FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Empirical Evaluation of Prediction Models for Smart City Applications

Maria Helena Viegas Oliveira Ferreira

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Dr. Ana Paula Rocha

Second Supervisor: Dr. Rosaldo José Fernandes Rossetti

July 31, 2022

Empirical Evaluation of Prediction Models for Smart City Applications

Maria Helena Viegas Oliveira Ferreira

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Jorge Alves da Silva
External Examiner: Prof. Pedro Campos
Supervisor: Prof. Ana Paula Rocha

July 31, 2022

Abstract

Prediction of traffic in cities is a problem within the scope of smart cities, which aims to minimize costs and improve the performance of cities and the lives of their citizens. Predicting traffic is not a trivial task as it deals with active citizens. Tourist pressure, the pandemic, and the increasingly ever-changing daily lives of citizens have made this challenge even more complex.

The problem addressed belongs to the traffic flow prediction type that corresponds to a governmental perspective and a network-based type, which uses loop detector sensors to monitor the number and movement of vehicles. Knowing mobility in the city, the City Chamber can make better decisions regarding traffic control through its traffic patterns. Specifically, the City Council can design and implement the right policies to minimize the intense influx of vehicles in the city by implementing appropriate traffic management policies. Therefore, this work has the potential to improve the lives of urban citizens.

In this dissertation, prediction models based on heterogeneous data related to mobility gathered from a series of inductive-loop sensors installed underneath the pavement of a highway were studied and implemented. In addition to mobility data, and with the intention of improving the prediction, prediction models also use meteorological data that here have missing values. Models are trained and evaluated either in predicting values in all sensors of the urban network under analysis, using the complete data set with inputs from all sensors, or in predicting only one sensor in the urban network, using only data from that sensor. Predictive models used include machine and deep learning, automated machine learning, and ensemble models.

Ensemble learning is a machine learning paradigm in which numerous models are trained to tackle the same learning problem and merge to predict accurately. Ensemble methods usually combine multiple learning algorithms to obtain exacter predictions than the models independently. In this work, the ensemble process allows the study and strategic combination of several prediction models to better deal with data diversity.

The understanding of traffic flow patterns can be complemented, in addition to predicting traffic in certain sensors in the network, by quantifying directional movement in certain areas and by using simulation. Therefore, this work also proposes an exact method for creating Origin-Destination (OD) matrices on which the entry and exit of vehicles on a highway are extrapolated using only information relating to some highway segments. Finally, this work proposes a method to implement a digital twin of a highway by applying meta-modelling approaches to calibrate the simulation model. An actual case study concerning the mobility segment is used to validate the proposed methods.

The main contribution of this work is the ensemble of prediction models for urban mobility considering heterogeneous data. The dissertation includes the study and implementation of prediction models applied to traffic conditions in urban networks, as well as an ensemble and empirical evaluation of models. Furthermore, as a second contribution, an exact method is proposed for creating OD matrices using information related to some segments of a highway. This OD matrix

thus created will be used by a simulator to generate a digital twin of a highway.

Keywords: model ensemble, traffic forecast, heterogeneous data, machine learning, auto ML

Resumo

A previsão de tráfego nas cidades é um problema no âmbito das cidades inteligentes, que visa minimizar custos e melhorar o desempenho das cidades e a vida dos seus cidadãos. A previsão de tráfego não é uma tarefa trivial, uma vez que lida com cidadãos ativos. A pressão turística, a pandemia e o quotidiano cada vez mais dinâmico dos cidadãos tornaram esse desafio ainda mais complexo.

O problema abordado pertence ao tipo de previsão de fluxo de tráfego que corresponde a uma perspetiva governamental e a um tipo baseado em rede, que utiliza sensores detetores de passagem para monitorizar a quantidade e movimentação dos veículos. Conhecendo a mobilidade na cidade, a Câmara Municipal pode tomar melhores decisões quanto ao controlo de tráfego através dos seus padrões de tráfego. Especificamente, a Câmara Municipal pode projetar e implementar as políticas corretas para minimizar o intenso afluxo de veículos na cidade através da implementação de políticas de gestão de tráfego adequadas. Portanto, este trabalho tem potencial para melhorar a vida dos cidadãos urbanos.

Nesta dissertação, foram estudados e implementados modelos de previsão baseados em dados heterogéneos relativos à mobilidade obtidos a partir de uma série de sensores indutivos instalados sob o pavimento de uma estrada. Para além dos dados de mobilidade, e com o intuito de melhorar a previsão, os modelos de previsão também utilizam dados meteorológicos que possuem valores em falta. Os modelos são treinados e avaliados quer na previsão de valores em todos os sensores da rede urbana em análise, utilizando o conjunto de dados completo com entradas de todos os sensores, quer na previsão de apenas um sensor na rede urbana, utilizando apenas os dados desse sensor. Os modelos preditivos usados incluem aprendizagem máquina e profunda (*machine e deep learning*), aprendizagem máquina automática (*automated machine learning*) e modelos *ensemble*.

A aprendizagem *ensemble* é um paradigma de aprendizagem computacional no qual vários modelos são treinados para resolver o mesmo problema de aprendizagem e combinados para obter uma previsão mais exata. Os métodos de *ensemble* geralmente combinam vários algoritmos de aprendizagem para obter previsões mais exatas do que os modelos independentemente. Neste trabalho, o processo de *ensemble* permite o estudo e combinação estratégica de diversos modelos de previsão para melhor lidar com a diversidade de dados.

O entendimento dos padrões de fluxo de tráfego pode ser complementado, além da previsão do tráfego em determinados sensores da rede, quantificando o movimento direcional em determinadas áreas e utilizando simulação. Portanto, este trabalho também propõe um método exato para a criação de matrizes Origem-Destino (OD) nas quais a entrada e saída de veículos numa autoestrada são extrapoladas usando apenas informações relativas a alguns segmentos da mesma. Por fim, este trabalho propõe um método para implementar um gémeo digital (*digital twin*) de uma autoestrada aplicando abordagens de meta-modelação para calibrar um modelo de simulação. Um caso de estudo real referente ao segmento de mobilidade é utilizado para validar os métodos propostos.

A principal contribuição deste trabalho é o *ensemble* de modelos de previsão da mobilidade urbana considerando dados heterogéneos. A dissertação inclui o estudo e implementação de mod-

elos de previsão aplicados às condições de tráfego em redes urbanas, bem como o *ensemble* e a avaliação empírica de modelos. Além disso, como segunda contribuição, é proposto um método exato para a criação de matrizes OD utilizando informações relacionadas a alguns segmentos de uma autoestrada. Por fim, esta matriz OD criada será utilizada por um simulador para criar um gémeo digital (*digital twin*) de uma autoestrada.

Palavras-chave: ensemble de modelos, previsão de tráfego, dados heterogéneos, aprendizagem máquina, auto ML

Acknowledgements

First, I thank Professor Ana Paula Rocha and Professor Rosaldo Rossetti, supervisors of this dissertation, for their guidance, time, support, and encouragement throughout the completion of this work. Their help was essential throughout the development of the dissertation because, without their ideas, support, and knowledge, this document would not have the quality and innovation it has. I want to thank Professor Ana Paula, co-author of my first published article, for all the knowledge in the Machine learning field that she transmitted to me, highlighting that she was the person who helped me the most in the development of this dissertation. I would also like to thank the two professors who launched me into the field of investigation, Professor Daniel Silva and Professor Dalila Fontes. I especially thank Professor Dalila Fontes for all the knowledge she shared with me.

Also, I want to thank my father for being a fantastic IT engineer who showed me the best of this fascinating field and fostered my engineering side. I want to thank my mother for teaching me everything she knows and showing me how happy one can be at work every day. Finally, I wanted to thank my sister for inspiring me to be better every day and showing me that perfectionism is not a defect. Finally, I want to thank all my friends, boyfriend and colleagues because life would not be so delightful without them.

Maria Helena Viegas Oliveira Ferreira

"People rarely succeed unless they have fun in what they are doing."

Dale Carnegie

Contents

List of Figures

List of Tables

Abbreviations

AI	Artificial intelligence
AITFP-WC	Artificial intelligence-based Traffic flow prediction considering weather conditions
ANN	Artificial Neural Network
AVs	Automated vehicles
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DL	Deep learning
DT	Decision Tree
ENN	Elman neural network
GBM	Gradient boosting framework
GCN	Graph convolutional network
GPS	Global Positioning System
GRU	Gated Recurrent Unit
IoV	Internet of Vehicles
ITS	Intelligent transportation systems
LSTM	Long Short Term Memory
MAE	Mean absolute error
MedAE	Median absolute error
ML	Machine learning
MLP	multi-layer perceptron
MSE	Mean squared error
NMME	North American multi-model ensemble
OD	Origin-Destination
PCA	Principal component analysis
POI	Points of interest
ReG-Rules	Ranked ensemble G-Rules
RF	Random Forest
RMSE	Root mean squared error
SSTD	Spatial Static Temporal Dynamic data
ST-GDN	Spatial-Temporal Graph Diffusion Network
SVM	Support Vector Machines
TDO	Traffic data observation
TFP	Traffic flow prediction
TP	Traffic prediction
TSA-FFNN	Tunicate swarm algorithm with feed-forward neural networks
TTP	Travel Time Prediction
VCI	Via de Cintura Interna
XAI	Explainable artificial intelligence
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

This chapter starts with a brief contextualization of the problem addressed in this dissertation, followed by a description of the objectives, motivation, and contributions. Afterwards, the structure of the document is outlined.

1.1 Context

The growth of cities and the complexity of urban living has been challenging for cities in their different segments: mobility, buildings, community, energy, and municipality. Smart cities are a trending option to combat these numerous challenges; therefore, the subject has been drawing the attention of researchers and others. A *smart city* is a technologically city that uses multiple sensors to extract data and analyze it to enhance the quality of life of its inhabitants while minimizing the resources and costs needed. The patterns of use of different resources in cities are increasingly varied, mainly due to the pandemic, tourist pressure, and citizens' increasingly active daily lives, which have changed the city paradigm.

1.2 Aim & Goals

This work emphasizes the mobility segment. The real case used in this dissertation provides data related to the "Via de Cintura Interna" (VCI) road. Throughout the inner-ring of the VCI road in the city of Porto, there are a series of inductive-loop sensors set up collecting data about traffic in the specific segment they are situated. This information makes it possible to find traffic patterns and predict network congestion. In short, we intend to study and implement forecasting models based on heterogeneous data related to Porto's mobility. Prediction of traffic in cities is a problem within the scope of smart cities and is not a trivial task as it deals with active citizens. After having different models, they are ensembled to obtain better results. Afterwards, there is speculation regarding traffic for the entire VCI based on a few points and the creation of a digital twin. The final aim is to accomplish intelligent transportation, i.e., make travel more efficient and convenient. The problem addressed belongs to the short-term traffic flow prediction type that corresponds to a

governmental perspective and a network-based type, which uses loop detector sensors to monitor the number and movement of vehicles. Finally, the last goal is to create a VCI digital twin using a micro-simulator.

1.3 Motivation and contributions

An excellent motivation for carrying out this work is dealing with actual data and problems and contributing to the city's knowledge. Another great motivation is the contribution of this dissertation since the ensemble of prediction models for urban mobility considering heterogeneous data is new in the literature. In short, this dissertation includes the study and implementation of prediction models applied to traffic conditions in urban networks, as well as an ensemble and empirical evaluation of models. Furthermore, an exact method is proposed for creating OD matrices using the information of traffic provided from count sensors located on some highway segments. The OD matrix created will be used by a simulator to create a digital twin of a highway.

Finally, understanding mobility patterns in urban environments allows better decision-making regarding traffic planning, management, and control. Therefore, by knowing the mobility in the city, the City Chamber can make better decisions regarding traffic control through its traffic patterns. Specifically, the City Council can design and implement the right policies to minimize the intense influx of cars in the city implementing appropriate traffic management policies. Therefore, given that congestion increases stress and has a very negative impact on the lives of all citizens, this thesis has the potential to improve the lives of urban citizens.

1.4 Outline of the Dissertation

This chapter introduces the problem that we aim to solve, providing some context to it and delineating the motivations, contributions, and goals behind our work. The rest of the dissertation is structured as follows: Chapter 2 provides an overview of the research topics addressed in the dissertation to explain the basic concepts necessary to understand the thesis. After the background chapter, chapter 3 presents the systematic literature review and the gap analysis, analyzing the methods and data sets used in the solution of similar sub-problems, the practical applications of the previously identified concepts, and the differentiation of this work from the existing literature. Next, chapter 4 consists of the methodological approach and includes the problem formalization and the description of the materials and methods. Chapter 5 presents a detailed specification of the development process of the prediction models for intelligent city applications with missing values, along with the description of the exact model implemented to obtain the OD matrix and the microscopic simulation developed. Chapter 6 details the empirical results from the applied tests, including the discussion and evaluation of the implemented solutions. Finally, chapter 7 presents the main conclusions obtained from this dissertation, as well as the identification of future work.

Chapter 2

Background

This background chapter provides an overview of the research topics addressed in the dissertation. The goal is to explain the basic concepts necessary to understand the thesis. This work comprise five main phases: the implementation of prediction models, the ensemble of models, the empirical evaluation, the speculation of traffic for the entire VCI based on a few points, and the creation of a digital twin. The results contribute to Intelligent transportation systems. Therefore, this chapter provides a background related to the five main themes of this thesis: prediction algorithms, intelligent transportation systems (ITS), model ensemble, empirical evaluation, and digital twin.

2.1 Prediction algorithms

Prediction algorithms are techniques that predict future or unknown events based on historical data. *Traffic prediction* is the task of precisely estimating traffic at a specific time interval in the future in a given region, based on historical traffic data and, optionally, floating car data. Numerous devices such as inductive loop sensors, video surveillance systems, and GPS can help to accurately predict traffic aiming to control and manage road traffic conditions. Inductive loop sensors are electromagnetic loops installed underneath the road pavement placed transversely to the road. Their function is to detect the passage of vehicles in a specific short temporal period and generate the respective data, namely, the number of vehicles of each class, the spacing between them, the average speed, and the sensor occupancy rate [32]. Nowadays, the exponential growth of vehicles has made it difficult to predict traffic in current scenarios with machine learning models [45].

In the literature three significant traffic prediction problems can be found: Traffic status prediction, traffic flow prediction (TFP), and travel demand prediction. The TFP category divides into two more specifically problems types: region-based and network-based. Region-based problems aim to simultaneously predict crowd flows, i.e., the forthcoming number of outgoing and incoming people in all city-regions. In contrast, the goal in the Network-Flow problem is to predict the flow

passing through individually intersections on road networks. Specifically, the network-based type uses loop detector sensors to infer the number of vehicles in a network [65].

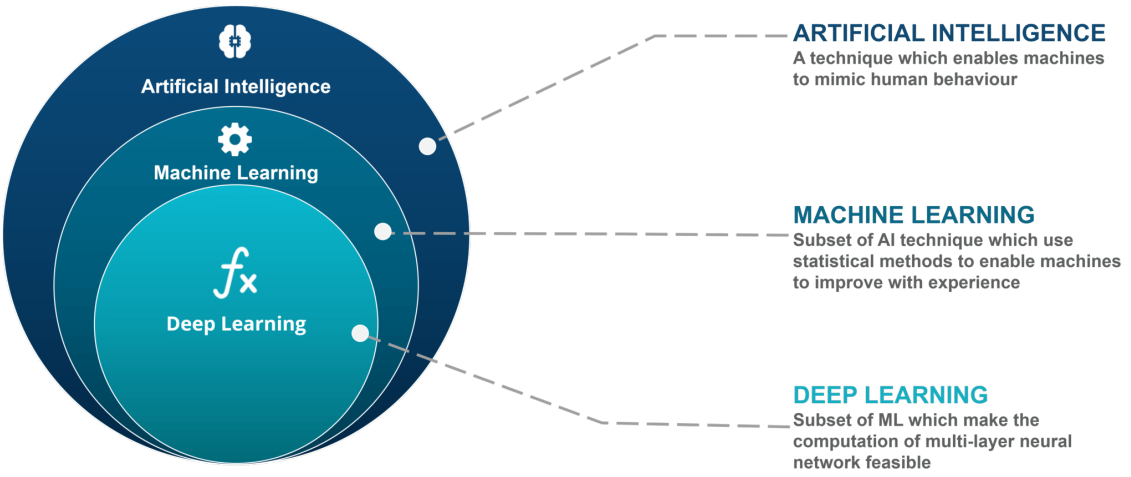
The data used by traffic predictive analytics algorithms is classified as spatial-only (such as roads), temporal-only, or spatio-temporal. The spatio-temporal is categorized in Spatio-Temporal Static data, Spatial Static Temporal Dynamic data (SSTD), and Spatial Dynamic Temporal Dynamic data. This dissertation's prediction algorithms are fed with SSTD. The prediction improves if the algorithms incorporate different context data, such as temporal data (for instance, holiday, date, and timestamp), meteorological data (for example, the weather), surrounding POI data [7], and event data (namely, a traffic accident) [65]. Like this thesis, most traffic forecasts' goal is to control the flow and prevent congestion.

2.1.1 Machine and deep learning

This subsection specifies and differentiates machine learning from deep learning while presenting the most usual algorithms of both models types. Machine intelligence is usually referred to as artificial intelligence (AI) and refers to the intelligence displayed by machines, i.e., AI systems perform tasks that, when performed by humans, require intelligence. ML achieved significant success in recent decades, and its application is still growing in a significant number of fields. Machine learning algorithms comprise not only linear varieties but also non-linear varieties. Although non-linear algorithms take longer to train than linear ones, non-linear methods are more optimized for non-linear problems. For a long time, traditional experiments used several machine intelligence techniques; however, these experiments were time-consuming and expensive. With the increasing availability of data, 'big data' has become commonplace in solving learning problems, and machine learning approaches have become less efficient. Therefore, machine learning methods were optimized for the increasingly common use of massive data, giving rise to deep learning. Deep learning is a more flexible, efficient, and robust manner to deal with the big data provided to the algorithms [66]. In addition to the outstanding performance of deep learning with big data, this learning method is more known for dealing with images, text, video, and audio [52], given that they can learn this feature representation automatically from the raw data [47]. Figure 2.1 can help to understand the difference between machine learning, deep learning, and artificial intelligence and their relation.

ML and DL also differ structurally. While the traditional machine learning methods, also known as shallow models, comprise none or only one hidden layer, the deep learning models include several hidden layers. The most well-known shallow models are support vector machine, naïve Bayes, logistic regression, clustering, artificial neural network, decision tree, K-nearest neighbor, and combined and hybrid models. Regarding deep learning, the more common supervised learning methods are recurrent neural networks, convolutional neural networks, deep neural networks, and deep brief networks. In the unsupervised learning category, the usual models are restricted Boltzmann machines, generative adversarial networks, and autoencoders [47]. Finally,

Figure 2.1: Difference between machine learning, deep learning and artificial intelligence. Image extracted from [35].



the design of both the machine and deep learning approaches includes the decision of the algorithm to use based on the provided raw data set, the pre-processing of the data set features, and the setting of the optimal model hyper-parameters.

In sum:

- **Machine learning** models allow machines to automatically learn valuable information and make predictions based on massive raw data provided to the algorithm [47]. Deep learning is a subset of AI.
- **Deep learning** models are a deep structure that consists of various deep networks, including several hidden layers. Deep learning grew out of improving ML algorithms, so it's a branch of ML.

2.1.2 Automated Machine Learning

Automated Machine Learning, more known as autoML, enables processes and methods to developers aiming to improve the efficiency of the Machine Learning models, obtain results in minutes, and accelerate research on ML [48]. Specifically, the main machine learning tasks that autoML can automate are:

1. Pre-process and clean the raw data,
2. Feature selection and feature engineering,
3. Model selection and evaluation,
4. Optimize model parameters and hyper-parameters,
5. Post-process/Ensemble ML models.

The pre-processing of the raw data usually includes the following steps in this order:

1. Acquire a relevant data set,
2. Import all the necessary libraries,
3. Import the data set(s),
4. Identify and handle the missing values,
5. Encode the data with specific categories,
6. Splitting the data set into two sets (training set and test set),
7. Feature scaling (i.e., limit the range of the independent variables).

Feature selection and feature engineering are the procedures that involve employing domain comprehension to respectively select and transform the significant features from raw data when developing a predictive model using statistical modeling or machine learning. The model selection step includes the selection of a suitable model family based on the data. Hyper-parameters are algorithm inputs that can lead to distinct predictive model instances; therefore, changing hyper-parameter values in the same algorithm alters the model behavior. Hyper-parameter optimization refers to automating the delivery of the best possible model instance among all the available algorithms by tunning the hyper-parameters of each algorithm. The post-processing of ML models is a procedure that enhances the results of the models and will be described in detail in section 2.3.

2.2 Intelligent transportation systems

Intelligent Transport Systems aim for a more intelligent, coordinated, and safer usage of transport networks. ITS propose to mitigate congestion using existing network infrastructures wisely and are critical to achieving smart cities [49]. Congestion leads to harmful effects such as decreased safety for pedestrians and vehicles and increased noise and air pollution [26]. To reduce congestion and journey time, one can make use of dynamic traffic monitoring and dispatch of traffic regulation departments. Thus, accurate short-term traffic prediction is essential to improve traffic efficiency and safety, especially concerning large-scale intercity road networks with a higher flow [31]. Therefore, traffic prediction is crucial for ITS.

2.3 Model ensemble

Given that traditional ML models generally deliver low accuracy and easily get over-fitting, many implement ensemble methods to enhance the accuracy of the predictions. Ensemble methods usually combine multiple learning algorithms (learners) to obtain exacter predictions than the models independently. *Ensemble learning*, also known as committee-based learning and multiple classifier systems [70], is a ML paradigm in which numerous models are trained to tackle the same

learning problem and merge in order to enhance the average prediction accurately. In addition, it is possible to obtain better results using fewer models as input in the ensemble models. Therefore, for the ensemble to be advantageous, it is necessary that the algorithms used with ensemble input are diverse and precise [70]. This technique can be applied to classification as well as regression algorithms. There are three main types of ensemble methods: Bagging, Boosting, and Combination Strategies. These ensemble types will be described in detail in the followings subsections.

Ensembles are categorized into homogeneous and heterogeneous. The homogeneous ensemble exclusively combines learners of the same type; for example, in a neural network ensemble, all learners belong to the neural network type, and a decision tree ensemble includes just decision trees. While individual learners are designated *base learners* in the homogeneous ensembles, the respective learning algorithms are designated *base learning algorithms*. In opposition, an ensemble is heterogeneous if it combines different learning algorithms and individual learners, and there is no single base learning algorithm or base learner [70]. In addition, learners are categorized as weak and strong learners. Weak learners are ensemble model input models with an accuracy slightly better than random guessing. For example, in binary classification problems, a weak learner would correspond to an algorithm with an accuracy insignificantly above 50%. In contrast, strong learners are more accurate models with arbitrarily good accuracy [42].

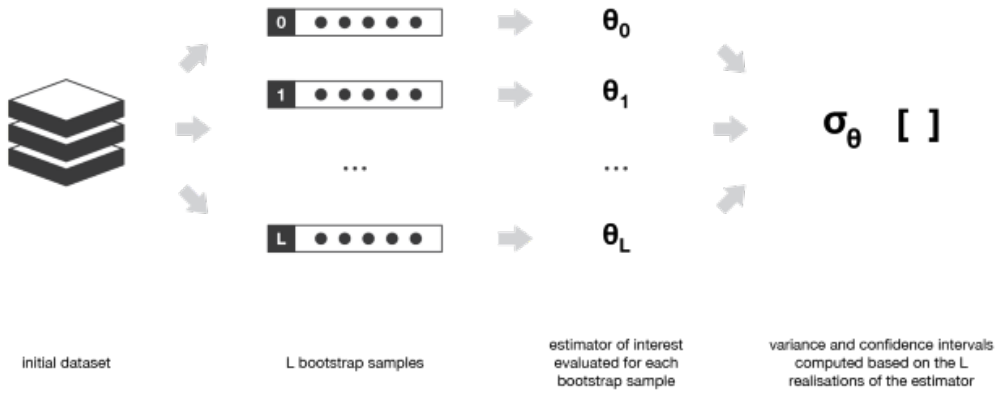
The explainable of a prediction model usually refers to presenting to the system analyst a conditioned view of the more influential model factors concerned in the forecast. Machine and ensemble models should be explainable to enhance the accuracy of the methods, avoid bias, and reduce the threat of irreversible wrong outputs in critical applications, namely automatic vehicle recognition, stock market analysis, intrusion detection, and credit risk evaluation. Therefore, explaining machine and ensemble learning models has recently gained attention from developers, data scientists, and researchers. *Explainable artificial intelligence* (XAI) is artificial intelligence in which human analysts can understand the solution results. The explainability of machine learning models is already a challenge because the models were usually treated as black-boxes [4]. Nevertheless, the explainability of ensemble models is even more challenging, given that it requires human analysts to analyse numerous decision models in order to acquire insights regarding the causality of the final output forecast [8]. In this work, the ensemble process allows the study and strategic combination of several prediction models to better deal with data diversity.

2.3.1 Bagging

Bagging [11] is short for Bootstrap AGGregaING, given that this ensemble learning method is based on bootstrap sampling. Figure 2.2 contains a schematic with a visual explanation of Bootstrapping. Bagging learns weak learners in parallel and independently from the other models and then merges them according to some deterministic averaging methodology. The main idea of Bagging is to obtain a model with a lower variance by fitting numerous independent models concurrently and making an "average" of their predictions. The algorithm starts by randomly picking a single sample from a data set with s samples and copying it to a new data set. The picked sample

remains in the original sampling set; therefore, it can still be picked up subsequently. The repetition of the process s times originates a data set containing s samples. Approximately 63.2% of the samples of the original data set will appear in the new data set, some more than once, while some of the original samples may never appear. The repetition of the entire process T times creates T data sets with s samples. The base learners are trained with these sampling sets and then combined [58]. Random Forest (RF) is considered an extension of Bagging. In the RF, randomized feature selection is inserted on top of Bagging [70].

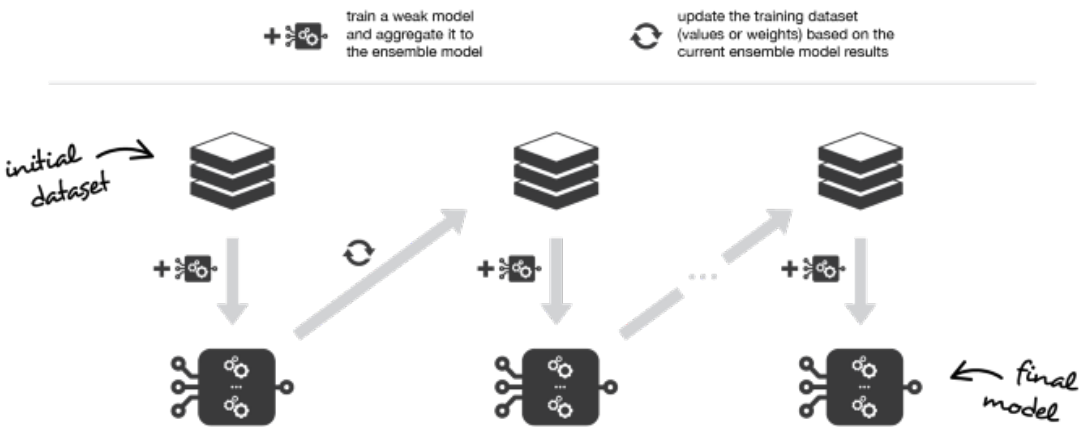
Figure 2.2: Explanation of Bootstrapping. Image extracted from [58].



2.3.2 Boosting

The *Boosting ensemble* is characterized by models that transform weak learners into strong learners. Boosting models can be used in regression and classification problems. In boosting ensemble, the method learns homogeneous weak learners sequentially in a highly adaptative way; therefore, a base learner depends on the previous ones. Finally, the algorithm combines the learnings following a deterministic process [58]. Specifically, the Boosting method's initial stage is training a weak base learner. Afterwards, the distribution of the training samples is tuned depending on the outcome of the base learner in a matter that subsequent base learners will give more attention to the inaccurately classified samples. When the first base learner finishes the training, the following base learner is trained with the tuned training samples, and the outcome is used to tune the training sample distribution a second time. This last procedure repeats until the count of base learners reaches a predetermined value. Lastly, the final stage corresponds to the weight and combination of these base learners [70]. Figure 2.3 contains a visual explanation of Boosting algorithms. Gradient Boosting[40], implemented with the GBM (gradient boosting framework), the Light GBM, and the XGBoost (eXtrem Gradient Boosting), is a common Boosting algorithm [70].

Figure 2.3: Explanation of Boosting. Image extracted from [58].

