve accuracy. There is a clear, and somewhat steep, decrease in the prediction error from tw=1 to tw=5, which is a similar behaviour as the one showed by the feed-forward network (see Fig. 6). However, from tw=8 to tw=10, the error, that was relatively stable before, diminished even further.

The values shown in Table. IV show a decrease in prediction power as the prediction horizon increases. 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. However, there was a much steeper increase in the error in the 60-min prediction, reaching 17.4%.

	MAPE (Train)	MAPE (Test)
15-min prediction	10.02%	8.48%
30-min prediction	10.16%	9.54%
45-min prediction	12.10%	11.95%
60-min prediction	19.47%	17.40%

TABLE IV
MAPE VARIATION WITH THE INCREASE IN THE PREDICTION HORIZON

B. Lisbon Sensor Data

One of the main goals of this project was to test the predictability of traffic flow in urban areas. Accordingly, in this section all the previously developed models are going to be applied to the Lisbon Sensor Data with the needed adaptations.

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. Due to the large amount of data and to the inefficiency showed by the ARIMA model in the previous section, only one year of data was used to train and test the model.

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 most fit was ARIMA(5,0,1)x(0,1,0)96. This means that we have an ARIMA model with a auto-regression coefficient (p) equal to 5. In practical terms this means that in order to predict one timestep ahead, 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 V-A1), which could indicate that in an urban environment, the repercussions of traffic flow events affect future traffic flow on a wider time span than in freeway environments.

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

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

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

TABLE V
METRIC RESULTS OBTAINED BY THE SARIMA MODEL.

2) Neural Network: Unlike the freeway data, the urban the data is much morevnoisy, 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 base model is composed by 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) and ADAM was the chosen optimizer.

In order to avoid overfitting, weight regularization and early stopping methods were applied. The plot shown in Fig. 8 shows the variation of the mean average percentage error in the increase in the size of the time window. By analysing the plot in Figure. 8 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 start to fade after 1 hour. However, theses 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.

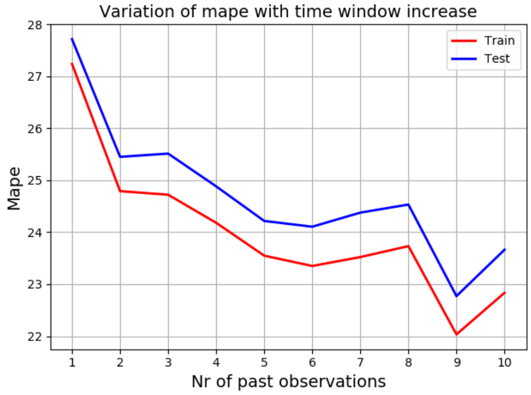

Fig. 8. Variation of the relative error according to the number of past observations used in the predictions.

3) Long Short-Term Memory Network: In order to try to capture the temporal dependencies present in the data, a LSTM network was developed and the procedure was similar to the one described in Section V-A3. The train-test split was 80%-20%, respectively. In order to avoid overfitting, 20% of the training set was used as a validation set during training. Still on the topic of overfitting, early stopping was also applied, monitoring the validation loss. The network has the same structure as the one developed for the freeway data.

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 shown in Fig. 9 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 timelags that produce the most accurate predictions is considerably higher than the one used in the freeway dataset.

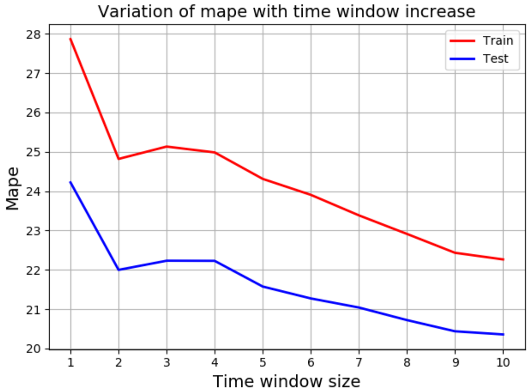

Fig. 9. Variation of the relative error according to the number of past observations used in the predictions.

VI. Result Analysis

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, in order to safely assume that this would be a stable method it should be tested for longer periods of time. This was tested just for a single day due to the large amount of time it takes to run.

The deep learning approach in this project consisted in 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 better in predicting within shorter time spans, as it is shown by the mean MAPE values for the 15 and 30-minute predictions. However, for larger time spans both networks performed similarly, achieving significantly close relative errors. Both showed the same tendency to perform better as the prediction horizon grew larger. This can be explained by the amount of noise present in the data and due to the fact that changing the granularity of the data may have acted as a smoothing method. Table. VI 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. Lets say that,

	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 VI

PERFORMANCE METRICS RESULTS OF THE DEEP LEARNING METHODS.

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.

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, perhaps it would not be unreasonable to emphasize the importance of the MAE. As it is shown on Table. VI, the MAE for both methods is around 6 cars per 15 minutes, which seems like a reasonable error when predicting high volume traffic flow.

VII. Discussion

The main purpose of this project was to test the feasibility of the application of a short-term prediction mechanism on an urban context. The developed methods achieved better results on the freeway data. This can be explained by two main reasons. First, an urban environment is much more unstable than a freeway environment, which as itself alone is a reason for 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 extreme values that are present in almost all of the sensors. 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.

VIII. 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. The development of other types of algorithm, such as K-NN based methods, as the one developed by Filmon G. Habtemichael and Mecit Cetin [16] would be interesting since it achieved promising results when tested with freeway data. Finally, the exploration of 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.

References

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

[2] 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.

[3] 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.

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

[5] 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

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

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

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

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

[10] 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.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] J. Scheffer, "Dealing with missing data," 2002.

[21] 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.

[22] 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