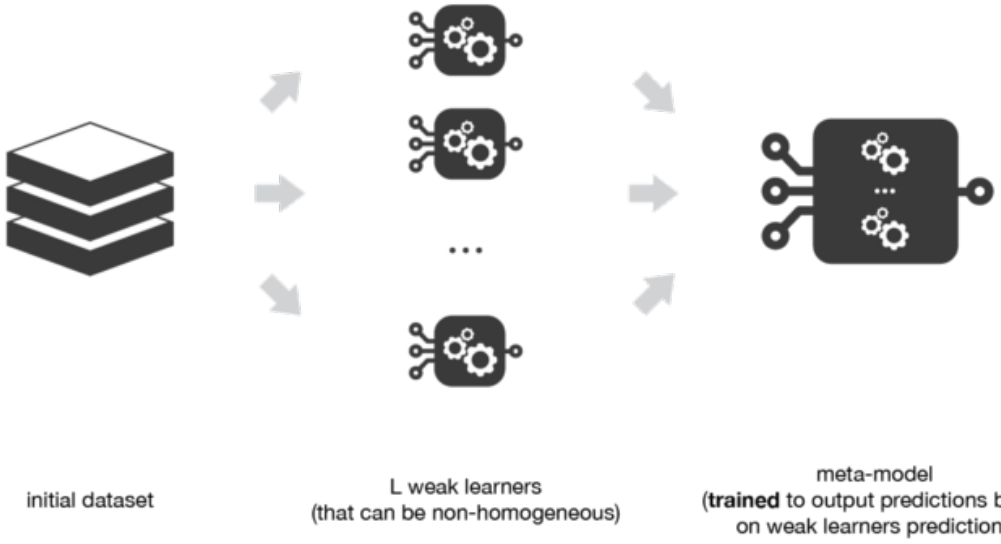


2.3.3 Combination Strategies

The combination strategies include averaging, voting, and combining by learning [70]. The averaging methods are divided into simple averaging and weighted averaging. All ensemble methods require the prediction of multiple models for the same data point. In *simple averaging*, the final prediction is the average of all models' forecasts. The simple averaging ensemble can be applied to classification as well as regression problems. In the first, it calculates probabilities, and in the latter, it makes predictions. The *weighted average* method is an extension of the simple averaging ensemble. In this ensemble type, instead of having an average of all predictions, the method returns a weighted average of all forecasts. The goal is for the most accurate predictions to have a higher weight in the final prediction. Finally, the voting or max voting ensemble is usually used in classification problems. In this ensemble model, each model prediction is considered a 'vote', and the final prediction corresponds to the majority of the models' predictions [58]. Of the three methods reviewed, the averaging method has the greatest potential.

The final combination strategy is *combining by learning*. Combining by learning is a more powerful combination approach employed if more abounding data for training is available. The combining by learning approach uses a meta-learner to combine the individual learners. The more well-known method of this type is Stacking. While Bagging and Boosting ensemble usually combine homogeneous weak learners, Stacking ensemble usually combines heterogeneous weak learners. Figure 2.4 contains a visual explanation of Stacking algorithms. Stacking approaches use a second-level learner to combine the individual learners. In this method, there are two types of learners. The individual learners used as input are designed as first-level learners, while the learners executing the combination are known as meta-learners or second-level learners.

Figure 2.4: Explanation of Stacking. Image extracted from [58].



2.4 Empirical Evaluation

Empirical evaluation is an assessment method where results are obtained with experiments or observation instead of theory. An adequate empirical evaluation starts with the appropriate design and execution of experiments to separate specific factors under testing from other confounding factors [17]. In sum, there is an empirical evaluation in this thesis as the study proposed is based on evidence rather than just theory.

2.5 Digital twins

A simulation model allows the abstraction of a real-world scenario and enables one to focus on intriguing phenomena. Micro-simulations are used for detailed analysis of activities such as disease spread and highway traffic flow, among others. However, the simulation study results are only valuable and eloquent if the simulation model correctly and closely represents the real system [53]. A digital twin is a digital replica of a physical object. Even though both simulation and the digital twin technology use virtual model-based simulations, the two concepts are not the same. Digital twins are used in a wide variety of industries and are crucial when the asset or environment being copied is dynamic, i.e., when the system changes over time [64]. Since the system under study is very dynamic, the microscopic simulation will be a digital twin.

Many traffic simulators that permit microscopic traffic simulation are available in the literature, such as PARAMICS, VanetMobiSim, VISSIM, and SUMO. Most simulators model only support

homogeneous vehicles. In opposition, SUMO supports customized vehicle types: cars, three-wheelers, and two-wheelers (like motorcycles), among others. Another distinctive characteristic of the SUMO simulator is that it enforces lane discipline on all vehicles. However, the compliance with the lane discipline is not always verified by some vehicles of small dimensions, such as the vehicles of class A, which have only two wheels [53].

Origin-Destination (OD) matrices are the standard way to express the demand for simulation. The volume of vehicles, referred to as the demand, is generally represented in OD matrices. In this type of matrix, a trip's starting and end points are defined, and each origin and destination is a node in the network. Generally, the OD matrix has a column for each destination and a row for each origin. The demand for the specific Origin-Destination pair is presented in each matrix cell. Further, a dynamic OD matrix is defined as a set of stacked OD matrices, each defining demand over time.

2.6 Summary

This background provided the necessary knowledge bases for the reader to understand the rest of the document, including the next literature review chapter. The chapter makes it possible to classify the problem to be addressed, in addition to giving an overview of the relevant techniques related to possible solutions of the problem.

Chapter 3

Related work

After the context provided by chapter 2, this chapter presents the related work. After reviewing and analyzing the work published previously, the novelty factors of this work and the differences between it and the others are detailed. In sum, this chapter includes a systematic literature review and a gap analysis.

3.1 Systematic literature review

This section reviews and analyses the methods used in similar problems or with some sub-problems in common. Therefore, the section provides a clear picture of the state of the art on the essential subjects.

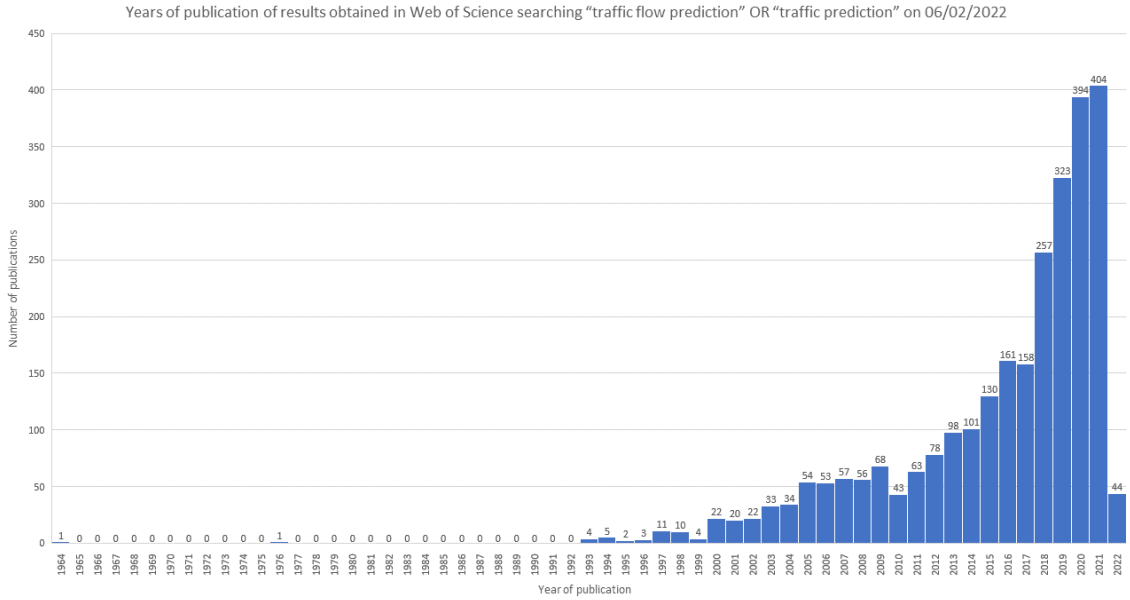
3.1.1 Prediction algorithms

This section analyses works that have proposed relevant and recent traffic forecasting methodologies. To systematically review traffic prediction algorithms, the terms “traffic flow prediction” and “traffic prediction” with quotation marks were searched in the Web of Science, obtaining 946 and 1924 results, respectively. The last search was performed with the terms: “traffic flow prediction” OR “traffic prediction” to eliminate overlapping results and obtained 2714 results. The years of publication of the results obtained on 6 February 2022 are shown in Figure 3.1.

The oldest result corresponds to an article published in 1963 [30] called "applications of the traffic model prediction", referring to the applications that traffic prediction could have, without mentioning strategies to predict it. The first article to apply a traffic forecasting method was from 1976 [41], therefore published more than 45 years ago. In this way, we conclude that the topic has been the subject of research for decades. The first approach used was a nonlinear-regression approach to estimate parameters in models for traffic prediction.

Traffic forecast can be highly affected by accidents and weather conditions since adverse conditions make the same paths longer and driving more difficult, altering the flow on the roads. Several authors have tried to include weather conditions in their predictions. Duhayyim et al. [33]

Figure 3.1: Number of publications per year of results obtained in Web of science searching "traffic flow prediction" OR "traffic prediction" on 06/02/2022.



developed an artificial intelligence-based TFP considering weather conditions designed AITFP-WC. Specifically, they implemented a model that incorporates an Elman neural network (ENN) method in the first stage to forecast short-term traffic flow in real-time. The aim is to improve the accuracy of traffic predictions, including under extreme weather conditions, for intelligent cities by incorporating weather-related conditions. In addition to the ENN, the novel technique incorporates a tunicate swarm algorithm with feed-forward neural networks (TSA-FFNN) and a data fusion procedure. In detail, the TSA-FFNN model is employed for the periodicity and weather analysis. Finally, traffic flow prediction and WPA procedure fusion use the FFNN model to establish the prediction's final output. Therefore, the presented AITFP-WC contains three high levels: ENN-based short-term traffic flow prediction, TSA-FFNN based WSA, and FFNN based fusion processes. The paper used a real traffic data set regarding the metro freeway in the Twin Cities. The data was gathered from more than 4,500 loop detectors. The AITFP-WC outperformed other techniques in the literature. In the previous year, Artin et al. [9] proposed a method to forecast urban traffic flow based on ensemble learning and weather criteria. The ensemble model uses as an input two deep learning methods: Linear Regression and Neural Architecture Search algorithm, implemented in Python 3.6. The data set used included 400 instances extracted from this article [25]. The data set features include six attributes regarding the day's weather: wind speed, cloud height, dew point, absolute humidity, temperature, and visibility. The linear regression method performed slightly better than the random forest, multilayer perceptron, and Support Vector Regression methods; therefore, it was the regression method used in ensemble learning. Other works such as Artin et al. related to both traffic forecasting and ensemble learning will be reviewed in sub-sub section 3.1.3. In addition to the works presented, some others studies also worked with

traffic forecasting and weather conditions, namely [46], [59], [2], [10], [61], [36] according to Artin et al. [9].

In addition to the adversities mentioned previously, the accuracy of the traffic forecast can also be affected by missing values, as the quality of the data is essential in the analysed learning methods [69]. Missing data can result from many factors, namely, communication errors, sensor failure, and weather problems, and usually negatively influence traffic prediction accuracy. Some authors try to deal with this adversity with different methods. Dong et al. [31] worked with a large-scale road network traffic prediction taking into account missing data and proposed a model named tensor combined temporal similarity revisited graph convolutional gate recurrent unit, which integrates data completion and traffic flow prediction through a graph representation specifically, the Graph Laplace. The Graph Laplace completeness of the data considers diverse kinds of time characteristics. The method integrates Gated Recurrent Unit (GRU), Attention network, and graph convolutional network (GCN) to enhance prediction performance. The spatial attributes were thoroughly considered for introducing the road network topology into GCN, and the method took extensive regard of time dependencies by employing multi-input fusion. They constructed a two-way network graph topology by considering the bidirectional connectivity of large-scale highway networks. In addition, the method can retain the network's topological information while adaptively extracting the Spatio-temporal attributes from diverse traffic periodicities. The data set used for validation was real and regarded the extensive two-way inter-city highway network in Jiangsu, China, during August 2019. Specifically, the data contains Pizhou City, Xuzhou City, Suqian City, Lianyungang City, and Xinxin City. Results show that the method outperformed the existing methods dealing with missing data. Furthermore, the model correctly determinates the road network's influential nodes and automatically learns to identify the significance of the prior traffic flow.

Although much of the literature still concentrates only on learning local spatial correlation, many others go beyond and demonstrate that prevision accuracy can be improved by using distanced spatial data in addition to local data. For instance, two geographically distant locations with identical urban functions (e.g., transportation hub and shopping zone) can be correlated regarding their traffic distribution [67]. Zhang et al. [67] presented a novel traffic flow prediction architecture based on graph neural networks named Spatial-Temporal Graph Diffusion Network (ST-GDN). This method goes beyond neighboring spatial correlations between adjoining regions, using the global geographic contextual information instead of most existing works. The traffic flow patterns are usually complicated and multi-periodic because each temporal resolution dimension (e.g., monthly, weekly, daily, hourly) demonstrate traffic flow dynamics from different temporal levels. The extracted temporal patterns of each level frequently complement each other. Therefore, this method also addresses the failure of some methods to consider only data related to one temporal dimension, ignoring patterns at diverse temporal levels and the temporal dependencies between them. Specifically, the ST-GDN starts by designing a resolution-aware self-attention network in order to encode the multidimensional temporal signals. Afterwards, the spatial-temporal pattern representations were improved with subsequent integration of the global traffic dependencies across diverse regions and local spatial contextual information. The framework was validated

using four real traffic data sets. Comparing the ST-GDN with 15 baselines using the data sets mentioned, it is notorious that the architecture outperforms the existing baselines. The authors provided the source code, which can be found at [44]. As Dong et al. [31], Feng et al. [37] incorporate a GCN with GRU. Feng et al. present a hybrid method for traffic flow prediction combining global and local spatial correlation, which effectively detects the spatial-temporal correlation of the traffic flow. The model comprises two components: local spatial-temporal element and global spatial-temporal part. The local Spatio-temporal component combines GCN and Fully Connected Layer to obtain the local spatial correlation. Furthermore, the global Spatio-temporal correlation derives from the stacking of the GRU. The final output of the model results from the two components mentioned. The model evaluated used two real-life highway data sets from the Caltrans Performance Evaluation System without any missing data: the PEMS08 and the PEMS04 data set. The results indicate that the method outpaces state-of-the-art baselines.

In opposition to this dissertation, Ali et al. [7] worked with region-based crowd flows prediction instead of network-based prediction. Their goal is to simultaneously predict crowd flows, i.e., the forthcoming number of outgoing and incoming people in all city regions. They propose a unified dynamic deep spatio-temporal neural network model based on long short-term memory and convolutional neural networks, named DHSTNet. They use both temporal and external features. The temporal features are categorized as weekly, daily, and recent and, together with the spatial data, are dynamically learned by the method. The external component refers to the distribution of points of interest (POI) and weather conditions. Different weights were assigned to the four features in the proposed method, and the results were combined, through a completely connected neural network, to obtain the final forecasts. Lastly, the DHSTNet was combined with Graph Convolutional Network based on Long Short Term Memory (LSTM), which handles dynamic spatio-temporal dependencies of traffic flows, in a model designed as GCN-DHSTNet. The GCN-DHSTNet captures the spatial patterns and reveals the influence of the previously mentioned features in the region-based prediction. The GCN-DHSTNet model's validation and comparison with other existing literature methods were made using two real-world data sets. After extensive evaluations, the authors conclude that the GCN-DHSTNet and the AAtt-DHSTNet model considerably outperform the current techniques.

Some articles in the literature present some innovative technologies such as connected vehicles [5] and Internet of Vehicles [43], [16], [60], [1]. The Internet of Vehicles applies the Internet of things technology in ITS. Tian et al. [60] proposed a method using an enhanced autoregressive moving average and Empirical Mode Decomposition for the network-based traffic prediction. Abdellah et al. [1] propose an artificial neural network (ANN) to forecast the IoT traffic; specifically, they presented a time-series traffic prediction using a multi-step ahead forecast with Time Series NARX Feedback Neural Networks.

Kashyap et al. [45] reviewed many of the newest deep learning models proposed to tackle the traffic flow prediction problem. This systematic review provides information on which models perform best in each scenario and how diverse factors impact these methods. Based on this section and the review of Kashyap et al., it can be concluded that numerous deep learning methods were

employed to solve the traffic flow prediction problem. The more usual methods are Stacked Auto Encoder, Restricted Boltzmann Machines, Long Short-Term Memory, Convolutional Neural Network, and Recurrent Neural Network. These deep learning architectures employ multiple layers; therefore, they can progressively extract higher-level attributes from the raw input.

3.1.2 Intelligent transportation systems

Even though traffic prediction is crucial for intelligent transportation systems [45], other works in the literature have also contributed to intelligent transportation without using forecasting models. This subsection reviews some of these works.

Lígia et al. [18] studied the impact of using reversible routes in urban areas since it is a way to improve the overall traffic system without additional lanes. In that work, they used mixed-integer non-linear mathematical programming in a problem named reversible lane network design. The same authors [19] explored the combination of smart cities, autonomous driving of levels 4 and 5, and V2X communication. The work concludes that reversible lanes can proportionate daily travel time savings of 11k€ if automated vehicles (AVs) minimize their own travel time, and 16k€ if AVs routes follow an optimal system scenario minimizing the overall total travel time. Both works use a Quasi-real case study of Delft, the Netherlands.

The data set used in this thesis has already been studied in other articles, and it will be specified in detail in chapter 4. Alam et al. [6] analysed the same historical data set to obtain expressive traffic flow patterns from big data. Specifically, they implemented a traffic data observation (TDO) tool using Java to visualise traffic data and encounter the historical averages of traffic flow. They developed and compared regression models applied to the sub-data sets generated by the TDO tool, which determine the annual average daily traffic. They implemented the following regression methods: M5 Base Regression Tree, Linear Regression, and Sequential Minimal Optimisation Regression. They apply regression trees to determine the traffic flow patterns. As in this work, their goal is to improve traffic flow and traffic management systems. Oliveira and Rocha [49] benchmarked numerous methods based on deep learning to fill in missing values, using the same case study. Specifically, they compared a baseline and a K-Nearest Neighbors algorithm against three deep learning methods (Simple Neural Network, Over complete Autoencoder Network, and Adversarial Network). They conclude that approximately 12% of all reads in the data set are missing It may be interesting to continue their future work related to using recurrent or convolutional neural networks to fill missing values in this thesis. According to the authors, the goal would be to leverage the temporal and spatial relationship between the sensor measures.

This subsection mentioned some relevant articles on the topic of intelligent transportation. The applications of intelligent transport are outside the scope of this dissertation and can be seen in detail in the rich survey in the article [65].

3.1.3 Model ensemble

Ensemble learning can be applied to a large diversity of problems. Deb et al. [22] published a paper when the exponential growth of covid-19 cases left populated areas like India without tests for the entire population. In this way, radiologists find a way to detect cases of covid-19 by classifying chest X-rays as images of patients with or without covid-19. They proposed a multi-model ensemble-based Deep Convolutional Neural Network (DCNN) structure, which comprises four pre-trained DCNN networks to detect covid-19 patients based on chest X-ray images. In this framework, an ensemble of four DCNN architectures extracts the low-level features of the X-ray images. Specifically, the DCNN architectures were GoogleNet, VGGNet, NASNet, and DenseNet. Afterwards, the features were fed to a fully connected layer for classification. The ensemble architecture empirical evaluation used two public data sets and a private Chest X-ray data set, the latter available at GitHub in the link [21] along with the source code. As expected, the ensemble model proposed obtained better results than any individual classifier used in the ensemble architecture. The results obtained by the private data set had an accuracy of 93.48%. Differently, the performance results on the other public data sets obtained an accuracy of 98.58% and 88.98% for binary class classification and three-class classification, respectively. Finally, the authors also compare the framework's performance with other models from the literature and conclude that the proposed architecture obtained promising results, improvable if more training data becomes available.

Pakdaman et al. [51] proposed a multi-model ensemble method on monthly precipitation predictions over the Iran region; precisely, they proposed neural network models to post-process over the North American multi-model ensemble (NMME) models that forecast precipitation monthly. The NMME is a seasonal prediction system comprising multiple models ensemble from North American modelling centers that regularly generate monthly precipitation forecasts for all earth. This paper considered precipitation hindcasts of eight separate NMME project models and demonstrated that, in all months, the proposed neural network high performances the individual NMME models used. The PERSIAN-CDR climatology data set was used to train the proposed neural network that employed multi-layer perceptron neural networks. Not only the paper raked NMME models for each month employing a multi-criteria decision-making technique, but it also proposed a ranking method for other climatic ensemble approaches. Later, the same authors proposed a European Multi-Model Ensemble for the same post-processing of outputs problem but applied to Europe instead of Iran [50]. The paper presented a model based on four machine learning techniques, specifically, Support Vector Machines (SVM), ANN, Random Forest, and Decision Tree (DT), to post-processing the outputs of four European models from Meteo-France, ECMWF, CMCC, and DWD. The neural network applications present a challenging training phase, which contains a solution of an unconstrained optimization problem to define the optimal model weights. The ANN models are typically significantly efficient at solving various problems, even though the algorithms generally converge to a locally optimal solution in the optimization problem mentioned. This paper used data from the Climatic Research Unit to train the ML models. The implemented

methods results show that the RF methods and neural network outpaced the SVM and DT in all months of the year. The four European models have distinct performances for each country's part, lead time, and month. Although the models usually present unsatisfactory performance, the RF and ANN enhanced these results. In sum, the new multi-model approach outpaced all independent European models as in the last paper.

Ensemble of prediction algorithms

Prediction problems that performed ensemble learning of any kind will be reviewed next.

Ahmed et al. [4] compare and explain Travel Time Prediction (TTP) methods, differentiating from works that use machine learning as a black box. They compared ten distinct prediction algorithms; from each, three belong to the TTP model type (hybrid models, ensemble tree-based learning, and deep neural networks). Moreover, the interpretation and understanding of the prediction algorithms derive from the application of Explainable Artificial Intelligence methods (LIME and SHAP), which can rationally explain model decisions and how different temporal and spatial features influence travel time. The work used three data sets (NextUp-1, NextUp-2, and PeMS) to evaluate and compare all models' prediction reliability and accuracy. The first two are available at the link [3], and the last is available at the link [13]. Their data preparation phase included identifying and removing outliers and duplicate data entries and filling missing values using interpolation. The experimental results demonstrate that the ensemble learning models based on Boosting used, the LightGBM and XGBoost, outperformed the other models for the three data sets.

Zhang et al. [68] used deep, machine, and ensemble learning methods applied to welding Robots; in detail, this paper proposed a model for welding robots to mutually predict the procedure parameters in the welding process, namely, bead geometry parameters and welding process parameters. The authors compared the prediction of four ML algorithms for these mentioned parameters; precisely, the algorithms picked were Convolutional neural network (CNN), CatBoost, and BP neural network, and were ensemble using an XGBoost model. The validation of the method used real welding processes data. The empirical results indicate that distinct deep and machine learning algorithms are more accurate on distinct kinds of prediction tasks; therefore, the selection of a method should be different regarding the task. The results also demonstrated the effectiveness of automatic parameters learning based on various learning methods.

ML algorithms are known for effectively recognizing hidden patterns regarding the financial market behaviour; thus, they have become valuable in financial applications, namely algorithmic trading. Carta et al. [14] proposed a method for risk-controlled trading to expand the current literature regarding algorithmic trading based on statistical and machine learning arbitrage. The presented approach is general and applicable to diverse assets and their underlying elements. The approach involved forecasting the asset returns with a highly heterogeneous ensemble of regression algorithms. The forecasting algorithms used were three ML algorithms: Support Vector Regression, Random Forest, and Light Gradient Boosting, and a statistical method, the Auto-Regressive Integrated Moving Average. The approach evaluation used historical data set regarding component stocks of the Standard & Poor's 500 index. Finally, the experimental results show that this

design outpaces competitive state-of-art trading strategies considered baselines by implementing in-depth risk-return analysis.

Many works used classification algorithms instead of regression methods. Almutairi et al. [8] implemented a novel predictive ensemble learner; specifically, a rule-based explainable ensemble classifier designed Ranked ensemble G-Rules (ReG-Rules). The goal was to present to the human analyst an explainable prediction model and enhance the predictive performance of ensemble learners with the explication. Therefore, the ReG-Rules combines classification rules from the base classifiers and presents a succinct and human-readable induced rule set defining the predictions made to the analyst. The ReG-Rules approach was evaluated theoretically concerning its computational complexity, qualitatively regarding the readability and complexity of the rule sets, and empirically with different benchmark data sets. Empirical results demonstrate that ReG-Rules is very accurate and has linear complexity. Cazañas-Gordón et al. [15] used ensemble learning for assessing retinal thickness, i.e., the automatic segmentation of retinal boundaries in optical coherence tomography images. The problem is a binary classification prediction problem. The evaluation of the segmentation accuracy of the novel method used two publicly accessible data sets, which contained some patients with severe retinal edema. The ensemble model was compared with two state-of-the-art models. The new algorithm outperformed the two widely referenced approaches at segmenting the retinal boundaries in pathological and ordinary images. Finally, a complete reliability analysis verified equivalency between the manual measurements estimated and the respective retinal thickness measurements obtained with the presented method. Deepthi and Jereesh [24] proposed a multi-source ensemble method to predict new drug-microRNA associations, which can be used to treat chronic human diseases. Specifically, they applied SVM, CNN, and principal component analysis (PCA) in their deep architecture-based classification. The feature creation stage used miRNA and drug similarities and experimentally verified drug-miRNA associations. Then, the PCA decreased the created feature dimensions. The resulting feature vectors were used to train the CNN in order to retrieve the high-level patterns and the hidden attributes of input data. Finally, the SVM classifier identifies the new drug-miRNA candidates by predicting the potential drug-miRNA associations after training. The evaluation of the model was made through a 5-fold cross-validation, miRNA-fixed local LOOCV, drug-fixed local LOOCV, and a global leave-one-out cross-validation. The data used in the paper is available at [23]. The empirical results and case studies demonstrate the efficacy of the proposed model in inferring novel drug-miRNA associations.

State of the art regarding ensemble learning has demonstrated a wide range of applications for ensemble learning, from predicting patients with covid to monthly weather forecasts. The systematic review demonstrated that the ensemble learning method tends to obtain better results compared to the individual models used as input. A gap analysis was carried out based on the section survey, which will be presented in the next section.

3.2 Gap analysis

Based on the systematic review of the literature presented in the previous section, the search gaps are identified and referred to in this section. The gap analysis performed can be found summarized in table 3.1.

Most of the articles reviewed in this chapter are part of the ITS area and present prediction algorithms. As shown in Table 3.1, most authors consider neither missing values nor weather conditions in their traffic forecasts, and some consider real case studies. Only two of the articles reviewed in Ensemble learning uses traffic prediction models. The first paper by Artin and others did not consider missing data, differing from the approach of this thesis. The other considered missing values in the data preparation phase; however, it only performed a simple interpolation for the values.

In sum, Duhayyimet al. [33], Ali et al. [7] and Artin et al. [9] worked with traffic forecasting considering weather conditions, and Dong et al. [31] and Ahmed et al. [4] considered missing data in their traffic prediction method. Ali et al. [7] differentiated from the rest by making a region-based crowd flow prediction instead of a network-based prediction. Even though Zhang et al. [67] made a network-based prediction, its method used spatial correlations between adjoining regions to enhance it. Feng et al. [37] present a hybrid method for traffic flow prediction combining global and local spatial correlation, which effectively detects the spatial-temporal correlation of the traffic flow. Some articles in the literature present some innovative technologies such as connected vehicles [5], Internet of Vehicles [43], [16], [60], [1], which are not be considered in this thesis. Lígia et al. [19] and Lígia et al. [18] studied the impact of using reversible routes in urban areas. Ahmed et al. [4] compare and explain Travel Time Prediction methods, differentiating from works that use machine learning as a black box. In opposition to the regression problem analysed, Deb et al. [22], Cazañas-Gordón et al. [15], Deepthi and Jereesh [24] and Almutairi et al. [8] dealt with classification problems, using methods utterly different from those reviewed. Pakdamanet al. [51] and Pakdamanet al. [50] ensemble models from existing models in other to predict monthly precipitation in Iran and Europe, respectively. While Carta et al. [14] applied ensemble models of prediction algorithms to the financial field, Zhang et al. [68] applied the same combination of methods to welding Robots. Finally, this dissertation carry out the future work presented in paper [6], which uses the same case study as this thesis. However, this work is expected to obtain better results because we perform an ensemble of models and fill missing values, the latter the goal of the work of Oliveira and Rocha [49] for the same data set.

3.3 Summary

The literature review showed that machine learning techniques have obtained good results in predicting network flow problems like this and are the most used at the moment. Works that perform ensemble learning obtained better results with the ensemble model than with any individual ensemble member. In addition, it is possible to obtain better results using fewer models as input in

the ensemble model. Finally, there are already works that used the case study of this thesis. One work has even applied regression algorithms to obtain flow patterns. Most of the articles reviewed are part of the area of Intelligent Transport Systems and work with forecasting algorithms. In addition, the literature review presented several data sets available for model validation.

Finally, this work differs from existing ones by performing an ensemble of prediction models for urban mobility leveraging on data provided by inductive-loop sensors and considering heterogeneous data from other sensors.

Table 3.1: The gap analysis table identifies the year of publication and whether the works use prediction algorithms (PA), connected vehicles (CV), Internet of things (IoV), regression algorithms (RA), and model ensemble (ME). The works that make forecasts are further classified in terms of whether or not they make traffic prediction (TP). In addition, it checks if the algorithms consider missing values (MV), weather conditions, and real data (RD). The works are classified as belonging to the ITS or not.

	Year	PA	CV	IoV	RA	ME	TP	MV	Weather	RD	ITS
Duhayyim et al. [33]	2022	*					*		*		*
Ali et al. [7]	2022	*			*		*		*	*	*
Dong et al. [31]	2022	*					*	*		*	*
Iranmanesh et al. [43]	2022	*		*			*				*
Deb et al. [22]	2022	*				*					
Pakdaman et al. [50]	2022	*			*	*					
Ahmed et al. [4]	2022	*			*	*	*	*		*	*
Feng et al. [37]	2022	*			*		*			*	*
Zhang et al. [68]	2022	*				*				*	
Carta et al. [14]	2021	*			*	*					
Artin et al. [9]	2021	*			*	*	*		*	*	*
Zhang et al. [67]	2021	*					*			*	*
Chen et al. [16]	2021	*		*			*			*	*
Tian et al. [60]	2021	*		*	*		*				*
Almutairi et al. [8]	2021	*				*					
Deepthi and Jereesh [24]	2021	*				*				*	
Cazañas-Gordón et al. [15]	2021	*				*				*	
Mallah et al. [5]	2020	*	*		*		*				*
Oliveira and Rocha [49]	2020	*			*			*		*	*
Pakdaman et al. [51]	2020	*			*	*					
Lígia et al. [18]	2020										*
Lígia et al. [19]	2019										*
Abdellah et al. [1]	2019	*		*	*		*				*
Alam et al. [6]	2017	*			*		*			*	*

Chapter 4

Methodological Approach

This chapter contains the methodological approach. In detail, it includes the problem formalization, the description of the materials (including the description of the actual data sets, their pre-processing, and their correlations), and the approach followed to create the prediction models and the microscopic simulation (including the evaluation metrics of both).

4.1 Problem formalization

The problem addressed in this dissertation arises from the context of smart cities, where traffic patterns are predicted based on information related to mobility. Therefore, based on data from inductive sensors on some road segments, the flow on the entire road is predicted. The ultimate goal is to provide the traffic prediction of the network based on historical data in a visual tool, a simulation. Therefore, the problem uses spatio-temporal data and belongs to the traffic flow prediction (TFP) type, which corresponds to a governmental perspective. Within TFP problems, it is classified as a network-based problem (see section 2.1).

The simulated model is descriptive and speculative. The model is descriptive as the goal of the simulation is to replicate the behaviour described by the sensor readings present in the data source. To this effect, the data was aggregated hourly by each sensor and compared with the simulation's observed value in the given sensor locations. The simulation is also speculative given the sensor observed readings are for few and fixed locations, and there can be multiple entries and exits between these sensors. As such, the traffic in the zones without sensors must be speculated using only the link count data mentioned.

The case study that validates the solution refers to *"Via de Cintura Interna"* (VCI), an inter-city highway network with considerable traffic, in the period between 2013 and 2015. With the historical information of the VCI flow and the meteorological data, the goal is to predict the flow for the sensor positions that the data set has information about and in the other segments that any information exists. In highways like the VCI understudy, the traffic is very complex because it

connects several regions containing a significant percentage of the population; precisely, in 2015, VCI contained 30% of the inhabitants of the Greater Metropolitan Area of the city Porto [6]. In addition, other factors make traffic forecasting a difficult task, such as the tourist pressure, the citizens' increasingly active daily lives, the growth of cities, the complexity of urban life, and other unexpected factors like the pandemic. The existing materials for solving the problem will be described in section 4.2. In short, the problem addressed in this work is the traffic forecast in the entire VCI with historical information regarding some inductive sensors and the weather, and the research question to be answered is the following: "How can traffic flow patterns be identified and predicted in urban networks?".

4.2 Materials and pre-processing

This subsection will describe the two data sets used. The first is related to mobility in VCI in the period between 2013 and 2015, and the second regards the meteorological data for the same dates. It also describes the pre-processing made to the data sets.

4.2.1 Traffic data set description

Regarding mobility data, the prediction models will use a data set with actual traffic data provided by 26 inner-loop sensors installed on the VCI highway in 2013 in Porto, Portugal [49]. The VCI has a ring-like shape, 21 km of total length, and two directions separated by a lane separator. Sensors are positioned transversely in the lane, under all pathways in both directions. The series of inductive-loop sensors that collect data about traffic throughout the inner-ring of the VCI in Porto extract information such as the type and the number of vehicles passing through every 5 minutes, regarding the specific segment they are situated.

The historical data set contains information regarding the mobility of the VCI and includes data for part of the year 2013 and the complete 2014 and 2015. If the data set had all instances related to the two directions of the 26 sensors mentioned, there would be 15,515,136 instances, given that the data set does not include data from January and February 2013. However, there are only 13,716,180 entries, with 12% of the instances missing. Table 4.1 contains the analysis of the missing values, including the number of missing instances, the expected number of entries, and the percentage of missing instances in the data set by month and year. The missing values refer to sensors that do not present information of any kind for specific time intervals. The data set does not have null values for any field or input.

The data set fields are described in detail in table 4.2. All entries in the *Agg_period_len_mins* field have a value of 5 as the data is aggregated in 5-minute periods. The total volume field always corresponds to each class's volume sum in the data set. The number of lanes in the segments for which the data set has information, represented by the *Nr_lanes* field, did not change over the three years under study. The position of the sensors for which the data set has information is identified with circles and numbers above a Google Maps map in Figure 4.1. The thirteen sensors positioned on VCI have ID numbers 5 to 12, 19 to 21, and 24 to 25. There are as many sensors in the VCI as

Table 4.1: Analysis of the missing values, including the number of missing instances, the expected number of entries, and the percentage of missing instances in the data set by month and year.

Year	Month	Missing Values	Expected number Nº	Missing %
2013	Mar	92883	464256	20%
	Apr	62515	449280	14%
	May	73733	464256	16%
	June	63265	449280	14%
	July	66950	464256	14%
	Aug	74109	464256	16%
	Sept	98323	449280	22%
	Oct	105599	464256	23%
	Nov	108516	449280	24%
	Dec	125879	464256	27%
Sum 2013		871772	4582656	19%
2014	Jan	103460	464256	22%
	Feb	98167	419328	23%
	Mar	122452	464256	26%
	Apr	105627	449280	24%
	May	84904	464256	18%
	June	48604	449280	11%
	July	39638	464256	9%
	Aug	17860	464256	4%
	Sept	6959	449280	2%
	Oct	11924	464256	3%
	Nov	10500	449280	2%
	Dec	6978	464256	2%
Sum 2014		657073	5466240	12%
2015	Jan	14263	464256	3%
	Feb	9977	419328	2%
	Mar	8337	464256	2%
	Apr	35136	449280	8%
	May	44148	464256	10%
	June	45390	449280	10%
	July	35876	464256	8%
	Aug	32802	464256	7%
	Sept	13432	449280	3%
	Oct	11025	464256	2%
	Nov	9215	449280	2%
	Dec	10510	464256	2%
Sum 2015		270111	5466240	5%
Sum		1798956	15515136	12%

outside it; therefore, the ids missing are placed outside VCI. There is usually more than one input and output road between sensors. Thus, there is no way to know where the cars entered and exited the highway.

By analyzing the data set, it seems to be relevant to study not only the total number of vehicles,

Table 4.2: Original data set fields [6].

Field Name	Type	Description
Agg_by_Lane _BundleID	Integer	ID that combines the lane Direction and the lane id.
Agg_Id	Integer	Street ID.
EquipmentID	Integer	Sensor unique ID.
Agg_period _start	Date & time	Time interval when the data was gathered in "Year-Month-Day Hour:Minutes:Seconds" format.
Agg_period _len_mins	Integer	Corresponds to the time between measurements made by the sensors. It is invariably of 5 minutes.
NR_Lanes	Integer	Number of lanes in the carriageway segment.
Lane_Bundle _Direction	String	The direction of the lane where the data was measured. Possible values: "D" or "C", corresponding to decreasing and ascending, respectively.
Total_VOL	Integer	Total number of vehicles that go over the sensor.
AVG_SPEED _ARITHMETIC	Decimal	Arithmetic average speed of vehicles that go over the sensor.
AVG_SPEED _HARMONIC	Decimal	Harmonic average speed of vehicles that go over the sensor.
AVG_LENGTH	Decimal	Average length of the vehicles that go over the sensor.
AVG_SPACING	Decimal	Average spacing separating vehicles that go over the sensor.
Occupancy	Decimal	Occupancy in respective Lane_NR and Lane_Direction.
LIGHT_ VEHICLE_RATE	Decimal	This percentage is the ratio of light vehicles that go over the sensor.
Vol_class_A	Integer	Nr of vehicles of class 1 or A that go over the sensor.
Vol_class_B	Integer	Nr of vehicles of class 2 or B that go over the sensor.
Vol_class_C	Integer	Nr of vehicles of class 3 or C that go over the sensor.
Vol_class_D	Integer	Nr of vehicles of class 4 or D that go over the sensor.
Vol_class_0	Integer	Nr of unidentified vehicles that go over the sensor.
AXLE _CLASS_VOL	String	Number of vehicles per vehicle axis number.

but also this number per vehicle class (A, B, C, D, or 0). Table 4.3 contains the analysis of the minimum, maximum and average values of these target variables: the total volume of vehicles and the volume of vehicles of each class by sensor. The minimum volume of vehicles of class A, B, C, D, and 0 was hidden from the table because it is zero in all sensors. Regarding the variables to be predicted, the minimum value in the data set is 0, and the maximum value is 885.

Figure 4.1: Location of the sensors in Google Maps.

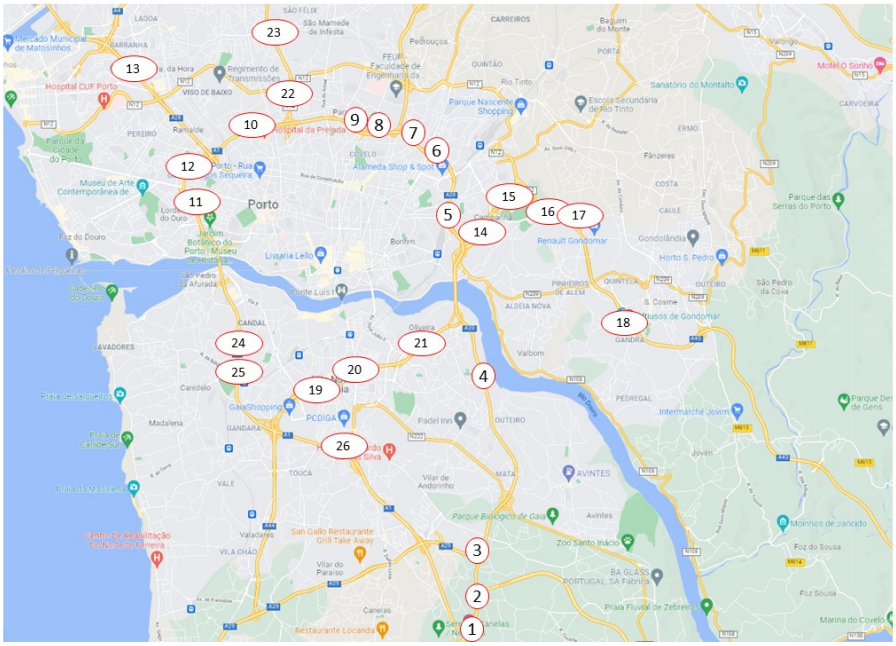


Table 4.3: Analysis of the minimum, maximum and average values of the target variables. The table contains the minimum total volume (tv m), the maximum total volume (tv mx), the total volume of cars of classes A (cA mx), B (cB mx), C (cC mx), D (cD mx) and unidentified classes (c0 mx) as well as the mean (mn) for the total volume and the volume for the same classes for each sensor (s id) and direction (d). In the column direction, the value of 0 corresponds to the ascending direction and 1 to the descending direction.

s id	d	tv m	tv mx	cA mx	cB mx	cC mx	cD mx	c0 mx	tv mn	cA mn	cB mn	cC mn	cD mn	c0 mn
1	0	0	301	105	281	29	21	84	62.9	0.3	57.5	4.4	0.5	0.2
1	1	0	225	51	223	28	35	91	55.7	0.3	50.3	4.3	0.6	0.3
2	0	0	885	83	291	33	22	870	75.4	0.3	66.6	4.9	0.5	3.1
2	1	0	248	51	240	27	41	63	65.7	0.3	60.5	4.1	0.7	0.2
3	0	0	381	59	354	41	23	51	100.3	0.5	92.6	6.4	0.8	0.1
3	1	0	328	47	310	36	39	6	96	0.5	89.4	5.2	0.9	0
4	0	0	405	72	383	38	26	79	100.4	0.4	93.2	5.8	0.9	0.2
4	1	0	370	47	356	42	37	96	95.4	0.4	88.9	5.3	0.7	0.1
5	0	1	603	86	574	46	28	175	183.6	1.1	171.2	7.2	1.1	3.1
5	1	0	632	45	606	43	40	119	181.7	1.2	171.9	6.8	0.8	1.1
6	0	0	598	86	573	54	25	184	201.6	1.4	176.1	7	0.7	16.5
6	1	0	531	46	509	48	42	153	192.6	1.6	176.9	7.4	1.1	5.6

Continued on next page

Table 4.3 – continued from previous page

s id	d	tv m	tv mx	cA mx	cB mx	cC mx	cD mx	c0 mx	tv mn	cA mn	cB mn	cC mn	cD mn	c0 mn
7	0	0	614	61	591	54	32	129	228	1.5	216.7	7.4	1.4	1.1
7	1	0	591	45	575	51	41	134	231.4	1.6	219.5	7.5	1	1.7
8	0	0	725	58	707	33	25	223	238.9	1.7	227.7	4.8	0.9	3.9
8	1	0	685	63	667	41	15	279	241	1.7	227.4	4.5	0.9	6.4
9	0	0	681	53	663	33	23	124	253.3	2	243.1	4.9	1.1	2.2
9	1	0	631	37	605	32	13	154	234.8	1.7	222.4	4.4	0.8	5.5
10	0	0	412	19	405	10	5	75	118.9	1.2	114.3	0.4	0	3
10	1	0	407	27	392	33	16	101	184.7	1.3	174.2	6	1.2	2.1
11	0	2	445	59	422	61	21	111	175.7	1.3	161.3	10	0.6	2.4
11	1	1	552	79	527	53	24	130	203.8	1.6	189.4	10	1.3	1.4
12	0	1	533	52	519	49	26	104	228.3	1.8	213.9	9.8	1.1	1.7
12	1	0	570	89	556	52	25	126	232.1	1.9	216.3	10.3	1.4	2.2
13	0	0	142	111	102	33	10	20	25.5	0.1	19.2	5.9	0.3	0
13	1	0	274	118	100	38	9	111	25.4	0.1	19.3	5.6	0.2	0.1
14	0	0	357	23	343	15	13	104	59.5	0.5	57.4	1.2	0.2	0.2
14	1	0	290	25	284	16	11	86	64.5	0.5	61.7	1.7	0.2	0.4
15	0	0	365	17	356	17	11	93	67.1	0.6	64	1.8	0.1	0.5
15	1	0	332	8	324	15	12	10	13	0.1	12.4	0.4	0	0.1
16	0	0	294	22	286	13	13	77	64	0.4	62.2	1.2	0.2	0
16	1	0	342	23	339	12	14	51	65	0.4	63.2	1.1	0.2	0
17	0	0	213	21	204	12	13	76	49.8	0.3	48.4	0.9	0.1	0
17	1	0	252	19	251	10	12	17	50.7	0.3	49.3	0.9	0.2	0
18	0	0	190	18	187	11	9	21	35.7	0.2	34.6	0.8	0.1	0
18	1	0	215	47	214	10	9	31	37.1	0.3	35.9	0.7	0.2	0.1
19	0	0	356	28	347	21	14	78	96.6	0.7	94.6	1.2	0.2	0.1
19	1	0	875	49	348	305	51	318	104.2	0.7	101.7	1.2	0.2	0.5
20	0	0	360	42	354	15	14	94	86.3	0.7	84.2	1.1	0.2	0.2
20	1	0	321	37	317	16	19	52	83	0.6	81.1	1.1	0.2	0
21	0	0	355	30	349	20	13	69	81.2	0.5	79	1.3	0.3	0.1
21	1	0	341	39	334	19	14	74	85.3	0.6	83.2	1.1	0.3	0
22	0	1	424	46	397	35	15	86	143.4	1.4	133.6	5.3	1	2.2
22	1	0	441	39	428	33	16	114	143.6	1.4	132.7	6	0.8	2.7
23	0	0	368	47	355	28	10	78	132.3	1.2	125.9	4.6	0.6	0.1
23	1	0	379	31	370	29	10	100	120.6	1	113.9	5	0.5	0.3
24	0	0	577	54	548	50	31	120	183.6	1.2	171.2	9.5	0.8	1

Continued on next page

Table 4.3 – continued from previous page

s id	d	tv m	tv mx	cA mx	cB mx	cC mx	cD mx	c0 mx	tv mn	cA mn	cB mn	cC mn	cD mn	c0 mn
24	1	0	498	99	471	51	26	104	182.1	1.3	169.3	9.7	1.4	0.4
25	0	0	474	59	457	44	23	89	189.2	1.2	176.8	9	1.3	0.9
25	1	0	514	101	489	51	24	109	193.3	1.3	179.9	10.6	1	0.6
26	0	0	361	38	350	32	18	84	109.3	0.6	103.6	4.5	0.5	0.2
26	1	0	305	56	289	31	25	110	101.2	0.5	82.9	4	0.4	13.4

Regarding the network topology, there are 26 entrances and 24 exits in the ascending direction. In the descending pathway, there are 24 entrances and 21 exits. Figure 4.2 numbers the inputs and outputs relative to the ascending direction, showing the direction of the road through the arrows. Figure 4.3 shows the same for the opposite direction. The figures show the location of inputs and outputs through the nomenclature Ox and Dy, corresponding to an entrance and exits, respectively, the origin and destination in the OD matrix that will be describe in the next chapter, in section 4.3.2.1. Figure 4.2 demonstrates that only the segment between sensors 8 and 9 contains a single exit or entry in the ascending direction. There are always one or more exits and entrances in the remaining segments. In figure 4.3, which shows the location of the entrances and exits in descending direction, it is possible to observe that two segments contain only one entry or exit: between sensors 8 and 9 and between sensors 11 and 12.

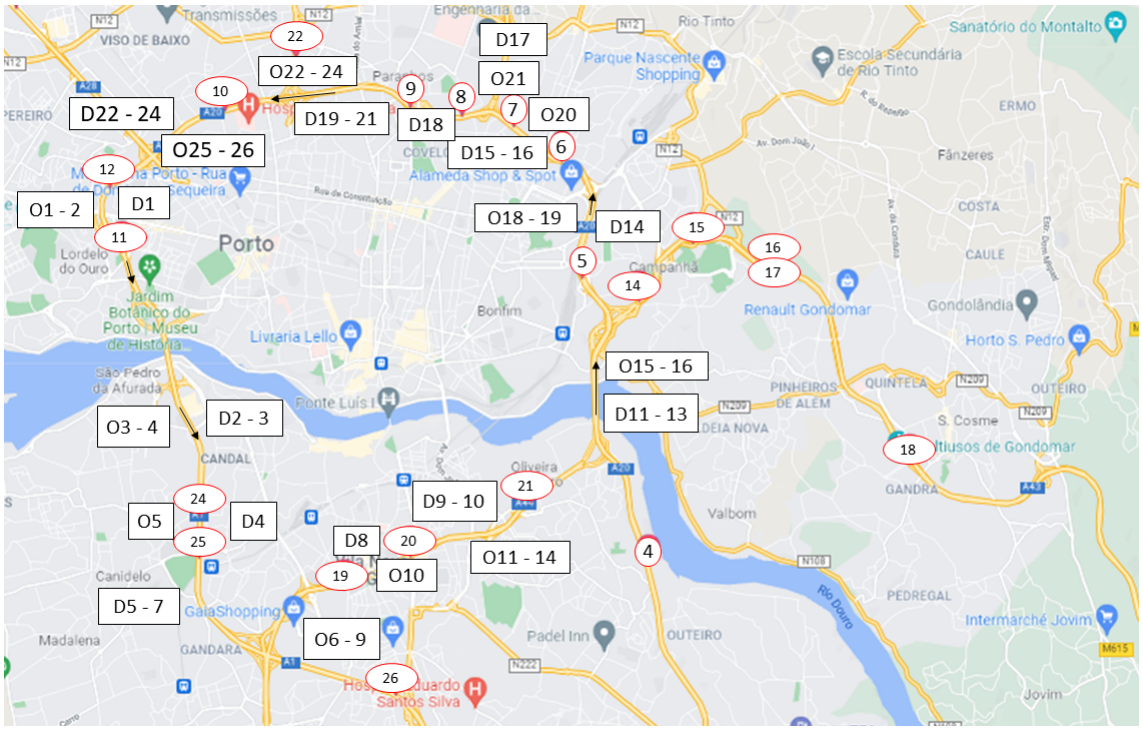
4.2.2 Weather data set description

In addition to mobility data, the prediction models also used meteorological data extracted from the Iowa Environmental Mesonet [63]. The Iowa Environmental Mesonet source contains data from various points across the world. Since the VCI has a high extension, data from 2 meteorological stations were used, one in Porto and another in Ovar. Table 4.4 enumerates all non-null fields from the weather data set mentioned, and includes, for each one, information about the field id, description, format/scale, minimum, maximum, and count of unique values.

The field *station* only has the values 'LPPR' and 'LPOV', corresponding to the id of the station in Porto and Ovar, respectively. The field *p01i* represents the value of precipitation measured during an hour for the period from the observation time to the time of the last hourly precipitation reset. The possible values for the sky coverage that report the cloud amount, at any level (*skyc1*,*skyc2*, *skyc3*, *skyc4*), are:

- NSC: No Significant Clouds,
- FEW: Few (1-2 oktas),
- SCT: Scattered (3-4 oktas),
- BKN: Broken (5-7 oktas),

Figure 4.2: Location of the entrances and exits between sensors in Google Maps in the ascending direction.



- OVC: Overcast (8 oktas),
- VV: Sky obscured.

Table 4.5 shows the number and percentage of missing values for the fields that, from March 2013 to the end of 2015, have some missing values. Only the fields *station*, *valid*, *lon*, *lat*, *elevation*, *p01i* and *metar*, did not show any null value.

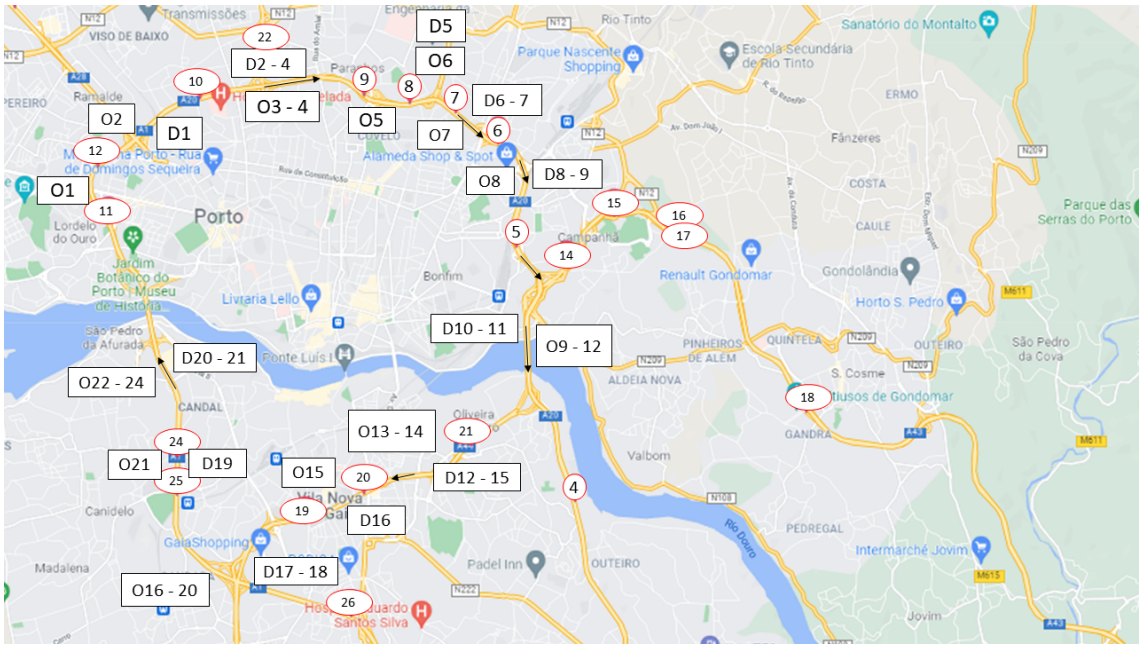
Some of the features enumerated above were not used for the prediction due to the high number of missing values. All fields with more than 35% of missing values were eliminated from the data set. An additional test was made to verify if the *skyc1* field could improve the predictions even with 34% of the missing values. The complete pre-processing is described in the next subsection.

In addition to the fields shown, the meteorological station also provides the *sea Level Pressure* in millibar, the *ice Accretion* over 1, 3, and 6 Hours in inches, the *peak Wind Gust* (from PK WND METAR remark) in knots, the *peak Wind Gust Direction* (from PK WND METAR remark) in degrees, the *peak Wind Gust Time* (from PK WND METAR remark), and the *snow-depth*. However, these fields were not considered, as they are null in all timestamps for the chosen stations.

4.2.3 Pre-processing of the data set

The pre-processing of the data set involves using feature engineering to enhance the accuracy of the ML models. The first pre-processing task concerns the handling of strings. The fields

Figure 4.3: Location of the entrances and exits between sensors in Google Maps in the descending direction.



Agg_period_start (from the traffic data set) and *valid* (from the meteorological data set) encode time using a string type. In addition, the fields *Axle_Class_Vol*, *Lane_Bundle_Direction* (from the traffic data set) and *station*, *skyc1*, *skyc2*, *skyc3*, *skyc4*, *wxcodes*, and *metar* (from the meteorological data set) are also strings. Given that the string type forbids the data set usage by machine learning models, a new encoding or deletion is applied to the mentioned fields [49]. Among the possible encodings are Integer Encoding and One-Hot Encoding [12]. On the one hand, Integer Encoding is a procedure that takes categorical variables and transforms them into a numerical representation. This procedure can be used to encode the day of the week and the direction of the lane. On the other hand, One-Hot Encoding is a procedure that also takes categorical variables; however, it transforms them into a binary numerical representation only. This procedure can also be used to encode the day of the week but using seven fields instead of one.

The following procedures have been then applied to the original data set regarding mobility:

1. Add a field with the day of the week, with integer values varying between 0 and 6. The lowest value corresponds to Monday, the highest to Sunday, and the remaining following the standard order of this type of data;

2. Add a field with the classification of the day as a holiday. The field is of type Boolean but represented by binary integers, with 0 representing false and 1 representing true;

3. Convert the *Lane_Bundle_Direction* field to an integer, with 0 corresponding to the ascending direction and 1 to the descending direction;

Table 4.4: All not null fields of the weather data set.

Field	Description	In	min	max	unique
station	Site identifier	string	LPOV	LPPR	2
lon	Longitude of the station	degrees	-8.6833	-8.6459	2
lat	Latitude of the station	degrees	40.9159	41.2333	2
elevation	Elevation of the station	meters	17.0	73.0	2
valid	Time interval when the data was observed	data & hour	2013-03-01 00:00	2015-12-31 23:30	50070
tmpf	Air temperature at 2 meters	Fahrenheit	26.6	100.4	43
dwpf	Dew point temperature at 2 meters	Fahrenheit	6.8	69.8	37
relh	Relative Humidity	Percentage	10.47	100.0	600
drct	Wind direction from geographic north	degrees	0.0	360.0	38
sknt	Wind Speed	knots	0.0	32.0	35
p01i	One-hour precipitation	inches	0.0	0.0	1
alti	Pressure altimeter	inches	28.85	31.51	70
vsby	Visibility	miles	0.06	6.21	63
gust	Wind Gust	knots	13.0	54.0	36
skyc1	Sky Level 1 Coverage	string			7
skyc2	Sky Level 2 Coverage	string			5
skyc3	Sky Level 3 Coverage	string			6
skyc4	Sky Level 4 Coverage	string			5
skyl1	Sky Level 1 Altitude	feet	0.0	10000.0	55
skyl2	Sky Level 2 Altitude	feet	100.0	10000.0	54
skyl3	Sky Level 3 Altitude	feet	300.0	7500.0	52
skyl4	Sky Level 4 Altitude	feet	1500.0	4900.0	26
wxcodes	Present Weather Codes	string			86
metar	Unprocessed reported observation	METAR format			65501
feel	Apparent Temperature (Wind Chill or Heat Index)	Fahrenheit	19.98	98.28	366

4. Delete the fields *Axle_Class_Vol*, *Light_Vehicle_Rate*, *Occupancy*, *Avg_Length*, *Avg_Spacing*, *Avg_Speed_Harmonic*, *Avg_Speed_Arithmetic*, *Agg_by_Lane_BundleId*, and *Agg_Id*;

5. Split the *Agg_period_start* field of type string into the year, month, day, hour, and minutes of type integer.

6. Encode the *Agg_period_start* field in four different variables inspired by the Fourier Transform to represent the period of the year and the period of the day. The variables were created by the formulas 4.1 to 4.4:

$$x_{year} = \sin\left(\frac{2 * \pi * month}{12}\right) \tag{4.1}$$

Table 4.5: Number and percentage of missing values per field. The table does not include the fields that do not contain null values nor the ones that do not have not null values.

field	Number of missing values	Percentage of missing values
dwpf	7	0.010684739 %
alti	7	0.010684739 %
tmpf	8	0.01221113 %
vsby	15	0.02289587 %
sknt	137	0.209115609 %
feel	147	0.224379522 %
relh	147	0.224379522 %
drct	6837	10.43593736 %
skyc1	22460	34.28274873 %
skyl1	25249	38.53985408 %
skyl2	43339	66.1522728 %
skyc2	43339	66.1522728 %
skyl3	59364	90.61269347 %
wxcodes	55149	84.17895412 %
skyc3	59363	90.61116708 %
skyl4	65370	99.78019965 %
skyc4	65370	99.78019965 %
gust	63857	97.47076961 %

$$y_{year} = \cos\left(\frac{2 * \pi * month}{12}\right) \tag{4.2}$$

$$x_{day} = \sin\left(\frac{2 * \pi * minute}{24 * 60}\right) \tag{4.3}$$

$$y_{day} = \cos\left(\frac{2 * \pi * minute}{24 * 60}\right) \tag{4.4}$$

In these equations, the month is the number of months that elapsed since the beginning of the year, and the minute represents the number of minutes passed since the beginning of the day. Finally, the variables x_{year} and y_{year} represent the time of the year when the extraction occur by the sensor, while the variables x_{day} and y_{day} encode the time of the day. These four variables used for the date encoding were proposed in another paper [49] for the same data set. The authors consider that the right time encoding is cyclical (i.e., 23:59 must be followed by 00:00) and allows the machine learning models to generalize correctly.

As can be seen, the procedure above included creating redundant fields to encode the time interval. Since the aggregation time is 5 minutes, the seconds in the *Agg_period_start* field are irrelevant as they are always zero; therefore, the seconds are removed in both encodings. In opposition to the work by Oliveira e Rocha [49], which only classified the days as working days or not, this work's methodology added two fields to the data set with additional information. The first field contains the day of the week, and the second identifies the day as a holiday or not. The

final fields of the mobility data set are present in Table 4.6; specifically, the inputs used are between the "Mobility data set" and the "Meteorological data" row and the targets below the Targets line.

A second pre-processing of the mobility data set was done. In this new pre-processing methodology, all the steps described before were done the same way, except the processing of the day of the week made in step 1. The new feature engineering uses the One-Hot Encoding method to encode the day of the week instead of Integer Encoding. Therefore, instead of having a field named day of the week that varies from 0 to 6, there are seven fields corresponding to each of the days of the week with binary values (1 or 0).

In addition, a third pre-processing task was performed, adding the fields related to the meteorological data. While the aggregation period for mobility data is 5 minutes, meteorological data has entries every 30 minutes. Therefore, it was considered that the meteorological data did not vary significantly during 30 minutes. In this way, the information from each entry of the weather data set was repeated five times at the junction. So, it can be considered that the percentage of missing values is the same, despite the number of missing values being six times higher. Numeric fields have been kept unchanged or were deleted due to the high percentage of missing values; however, nominal fields have changed. This pre-processing included the following procedures related to meteorological data:

1. Split the *valid* field of type string into the year, month, day, hour, and minutes of type integer. The processing done to the *valid* field in the meteorological data, which corresponds to the *Agg_period_start* field in the mobility data, was the same as the one made in the mobility data set.

2. The *station* field with the LPOV and LPPR values took on binary values instead of nominal values, 0 and 1, respectively.

3. The field related to Sky coverage at level 1, *skyc1*, with possible values NSC, FEW, SCT, BKN, OVC, and VV started to take a value from 0 to 5, with zero corresponding to no Significant Cloud (NSC) and five to Sky obscured (VV).

4. Delete the nominal fields *wxcodes* and *metar*.

5. Delete the eight fields that are always null: *peak wind gust*, *peak wind drct*, *peak wind time*, *mslp*, *ice accretion 1hr*, *ice accretion 3hr*, *ice accretion 6hr*, and *snow-depth*.

6. Delete the numerical fields *skyl1*, *skyl2*, *skyc2*, *skyl3*, *skyc3*, *skyl4*, *skyc4*, and *gust*.

7. Keep the numerical fields *lon*, *lat*, *elevation*, *p01i*, *tmp*f, *feel*, *dwpf*, *alti*, *relh*, *drct*, *vsby*, and *sknt*.

8. Fill in the null values with the previously observed value, i.e., the meteorological values of the previous half hour are assigned to the missing values.

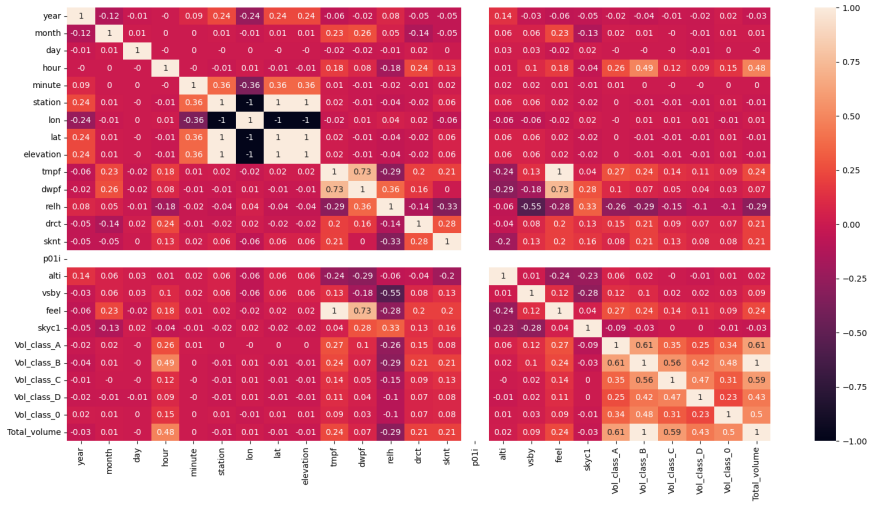
9. Join this data set with the first pre-processed data set by the date and time fields.

The prediction algorithms will start by using all features, and the features that do not contribute to a more accurate prediction will be eliminated. The results using the three different pre-processing and applying feature engineering will be compared in chapter 6. The fields resulting from the third pre-processing of the data, which joins the mobility data to the meteorological data, are found in table 4.6, with the respective name, type and description.

4.2.4 Data set correlations

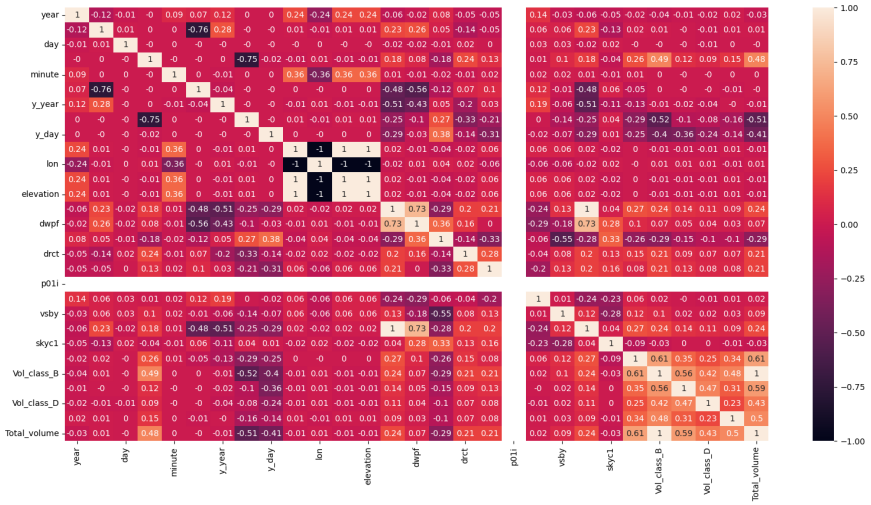
The first step made was the Gaussian hypothesis testing. Given that the p-value of all variables of the final data set was always 0 (lower than 0.05), the null hypothesis is rejected; therefore, it is assumed that the distribution of the variables is not normal/gaussian. The correlations of the variables can be seen in the Heat Maps 4.5 and 4.4.

Figure 4.4: Heat map with the correlations of the mobility data.



There are some very predictable correlations, such as the high correlation between the *month* and the *x_year* field, as well as the *hour* and the *x_day*, given that the *x_year* and the *x_day* come from calculations made with the *month* and *day* fields, respectively. There is also a high correlation between the feature *Vol_class_B* and *total_volume* since most of the volume corresponds to cars and not to other types of vehicles. For feature *Vol_class_A*, the correlation is higher than 0.01 for all variables in the mobility data set, except for the fields minutes and day. These two variables did not show a correlation higher than 0.01 for any of the vehicle volumes. For features *Vol_class_B*, *Vol_class_C*, *vol_class_D*, and *total_volume*, the correlation is higher than 0.01 for all variables in the mobility data set, except for *x_year*, *minute*, and *day*. Finally, for feature *vol_class_0*, therefore, the prevision of unidentified vehicles, the correlation is higher than 0.01 for all variables in the mobility data set, except for *y_year*, *minute*, and *day*.

Figure 4.5: Heat map with the correlations of the meteorological data.



In Heat Maps 4.5 and 4.4, the features *p01i* and *Agg_period_len_mins*, corresponding to the one-hour precipitation value and the time between measurements made by the sensors, respectively, constantly had the vales 0.0 inches and 5 minutes; therefore, there is no correction with these variables and any other. Since longitude, latitude and elevation correspond to the characteristics of the weather station, there is a strong association between the station id and these three variables. For feature *Vol_class_A*, the correlation is higher than 0.01 for all variables in the meteorological data set, except for the fields *p01i*, *elevation*, *lat* (latitude), *lon* (longitude), *station*, and *day*. For features *Vol_class_B*, *vol_class_0*, and *total_volume* the correlation is higher than 0.01 for all variables in the meteorological data set, except for the fields *p01i*, *minute*, and *day*. When observing the correlations in the meteorological data set, for the column *Vol_class_C*, the correlation is higher than 0.01 for all variables in the data set, except for the *skyc1*, *alti*, *p01i*, *minute*, *day* and *month*. Finally, for feature *vol_class_D*, the correlation is higher than 0.01 for all variables in the meteorological data set, except for the fields *skyc1*, *p01i*, and *minute*.

4.3 Methods

As mentioned before, the main concern of this work is the prediction of traffic flow using machine learning methods. Alam et al. [6] already applied Sequential Minimal Optimization Regression, Regression Tree (M5Base classifier), and Linear Regression on the data used here, so it would be interesting to explore other algorithms.

Figure 4.6 illustrates the methodology followed, and summarizes the steps necessary to achieve the objective of obtaining the flow prediction in the segments of the urban network that contains sensors. The methodology does not include the steps for creating the simulation.

Figure 4.6: Summary of the methodology to create prediction models

The first step regards the data analysis and the detection of the missing values, including identifying the 12% of missing values in the mobility data mentioned in the literature. The analysis of missing values was already discussed in subsections 4.2.1 and 4.2.2 for mobility and meteorological data respectively.

The second step is related to data preparation, it comprises dealing with missing values and adding or removing fields to the data set, and that was already described in subsection 4.2.3 for the current case study. In the problem in question, there are three types of missing values: i) some sensors have missing values at unpredictable time intervals; ii) while the meteorological data set contains all temporal instants and some null values, the mobility data set does not have null values, but some time intervals do not exist; iii) given that the data set only has information on some network segments, there is a lack of information regarding the remaining segments of the road. The proposed solution will handle the last two types of missing values. Although the first type of missing values is not addressed, short-term traffic prediction models on the VCI and outside can be used to predict missing values in the mobility data set. Regarding the second type of missing values, the missing data from the meteorological data set are filled in with the previous value. Finally, The missing values due to the lack of information regarding the remaining road segments are not treated at this stage but later.

After the data preparation, the next step is implementing, training, and tuning the prediction models. The traffic prediction models are trained using 80% of the data set and tested with the remaining 20%. If there is a tunning of the parameters and hyper-parameters, the percentage of the train data set used is 60%, and the validation and the test data set have the same size, using the 40% of the remaining data. The next step regards the ensemble of the accurate models that is carried out to improve the results. Specifically, the simple averaging, the weighted average, and stacking were used. This will be further detailed in section 4.3.1.

The next step is the evaluation of the model, which will be carried out with the aid of microscopic simulation. After obtaining the ensemble model, there will be a traffic extrapolation across the entire highway based on the prediction for the sensors the data set has information since the only way to create a simulation is to provide a Origin-Destination matrix as input. Specifically, in the case study, the origin and destination of all the vehicles that pass through the VCI must be established. There is usually more than one input and output road between sensors. Thus, there is no way to know where the vehicles entered and exited, so it is necessary to speculate. What is exact is that, for example, a vehicle that enters before sensor eleven and leaves after sensor ten will pass sensors eleven, twelve, and ten if it is going in the descending direction (i.e., driving on the

inner ring), and so on and so forth. Therefore, it is possible to get the count of vehicles passing the sensors according to an OD matrix. The process of evaluating the microscopic simulation results using the created OD matrix is done by comparing the results of the real radars located on VCI to what would be virtual radars positioned on the SUMO simulation mirroring the real ones, and is described in detail in subsection 4.3.2.

The next sub-sections of this section will describe in more detail the core tasks referred to in the presented methodology, namely the ensemble of models, the microscopic simulation and the selection of the evaluation metrics (subsection 4.3.3) for both the prediction models and the microscopic simulation.

4.3.1 Ensemble model

The creation of the ensemble model involves obtaining the training, validation, and test sample. Therefore, the data set is split into train and test sets. Then the whole train set is split into train and validation sets. In the second phase, the models are created with the hyper-parameters, which obtain the best results. The models are individually trained and evaluated with the MAE metric, and the performance of each one is stored. The ranking of the models is calculated through the individual evaluation of the models using the training and validation data. Alternatively, the best previously created models are loaded in the second phase; however, it is necessary to provide the MAE obtained with the model. The ensemble model is created in the third phase, passing the models and the ranking as an argument, both obtained in the previous phase. With the ensemble model created, this model is fitted with the training and validation data set (also referred to as the whole training data set). Furthermore, the predictions with the test set are made and evaluated.

4.3.2 Microscopic simulation

Simulators cannot perform a micro-simulation based on segment information, only with Origin-Destination (OD) matrices or routes. Implementing a simulation implies the creation of a method to transform the information regarding the traffic in some sensors into OD matrices accepted by simulators. This subsection presents an innovative method to create OD matrices and the implementation of an OD matrix in a simulator.

4.3.2.1 Exact model to obtain the OD matrix

After obtaining the prediction of the volume of vehicles that pass each sensor, it is necessary to create an OD matrix that indicates the trajectories of the automobiles. One can also aggregate the data for a day of the week that is not a holiday and make an OD matrix representing an hour of a typical weekday. Therefore, an exact model was created, which will be described in this section. Although it is possible to predict the volume of cars in all sensors, the focus will be only on using predictions regarding the sensors located in the VCI. The exact model has the cells of the OD matrix as decision variables, namely, the cell that indicates the number of vehicles with origin in 1 and destination in 1, among others. Therefore, the decision variables O_iD_j represent the number

of vehicles that enter the VCI at origin i and leave the VCI at destination j. Auxiliary variables were created to facilitate the formulation. Formally:

Set of sensors: $S = 11,24,25,19,20,21,5,6,7,8,9,10,12$

Decision variables: O_iD_j – number of vehicles that enter the VCI at origin i, and leave the VCI at destination j

Auxiliary variables: $sensor_j$ - number of vehicles that pass through sensor j $(j \in S)$

$$\text{maximize} \sum_{j \in S} sensor_j \tag{4.5}$$

The objective function for this problem is to maximize the number of vehicles that pass the sensors in order to minimize the error, as stated in expression 4.5. Regarding the constraints, the model has three main types of constraints. The model is subject to:

$$sensor_j = \sum O_iD_j \quad that \quad influence \quad variable \quad sensor_j, \quad \forall j \in S; \tag{4.6}$$

$$sensor_j <= observedFlowInSensor, \quad \forall j \in S; \tag{4.7}$$

$$O_iD_j <= MinimumValueThatGuaranteestheOptimal, \quad \forall O_iD_j variables; \tag{4.8}$$

Constraints 4.6 define auxiliary variables. Constraints 4.7 ensures that the flow passing through each sensor is not higher than the average flow value observed. Ideally, the flow will be the same as recorded. To ensure that the matrix is as dense distributed as possible, Constraint 4.8 defines a maximum for all decision variables. The lowest value that guarantees the optimal value should be used on the right side of this type of constraint. Therefore, the model will reduce the zero value variables and obtain a less sparse distribution. The OD matrix corresponds to the flow of cars that occurred during one hour. Therefore, 24 exact models are needed to simulate the VCI traffic 24 hours, one for each hour. The matrix OD using the predicted values was used to build a microscopic simulation of the traffic flow behaviour on Porto's VCI inner-ring. The use of the OD matrix in the digital twin is described in the next section.

4.3.2.2 Digital twin

In this dissertation, the SUMO simulator was selected to create a digital twin of VCI in Portugal. This simulation is also the best way to visualize and analyze the results produced by the OD matrix estimation. The simulation can both be used to graphically show the predictions made and to profile the VCI traffic. The methodology for creating a microscopic simulation in Sumo will be described in detail in this section.

A more practical description of the methodology can be found in appendix C. The methodology in this project does not include creating the city network. Therefore, if the file with the desired

network in the SUMO format does not exist, it must be developed. The networks have two representations in the SUMO simulator: the *.net.xml* file and a set of plain-XML files. Whereas the *.net.xml* file includes tons of generated information, namely, the right-of-way logic and the structures within an intersection, the plain-XML files describe the network geometry and topology. Sumo provides the *netconvert* package, which can convert these two formats freely and without information loss. In addition, while the *.net.xml* format has plenty of subtle inter-dependencies between its elements and should never be edited by hand; the plain-XML format was designed to be edited by the users. This dissertation used the Porto network made for a SUMO simulation in a previous project named C-ROADS. This network file is present in a *.net.xml* file and represents Porto's geographic zone containing the VCI network configuration. Although the original network contains all the streets of Porto, this work will only focus on the exact simulation of vehicles in the VCI and the calibration of the micro-simulation; therefore, the rest of Porto's streets will not be simulated. The methodological approach starts with the comprehension of SUMO simulations and the network.

The second stage of the methodology consists of assigning the origins and destinations defined in the created OD matrix to the simulator. Therefore, the second step in building the simulation included creating a traffic assignment zone (taz) file in XML format containing small areas where the traffic can start or end on the network. Netedit is a graphical network editor incorporated in SUMO that allows aggregating the entrances and exits of the VCI network on a taz file. With this in mind, to mirror VCI as well as possible and using the SUMO Netedit editor, these zones were manually drawn, copying the location of the actual exits and entrances, in both directions, to the SUMO network representation of the VCI. The entrances and exists assigned also match the ones used on the OD matrices. For this process, the Google Maps was used as an auxiliary in order to select the entrances and exists marked as 'OX' and 'DX', respectively in figures 4.3 and 4.2. On the figures, the 'X' represents a count number for the exit or entrance. Figure 4.3 represents the entrances and exists for the descending direction (inner ring), while figure 4.2 shows the same for the other direction (outer ring).

After having the network environment configured, the third part of the simulation generation process consisted of parsing the OD matrix to the Sumo OD matrix file type. This step was the last to start the simulation generation using Sumo commands. The simulation only considers one vehicle type, as the total volume of vehicles was used when constructing the OD matrix. Instead of using the total volume of vehicles in the simulation, the forecast of vehicles by class can be used; however, this requires the creation of five OD matrices. The *od2trips* [28] command imports the OD matrix and splits it into single vehicle trips. Afterward, routes are generated using *duarouter* [27] command. Specifically, The *duarouter* [27] command imports different demand definitions and builds vehicle routes from these definitions. In addition, it computes vehicle routes during the user assignment, which Sumo can use by employing the shortest path computation. If it is called iteratively, the duarouter executes dynamic user assignments. Finally, even though it is unnecessary for the project, this command can also repair any connectivity problems in existing route files. Finally, the simulation may be ran after this step. The commands inputs and flags used,

as well as the tests made, can be viewed in appendix C.

The last step is to run the simulation. To visualize the simulation it is only necessary to provide to the sumo graphical user interface the network and the routes generated file as input files. By importing this file on sumo, it is possible to run the simulation based on the OD matrix generated. The process described is a generic process to design simulations using OD matrices created by the exact model. Therefore, this process was executed more than once to test different days and times of the day, as the constructed simulations correspond to a simulation of a period of an hour going from 0 to 3600 seconds.

In sum, the steps for the simulation creation are the following:

1. Comprehension and adaptation of the network in SUMO;
2. Assign the origins and destinations provided by the created OD matrix;
3. Parse of the OD matrix to the OD matrix sumo file type;
4. Imports the OD matrix and splits it into single vehicle trip using the *od2trips* [28] command;
5. Generated routes using *duarouter* [27] command;
6. Provide to Sumo the network and the routes generated file as input files to run the simulation.

4.3.3 Evaluation metrics

The empirical evaluation uses seven different model evaluation metrics to compare the prediction and ensemble models, which follow:

1. The R^2 score.
2. Explained variance (r^2).
3. The mean absolute error (MAE).
4. The mean squared error (MSE).
5. The root mean squared error (RMSE).
6. Max error.
7. Median absolute error (MedAE).

When analyzing empirical evaluation results and model errors, it is usually relevant to refer to the definition of variance. In linear regression, variance measures the observed values' (y_1, y_2, y_3, ...) difference relative to the mean of the predicted values ($\hat{y_1}$, $\hat{y_2}$, $\hat{y_3}$, ...) assuming that y_i is the true value of the i-th sample and $\hat{y}_i$ is the corresponding predicted value for the total n samples. The models aim to have low variance, and the R^2 score quantifies the significance of a low value. The R^2 score [57], also known as the coefficient of determination, varies between 0 and 1 and

calculates unadjusted R^2 without correcting for bias in sample variance of y. The highest value of 1.0 is the best possible R^2 score. In opposition, the lowest value corresponds to a constant model with a low level of correlation that consistently predicts the output value without considering the input features. The R^2 score is calculated as in equation 4.9.

$$R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.9}$$

given that $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \varepsilon_i^2$.

The R^2 score was used on the prediction relative to the training and test data. As the R^2 score, the best possible score of the explained variance, also referred to by explained variance regression score function, is 1.0, and lower values correspond to worse models. If y is the correct target output, $\hat{y}$ the corresponding estimated target output, and *Var* is Variance, the square of the standard deviation, then the explained variance is estimated as in equation 4.10.

$$explained\_variance(y,\hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}} \tag{4.10}$$

While in the R^2 score and the explained variance metrics the goal is to obtain the highest value possible, in the other metrics regarding the error, the aim is to obtain the lowest value possible: 0.0. The error is the difference between the observed and the predicted values for the same phenomenon. The mean absolute error is the average of the errors, and it is calculated as in formula 4.11.

$$\text{MAE}(y,\hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i| \tag{4.11}$$

The mean square error corresponds to the average of the square of the errors; therefore, the square of each difference is made with $(\hat{y_n} - y_n))^2$ in a way that positive and negative values do not cancel each other out; therefore, the formula is in equation 4.12.

$$\text{MSE}(y,\hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2 \tag{4.12}$$

Finally, the root mean squared error is the root of the mentioned MSE value. The MAE, MSE, RMSE are proportional to the error. The max error metric corresponds to the calculation of the maximum residual error with the formula is in equation 4.13.

$$\text{Max Error}(y,\hat{y}) = max(|y_i - \hat{y}_i|) \tag{4.13}$$

The median absolute error output, also known as the median absolute error regression loss, is a non-negative floating point calculated as in equation 4.14.

$$\text{MedAE}(y,\hat{y}) = \text{median}(|\, y_1 - \hat{y}_1 \,|, \ldots, |\, y_n - \hat{y}_n \,|) \tag{4.14}$$

As for the MAE, MSE, RMSE, and Max error, the median absolute error best possible value is also 0.0. The metrics mean absolute percentage error and symmetric mean absolute percentage error are relative metrics. Both metrics express the prediction accuracy as a ratio. Specifically, the formula of the mean absolute percentage error is the one in equation 4.15.

$$\text{MAPE}(y,\hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{|y_i - \hat{y}_i|}{max(\varepsilon, |y_i|)} \tag{4.15}$$

The symmetric mean absolute percentage error is calculated as in formula 4.16.

$$\text{SMAPE} = \frac{1}{n} \sum_{t=1}^{n} \frac{|\hat{y}_t - y_t|}{(y_t + \hat{y}_t)/2} \tag{4.16}$$

As demonstrated by the formulas, null predictions and actual values would make the error infinite. The data set has 87312 entries with a null value just for the total volume of vehicles. Therefore, even though a relative metric would be interesting for the problem in question, given the impossibility of using the null variables and the existence of many entries with this value, no relative metric was used.

The simulation lasts for one hour and only uses information from sensors located on the VCI. The data, when aggregated hourly, have few or no zeros depending on the periods used for the aggregation. In addition, the simulation evaluation also used a relative metric since hectic hours are more interesting for the evaluation and, therefore, traffic was not zero at any 5 minutes simulated. In this way, a relative metric was used to evaluate the simulation error. In addition to the relative error, the absolute error was also used.

4.4 Summary

In this chapter, the flow prediction problem addressed has been formalized, making the problem explicit and specific. The chapter also describes the materials used in this thesis, the two data sets: the mobility data related to the VCI and the corresponding meteorological data. Additionally, the three different pre-processing of the data sets implemented and used in the different tests are specified. Finally, a description of the methods used and the evaluation metrics of the machine learning and micro-simulation models is made.

Table 4.6: Fields resulting from the first and third pre-processing of the data.

Field Name	Type	Description
Mobility data set		
EquipmentID	Integer	Sensor unique ID.
Agg_Period _Len_Mins	Integer	Corresponds to the time between measurements made by the sensors. It is invariably of 5 minutes.
Nr_Lanes	Integer	Number of lanes in the carriageway segment.
Lane_Bundle _Direction	Integer	The direction of the lane where the data was measured. Possible values: “1” or “0”, corresponding to decreasing and ascending, respectively.
holiday	Integer	Classification of the day as a holiday, with 0 representing false and 1 representing true.
year	Integer	Year in which the measurement occurred.
month	Integer	Month in which the measurement occurred.
day	Integer	Day in which the measurement occurred.
hour	Integer	Hour in which the measurement occurred.
minute	Integer	Minute in which the measurement occurred.
Week_Day	Integer	Day of the week, with values varying between 0 and 6
x_year, y_year	Decimal	Time of the year when the extraction occur by the sensor.
x_day, y_day	Decimal	Encode the time of the day
Meteorological data		
station	Integer	Site identifier with the LPOV and LPPR values taking the values 0 and 1, respectively.
lon	Decimal	Longitude of the station. Values: -8.6833º and -8.6459º.
lat	Decimal	Latitude of the station. Values: 40.9159º and 41.2333º.
elevation	Integer	Elevation of the station. Values: 17m and 73m.
tmpf	Decimal	Air temperature at 2 meters in Fahrenheit
dwpf	Decimal	Dew point temperature at 2 meters in Fahrenheit.
relh	Decimal	Relative Humidity in Percentage
drct	Decimal	Wind direction from geographic north. Varies from 0.0º to 360.0º
sknt	Decimal	Wind Speed in knots
p01i	Decimal	One-hour precipitation. Always 0.0.
alti	Decimal	Pressure altimeter in inches.
vsby	Decimal	Visibility in miles.
skyc1	Integer	Sky level 1 coverage. Varies from 0 to 5.
feel	Decimal	Apparent temperature in Fahrenheit (Wind Chill or Heat Index)
Targets		
Total_Vol	Integer	Total number of vehicles that go over the sensor.
Vol_class_A	Integer	Nr of vehicles of class 1 or A that go over the sensor.
Vol_class_B	Integer	Nr of vehicles of class 2 or B that go over the sensor.
Vol_class_C	Integer	Nr of vehicles of class 3 or C that go over the sensor.
Vol_class_D	Integer	Nr of vehicles of class 4 or D that go over the sensor.
Vol_class_0	Integer	Nr of unidentified vehicles that go over the sensor.

Chapter 5

Implementation

This chapter contains the description of the algorithms implemented for the several issues identified before, namely to predict the missing values and solve the traffic prediction problem, as well as the specification of the exact model implemented to obtain the OD matrix and the microscopic simulation developed. Several algorithms implemented in python were used to predict the missing values and solve the initial traffic prediction problem, including machine and deep learning, ensemble, and Auto ML algorithms. The machine and deep learning algorithms used belong to the supervised learning category. All algorithms and the mentioned metrics were implemented using scikit-learn libraries. Specifically, all the seven metrics were implemented using the sklearn.metrics library. Finally, all tests were performed on a computer with an i7-10510U quad-core processor with 16GB of RAM, 1.80GHz.

5.1 Machine and deep learning algorithms

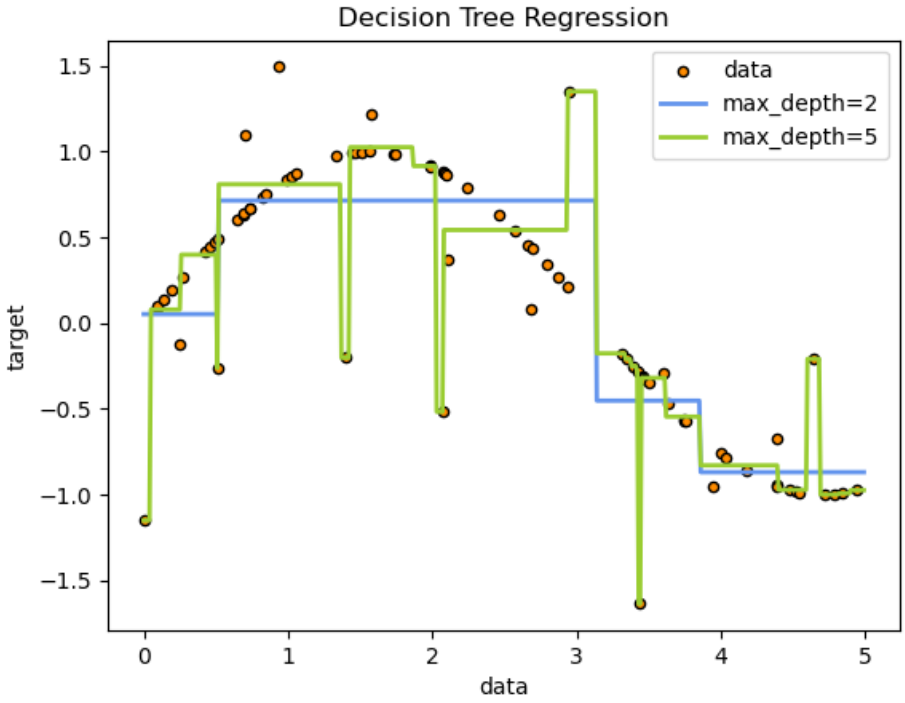
The implemented machine learning models that will be used as baselines are the Linear Regression, the Regression Tree, the Multi-Output Regressor, and the Regressor Chain. In addition, another important method was also implemented, the neural networks model.

The Linear Regression model belongs to the linear classifiers type algorithms and supports both single output and multi-output. In the multi-output (also known as multi-target) regression, a sample can have more than one target. The Linear Regression model was implemented using the sklearn.linear_model.LinearRegression library. This library implements a standard least-squares linear regression. Precisely, the Linear Regression algorithm fits a linear model with coefficients c = ($c1, \ldots, c_n$) in order to minimize the residual sum of squares between the observed targets in the data set and the targets predicted by the linear approximation.

The Regression Tree employed uses the scikit-learn library and is a simple one-dimensional regression with a decision tree capable of handling multi-output problems [54]. Regression Trees are a non-parametric method designed to create a model that predicts the value of a target variable by learning straightforward decisions with a set of if-then-else decision rules inferred from the input data features. A tree is usually defined as a piecewise constant approximation. Regression

Trees are employed to fit a sine curve with additional noisy observation. Consequently, the method learns local linear regressions by approximating the sine curve. As the tree grows, the decision rules inferred get more complex, and the decision tree model gets fitter. In practice, the max_depth parameter in the chosen library controls the maximum depth of the tree. Figure 5.1 shows that if the value of the max_depth parameter of the algorithm is set too high, the Regression trees will over-fit because they learn too fine details of the training data; therefore, the model also learns from the additional noise. In sum, the deeper the tree, the more probable it is for the model to be over-fitting. This supervised learning method can be applied to regression and classification problems [56].

Figure 5.1: Explanation of the effect of the max_depth parameter in Decision Trees. Image extracted from [56].



The Multi-Output Regressor model was implemented using the sklearn.multioutput library, and it is a prediction multi-output regression model where one regressor is fitted per output. This model is a straightforward strategy for extending regressors that only support single-output regression natively.

The Regressor Chain model was implemented using the sklearn.multioutput library and is also a prediction multi-target model, but, in opposition to the Multi-Output Regressor model, it organizes regressions into a chain. In detail, each model predicts in the order specified by the chain passed as a parameter. The method uses not only all of the available features provided to the model but also the forecasts of models that are before in the chain.

In addition to the four methods mentioned, a neural network was also implemented. Neural

networks can be used in regression and classification problems. The neural network implemented is a multi-layer perceptron (MLP) [55], a model that comprises layers of neurons linked together by synapses with weights. This type of neural network is similar to the perceptron; however, the latter has a single layer of neurons in direct feed. The Sklearn library implements the regression multi-layer perceptron algorithm in the class MLPRegressor, which supports single output and multi-output. Specifically, this class implements an MLP that trains using back-propagation with the identity function as an activation function (i.e., with no activation function) in the output layer. Thus, the output is a set of continuous values, and the loss function employed is the square error.

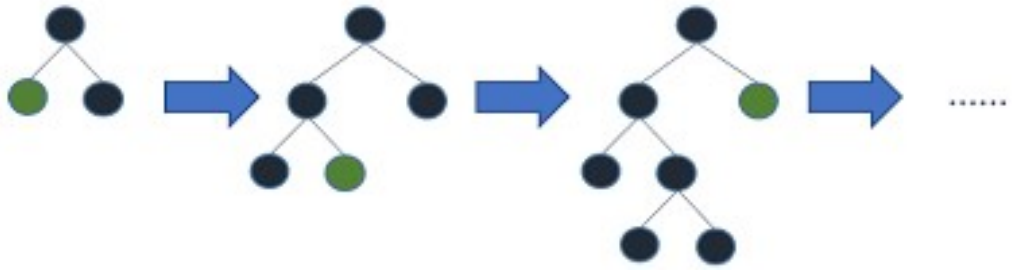
5.2 Ensemble Models

This section describes in detail the implemented ensemble learning algorithms. Specifically, three decision tree-based ensemble methods were applied: the Random Forest, XGBoost, and LightGBM, along with three methods that ensemble models from different families: simple averaging, weighted averaging, and stacking. Methods that combine models from different families were explained in detail in section 2.3 and were implemented with Neural Network, Random Forest, XGBoost, LightGBM, and Decision Trees models.

The random forest regressor model is employed in regression problems and supports multi-output. The random forest is a meta estimator that fits a certain number of regression trees using several sub-samples of the data set and uses averaging to control over-fitting and enhance the predictive accuracy. In the Sklearn library, the sub-sample size is regulated with the parameter max_samples and the number of ensemble tree by the parameter n_estimators. In addition, the parameter bootstrap is a flag that determines if the bootstrap samples are utilized when building trees, and, by default, the bootstrap flag is true. If the bootstrap flag is false, the whole data set is used to create each tree.

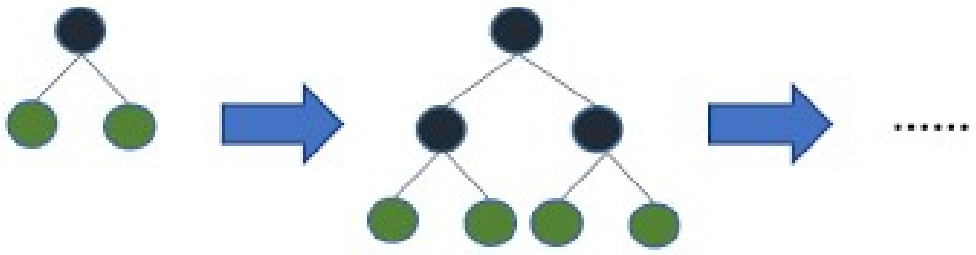
The LightGBM is an extremely powerful machine learning model. This fast processing algorithm makes use of tree-based learning algorithms. In contrast to other tree-based algorithms in which trees grow horizontally (also known as level-wise), the LightGBM [34] algorithm grows vertically (or leaf-wise). The explanation of LightGBM leaf-wise growth can be found in figure 5.2. In detail, the LightGBM models select the leaf with a large loss to grow.

Figure 5.2: Explanation of LightGBM leaf-wise growth. Image extracted from [20].



Another popular and efficient ensemble algorithm is the XGBoost (Extreme Gradient Boosting). This model is an implementation of gradient boosting trees algorithm and its power dues to some key characteristics namely the parallelization, performance, and computation speed. In opposition to the LightGBM, this ensemble algorithm grows level-wise. Both algorithms Handle missing values.

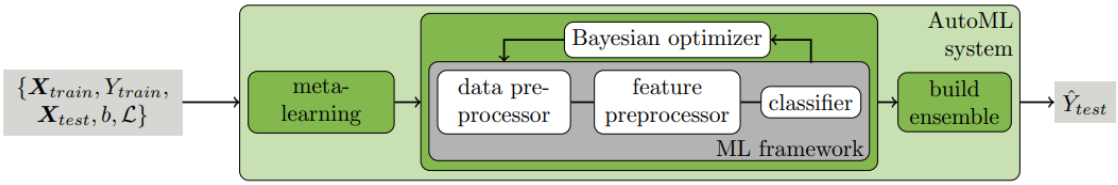
Figure 5.3: Explanation of XGBoost level-wise growth. Image extracted from [20].



5.3 Auto ML algorithms with the Auto-Sklearn library

The functioning of the Auto-Sklearn system is shown in figure 5.4. The steps involved will be described in this section.

Figure 5.4: Auto-Sklearn system. Image from [38].



Meta-Learning aims to find an appropriate instantiation of hyper-parameters for the Bayesian optimization in order to perform better than random at the beginning. Regarding the data pre-processor mention in figure 5.4, the pre-processing described in subsection 4.2.3 is entirely different from the pre-processing done by Auto-Sklearn. This library pre-processes the data in the order that follows [39].

1. Replaces categorical features using One Hot Encoding
2. Deals with missing values by replacing them with the median, mean, or most frequent value.
3. Re-scales features into the range [0, 1] or standardizes them with zero mean and unit variance. Optionally, the algorithm can normalize samples to have a unit length or even leave the features unscaled.

4. In classification, balances the data set using class weights.

Regarding the feature pre-processing algorithms, the auto-Sklearn library comprises an extensive collection of various feature pre-processing algorithms belonging to the eight categories that follow [39]:

1. Matrix decomposition - Decomposing the data set provided into maximally descriptive components using Principal component analysis, truncated SCV, kernel PCA, or Independent component analysis.

2. Features selection based on univariate statistical tests on the data .

3. Feature selection.

4. Feature clustering - Rather than feature selection, the highly correlated features can be merged, i.e., added together.

5. Kernel approximations - Using kernel functions such as the RBF kernel.

6. Polynomial feature expansion

7. Feature embeddings - Even though there are many embedding procedures, the auto-Sklearn only considers embedding through random forests.

8. Sparse representation and transformation - The sparse to a dense transformation permits the usage of algorithms on sparse data that cannot natively manage sparsely represented inputs.

The Bayesian optimization algorithm optionally selects the algorithms among the eight categories. Finally, the ensemble performed by Auto-Sklearn in the stage "build ensemble" is the same as described in section 2.3.

The first auto ML approach used the Auto-Sklearn library in version 0.14.6. The auto-sklearn library, the novel AutoML system based on scikit-learn, includes:

- 4 data pre-processing methods.

- 14 feature pre-processing methods.

- 15 classifiers.

- 110 hyperparameters [38].

Having reviewed the implementations of the prediction algorithms, the exact model employed to create an estimated OD matrix based on the predictions obtained will be described in the next section.

5.4 Microscopic simulation

In this subsection, the implementation of the simulation performed will be described. Although it is possible to do a simulation with the data obtained from the ensemble forecast model, a simulation of a day of the week at a busy time was carried out, to make a profile of the VCI at a certain time.

The exact model to obtain the OD matrix was implemented in python, using the Gurobi library. After obtaining the matrix, the digital twin begins to be implemented. The first step of the implementation is editing the file containing additions to the network by adding induction loops in order to count how many cars would pass through certain lanes on the VCI during that hour. A detail description of Induction Loops Detectors, including the steps for the creation of it can be found in the official SUMO documentation at [29]. Based on the figure 4.1, the *induction Loops* were created for each lane and direction with a frequency of 3600 to store the whole hour period on a specific XML file. Therefore, each radar counted the vehicles on each lane during one hour, saving it on a specific XML file that was parsed aggregating all radars info on an csv file, i.e., the XML file was parsed condensing all lane values observed into one, to later compare with the results collected in real sensors. The initial data set also provides aggregated results per lane, that is, the total number of vehicles that passed the segment on any lane is provided. In addition, the only radars drawn as induction loops were the ones in the VCI. Therefore, the radars situated outside the VCI rings were neither considered in the simulation results nor when generating OD matrices, nor were they drawn on the traffic assignment zones.

5.5 Summary

This section specified the scikit-learn libraries employed and how they operate in detail. In addition to specifying the libraries used in the prediction algorithms, the tools for implementing a simulation and exact models, SUMO and Gurobi, were also detailed. Finally, the steps required to implement a digital twin were presented.

Chapter 6

Empirical Evaluation

This chapter contains the description of the experiments carried out, as well as the analysis of the results. The experiments were carried out using the three pre-processed data sets mentioned in subsection 4.2.3. The first pre-processed data set uses only the mobility data with a single field to represent the day of the week. The second pre-processed data set differs from this by the use of seven fields with binary values to represent the day of the week. In the third pre-processed data set, the meteorological data fields with less than 35% missing values are added to the mobility data using Integer Encoding to encode the day of the week through a merge of the entries with the same day and time. The experiments were carried out using the types of algorithms already mentioned: Machine and deep learning algorithms, Ensemble models and Auto ML algorithms Although the Support Vector Machines and KNeighbors algorithms were implemented (belonging to the Machine learning type), due to their inefficiency in dealing with large amounts of data, they did not obtain results in a reasonable time. Therefore, the results of these two models are not present in this chapter.

The process of selecting the relevant features (known as feature selection) that occurs automatically with auto ML models will be done manually with the machine and deep learning and ensemble models. The selection of features includes three concrete experiments, all using the third pre-processed data set. The first test did not use the sky coverage level 1 values in order to understand if this input improves the predictions, given the 34% of the missing values. The last two tests were carried out in order to remove the mentioned redundant variables. In the second test, features were selected without including the year, month, day, hour, and minute and keeping the four different variables inspired by the Fourier Transform. In the third test, the opposite of the second test was done by excluding the four different variables and keeping the five other time variables mentioned.

Thus, the experiences considered are then the following (enumerated for easier reference throughout the chapter):

1. Md1: using mobility data set with a single field for the day of the week

2. Md7: using mobility data set with a field for each day of the week

3. MW: using mobility data set combine with the weather data set
4. MWwS: using mobility data set combine with the weather data set, excluding the feature sky Level 1 Coverage
5. MWw4: using mobility data set combine with the weather data set (MW), including the sky Level 1 Coverage and without the 4 redundant variables (x_{year}, y_{year}, x_{day} and y_{day})
6. MWw5: using mobility data set combine with the weather data set (MW), including the sky Level 1 Coverage and without the 5 redundant variables (year, month, day, hour and minute)
7. SMd1: using mobility data set with a single field for the day of the week from just one sensor
8. SMW: using mobility data set combine with the weather data set, including the sky Level 1 Coverage, from just one sensor

Regarding the targets, the prediction results concerned the total volume of vehicles and the volume of vehicles belonging to each class. The volume refers to the number of vehicles that pass through the sensors. Two different approaches were used in this: i) each of the six variables (total, class A, class B, class C, class D, class 0) is predicted individually; ii) the variables are forecast simultaneously, considering all y's and considering only the volume of vehicles related to each class, i.e., without considering the total volume, which can be obtained from the sum of all classes' volume.

All algorithms were evaluated with the same metrics: R^2 score, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Max error, Explained variance (r^2), and median absolute error (MedAE). The description of these metrics can be seen in section 4.3. The results also contain the test sample mean.

For the first pre-processed data set (Md1), two different data samples were used named sample A and B; however, since the metric values did not change significantly in many models, the same test set (sample A) was used for the following experiments. In both samples the data set was sorted by year, month, day, hour, and minute in this sequence. Since the data was not sorted by sensors, the samples only differ by dealing with information from different sensors. The use of two samples was made to prove that the results shown are not an exception and, therefore, can be generalized. The average of the two chosen test data sets is not significantly different.

6.1 Machine and deep learning algorithms

As mentioned before, two approaches were used in the prediction: i) each of the six variables considered is predict individually and ii) the six variables are predict simultaneously. For i) three algorithms that support single-output were tested: a Linear Regression, a Decision Tree, and a Neural Network model. For ii) other two algorithms, that are multi-output, the Multi-Output Regressor and the Chain Regressor, are tested.

6.1.1 Linear Regression

All linear regression models were created and trained in a few seconds. In the first and second sub-subsections of this subsection, the data sets used by the linear regression models are those exclusively related to mobility, Md1 and Md7, respectively. In the third and forth sub-subsection, data related to mobility and meteorology were used, MW and MWw (MWw4 and MWw5), respectively. In the final section, the data sets that obtained the best results are employed, and the data used is only related to sensors on the same path (i.e., with the same id) and in the same direction, S1.

6.1.1.1 Using data set Md1

The first machine learning algorithm implemented was the linear regression model using the Md1 data set.

Table 6.1: Results of Linear Regression model with single output and multi-output using the Md1 data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.19	0.48	0.29	0.11	0.05	0.48	0.22	0.27
Test R^2 score	0.19	0.47	0.29	0.11	0.05	0.48	0.22	0.27
MAE	0.90	60.85	3.81	0.74	2.27	64.22	13.71	22.13
MSE	1.82	6573.32	26.84	1.13	52.72	7267.88	1331.17	2320.62
RMSE	1.35	81.08	5.18	1.06	7.26	85.25	36.49	48.17
Max error	115.07	535.70	299.81	50.39	770.39	727.79		
r^2	0.19	0.47	0.29	0.11	0.05	0.48	0.22	0.27
MedAE	0.66	46.84	2.95	0.58	1.26	49.83	10.46	17.02
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

Tables 6.1 and 6.2 contain the results obtained by the linear regression models using the first pre-processed data set, and using in testing sample A and sample B, respectively. The results related to the prediction of each model's outcome individually demonstrate that the R^2 score is close to zero; therefore, the model is closer to a constant model than an optimal model. The R^2 score results practically do not differ using the training and test samples for both sample A and sample B, so the model is not over-fitting to the train data. The mean of test sample A and B is not significantly different. In both tables, the highest MAE, of approximately 64, corresponds to the total volume with the largest value average, of approximately 132. In addition, the absolute error is more significant for the total volume field as it has a more extensive range of values. The minor absolute error corresponds to the volume relative to class D, the output with the lower mean. Viewing the error in a more proportional way, that is, dividing the mean absolute error by the mean value of y, the larger error refers to the volume of class C vehicles. The max error is very high in comparison to the test mean.

Table 6.2: Results of Linear Regression model with single output and multi-output using the Md1 data set (sample B).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.19	0.48	0.30	0.11	0.05	0.48	0.22	0.27
Test R^2 score	0.19	0.48	0.30	0.11	0.05	0.48	0.22	0.27
MAE	0.91	60.78	3.79	0.74	2.49	64.16	13.74	22.14
MSE	1.88	6562.64	26.63	1.13	60.59	7260.36	1330.57	2318.87
RMSE	1.37	81.01	5.16	1.06	7.78	85.21	36.48	48.15
Max error	116.10	555.43	192.74	38.43	770.18	649.62		
r^2	0.19	0.48	0.30	0.11	0.05	0.48	0.22	0.27
MedAE	0.66	46.71	2.93	0.57	1.39	49.70	10.45	16.99
test mean	0.92	123.39	4.89	0.66	1.70	131.56	26.31	43.85

Regarding the results of the multi-output model, these results correspond to the average of the results obtained by the single output models. The maximum error metric is not displayed as this metric in the sklearn is not supported with multi-outputs. However, the average max error was 334.57 and the maximum max error was 770.18. Although the maximum error was very high, the average and median errors were only relatively high relative to the average of the test values. The forecast error of all outputs is always higher than the forecast error of all classes since the former corresponds to the latter's error summed with the prediction error of the total volume.

6.1.1.2 Using data set Md7

In this sub-subsection the results of the Linear Regression model using the Md7 data set are present.

Table 6.3: Results of Linear Regression model with single output and multi-output using the Md7 data set (sample B).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.19	0.47	0.21	0.10	0.05	0.47	0.20	0.25
Test R^2 score	0.19	0.47	0.21	0.10	0.05	0.47	0.20	0.25
MAE	0.91	60.93	3.99	0.74	2.49	64.42	13.81	22.25
MSE	1.88	6625.68	29.68	1.14	60.64	7361.01	1343.80	2346.67
RMSE	1.37	81.40	5.45	1.07	7.79	85.80	36.66	48.44
Max error	116.03	551.55	194.45	38.37	770.17	649.55		
r^2	0.19	0.47	0.21	0.10	0.05	0.47	0.20	0.25
MedAE	0.66	46.24	3.03	0.57	1.39	49.20	10.38	16.85
test mean	0.92	123.39	4.89	0.66	1.70	131.56	26.31	43.85

The results obtained by the linear regression model using the data set with seven fields for the day of the week instead of a single field are shown in Table 6.3. This results were compared with previous results for the same sample, the sample test B. The results obtained by the same linear regression model using the data set with One-Hot Encoding instead of Integer Encoding are very close; however, the results are not significantly always worse. The R^2 score remained the same or was lower than that obtained previously. The train and test R^2 score results are equal when rounded to two decimal places; therefore, the model is not over-fitting to the data. The most significant difference in the R^2 scores was found in class C. The absolute error is quite close to or higher than that obtained previously; however, the differences between the errors obtained by the model become more visible by viewing the mean squared error. The maximum error was both greater and lesser, depending on the target. Therefore, using one-hot encoding did not improve the results in comparison with using Integer Encoding.

6.1.1.3 Using data sets MWwS and MW

In this sub-subsection, both mobility and meteorological data were used. Since the coding of the days of the week in six fields did not improve the results, a coding of the days of the week was maintained using a field ranging from 0 to 6.

Table 6.4: Results of Linear Regression model with single output and multi-output using the MWwS data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.20	0.48	0.30	0.11	0.05	0.48	0.23	0.27
Test R^2 score	0.21	0.48	0.29	0.11	0.05	0.48	0.23	0.27
MAE	0.89	60.83	3.81	0.74	2.27	64.20	13.71	22.12
MSE	1.78	6562.28	26.83	1.13	52.71	7255.97	1328.95	2316.78
RMSE	1.34	81.01	5.18	1.06	7.26	85.18	36.45	48.13
Max error	115.22	532.95	300.12	50.44	770.37	733.15		
r^2	0.21	0.48	0.29	0.11	0.05	0.48	0.23	0.27
MedAE	0.65	46.85	2.95	0.58	1.27	49.83	10.46	17.02
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

The Linear Regression model results with single output and multi-output using the MWwS data set, which does not include the sky Level 1 Coverage field regarding the sample A are presented in Table 6.4. These results were compared with initial results using the Md1 data set and the same sample A. The values of the R^2 score and the explained variance with the test sample rounded to 2 decimal places are the same for all targets. Both the r^2 and the R^2 score were equal or higher for the third data set than for the other two. Specifically, these metrics were slightly higher in classes A and B and remained the same in the other classes. However, the addition of meteorological data did not significantly improve the results. Absolute errors were minor for the

prediction of the volume of classes A and B and the total volume. Once again, the only difference between the mean value of all metrics and the multi-output prediction is the Root Mean Squared Error value and the existence of a max error. The Root Mean Squared Error was 30.17 on average, while in the multi-output results was 48.13.

Table 6.5: Linear Regression model results with single output and multi-output using the MW data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.20	0.48	0.30	0.11	0.05	0.48	0.23	0.27
Test R^2 score	0.21	0.48	0.29	0.11	0.05	0.48	0.23	0.27
MAE	0.89	60.83	3.81	0.74	2.27	64.20	13.71	22.12
MSE	1.78	6562.25	26.83	1.13	52.71	7255.92	1328.94	2316.77
RMSE	1.34	81.01	5.18	1.06	7.26	85.18	36.45	48.13
Max error	115.19	533.08	300.13	50.44	770.38	733.26		
r^2	0.21	0.48	0.29	0.11	0.05	0.48	0.23	0.27
MedAE	0.65	46.85	2.95	0.58	1.27	49.82	10.46	17.02
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

The results obtained with the addition of the field related to sky Level 1 (presented in table 6.5) only differ from the originals (table 6.4) concerning the mean squared error, max error, and median absolute error metrics. In table 6.5, the mean square error is lower by three hundredths for the forecast of the volume of vehicles of class B and five hundredths lower for class 0, affecting the average squared error of all classes and outputs. The maximum error was also not significantly different for some classes at different amounts. Finally, the median error was only different for the prediction of the volume of vehicles in class 0, being the difference one hundredth less. Therefore, mean squared error and MedAE were not significantly lower with the addition of the *skyc1* field. However, the maximum error was both superior and inferior, being on average superior.

Finally, the results obtained by the multi-output models correspond to the average of the results, except for the max error and the Root Mean Squared Error value. The MSE of the multi-output models, compared with the results with the sky Level 1 Coverage, was the only metric with a lower value.

6.1.1.4 Using data sets MWw4 and MWw5

This sub-subsection uses MW data set, including the *skyc1* field. However, contrary to the previous sub-subsection, the data set was employed without the redundant variables (concerning the date and time); therefore, it uses the MWw4 and MWw5 data sets. To do so, initially the 4 variables were not used: x_{year}, y_{year}, x_{day} and y_{day}, and then the fields year, month, day, hour and minute were not used.

Table 6.6: Linear Regression model results with single output and multi-output using the MWw4 data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.18	0.35	0.18	0.06	0.05	0.36	0.17	0.20
Test R^2 score	0.18	0.35	0.18	0.06	0.05	0.36	0.17	0.20
MAE	0.92	69.71	4.05	0.77	2.25	73.76	15.54	25.24
MSE	1.84	8084.02	31.13	1.18	52.96	8997.52	1634.23	2861.44
RMSE	1.36	89.91	5.58	1.09	7.28	94.86	40.43	53.49
Max error	114.96	579.03	297.96	50.21	770.62	706.34		
r^2	0.18	0.35	0.18	0.06	0.05	0.36	0.17	0.20
MedAE	0.68	57.37	3.12	0.59	1.22	60.93	12.60	20.65
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

Table 6.6 presents the results obtained with the mobility data set combine with the weather data set and without the four variables used for the date and hour encoding. The results obtained were significantly worse than those obtained previously using meteorological data and even worse than the initial results using only data related to mobility in the VCI.

Table 6.7: Linear Regression model results with single output and multi-output using the MWw5 data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	0.20	0.47	0.29	0.10	0.04	0.48	0.22	0.27
Test R^2 score	0.20	0.47	0.29	0.10	0.04	0.47	0.22	0.27
MAE	0.89	60.81	3.81	0.74	2.20	64.19	13.69	22.11
MSE	1.79	6633.86	26.86	1.13	53.13	7333.19	1343.35	2341.66
RMSE	1.34	81.45	5.18	1.06	7.29	85.63	36.65	48.39
Max error	115.09	534.74	300.28	50.46	771.23	736.50		
r^2	0.20	0.47	0.29	0.10	0.04	0.47	0.22	0.27
MedAE	0.65	47.16	2.95	0.57	1.17	49.99	10.50	17.08
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

Table 6.7 presents the results obtained with the mobility data set combine with the weather data set and without the five variables used for the date and hour encoding: year, month, day, hour and minute. The results obtained are similar to the results obtained with the redundant variables and including the sky Level 1 Coverage. However, the r^2 and the R^2 score with the test data was almost always a hundredth lower and the MSE was always higher without the five variables.

Regarding the results obtained by linear regression in general, although the results were almost always similar, the smallest error was obtained by the third data set with the sky Level 1 Coverage

field.

6.1.1.5 Using data sets SMd1 and SMW

Since the previous data sets contained data from all sensors, inside and outside the VCI, tests were performed in which the training and test data are related only to a specific sensor and direction chosen randomly. Since the results with the mobility data exclusively and the mobility data combined with the meteorology data obtained similar results, both data sets were used in this sub-section.

Table 6.8: Results of Linear Regression model with single output using the SMd1 data set regarding the data from sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
Train R^2 score	0.2478	0.7267	0.5168	0.2550	0.0317	0.7303
Test R^2 score	0.2527	0.7274	0.5208	0.2570	0.0317	0.7311
MAE	1.1770	46.4720	4.7844	1.0164	1.8868	49.1652
MSE	2.8922	3760.4460	36.8763	1.8704	38.8978	4131.7537
Root MSE	1.7006	61.3225	6.0726	1.3676	6.2368	64.2787
Max error	69.5385	323.9382	34.3924	18.2419	127.4215	329.2121
explained variance	0.2527	0.7274	0.5208	0.2570	0.0318	0.7311
MedAE	0.8653	37.2515	3.9500	0.7953	1.0245	39.5120
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

The results of the Linear Regression model with single output using the mobility data set with a single field for the day of the week and using only data from sensor 11 in the descending direction are present in Table 6.8.

Table 6.9: Results of Linear Regression model with single output using the SMW data set, with data only regarding sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
Train R^2 score	0.2837	0.7283	0.5173	0.2562	0.0336	0.7319
Test R^2 score	0.2914	0.7291	0.5211	0.2581	0.0332	0.7328
MAE	1.1465	46.2816	4.7819	1.0156	1.8976	48.9841
MSE	2.7423	3736.8720	36.8554	1.8675	38.8385	4106.3375
Root MSE	1.6560	61.1300	6.0709	1.3666	6.2321	64.0807
Max error	69.1912	330.3724	34.2141	18.2624	127.8381	329.9734
explained variance	0.2914	0.7291	0.5211	0.2581	0.0332	0.7328
MedAE	0.8383	36.8974	3.9499	0.7949	1.0195	39.2509
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

The results obtained with the third data set are shown in Table 6.9, and present smaller errors than those obtained with the first data set, also with information from only sensor 11 in the descending direction. Both in Tables 6.8 and 6.9, the R^2 score is superior for all targets, except class 0. For classes A, C, and D, the mean absolute error was higher than the previously obtained with

the data from all sensors. The error was significantly lower for the remaining classes, especially for the two classes with higher volumes, the volume of vehicles in class B, and the total volume.

6.1.2 Decision Tree

This subsection presents the results of the regression decision tree models. These models took significantly longer to train than linear regression models. As in the previous section, the first sub-subsection uses the first pre-processed data set (Md1), while the second sub-subsection uses the second pre-processed data set (Md7). The MWw4 and MWw5 data set was used in the third and four sub-subsections. Finally, in the fifth section, the data sets that obtained the best results are employed, and the data used is only related to sensors on the same path (i.e., with the same id) and in the same direction, S1.

6.1.2.1 Using data set Md1

The results obtained by the decision trees using the mobility data set with a single field for the day of the week are presented in Tables 6.10 and 6.11

Table 6.10: Results of Decision Tree model with single output and multi-output using the Md1 data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.40	0.98	0.83	0.32	0.88	0.98	0.67	0.72
MAE	0.51	8.21	1.28	0.40	0.58	8.47	2.21	3.25
MSE	1.35	273.84	6.64	0.86	6.58	278.56	58.34	94.51
RMSE	1.16	16.55	2.58	0.93	2.56	16.69	7.64	9.72
Max error	69.0	594.0	156.0	32.0	866.0	727.0		
r^2	0.40	0.98	0.83	0.32	0.88	0.98	0.67	0.72
MedAE	0.00	2.00	0.00	0.00	0.00	3.00	0.40	0.67
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

Table 6.10 and 6.11 show that the R^2 score with the training data was 1, the highest possible value. Unfortunately, the same metric with the test data obtained much lower values. The results obtained with test data A are entirely different from those obtained with test sample B. This large variation in the results depending on the training and test data presents itself as a disadvantage of this method. The results obtained by the decision trees for sample A were much better than the results obtained by the best linear regression model for the same sample. Although the results obtained by the decision trees using sample B had lower absolute errors and the R^2 score was very irregular, both higher and lower in comparison depending on the output to be predicted. For both samples, the median absolute error was often zero. In table 6.11, the R^2 score of the prediction of the volume of vehicles belonging to class D is a negative value as this model is arbitrarily worse.

Table 6.11: Results of Decision Tree model with single output and multi-output using the Md1 data set (sample B).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.05	0.96	0.71	-0.11	0.84	0.97	0.48	0.56
MAE	0.83	13.32	2.08	0.65	0.92	13.74	3.58	5.26
MSE	2.19	448.04	10.83	1.40	10.45	455.99	94.60	153.78
RMSE	1.48	21.17	3.29	1.18	3.23	21.35	9.73	12.40
Max error	104.00	615.00	107.00	50.00	864.00	754.00		
r^2	0.05	0.96	0.71	-0.10	0.84	0.97	0.48	0.56
MedAE	0.00	8.00	1.00	0.00	0.00	9.00	1.80	3.00
test mean	0.92	123.39	4.89	0.66	1.70	131.56	26.31	43.85

However, the model is not a constant, i.e., it does not predict the output disregarding the input features given the R^2 score is different from 0.

6.1.2.2 Using data set Md7

The results obtained by the decision trees using the mobility data set with a field for each day of the week are presented in table 6.12.

Table 6.12: Results of Decision Tree model with single output and multi-output using the Md7 data set (sample B).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.02	0.93	0.66	-0.21	0.84	0.94	0.44	0.52
MAE	0.84	16.42	2.25	0.69	0.95	17.02	4.26	6.38
MSE	2.26	823.02	13.01	1.53	10.45	868.28	169.83	286.24
RMSE	1.50	28.69	3.61	1.24	3.23	29.47	13.03	16.92
Max error	118.00	598.00	189.00	51.00	752.00	868.00		
r^2	0.03	0.93	0.66	-0.21	0.84	0.94	0.44	0.52
MedAE	0.00	9.00	1.00	0.00	0.00	10.00	2.00	3.33
test mean	0.92	123.39	4.89	0.66	1.70	131.56	26.31	43.85

Changing the field related to the day of the week worsened the R^2 score with the test data as well as the explained variance for all vehicle class volume predictions except class 0 and increased the error or kept it the same. The result obtained by the R^2 score metric using the training data remained 1. The median error was higher for the prediction of the volume of vehicles of class B and the total volume and equal in the rest of the outputs in comparison with the results obtained

with Integer encoding for the same sample. The max error was both higher and lower in comparison with the previous model. Overall, the results were worse with the Md7 data set in comparison with the Md1 data set.

6.1.2.3 Using data sets MWwS and MW

The mobility data concerning the VCI and the respective meteorological data were used in this sub-subsection. Since the coding of the days of the week in six binary fields did not improve the results, the coding of the days of the week was maintained using a single field. The results obtained with the second data set without the sky Level 1 Coverage field are presented in table 6.13, whereas the results obtained with the same data set with the sky Level 1 Coverage field are presented in table 6.14.

Table 6.13: Results of Decision Tree model with single output and multi-output using the MWwS data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.14	0.97	0.76	0.05	0.85	0.97	0.55	0.62
MAE	0.75	11.19	1.79	0.57	0.79	11.55	3.00	4.42
MSE	1.93	362.37	9.21	1.20	8.39	368.65	76.47	125.04
RMSE	1.39	19.04	3.03	1.09	2.90	19.20	8.74	11.18
Max error	83.00	561.00	107.00	33.00	866.00	863.00		
r^2	0.14	0.97	0.76	0.05	0.85	0.97	0.55	0.62
MedAE	0.00	6.00	1.00	0.00	0.00	7.00	1.40	2.33
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

The R^2 score using the training data remains 1.0; therefore, an over-fit from the decision trees to the data still occurs. The median absolute error is the same for the prediction of the volume of class A, D and 0 having a value of zero. When the median error is not equal, this error is significantly higher for forecasts using meteorological data. The MAE, the MSE, and the RMSE are always inferior for the models trained only with the mobility data; on the contrary, the R^2 score using the test data and the explained variance were always superior. The maximum error varied greatly depending on the predicted output, being lower, equal, and higher comparatively. Therefore, the addition of meteorological data did not improve the results obtained.

The R^2 score with the test data ranged from 0 to 0.97, therefore getting both the worst possible value and almost the maximum value. The differences in the results obtained with the addition of the sky Level 1 Coverage field were not significantly noticeable. Results rounded to two decimal places were often the same. The errors were both higher and lower, therefore, it was not possible to conclude whether the feature is contributing positively to the results.

Table 6.14: Decision Tree model results with single output and multi-output using the MW data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.14	0.97	0.76	0.05	0.85	0.97	0.55	0.62
MAE	0.75	11.22	1.80	0.57	0.79	11.58	3.01	4.43
MSE	1.94	362.86	9.23	1.20	8.20	368.05	76.57	125.49
RMSE	1.39	19.05	3.04	1.10	2.86	19.18	8.75	11.20
Max error	105.00	531.00	107.00	32.00	724.00	863.00		
r^2	0.14	0.97	0.76	0.05	0.85	0.97	0.55	0.62
MedAE	0.00	6.00	1.00	0.00	0.00	7.00	1.40	2.33
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

6.1.2.4 Using data sets MWw4 and MWw5

Finally, the effects of redundant variables were tested and the results are present in tables 6.15 and 6.16.

Table 6.15: Decision Tree model results with single output and multi-output using the MWw4 data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.10	0.97	0.75	0.02	0.86	0.97	0.55	0.62
MAE	0.77	11.31	1.83	0.58	0.81	11.65	3.02	4.46
MSE	2.02	365.49	9.41	1.24	8.01	370.89	76.39	126.00
RMSE	1.42	19.12	3.07	1.11	2.83	19.26	8.74	11.23
Max error	105.00	568.00	107.00	36.00	685.00	676.00		
r^2	0.10	0.97	0.75	0.02	0.86	0.97	0.55	0.62
MedAE	0.00	6.00	1.00	0.00	0.00	7.00	1.40	2.33
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

Table 6.15 contains the results Decision Tree model using the MWw4 data set. The results without the 4 time variables always obtained a higher MAE, MSE and RMSE and a lower R^2 score, therefore, this novel model is worse than the one with redundant time variables. The maximum error was either higher or lower depending on the output to be predicted. The errors obtained by the model without the four redundant variables were higher than those obtained only using the mobility data.

Table 6.16 contains the results Decision Tree model using the third data set without the five variables used for the date encoding: year, month, day, hour and minute. The model using the

Table 6.16: Decision Tree model results with single output and multi-output using the MWw5 data set (sample A).

	class A	class B	class C	class D	class 0	total volume	all classes	all y
Train R^2 score	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Test R^2 score	0.12	0.97	0.74	0.00	0.84	0.97	0.51	0.58
MAE	0.77	12.16	1.90	0.60	0.83	12.53	3.26	4.81
MSE	1.98	413.90	9.79	1.27	8.77	420.12	87.52	143.69
RMSE	1.41	20.34	3.13	1.13	2.96	20.50	9.36	11.99
Max error	105.00	560.00	107.00	32.00	866.00	863.00		
r^2	0.12	0.97	0.74	0.00	0.84	0.97	0.51	0.58
MedAE	0.00	7.00	1.00	0.00	0.00	7.00	1.60	2.50
test mean	0.92	124.09	4.92	0.66	1.58	132.18	26.44	44.06

MWw5 data set obtained the worst results, being the results worse than those obtained without the 4 variables that encode the time for all outputs except the forecast of the volume of class A vehicles.

6.1.2.5 Using data sets SMd1 and SMW

Since the previous data sets contained data from all sensors, inside and outside the VCI, tests were performed in which the training and test data are related only to a specific sensor and direction. Although the addition of meteorological data did not improve the results obtained with information from all sensors, data sets one and three were tested to verify if the same happened with information from only one lane.

Table 6.17: Results of Decision Tree model with single output using the SMd1 data set, regarding sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
Train R^2 score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Test R^2 score	0.3475	0.9666	0.8065	0.2645	0.7888	0.9709
MAE	0.8092	11.2712	2.1739	0.7067	0.6988	11.4202
MSE	2.5254	460.2718	14.8932	1.8516	8.4857	447.5335
Root MSE	1.5891	21.4539	3.8592	1.3607	2.9130	21.1550
Max error	35.0000	346.0000	31.0000	13.0000	113.0000	304.0000
explained variance	0.3477	0.9666	0.8065	0.2647	0.7888	0.9709
MedAE	0.0000	4.0000	1.0000	0.0000	0.0000	4.0000
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

Tables 6.17 and 6.18 present the model results using mobility data and the same data combined with all meteorology data, respectively. As in all models that used all existing data, the R^2 score

Table 6.18: Results of Decision Tree model with single output using the SMW data set, regarding sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
Train R^2 score	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Test R^2 score	0.0688	0.9550	0.7244	-0.0497	0.6951	0.9613
MAE	1.1888	15.4332	3.1160	1.0214	1.0068	15.8071
MSE	3.6038	621.0344	21.2086	2.6425	12.2495	594.0911
Root MSE	1.8984	24.9206	4.6053	1.6256	3.4999	24.3740
Max error	35.0000	380.0000	32.0000	18.0000	109.0000	316.0000
explained variance	0.0689	0.9550	0.7244	-0.0495	0.6951	0.9613
MedAE	1.0000	9.0000	2.0000	1.0000	0.0000	10.0000
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

in the models trained with data from only one lane was also 1.00, the highest possible value, for all targets and with both data sets. Regarding both tables, the R^2 score with the test data was consistently lower than that obtained previously, with information from all sensors and the mobility data only. In addition, the errors obtained on average were higher without information from the remaining sensors for all targets.

6.1.3 Neural Networks

In this subsection, the results obtained with the neural networks will be presented. In addition to the number of iterations, the number of hidden layers and the size of these layers are crucial for the implementation of neural networks. Therefore, various sizes and number of layers were chosen as well as various numbers of maximum iterations. Tests were made concerning the volume of vehicles of class A, B, and C; as well as in relation to the total volume.

The results obtained with the neural networks with only one hidden layer can be found in Annex B. In sum, the results obtained with only one hidden layer, several sizes and different maximum iterations were close and unpromising, as they obtained worse results than the linear regression model and decision trees with the same data set. As the results were not promising, only class A and total volume targets were predicted with a hidden layer. Since the amount of data is very high, the size of the network has been increased for better results.

6.1.3.1 Two hidden layers

In this sub-subsection, only the best results obtained with two hidden layers will be presented. The remaining results can be found in Annex B.

Regarding the prediction of the volume of class A vehicles, the results obtained with two hidden layers were slightly better than those obtained with linear regression. While four models stood out in the prediction of the volume of class A vehicles, in the prediction of the volume of class B vehicles and the total volume, only one of the models distinguished itself. The network

Table 6.19: Best Neural Networks model with two hidden layers results regarding the volume of vehicles of class A, B and total volume using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

	class A	class A	class A	class A	class B	Total Volume
max iter	5000	5000	5000	5000	5000	3000
hidden layers	(5, 2)	(12, 4)	(9, 3)	(10, 3)	(10, 2)	(4, 3)
Train R^2 score	0.20	0.18	0.18	0.20	0.60	0.56
Test R^2 score	0.20	0.18	0.18	0.19	0.60	0.56
MAE	0.89	0.85	0.85	0.91	51.72	57.48
MSE	1.86	1.91	1.90	1.87	5044.77	6161.23
Root MSE	1.36	1.38	1.38	1.37	71.03	78.49
Max error	114.82	115.56	114.80	114.81	534.07	604.92
r^2	0.20	0.19	0.19	0.20	0.60	0.59
MedAE	0.61	0.51	0.52	0.67	34.45	39.87
test mean	0.92	0.92	0.92	0.92	123.39	131.56

results for the prediction of class B vehicles and total volume were significantly better than those obtained with the linear regression model.

Table 6.20: Best Neural Networks model with two hidden layers results regarding the volume of vehicles of class C using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

	class C	class C	class C	class C	class C	class C	class C
max iter	5000	5000	5000	5000	5000	5000	5000
hidden layers	(5, 2)	(6, 3)	(6, 4)	(6, 5)	(7, 3)	(7, 4)	(7, 5)
Train R^2 score	0.32	0.31	0.32	0.32	0.32	0.32	0.31
Test R^2 score	0.32	0.31	0.32	0.32	0.32	0.32	0.31
MAE	3.52	3.50	3.67	3.60	3.66	3.66	3.52
MSE	25.78	25.98	25.83	25.62	25.76	25.81	26.11
Root MSE	5.08	5.10	5.08	5.06	5.08	5.08	5.11
Max error	193.76	194.04	192.78	193.31	192.96	192.82	193.85
r^2	0.32	0.32	0.32	0.32	0.32	0.32	0.32
MedAE	2.31	2.25	2.56	2.45	2.59	2.56	2.17
test mean	4.89	4.89	4.89	4.89	4.89	4.89	4.89

Table 6.20 presents the results of the best Neural Networks models with two hidden layers regarding the volume of class C vehicles using only the data concerning the VCI. The results of the best models were very close and slightly better than those obtained by the linear regression models. Specifically, the R^2 score values were consistently higher for all neural networks than those obtained with the linear regression model. In addition, the only error that was not always smaller for all neural networks in relation to the linear regression model was the Max error.

6.1.3.2 Three hidden layers

Since the amount of data is very high, the size of the network has been increased for better results. This sub-subsection presents the results of models with three hidden layers. Mobility data has 15 input features, so tests were done with a hidden layer 1 of size ranging from 4 to 14, a hidden layer 2 of size ranging from 3 to hidden layer 1 size exclusive, and a hidden layer 3 of size ranging from 2 to hidden layer 2 size exclusive. In all tests, the mobility data set and sample B were used, as in the other sub-subsections. In this sub-subsection, two targets were predicted: the volume of class A vehicles and the total volume of vehicles. Whenever the results obtained were the same, they were aggregated.

Table 6.21: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class A using the MW data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	4000	4000	4000	4000	4000	4000
hidden layers	(4, 3, 2); (5, 4, 2); (8, 4, 3); (8, 5, 4); (8, 6, 2 / 4); (8, 7, 2 / 6); (9, 4, 3); (9, 7, 6); (9, 8, 5 / 6)	(5, 3, 2); (6, 3, 2); (7, 4, 2/3); (7, 5, 2/4); (7,6,2/3/4); (8, 4, 2); (8, 5, 2); (8, 6, 3/5); (8, 7, 4); (9, 4, 2); (9, 5, 2/ 3); (9,6,2/5); (9, 7, 2/3/4); (9, 7, 4); (9, 8, 3); (10, 3, 2); (10, 4,3); (10,5,3); (10, 6, 2/4/5)	(5, 4, 3); (6, 4, 2); (6, 4, 3); (6, 5, 2); (7, 3, 2); (7, 5, 3); (8, 3, 2); (8, 7, 5); (9, 5, 4); (9, 7, 5)	(6, 5, 3)	(6, 5, 4)	(7, 6, 5)
Train R^2 score	0.00	0.00	0.00	0.21	0.22	0.20
Test R^2 score	0.00	0.00	0.00	0.21	0.22	0.20
MAE	1.04	1.04	1.04	0.85	0.86	0.92
MSE	2.25	2.25	2.25	1.77	1.76	1.79
RMSE	1.50	1.50	1.50	1.33	1.33	1.34
Max error	113.07	113.08	113.09	113.85	113.82	114.40
r^2	0.00	0.00	0.00	0.22	0.22	0.22
MedAE	0.93	0.92	0.91	0.56	0.57	0.60
test mean	0.92	0.92	0.92	0.92	0.92	0.92

In table 6.21, the training and testing R^2 scores were invariably the same. The first three columns of the table, which correspond to multiple neural networks, obtained a R^2 score of zero, which means the model is constant. The remaining neural networks obtained better results and non-constant models. The set of models that had the lowest mean and median absolute error had the highest maximum error.

Table 6.22: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class A using the MW data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	4000	4000	4000	4000	4000	4000	4000
hidden layers	(8, 5, 3)	(8, 7, 3)	(9, 3, 2)	(9, 6, 3)	(9, 6, 4)	(9, 8, 2)	(9, 8, 4); (10, 7, 6)
Train R^2 score	0.21	0.20	0.22	0.22	0.22	0.22	0.00
Test R^2 score	0.22	0.21	0.22	0.22	0.22	0.22	0.00
MAE	0.91	0.91	0.86	0.88	0.88	0.87	1.04
MSE	1.76	1.78	1.76	1.76	1.76	1.76	2.25
Root MSE	1.33	1.34	1.33	1.33	1.33	1.33	1.50
Max error	114.18	113.78	114.24	114.22	113.88	113.85	113.06
r^2	0.22	0.21	0.22	0.22	0.22	0.22	0.00
MedAE	0.61	0.67	0.59	0.64	0.61	0.61	0.94
test mean	0.92	0.92	0.92	0.92	0.92	0.92	0.92

Table 6.23: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class A using the MW data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	4000	4000	4000	4000	4000	4000
hidden layers	(9, 8, 7); (10, 8, 2)	(10, 4, 2); (10, 8, 5/6); (11, 4, 3); (11, 5, 2); (11, 7, 3); (11, 8, 3)	(10, 5, 2)	(10, 5, 4); (10, 7, 3); (10, 7 ,5); (10, 9, 2)	(10, 6, 3)	(10, 7, 2/4); (10, 9, 3/7); (11, 3, 2); (11, 8, 5/6); (11, 8, 7); (11, 9, 4/5)
Train R^2 score	0.00	0.00	0.22	0.00	0.21	0.00
Test R^2 score	0.00	0.00	0.22	0.00	0.22	0.00
MAE	1.04	1.04	0.86	1.04	0.87	1.04
MSE	2.25	2.25	1.76	2.25	1.76	2.25
Root MSE	1.50	1.50	1.33	1.50	1.33	1.50
Max error	113.10	113.09	113.89	113.09	113.77	113.08
r^2	0.00	0.00	0.22	0.00	0.22	0.00
MedAE	0.90	0.91	0.60	0.91	0.59	0.92
test mean	0.92	0.92	0.92	0.92	0.92	0.92

None of the networks present in Tables 6.22 and 6.23 obtained a mean absolute error as low as the 0.85 obtained by the set of neural networks present in Table 6.21. In opposition to what was observed in most models, the network with size (10, 6, 3) obtained a better R^2 score with the test data than with the training data.

For the target prediction "total vehicle volume", the max iterations parameter took the value of 5000 instead of 4000.

The results obtained by the models with the parameter max iterations with value 5000 are presented in Tables 6.27, 6.28, 6.29, and 6.30. The lowest mean absolute error obtained was

Table 6.24: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class A using the MW data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	4000	4000	4000	4000	4000	4000
hidden layers	(10, 8, 3); (10, 9,5/8); (11, 4, 2); (11, 9, 2); (11, 5, 3); (11, 6, 2); (11, 7, 4); (11, 8, 2); (11, 9, 6/7); (12, 3, 2); (12, 6, 5); (12,7,2/3/6); (12, 8, 4)	(10,8,4)	(10,8,7)	(10, 9, 4); (11, 9, 8)	(10,9,6)	(11, 5, 4); (11, 6,3/4/5); (11, 7, 2/5/6); (11, 8,4); (11,10,2/4/5); (11,10,6/7/9); (12, 4, 2); (12, 5, 2/3/4); (12, 6, 4); (12, 7, 5); (12, 8, 2/3); (12, 9, 2/3)
Train R^2 score	0.00	0.21	0.21	0.00	0.21	0.00
Test R^2 score	0.00	0.21	0.21	0.00	0.22	0.00
MAE	1.04	0.86	0.84	1.04	0.89	1.04
MSE	2.25	1.77	1.77	2.25	1.76	2.25
Root MSE	1.50	1.33	1.33	1.50	1.33	1.50
Max error	113.07	113.77	113.85	113.06	113.83	113.08
r^2	0.00	0.21	0.22	0.00	0.22	0.00
MedAE	0.93	0.55	0.54	0.94	0.62	0.92
test mean	0.92	0.92	0.92	0.92	0.92	0.92

52.15, while the highest value obtained was 96.97. For the same test sample B, the error obtained by the linear regression model was 64.16, therefore, higher than the error obtained by the best model of neural networks. Given that the decision trees obtained an error of 13.74, the results of the neural networks were much lower than expected. Of the tests performed, the three-layer neural network that obtained the best results was the one with size (6, 5, 3). Since several tests were not performed, the results can still be greatly improved.

6.1.4 Multi-Output Regressor

The Multi-Output Regressor only predicts multi-outputs. Therefore, test samples A and B were used to evaluate the predictions regarding all outputs and the volume of vehicles of all classes compared with the real targets.

Tables 6.31 and 6.32 show the results of all tests obtained by the Multi-Output Regression model when the model predicts all vehicle categories simultaneously. In all tests, the value of the R^2 score was equal for the test and training samples and was also equal to the explained variance. The best results were obtained by the third data set with redundant variables and with the *skyc1* field. As one can see in table 6.32, which presents the results of the multi-output regressor

Table 6.25: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class A using the MW data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	4000	4000	4000	4000	4000
hidden layers	(11, 9, 3); (12, 11, 5); (13, 4, 3); (13, 10, 8)	(11, 10, 3)	(11, 10, 8); (12, 4, 3); (12, 6, 2/3); (12, 8, 7); (12, 9, 4/5); (12, 10, 3/4); (12, 11, 7/8); (13, 5, 4); (13, 10, 3)	(12, 7, 4)	(12, 8, 5/6); (12, 9, 6/8); (12, 10, 2); (12, 11, 2/3); (12, 11, 9); (13, 6, 5); (13, 7, 3/4); (13, 8, 7); (13, 9, 6/8)
Train R^2 score	0.00	0.21	0.00	0.21	0.00
Test R^2 score	0.00	0.21	0.00	0.21	0.00
MAE	1.04	0.92	1.04	0.92	1.04
MSE	2.25	1.77	2.25	1.78	2.25
Root MSE	1.50	1.33	1.50	1.33	1.50
Max error	113.10	113.87	113.09	113.86	113.07
r^2	0.00	0.22	0.00	0.22	0.00
MedAE	0.90	0.67	0.91	0.69	0.93
test mean	0.92	0.92	0.92	0.92	0.92

model relative to the prediction of all classes with sample B, the use of different samples did not change significantly the results. Using one field to represent the days of the week obtained better results than using seven binary fields, mainly observing the R^2 score and the explained variance. Therefore, the use of more than one field to encode the day of the week had worse results since the R^2 score got worse with One-hot encoding and the errors were higher. In opposition, the worst results were obtained with the same data set without the four variables inspired by the Fourier Transform.

Tables 6.33 and 6.34 show the results of all tests obtained by the Multi-Output Regression model when the model predicts all outputs simultaneously. The forecast errors for all outputs are higher than those regarding the predictions of all vehicle classes only, as the total volume forecast error is added. In opposition to the forecast errors, the value of the R^2 score was much higher when predicting all outputs. The best and worst results continued to be obtained by the same data sets with the same features.

In general, the R^2 score results practically do not differ using the training and test samples for samples A and B, so the Multi Output Regressor model is not over-fitting the data. The results obtained by the Multi-Output Regressor were the same as the Linear Regression.

Table 6.26: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class A using the MW data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	4000	4000	4000	4000	4000	4000
hidden layers	(12,9,7); (12,10,6/7/8/9); (12, 11, 4); (13, 3, 2); (13, 4, 2); (13, 5, 2/3); (13, 6, 2/3); (13, 7, 5); (13, 8, 2/6); (13, 9, 3/4); (13, 9, 5/7); (13, 10, 2/4); (13, 10, 5/6); (13, 10, 7/9)	(12,10,5)	(12,11,6)	(12,11,10)	(13, 6, 4); (13, 7, 2/6); (13, 8, 3/4/5)	(13,9,2)
Train R^2 score	0.00	0.21	0.22	0.00	0.00	0.20
Test R^2 score	0.00	0.21	0.22	0.00	0.00	0.20
MAE	1.04	0.90	0.86	1.04	1.04	0.91
MSE	2.25	1.77	1.76	2.25	2.25	1.79
RMSE	1.50	1.33	1.33	1.50	1.50	1.34
Max error	113.08	113.82	113.84	113.06	113.09	113.78
r^2	0.00	0.22	0.22	0.00	0.00	0.21
MedAE	0.92	0.66	0.58	0.94	0.91	0.66
test mean	0.92	0.92	0.92	0.92	0.92	0.92

6.1.5 Regressor Chain

This algorithm can predict the outputs in any given order. Therefore, one can predict the volume of cars of one class and use these predictions to predict the volume of another distinct class, and so on. All possible orders were tested for each data set to ensure that the best possible result is obtained. Although the results are not the same, the values obtained for all metrics are the same when rounded to five decimal places.

Table 6.35 presents the results of all experiments related to the prediction of all classes from A to 0 using the Chain Regressor model. The R^2 score values using the training data were equal to the R^2 score values using the test data. On the one hand, the best results were obtained with the data set that includes meteorological data and the *skyc1* feature. On the other hand, the worst result was obtained with the same data set without the four variables inspired by the Fourier Transform.

The results of the Chain Regressor models using the first and second data sets are shown in Table 6.36. For the same data, sample B obtained an R^2 score, an r^2 and a MAE higher; however,

Table 6.27: Neural Networks model with three hidden layers results regarding the total volume of vehicles using the Md1 data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	5000	5000	5000	5000	5000	5000	5000
hidden layers	(4, 3, 2)	(5, 3, 2)	(5, 4, 2)	(5, 4, 3)	(6, 3, 2)	(6, 4, 2)	(6, 4, 3)
train R^2 score	0.50	0.50	0.61	0.52	0.60	0.62	0.63
test R^2 score	0.50	0.50	0.61	0.52	0.60	0.62	0.63
MAE	60.50	60.69	53.32	60.02	54.03	53.24	52.29
MSE	7020.81	6974.98	5456.51	6754.65	5534.00	5291.61	5207.11
Root MSE	83.79	83.52	73.87	82.19	74.39	72.74	72.16
Max error	747.14	742.03	684.14	741.24	687.91	685.15	724.50
r^2	0.50	0.50	0.61	0.52	0.61	0.62	0.63
MedAE	41.87	42.56	35.31	42.18	36.13	36.31	35.23
test mean	132.18	132.18	132.18	132.18	132.18	132.18	132.18

Table 6.28: Neural Networks model with three hidden layers results regarding the total volume of vehicles using the Md1 data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	5000	5000	5000	5000	5000	5000	5000
hidden layers	(6, 5, 2)	(6, 5, 3)	(6, 5, 4)	(7, 3, 2)	(7, 4, 2)	(7, 4, 3)	(7, 5, 2)
train R^2 score	0.00	0.62	0.59	0.00	0.50	0.52	0.00
test R^2 score	0.00	0.62	0.59	0.00	0.50	0.52	0.00
MAE	96.85	52.15	55.65	96.84	60.64	59.51	96.86
MSE	13959.99	5277.68	5660.08	13959.99	6977.44	6740.78	13959.98
Root MSE	118.15	72.65	75.23	118.15	83.53	82.10	118.15
Max error	744.89	718.16	698.18	744.92	742.63	747.67	744.81
r^2	0.00	0.63	0.59	0.00	0.50	0.52	0.00
MedAE	92.11	35.07	39.38	92.08	42.43	41.37	92.19
test mean	132.18	132.18	132.18	132.18	132.18	132.18	132.18

the MSE, the RMSE and the MedAE were lower. The results obtained with a single field for the day of the week were better than those obtained with more than one binary field.

As with the Regressor Chain model, the best and worst results continued to be obtained by the same data set with the same features. For all Chain Regressor models, the value of the R^2 score and explained variance were the same. The results obtained by the Regressor Chain rounded to two decimal places were the same as the Linear Regression and Multi-Output Regressor.

Table 6.29: Neural Networks model with three hidden layers results regarding the total volume of vehicles using the Md1 data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	5000	5000	5000	5000	5000	5000	5000
hidden layers	(7, 5, 3)	(7, 5, 4)	(7, 6,2)	(7, 6, 3)	(7, 6, 4)	(7, 6, 5)	(8, 3, 2)
train R^2 score	0.61	0.57	0.49	0.00	0.00	0.50	0.52
test R^2 score	0.61	0.57	0.48	0.00	0.00	0.50	0.52
MAE	53.95	56.18	60.43	96.85	96.80	60.78	59.51
MSE	5467.94	6000.23	7192.28	13959.99	13960.08	6978.26	6739.26
Root MSE	73.95	77.46	84.81	118.15	118.15	83.54	82.09
Max error	665.70	726.40	752.36	744.88	745.14	740.62	747.59
r^2	0.61	0.57	0.50	0.00	0.00	0.50	0.52
MedAE	35.38	39.86	41.37	92.12	91.86	43.40	41.39
test mean	132.18	132.18	132.18	132.18	132.18	132.18	132.18

Table 6.30: Neural Networks model with three hidden layers results regarding the total volume of vehicles using the Md1 data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size, layer 3 size).

max iter	5000	5000	5000	5000	5000	5000	5000
hidden layers	(8, 4, 2)	(8, 4, 3)	(8, 5, 2)	(8, 5, 3)	(8, 5, 4)	(8, 6, 2)	(8, 6, 3)
train R^2 score	0.60	0.49	0.00	0.60	0.50	0.00	0.50
test R^2 score	0.60	0.49	0.00	0.60	0.50	0.00	0.50
MAE	54.76	62.59	96.86	54.34	60.44	96.97	60.83
MSE	5569.26	7097.86	13959.98	5576.32	7007.17	13960.26	6979.07
Root MSE	74.63	84.25	118.15	74.67	83.71	118.15	83.54
Max error	659.45	725.51	744.84	694.77	744.29	744.30	740.21
r^2	0.61	0.50	0.00	0.60	0.50	0.00	0.50
MedAE	36.37	46.29	92.16	35.68	41.84	92.70	43.37
test mean	132.18	132.18	132.18	132.18	132.18	132.18	132.18

Table 6.31: Multi-output Regressor model results relative to the prediction of all classes (sample A). The 4 time variables correspond to x_{year}, y_{year}, x_{day}, and y_{day}. The five variables correspond to year, month, day, hour and minute.

Data set	Md1	MW w/o skyc1	MW w/ skyc1	MWw4 w/ skyc1	MWw5 w/ skyc1
Train R^2 score	0.2227	0.2266	0.2268	0.1652	0.2234
Test R^2 score	0.2227	0.2266	0.2268	0.1652	0.2234
MAE	13.7148	13.7087	13.7087	15.5390	13.6909
MSE	1331.1676	1328.9466	1328.9403	1634.2262	1343.3535
RMSE	36.4852	36.4547	36.4546	40.4256	36.6518
r^2	0.2227	0.2266	0.2268	0.1652	0.2234
MedAE	10.4591	10.4581	10.4581	12.5954	10.5002
test mean	26.4358	26.4358	26.4358	26.4358	26.4358

Table 6.32: Multi-output regressor model results relative to the prediction of all classes (sample B).

	Md1	Md7
Train R^2 score	0.2239	0.2049
Test R^2 score	0.2236	0.2047
MAE	13.7404	13.8107
MSE	1330.5744	1343.8019
RMSE	36.4770	36.6579
r^2	0.2236	0.2047
MedAE	10.4524	10.3766
test mean	26.3115	26.3115

Table 6.33: Multi Output Regressor model results relative to the prediction of all outputs (sample A). The 4 time variables correspond to x_{year}, y_{year}, x_{day}, and y_{day}. The five variables correspond to year, month, day, hour and minute.

Data set	Md1	MW w/o skyc1	MW w/ skyc1	MWw4 w/ skyc1	MWw5 w/ skyc1
Train R^2 score	0.2656	0.2690	0.2692	0.1971	0.2655
Test R^2 score	0.2655	0.2689	0.2690	0.1969	0.2653
MAE	22.1330	22.1238	22.1237	25.2421	22.1072
MSE	2320.6198	2316.7838	2316.7710	2861.4421	2341.6599
RMSE	48.1728	48.1330	48.1328	53.4924	48.3907
r^2	0.2655	0.2689	0.2690	0.1969	0.2653
MedAE	17.0204	17.0196	17.0186	20.6505	17.0821
test mean	44.0597	44.0597	44.0597	44.0597	44.0597

Table 6.34: Multi-Output Regressor model results relative to the prediction of all outputs (sample B).

	Md1	Md7
Train R^2 score	0.2669	0.2499
Test R^2 score	0.2667	0.2497
MAE	22.1430	22.2454
MSE	2318.8719	2346.6702
RMSE	48.1547	48.4424
r^2	0.2667	0.2497
MedAE	16.9935	16.8473
test mean	43.8524	43.8524

6.2 Ensemble Models

In addition to the machine and deep learning methods, three tree-based ensemble algorithms were used: Light GBM, XGB Regressor, and Random Forest. Furthermore, in addition to these three types of models that make a homogeneous ensemble, models that make a heterogeneous ensemble were also created, the simple averaging, weighted averaging, and stacking.

Table 6.35: Regressor Chain model results relative to the prediction of all classes (sample A). The 4 time variables correspond to x_{year}, y_{year}, x_{day}, and y_{day}. The five variables correspond to year, month, day, hour and minute.

Data set	Md1	MW w/o skyc1	MW w/ skyc1	MWw4 w/ skyc1	MWw5 w/ skyc1
Train R^2 score	0.2227	0.2266	0.2268	0.1652	0.2234
Test R^2 score	0.2227	0.2266	0.2268	0.1652	0.2234
MAE	13.7148	13.7087	13.7087	15.5390	13.6909
MSE	1331.1676	1328.9466	1328.9403	1634.2262	1343.3535
RMSE	36.4852	36.4547	36.4546	40.4256	36.6518
r^2	0.2227	0.2266	0.2268	0.1652	0.2234
MedAE	10.4591	10.4581	10.4581	12.5954	10.5002
test mean	26.4358	26.4358	26.4358	26.4358	26.4358

Table 6.36: Regressor Chain model results relative to the prediction of all classes (sample B).

	Md1	Md7
Train R^2 score	0.22393	0.20488
Test R^2 score	0.22364	0.20470
MAE	13.74044	13.81069
MSE	1330.57439	1343.80191
RMSE	36.47704	36.65790
r^2	0.22364	0.20470
MedAE	10.45236	10.37657
test mean	26.31145	26.31145

Table 6.37: Regressor Chain model results relative to the prediction of all outputs (sample A).

Data set	Md1	MW w/o skyc1	MW w/ skyc1	MWw4 w/ skyc1	MWw5 w/ skyc1
Train R^2 score	0.2656	0.2690	0.2692	0.1971	0.2655
Test R^2 score	0.2655	0.2689	0.2690	0.1969	0.2653
MAE	22.1330	22.1238	22.1237	25.2421	22.1072
MSE	2320.6198	2316.7838	2316.7710	2861.4421	2341.6599
RMSE	48.1728	48.1330	48.1328	53.4924	48.3907
r^2	0.2655	0.2689	0.2690	0.1969	0.2653
MedAE	17.0204	17.0196	17.0186	20.6505	17.0821
test mean	44.0597	44.0597	44.0597	44.0597	44.0597

6.2.1 Light GBM

This subsection presents the results of the Light GBM models. As in the Linear Regression and Decision Tree subsections, in the first and second sub-subsections of this subsection, the data sets used by the models are those exclusively related to mobility (Md1 and Md7). In the third and four sub-subsection, data related to mobility and meteorology were used (MW, MWw4 and

Table 6.38: Regressor Chain model results relative to the prediction of all outputs (sample B).

	Md1	Md7
Train R^2 score	0.26692	0.24985
Test R^2 score	0.26670	0.24973
MAE	22.14302	22.24536
MSE	2318.87194	2346.67018
RMSE	48.15467	48.44244
r^2	0.26670	0.24973
MedAE	16.99351	16.84731
test mean	43.85242	43.85242

MWw5). Specifically, in the four sub-subsection, the test results without redundant time variables are presented. Finally, in the fifth section, the data sets that obtained the best results are employed, and the data used is only related to sensors on the same path (i.e., with the same id) and in the same direction, S1.

6.2.1.1 Using data set Md1

The results obtained by the Light GBM models using the mobility data set with a single field for the day of the week and a variable number of leaves are presented in Tables 6.39 and 6.40.

Table 6.39: Results of the Light GBM model with single output regarding the volume of vehicles of class A using the Md1 data set (sample A).

num leaves	10	100	1000	10000	100000	131072
Train R^2 score	0.3678	0.4678	0.5258	0.6046	0.7724	0.7950
Test R^2 score	0.3717	0.4641	0.5131	0.5528	0.6005	0.6042
MAE	0.7577	0.6969	0.6654	0.6410	0.5967	0.5902
MSE	1.4111	1.2037	1.0937	1.0044	0.8974	0.8889
RMSE	1.1879	1.0972	1.0458	1.0022	0.9473	0.9428
Max error	114.0903	89.9402	77.7109	75.1600	72.7589	74.4686
r^2	0.3717	0.4641	0.5131	0.5528	0.6005	0.6042
MedAE	0.4922	0.4433	0.4220	0.4057	0.3630	0.3544
test mean	0.9178	0.9178	0.9178	0.9178	0.9178	0.9178

Table 6.39 presents the results of the Light GBM model regarding the volume of vehicles of class A using only the mobility data set with a single field for the day of the week, varying the number of leaves passed as an argument. Even with only ten leaves, the results obtained were better than the results obtained by the linear regression model with the same data and test sample. With the increase in the number of leaves, the errors were significantly smaller. Specifically, the MSE is smaller than one with 100000 and 131072 leaves, the last being the maximum value of leaves that can be used. Although the R^2 score values are not the same for the test and training data, the values are close, suggesting that over-fitting does not occur. The explained variance obtained the same values as the R^2 score with the same test data.

Table 6.40: Results of the Light GBM model with single output regarding the total volume of vehicles using the Md1 data set (sample A).

num leaves	10	100	1000	10000	131072
Train R^2 score	0.9130	0.9615	0.9773	0.9865	0.9938
Test R^2 score	0.9130	0.9614	0.9769	0.9849	0.9880
MAE	23.2236	14.6295	11.4545	9.7196	8.5981
MSE	1214.3705	538.4995	322.8147	210.8396	167.2301
RMSE	34.8478	23.2056	17.9670	14.5203	12.9317
Max error	742.0892	749.3954	740.0954	721.3041	707.4645
r^2	0.9130	0.9614	0.9769	0.9849	0.9880
MedAE	15.2643	9.3553	7.2913	6.4159	5.6638
test mean	132.1792	132.1792	132.1792	132.1792	132.1792

Table 6.40 presents the results of the Light GBM model regarding the total volume of vehicles using the mobility data set with a single field for the day of the week, including the evolution of metrics with the increase in the number of leaves passed as a parameter to the algorithm. As the number of leaves increased, the errors decreased, and the R^2 score increased for the training and test data. Although the training and test R^2 scores are not always equal values, these values are similar, and the difference between these two values increases with the increase in the number of leaves and, consequently, the model over-fits the training data.

As the machine could not use more than 131072 leaves and increasing the number of leaves improved the results, tests were carried out with the maximum number of leaves. Tables 6.41 and 6.42 present the Light GBM model results with single output using the mobility data set with a single field for the day of the week and with the maximum number of leaves as a parameter.

Table 6.41: Results of Light GBM model with single output using the Md1 data set (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.7950	0.9933	0.9381	0.7652	0.9713	0.9938
Test R^2 score	0.6042	0.9867	0.8828	0.5556	0.9167	0.9880
MAE	0.5902	8.3833	1.3954	0.4750	0.6880	8.5981
MSE	0.8889	166.0109	4.4592	0.5624	4.6319	167.2301
RMSE	0.9428	12.8845	2.1117	0.7499	2.1522	12.9317
Max error	74.4686	356.0174	189.9283	38.2792	718.1866	707.4645
r^2	0.6042	0.9867	0.8828	0.5556	0.9167	0.9880
MedAE	0.3544	5.4179	0.8649	0.2728	0.1739	5.6638
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Table 6.42: Results of Light GBM model with single output using the Md1 data set (sample B).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.7876	0.9927	0.9341	0.7499	0.9579	0.9935
Test R^2 score	0.5226	0.9832	0.8536	0.4519	0.8993	0.9852
MAE	0.6542	9.3136	1.5455	0.5260	0.7934	9.4910
MSE	1.1053	210.5634	5.5329	0.6918	6.4402	207.0997
RMSE	1.0513	14.5108	2.3522	0.8318	2.5378	14.3910
Max error	104.3790	352.7842	164.7019	24.9488	713.8559	695.6560
r^2	0.5226	0.9832	0.8536	0.4519	0.8993	0.9852
MedAE	0.3881	5.8313	0.9313	0.2976	0.1778	6.1082
test mean	0.9241	123.3873	4.8901	0.6557	1.7000	131.5573

While in Table 6.41 the results presented regard sample A, in Table 6.42 the test sample used was the sample B. For both sample A and sample B, the difference between the R^2 score with the test and training data was always notable; however less significant in the prediction of class B vehicles and the total volume. Although the Light GBM models over-fitted to the data, the difference between the R^2 score with the training and test data is smaller than the difference between the decision trees. In addition, the results obtained by sample A with the R^2 score metric were similar to those obtained for sample B for the training set but significantly different for sample B.

Table 6.43: Results of the Light GBM model with single output using the Md1 data set and with a validation sample (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8119	0.9936	0.9426	0.7849	0.9694	0.9942
Validation R^2 score	0.5827	0.9851	0.8743	0.5241	0.9035	0.9868
validation mean	0.9201	124.0663	4.9213	0.6584	1.5891	132.1551
Test R^2 score	0.5794	0.9851	0.8739	0.5250	0.9023	0.9868
MAE test	0.6039	8.7084	1.4351	0.4873	0.7301	8.9031
MSE test	0.9446	186.2492	4.7963	0.6011	5.4304	184.4244
RMSE test	0.9719	13.6473	2.1900	0.7753	2.3303	13.5803
Max error test	76.1052	356.5799	190.5258	37.4601	714.4388	686.5855
r^2 test	0.5794	0.9851	0.8739	0.5250	0.9023	0.9868
MedAE test	0.3573	5.5015	0.8754	0.2763	0.1783	5.7547
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Since the algorithm was using the test data as validation data, other tests were done. The algorithm trained with fewer data entries in the new tests and used the remaining training data for

validation. However, the test data are the same, using sample A. The test results are shown in Table 6.43. With less training data, the training R^2 score was almost always higher. The mean of the validation sample is close to the mean of the test sample. Despite the algorithm using fewer data in the training phase and having a test data set different from the validation data set, the results were similar, although worse.

6.2.1.2 Using data set Md7

In this sub-section, the results of the Light GBM model with seven binary fields encoding the day of the week instead of one that varies from 0 to 6 (the second data set) will be presented. The results always used the maximum number of leaves allowed by the machine.

Table 6.44: Results of the Light GBM model with single output using the Md7 data set (sample B).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.7593	0.9897	0.9213	0.7119	0.9484	0.9906
Test R^2 score	0.5103	0.9777	0.8440	0.4328	0.8872	0.9798
MAE	0.6622	10.5975	1.6145	0.5378	0.8387	10.8799
MSE	1.1338	279.7697	5.8949	0.7159	7.2155	283.7247
RMSE	1.0648	16.7263	2.4279	0.8461	2.6862	16.8441
Max error	104.3474	373.5760	168.5200	30.0206	722.2499	720.9467
r^2	0.5103	0.9777	0.8440	0.4328	0.8872	0.9798
MedAE	0.3938	6.4343	1.0030	0.3085	0.1840	6.7428
test mean	0.9241	123.3873	4.8901	0.6557	1.7000	131.5573

All errors obtained with the second data set compared to the first data set were higher, except for the max error of the class A prediction. In contrast, the results obtained by the R^2 score and the explained variance were lower for the Light GBM model trained with the second data set. Therefore, once again, the results with the second data set were worse than the results obtained with the first data set.

6.2.1.3 Using data sets MWwS and MW

After exclusively testing the mobility data set, tests were carried out in which, in addition to the mobility data set, meteorological data were added. Light GBM model results using the mobility data set combine with the weather data set are presented in Tables 6.45 and 6.47. Since the algorithm was using the test data as validation data, other tests were done. The algorithm trained with fewer data entries in the new tests and used the remaining training data for validation. However, the test data are the same and the mean of the validation sample is close to the mean of the test sample. The results are presented in Tables 6.46 and 6.48

Table 6.45: Light GBM model results with single output using the MWwS data set (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8188	0.9944	0.9469	0.7969	0.9663	0.9948
Test R^2 score	0.5531	0.9857	0.8705	0.5032	0.9058	0.9871
MAE	0.6277	8.7165	1.4648	0.5025	0.7274	8.9401
MSE	1.0038	178.9308	4.9263	0.6287	5.2390	180.0414
RMSE	1.0019	13.3765	2.2195	0.7929	2.2889	13.4180
Max error	70.7178	375.8839	228.9734	44.0922	702.5228	695.8544
r^2	0.5531	0.9857	0.8705	0.5032	0.9058	0.9871
MedAE	0.3745	5.6008	0.8977	0.2878	0.1762	5.8651
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

From the analysis of Table 6.45, it can be concluded that the addition of meteorological data did not improve the model's results since, with them, the errors are generally higher, and the R^2 score is lower than only with the mobility data set regarding the VCI.

Table 6.46: Light GBM model results with single output using the MWwS data set and with a validation sample (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8422	0.9950	0.9539	0.8008	0.9606	0.9954
Validation R^2 score	0.5338	0.9841	0.8616	0.4725	0.8934	0.9858
validation mean	0.9201	124.0663	4.9213	0.6584	1.5891	132.1551
Test R^2 score	0.5298	0.9842	0.8612	0.4739	0.8914	0.9857
MAE test	0.6412	9.0588	1.5108	0.5162	0.7606	9.2973
MSE test	1.0560	197.9754	5.2824	0.6658	6.0353	199.1191
RMSE test	1.0276	14.0704	2.2983	0.8160	2.4567	14.1110
Max error test	69.9268	395.1330	220.6511	44.7335	705.4269	681.7945
r^2 test	0.5298	0.9842	0.8612	0.4739	0.8914	0.9857
MedAE test	0.3785	5.7291	0.9161	0.2927	0.1767	6.0053
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

The results obtained with the model trained with the reduced training data set were similar to those obtained previously without the validation data set. Then, it was tested if the addition of the sky Level 1 Coverage field improves the results obtained. The results are presented in Tables 6.47 and 6.48.

Table 6.47 presents the results of the Light GBM model with single output using the mobility data set with One-Hot Encoding for the day of the week and the weather data set, including the

Table 6.47: Results of Light GBM model with single output using the MW data set (sample A).

	class A	class B	class C	class D	class 0	total volume
num_leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8201	0.9945	0.9469	0.7986	0.9668	0.9949
Test R^2 score	0.5551	0.9857	0.8701	0.5027	0.9060	0.9871
MAE	0.6267	8.7224	1.4678	0.5030	0.7272	8.9404
MSE	0.9992	179.0560	4.9439	0.6294	5.2247	179.8464
RMSE	0.9996	13.3812	2.2235	0.7933	2.2858	13.4107
Max error	70.3267	383.9546	227.9040	43.6712	701.7254	696.1292
r^2	0.5551	0.9857	0.8701	0.5027	0.9060	0.9871
MedAE	0.3738	5.6015	0.8987	0.2881	0.1761	5.8665
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

sky Level 1 Coverage and regarding sample A. The results obtained with the addition of the sky Level 1 Coverage feature were similar to those obtained without this feature. The errors were either bigger or smaller, depending on the target being predicted.

Table 6.48: Light GBM model results with single output using the MW data set and with a validation sample (sample A).

	class A	class B	class C	class D	class 0	total volume
num_leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8434	0.9951	0.9541	0.7825	0.9627	0.9954
Validation R^2 score	0.5327	0.9841	0.8610	0.4716	0.8944	0.9857
MAE validation	0.6419	9.0695	1.5127	0.5177	0.7620	9.2989
MSE validation	1.0688	198.6732	5.2894	0.6667	5.8952	198.9432
RMSE validation	1.0338	14.0951	2.2999	0.8165	2.4280	14.1047
Max error validation	104.7016	494.5125	87.6359	24.3596	582.9628	579.1107
r^2 validation	0.5327	0.9841	0.8610	0.4716	0.8944	0.9857
MedAE validation	0.3778	5.7309	0.9156	0.2929	0.1779	6.0083
validation mean	0.9201	124.0663	4.9213	0.6584	1.5891	132.1551
Test R^2 score	0.5292	0.9841	0.8607	0.4727	0.8924	0.9858
MAE test	0.6416	9.0661	1.5130	0.5176	0.7610	9.2945
MSE test	1.0574	198.2418	5.3006	0.6673	5.9821	198.8286
RMSE test	1.0283	14.0798	2.3023	0.8169	2.4458	14.1007
Max error test	68.9851	395.2116	214.6260	45.2525	701.5250	683.5436
r^2 test	0.5292	0.9841	0.8607	0.4727	0.8924	0.9858
MedAE test	0.3782	5.7324	0.9175	0.2928	0.1778	6.0080
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Since the model was using test data for parameter validation, the trained model was studied with less training data, a validation and test data set. The results of this model are presented in Table 6.48. Although worse, the results obtained with less training data were worse, however, not significantly worse. In the next sub-subsection, the same data set was used, obtained with the third pre-processing, but with an different engineering feature.

6.2.1.4 Using data sets MWw4 and MWw5

Tables 6.49 and 6.51 present the results of the Light GBM models using the meteorological data set without redundant variables. The number of leaves passed as a parameter was always the maximum value supported by the machine, 131072.

Table 6.49: Light GBM model results with single output using the MWw4 data set (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8065	0.9940	0.9439	0.7854	0.9552	0.9945
Test R^2 score	0.5440	0.9851	0.8671	0.4883	0.8974	0.9867
MAE	0.6341	8.8597	1.4842	0.5104	0.7430	9.0699
MSE	1.0242	185.8368	5.0554	0.6475	5.7027	186.3153
RMSE	1.0121	13.6322	2.2484	0.8047	2.3880	13.6497
Max error	70.3529	437.3387	233.5568	45.1423	676.3933	691.3775
r^2	0.5440	0.9851	0.8671	0.4883	0.8974	0.9867
MedAE	0.3780	5.6763	0.9080	0.2926	0.1776	5.9323
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

The results obtained by the Light GBM model without the four different variables inspired by the Fourier Transform to represent the year and the day, present in Table 1, were worst than the previous models. Specifically, the results were much worse than those obtained with the model trained with the mobility data set and slightly worse than those obtained with the addition of meteorological data. Then, a new Light GBM model was trained with 60% of the same data and tested with 20% of the data, with the rest used as validation data.

As in the other Light GBM models, the results were worse but similar to those obtained with the model trained with 80% of the same data set and without the validation data set.

Removing the five redundant integer variables negatively affected the model since all model errors, which did not train with redundant variables, were greater than the maximum error. Since the differences in the results of the models using the validation data set and without it were not significant, the model with less training data was not tested for the model trained without the five variables.

In sum, the best results using the Light GBM model were obtained using the mobility data when using the data sets with information from all sensors. Therefore, the best results were then obtained by the first data set, which does not contain the meteorological information. In addition,

Table 6.50: Light GBM model results with single output using the MWw4 data set and with a validation sample (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
train R^2 score	0.8174	0.9946	0.9488	0.7625	0.9478	0.9950
validation R^2 score	0.5222	0.9835	0.8584	0.4607	0.8849	0.9852
MAE validation	0.6494	9.1972	1.5275	0.5235	0.7751	9.4391
MSE validation	1.0926	205.6572	5.3901	0.6805	6.4244	206.4445
RMSE validation	1.0453	14.3408	2.3217	0.8249	2.5346	14.3682
Max error validation	104.3499	478.6222	89.2491	22.8610	582.2400	577.3008
r^2 validation	0.5222	0.9835	0.8584	0.4607	0.8849	0.9852
MedAE validation	0.3833	5.8010	0.9244	0.2967	0.1788	6.0788
validation mean	0.9201	124.0663	4.9213	0.6584	1.5891	132.1551
Test R^2 score	0.5185	0.9836	0.8580	0.4615	0.8830	0.9852
MAE test	0.6491	9.1949	1.5277	0.5235	0.7739	9.4393
MSE test	1.0814	205.1366	5.4036	0.6815	6.5019	206.4465
RMSE test	1.0399	14.3226	2.3246	0.8255	2.5499	14.3682
Max error test	70.1727	411.7355	226.6375	45.9073	684.4989	681.1317
r^2 test	0.5185	0.9836	0.8580	0.4615	0.8830	0.9852
MedAE test	0.3836	5.8018	0.9244	0.2966	0.1785	6.0815
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Table 6.51: Light GBM model results with single output using the MWw5 data set (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.7940	0.9934	0.9367	0.7326	0.9704	0.9939
Test R^2 score	0.5360	0.9844	0.8612	0.4733	0.9025	0.9860
MAE	0.6401	9.1047	1.5167	0.5196	0.7634	9.3189
MSE	1.0422	194.8500	5.2812	0.6665	5.4192	196.0992
RMSE	1.0209	13.9589	2.2981	0.8164	2.3279	14.0035
Max error	71.4523	412.2833	223.4705	44.9937	712.3723	706.5288
r^2	0.5360	0.9844	0.8612	0.4733	0.9025	0.9860
MedAE	0.3819	5.8353	0.9227	0.2961	0.1933	6.0924
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

the results obtained by the Light GBM model demonstrate that the R^2 score of training and testing differ significantly in the prediction of class A and D vehicles.

6.2.1.5 Using data sets SMd1 and SMW

Since the models in the previous subsections used data sets containing data from all sensors, inside and outside the VCI, tests were performed in which the training and test data are related only to a specific sensor and direction. Both the mobility data set and the same data set combined with the meteorology data set were used in this sub-section.

Table 6.52: Results of Light GBM model with single output using the SMd1 data set and using data from sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.8631	0.9932	0.9590	0.8480	0.9502	0.9942
Test R^2 score	0.5617	0.9792	0.8691	0.5077	0.8426	0.9818
MAE	0.8641	11.3373	2.2313	0.7683	0.9200	11.5241
MSE	1.6964	286.5567	10.0747	1.2394	6.3229	279.5906
Root MSE	1.3025	16.9280	3.1741	1.1133	2.5145	16.7210
Max error	36.3241	225.7862	24.0630	11.8705	86.5515	183.0996
explained variance	0.5617	0.9792	0.8691	0.5077	0.8426	0.9818
MedAE	0.5615	7.4960	1.5191	0.5137	0.4125	7.8419
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

Table 6.53: Results of the Light GBM model with single output using the SMW data set, regarding sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
num leaves	131072	131072	131072	131072	131072	131072
Train R^2 score	0.7225	0.9959	0.9261	0.6808	0.9609	0.9965
Test R^2 score	0.4882	0.9763	0.8489	0.4475	0.8162	0.9793
MAE	0.9642	12.4418	2.4715	0.8526	0.9945	12.6530
MSE	1.9808	327.4512	11.6251	1.3907	7.3829	318.1103
Root MSE	1.4074	18.0956	3.4096	1.1793	2.7172	17.8356
Max error	38.0619	287.3017	27.2190	16.8893	102.5714	221.5905
explained variance	0.4882	0.9763	0.8489	0.4476	0.8162	0.9793
MedAE	0.6563	8.5346	1.7525	0.5898	0.4837	8.9378
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

The results obtained using information from only one sensor are shown in Tables 6.52 and 6.53. Compared to the results obtained with the entire data set, the results obtained with the data from one sensor show a higher error for all targets and a lower R^2 score with both data sets. When comparing the results obtained by the Light GBM models, it is observed that the best results were obtained using the Md1 data set (as before).

6.2.2 XGB Regressor

This subsection presents the results of the XGB Regressor models. As in the Linear Regression, Decision Tree, and Light GBM subsections, in the first and second sub-subsections of this subsection, the Md1 and Md7 data sets were used by the models. In addition, in the third and four sub-subsection, data related to mobility and meteorology were used (MW, MWw4, and MWw5). In the final sub-subsection, the data sets that obtained the best results are employed, and the data used is only related to sensors on the same path (i.e., with the same id) and in the same direction, S1.

6.2.2.1 Using data set Md1

The results obtained by the XGB Regressor with a variable number of estimators are presented in Annex B. From the present results of the Annex it is concluded that as the number of estimators increased, the difference between the R^2 score using the training and test data increased. Although XGB Regressor supports more than 1000 estimators, the computational time becomes very high as the number of estimators increases beyond that number. Therefore, the tests performed always used the same number of estimators (1000) to allow a fair comparison of them. While the initial tests present in Annex B were done only with the mobility data and sample B, the following tests used all data sets.

Table 6.54: Results of the XGB Regressor model with single output using the Md1 data set (sample A). In this table, the number of estimators of the model was always 1000.

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.4884	0.9668	0.8444	0.4376	0.7249	0.9714
Test R^2 score	0.4849	0.9665	0.8432	0.4347	0.7155	0.9711
MAE	0.6802	12.1721	1.6121	0.5459	1.1716	12.3259
MSE	1.1569	419.1916	5.9663	0.7153	15.8174	403.1959
RMSE	1.0756	20.4742	2.4426	0.8458	3.9771	20.0797
Max error	100.8043	527.6476	301.4577	50.6621	749.8588	735.2572
r^2	0.4849	0.9665	0.8432	0.4347	0.7155	0.9711
MedAE	0.4279	7.5043	0.9980	0.3298	0.3733	7.7940
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

The results of the XGB Regressor algorithms with test samples A and B were close. However, the MAE, the MSE, the RMSE, and the MedAE relative to the forecasts of the volume of vehicles of classes A, B, C, D and the total volume was bigger for the sample A than for the sample B. The MAE, the MSE, the RMSE, and the MedAE relative to forecasts of the volume of class 0 vehicles were lower for sample A than for sample B. Given that the forecast of the number of cars in class 0 is of little significance, the model generally obtained better results for sample A than for sample B using the mobility data set.

Table 6.55: Results of the XGB Regressor model with single output regarding the volume of vehicles of class A using the Md1 data set (sample B). In this table, the number of estimators of the model was always 1000.

	class A	class B	class C	class D	class 0	total volume
n estimators	10000	10000	1000	1000	1000	1000
Train R^2 score	0.5337	0.9750	0.8443	0.4384	0.7519	0.9717
Test R^2 score	0.5055	0.9738	0.8428	0.4340	0.7421	0.9714
MAE	0.6706	10.9285	1.6083	0.5444	1.2057	12.2903
MSE	1.1449	328.4855	5.9388	0.7144	16.4936	400.3747
RMSE	1.0700	18.1242	2.4370	0.8452	4.0612	20.0094
Max error	104.7760	482.6334	195.2000	33.4815	749.5424	736.4468
r^2	0.5055	0.9738	0.8428	0.4340	0.7421	0.9714
MedAE	0.4152	6.7766	0.9947	0.3287	0.3786	7.7622
test mean	0.9241	123.3873	4.8901	0.6557	1.7000	131.5573

6.2.2.2 Using data set Md7

In this sub-section, the results of the XGB Regressor model using the same data set presented in sub-section 6.2.2.1 but with seven binary fields encoding the day of the week instead of an Integer field that varies from 0 to 6 will be presented.

Table 6.56: Results of the XGB Regressor model with single output using the Md7 data set (sample B).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.4649	0.9350	0.7852	0.4000	0.7308	0.9406
Test R^2 score	0.4569	0.9346	0.7837	0.3960	0.7209	0.9402
MAE	0.6991	17.7662	1.9897	0.5662	1.2443	18.2756
MSE	1.2575	819.3914	8.1745	0.7624	17.8503	838.2238
RMSE	1.1214	28.6250	2.8591	0.8731	4.2250	28.9521
Max error	104.4100	495.6272	194.5280	34.5929	750.5505	731.4254
r^2	0.4569	0.9346	0.7837	0.3960	0.7209	0.9402
MedAE	0.4370	10.9469	1.3730	0.3436	0.3940	11.5300
test mean	0.9241	123.3873	4.8901	0.6557	1.7000	131.5573

As with the other algorithms presented above, using more than one field to represent the day of the week had worse results than using a feature with Integer encoding.

6.2.2.3 Using data sets MWwS and MW

In this section, the results of the XGB Regressor models trained and tested with the third data set will be presented, including and excluding the sky Level 1 Coverage field from the meteorological data set.

Table 6.57: XGB Regressor model results with single output using the MWwS data set (sample A).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.4977	0.9669	0.8447	0.4373	0.7393	0.9715
Test R^2 score	0.4907	0.9665	0.8430	0.4323	0.7263	0.9711
MAE	0.6750	12.4180	1.6183	0.5487	1.1907	12.5695
MSE	1.1439	419.1970	5.9736	0.7184	15.2137	403.4192
RMSE	1.0696	20.4743	2.4441	0.8476	3.9005	20.0853
Max error	88.1763	518.1797	301.6737	50.3688	748.4849	741.1038
r^2	0.4907	0.9665	0.8430	0.4323	0.7263	0.9711
MedAE	0.4217	7.7710	1.0065	0.3338	0.3888	8.0662
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Although the R^2 score was not significantly higher for classes A, B, C, and 0 and total volume using the train data set, the same metric was lower for classes C and D in comparison with the first data set. The MAE error was lower exclusively for class A and higher for the remaining outputs, again in comparison with the results using the first pre-processed data set. The maximum error was lower for the forecast of all outputs except for the total volume and the class C vehicles forecast. With only these results for sample A, it is not possible to say whether the use of meteorological data improved the forecasts or not.

Table 6.58: XGB Regressor model results with single output using the MW data set (sample A).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.4986	0.9673	0.8447	0.4370	0.7388	0.9716
Test R^2 score	0.4912	0.9668	0.8430	0.4320	0.7255	0.9712
MAE	0.6750	12.3542	1.6179	0.5490	1.1936	12.5531
MSE	1.1427	415.0029	5.9739	0.7188	15.2587	402.2056
RMSE	1.0690	20.3716	2.4442	0.8478	3.9062	20.0551
Max error	80.4873	515.8823	292.7329	50.4601	748.2324	740.3596
r^2	0.4912	0.9668	0.8430	0.4320	0.7255	0.9712
MedAE	0.4218	7.7261	1.0065	0.3352	0.3862	8.0616
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

The MAE and MedAE obtained with the MW data set with the sky Level 1 Coverage field were lower, equal, or higher than those obtained with the MWwS data set without this field, depending on the predicted outputs.

6.2.2.4 Using data sets MWw4 and MWw5

Since, in both tests, the R^2 score metric values with the third data set without redundant variables are the same as those obtained by the explained variance, the row with this metric does not appear in the tables.

Table 6.59: XGB Regressor model results with single output using the MWw4 data set (sample A).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.4894	0.9650	0.8424	0.4267	0.7289	0.9699
Test R^2 score	0.4857	0.9646	0.8411	0.4230	0.7179	0.9695
MAE	0.6779	12.8092	1.6278	0.5535	1.1996	12.9251
MSE	1.1552	442.1156	6.0471	0.7301	15.6801	425.1963
Root MSE	1.0748	21.0265	2.4591	0.8545	3.9598	20.6203
Max error	79.3432	546.0850	298.5392	50.5468	750.5079	743.2329
MedAE	0.4235	8.0205	1.0125	0.3384	0.3870	8.2778
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Table 6.60: XGB Regressor model results with single output using the MWw5 data set (sample A).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.4915	0.9632	0.8390	0.4321	0.5539	0.9684
Test R^2 score	0.4848	0.9628	0.8373	0.4270	0.5394	0.9680
MAE	0.6791	13.1182	1.6454	0.5516	1.5822	13.1870
MSE	1.1572	465.4086	6.1890	0.7251	25.6062	446.3144
RMSE	1.0757	21.5733	2.4878	0.8515	5.0603	21.1262
Max error	85.5686	536.6219	301.5229	50.4798	762.7968	750.8078
MedAE	0.4250	8.1811	1.0247	0.3379	0.4555	8.4335
test mean	0.9178	124.0935	4.9245	0.6596	1.5838	132.1792

Removing the year, month, day, hour, and minute columns from the data set increased the errors obtained by all metrics. Although less significantly, the R^2 score was always lower with the data set with fewer features. The results obtained without the four variables used for the date encoding were better than those obtained without the five variables: year, month, day, hour and minute.

6.2.2.5 Using data sets SMd1 and SMW

Since the models in the previous subsections used data sets containing data from all sensors, inside and outside the VCI, tests were performed in which the training and test data are related only to a specific sensor and direction; sensor 11 in the descending direction.

Table 6.61: Results of the XGB Regressor model with single output using the SMd1 data set and using only data regarding sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.6312	0.9769	0.8834	0.5781	0.8132	0.9806
Test R^2 score	0.5101	0.9700	0.8522	0.4597	0.6972	0.9751
MAE	0.9476	13.4906	2.4516	0.8366	1.3651	13.5189
MSE	1.8960	414.3300	11.3769	1.3601	12.1653	382.1077
Root MSE	1.3770	20.3551	3.3730	1.1662	3.4879	19.5476
Max error	38.3206	324.3770	24.9210	16.5267	92.0637	261.6169
explained variance	0.5101	0.9700	0.8522	0.4597	0.6972	0.9751
MedAE	0.6520	9.0716	1.7490	0.5807	0.6067	9.3759
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

The results obtained by the XGB Regressor using the mobility data, present in Table 6.61, obtained higher R^2 scores and errors. In the same table, the value of the R^2 score with the test data was equal to the explained variance, both constantly lower than the R^2 score with the training data.

Table 6.62: Results of the XGB Regressor model with single output using the SMW data set, regarding sensor 11 in the descending direction (sample A).

	class A	class B	class C	class D	class 0	total volume
n estimators	1000	1000	1000	1000	1000	1000
Train R^2 score	0.6655	0.9815	0.8927	0.6097	0.9085	0.9836
Test R^2 score	0.4858	0.9725	0.8461	0.4294	0.7713	0.9762
MAE	0.9664	13.2994	2.5013	0.8601	1.1811	13.4785
MSE	1.9902	379.1888	11.8434	1.4364	9.1879	365.9534
Root MSE	1.4107	19.4728	3.4414	1.1985	3.0312	19.1299
Max error	35.4860	362.9297	24.3190	17.7936	105.6115	243.4131
explained variance	0.4858	0.9725	0.8461	0.4294	0.7713	0.9762
MedAE	0.6650	9.1484	1.7780	0.5970	0.5710	9.5628
test mean	1.5999	190.0465	10.0871	1.3285	1.3767	204.4388

In addition to the tests carried out with the mobility data, the same tests were carried out with the mobility data combined with the meteorological data, which are present in Table 6.62. The R^2 with training data was invariably higher with the addition of meteorological data. The R^2 with the test data and the error was both lower and higher, depending on the target, with the addition of meteorological data.

6.2.3 Random Forest

This subsection presents the results of the random forest models. In the first and second sub-subsections of this subsection, the data sets used by the models are those exclusively related to

mobility, Md1 and Md7, respectively. The third sub-subsection used data related to mobility and meteorology, the MW data set. Since all other models obtained worse results without the redundant variables, tests were not performed without the redundant variables for the Random Forest model.

6.2.3.1 Using data set Md1

The results obtained by the Random Forest models regarding all targets and using the Md1 data set will be discussed in this sub-subsection.

Table 6.63: The Random Forest model results regarding the volume of vehicles of class A using the Md1 data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9301	0.9416	0.9455	0.9474	0.9486	0.9493
Test R^2 score	0.6106	0.6313	0.6380	0.6412	0.6431	0.6444
MAE	0.5316	0.5215	0.5180	0.5163	0.5152	0.5145
MSE	0.8745	0.8281	0.8132	0.8059	0.8016	0.7987
RMSE	0.9352	0.9100	0.9018	0.8977	0.8953	0.8937
Max error	68.9000	69.6000	69.4667	69.7250	69.3400	69.3167
r^2	0.6112	0.6319	0.6385	0.6417	0.6436	0.6449
MedAE	0.3000	0.2500	0.2667	0.2750	0.2600	0.2667
test mean	0.9178	0.9178	0.9178	0.9178	0.9178	0.9178

Table 6.64: The Random Forest model results regarding the volume of vehicles of class A using the Md1 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9018	0.9166	0.9215	0.9240	0.9254	0.9264
Test R^2 score	0.4455	0.4693	0.4772	0.4813	0.4838	0.4854
MAE	0.6973	0.6864	0.6824	0.6805	0.6794	0.6786
MSE	1.2839	1.2287	1.2104	1.2009	1.1953	1.1914
RMSE	1.1331	1.1085	1.1002	1.0959	1.0933	1.0915
Max error	104.2000	104.2500	104.1333	104.0500	104.0600	104.0667
r^2	0.4464	0.4702	0.4781	0.4822	0.4846	0.4863
MedAE	0.4000	0.4000	0.4333	0.4250	0.4200	0.4167
test mean	0.9241	0.9241	0.9241	0.9241	0.9241	0.9241

The results obtained by the random forest models regarding the prediction of vehicles of class A are visible in tables 6.63 and 6.64. Up to 60 estimators were used since the machine did not have the computational power to use 70 estimators. The results obtained by the same Random forest models using test data A were better than those obtained with test sample B. Although table 6.63 shows that the difference between the R^2 score using the training data and the test data is significant, the results with sample B greatly accentuate this difference. In table 6.64, the R^2 score with the test data is almost half that obtained with the training data. Therefore, there is an over-fit of the models to the training data. The R^2 score and the explained variance with the test data

obtained by Random forest, even with only ten estimators, were better than the values obtained by the Linear Regression models and the decision tree using the same data.

Table 6.65: The Random Forest model results regarding the volume of vehicles of class B using the Md1 data set (sample A).

estimators nº	10	20	30	40
Train R^2 score	0.9973	0.9978	0.9980	0.9980
Test R^2 score	0.9854	0.9863	0.9866	0.9868
MAE	7.9623	7.7316	7.6519	7.6106
MSE	182.1450	170.6857	166.9359	165.0363
RMSE	13.4961	13.0647	12.9204	12.8466
Max error	403.6000	441.9500	454.6667	453.9250
r^2	0.9854	0.9863	0.9866	0.9868
MedAE	4.4000	4.2500	4.2333	4.2000
test mean	124.0935	124.0935	124.0935	124.0935

Table 6.66: The Random Forest model results regarding the volume of vehicles of class B using the Md1 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9961	0.9968	0.9970	0.9971	0.9971	0.9972
Test R^2 score	0.9785	0.9797	0.9801	0.9803	0.9804	0.9805
MAE	10.4192	10.1537	10.0623	10.0179	9.9912	9.9728
MSE	269.3862	254.8984	249.8172	247.4847	246.0511	244.9987
RMSE	16.4130	15.9655	15.8056	15.7316	15.6860	15.6524
Max error	481.2000	491.5500	489.7000	493.3500	496.8000	495.8000
r^2	0.9785	0.9797	0.9801	0.9803	0.9804	0.9805
MedAE	6.5000	6.4000	6.3333	6.3000	6.2800	6.2833
test mean	123.3873	123.3873	123.3873	123.3873	123.3873	123.3873

The same samples were used to predict the volume of class B vehicles, and the results are shown in Tables 6.65 and 6.66. While the model using training sample A managed to use only 40 estimators, with training sample B the maximum multiple of 10 used was 60. In the prediction of class B vehicles, the values obtained with the training and test data were close to those obtained by the R^2 score. For sample A, the errors obtained by all Random Forest models were lower than those obtained with the Linear Regression, Decision Tree, and Light GBM models. In addition, the R^2 score was also higher than the mention models in comparison with the Random Forest model. For sample B, the minimum error of the Random Forest model was 9.9728, higher than the Light GBM error of 9.3136 for the same sample. The mean absolute error of the Random Forest model was, however, smaller than the error of the decision trees and the linear regression.

In the prediction regarding the volume of vehicles of classes C, D, and 0, it was possible to use 60 estimators, and the results for samples A and B are in tables 6.67, 6.68, 6.69, 6.70, 6.71, and 6.72, respectively. The errors obtained with sample B related to the volume of class D vehicles

Table 6.67: The Random Forest model results regarding the volume of vehicles of class C using the Md1 data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9796	0.9830	0.9842	0.9847	0.9851	0.9853
Test R^2 score	0.8877	0.8938	0.8959	0.8969	0.8975	0.8979
MAE	1.2520	1.2206	1.2093	1.2039	1.2005	1.1982
MSE	4.2715	4.0389	3.9604	3.9233	3.9002	3.8853
RMSE	2.0668	2.0097	1.9901	1.9807	1.9749	1.9711
Max error	69.7000	68.4500	69.4333	69.0000	69.4200	69.4167
r^2	0.8878	0.8939	0.8959	0.8969	0.8975	0.8979
MedAE	0.7000	0.6500	0.6667	0.6500	0.6400	0.6500
test mean	4.9245	4.9245	4.9245	4.9245	4.9245	4.9245

Table 6.68: The Random Forest model results regarding the volume of vehicles of class C using the Md1 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9705	0.9751	0.9766	0.9773	0.9778	0.9781
Test R^2 score	0.8345	0.8419	0.8444	0.8457	0.8465	0.8469
MAE	1.6390	1.6044	1.5925	1.5865	1.5830	1.5808
MSE	6.2551	5.9731	5.8799	5.8295	5.8016	5.7844
RMSE	2.5010	2.4440	2.4249	2.4144	2.4086	2.4051
Max error	131.2000	129.8000	139.2667	129.8000	117.9200	120.0167
r^2	0.8345	0.8420	0.8444	0.8458	0.8465	0.8470
MedAE	1.0000	1.0000	0.9667	0.9750	0.9600	0.9667
test mean	4.8901	4.8901	4.8901	4.8901	4.8901	4.8901

Table 6.69: The Random Forest model results regarding the volume of vehicles of class D using the Md1 data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9215	0.9345	0.9389	0.9411	0.9424	0.9432
Test R^2 score	0.5673	0.5903	0.5982	0.6019	0.6042	0.6057
MAE	0.4265	0.4192	0.4166	0.4153	0.4145	0.4140
MSE	0.5476	0.5185	0.5085	0.5038	0.5009	0.4990
RMSE	0.7400	0.7201	0.7131	0.7098	0.7077	0.7064
Max error	30.2000	29.2000	28.9000	29.0750	28.9800	28.5333
r^2	0.5680	0.5910	0.5990	0.6027	0.6049	0.6064
MedAE	0.2000	0.2000	0.2000	0.2000	0.2200	0.2167
test mean	0.6596	0.6596	0.6596	0.6596	0.6596	0.6596

were higher than those obtained with sample A. Furthermore, the R^2 score was also lower for sample B.

Finally, in the prediction of the total volume, it was found that the maximum number of estimators used was not the same for both samples. The results of the Random Forest model regarding the total volume of vehicles using only the mobility data set are present in Tables 6.73 and 6.74.

Table 6.70: The Random Forest model results regarding the volume of vehicles of class D using the Md1 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.8885	0.9054	0.9111	0.9139	0.9156	0.9167
Test R^2 score	0.3710	0.3992	0.4085	0.4130	0.4157	0.4173
MAE	0.5588	0.5509	0.5481	0.5468	0.5459	0.5455
MSE	0.7939	0.7583	0.7466	0.7410	0.7374	0.7354
RMSE	0.8910	0.8708	0.8641	0.8608	0.8587	0.8576
Max error	26.9000	25.6000	26.0667	25.6000	25.6600	25.8667
r^2	0.3723	0.4005	0.4097	0.4142	0.4170	0.4186
MedAE	0.3000	0.3500	0.3333	0.3250	0.3400	0.3333
test mean	0.6557	0.6557	0.6557	0.6557	0.6557	0.6557

Table 6.71: The Random Forest model results regarding the volume of vehicles of class 0 using the Md1 data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9854	0.9882	0.9891	0.9895	0.9897	0.9899
Test R^2 score	0.9188	0.9238	0.9255	0.9264	0.9267	0.9269
MAE	0.5965	0.5833	0.5783	0.5759	0.5745	0.5736
MSE	4.5137	4.2356	4.1434	4.0912	4.0723	4.0664
RMSE	2.1245	2.0580	2.0355	2.0227	2.0180	2.0165
Max error	708.9000	696.5500	677.5000	673.1750	676.3400	681.0333
r^2	0.9188	0.9238	0.9255	0.9264	0.9268	0.9269
MedAE	0.1000	0.1500	0.1333	0.1250	0.1200	0.1333
test mean	1.5838	1.5838	1.5838	1.5838	1.5838	1.5838

Table 6.72: The Random Forest model results regarding the volume of vehicles of class 0 using the Md1 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9817	0.9850	0.9860	0.9865	0.9868	0.9871
Test R^2 score	0.8954	0.9011	0.9037	0.9051	0.9056	0.9058
MAE	0.7897	0.7737	0.7677	0.7649	0.7632	0.7620
MSE	6.6914	6.3247	6.1579	6.0713	6.0359	6.0208
RMSE	2.5868	2.5149	2.4815	2.4640	2.4568	2.4537
Max error	692.9000	684.2000	694.7000	688.9500	686.2200	689.0500
r^2	0.8954	0.9011	0.9037	0.9051	0.9057	0.9059
MedAE	0.2000	0.2000	0.2000	0.2000	0.2000	0.2000
test mean	1.7000	1.7000	1.7000	1.7000	1.7000	1.7000

While only 40 estimates were used in sample A, for sample B, 50 were used. The Random Forest model that predicted the total volume with 60 estimators was aborted due to a lack of memory problem, given the machine could not allocate 939524096 bytes.

Random Forest results can be generalized. For test sample A and in all target, all errors obtained with the Light GBM model using the maximum number of leaves possible by the machine

Table 6.73: The Random Forest model results regarding the total volume of vehicles using the Md1 data set (sample A).

estimators nº	10	20	30	40
Train R^2 score	0.9976	0.9980	0.9981	0.9982
Test R^2 score	0.9868	0.9876	0.9879	0.9880
MAE	8.1963	7.9600	7.8761	7.8357
MSE	184.0324	172.6777	168.7344	166.9070
RMSE	13.5659	13.1407	12.9898	12.9193
Max error	721.7000	695.8500	692.3333	682.3500
r^2	0.9868	0.9876	0.9879	0.9880
MedAE	4.6000	4.5000	4.4333	4.4250
test mean	132.1792	132.1792	132.1792	132.1792

Table 6.74: The Random Forest model results regarding the total volume of vehicles using the Md1 data set (sample B).

estimators nº	10	20	30	40	50
Train R^2 score	0.9965	0.9971	0.9973	0.9974	0.9974
Test R^2 score	0.9806	0.9816	0.9819	0.9821	0.9822
MAE	10.7192	10.4522	10.3623	10.3153	10.2875
MSE	272.3244	257.7847	253.1893	250.7937	249.2838
RMSE	16.5023	16.0557	15.9119	15.8365	15.7887
Max error	697.4000	698.8000	693.9333	689.3000	695.9200
r^2	0.9806	0.9816	0.9819	0.9821	0.9822
MedAE	6.9000	6.7000	6.6667	6.6250	6.6200
test mean	131.5573	131.5573	131.5573	131.5573	131.5573

were higher than those obtained by Random forest with the number of estimators multiple of 10 maximum and the same mobility data. Furthermore, the R^2 scores and the explained variance obtained by Light GBM were lower than the values obtained by the best Random forest model with the same data. Therefore, for the first data set, all outputs, and test sample A, the results of the Random Forest model were better than the results previously obtained by the Light GBM. Although in class 0 with sample B, the random forest continued to be the best model, the same was not valid for the remaining outputs. Therefore, in almost all predicted outputs, the opposite of the results observed with sample A occurred for sample B, in which Light GBM obtained better results given the smallest errors and the highest R^2 scores and r^2.

6.2.3.2 Using data set Md7

This sub-sub section presents the results of the VCI data with separate fields for the day of the week.

The maximum number of estimators multiple of 10 that the model predicting the volume of vehicles of class A could use was 60, the same number as the prediction model for the same class of vehicles using mobility data with only one field for the day of the week. With the same 60

Table 6.75: The Random Forest model results regarding the volume of vehicles of class A using the Md7 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.8973	0.9125	0.9177	0.9202	0.9217	0.9227
Test R^2 score	0.4177	0.4423	0.4503	0.4543	0.4569	0.4585
MAE	0.7141	0.7032	0.6993	0.6975	0.6962	0.6956
MSE	1.3481	1.2913	1.2728	1.2634	1.2575	1.2538
RMSE	1.1611	1.1363	1.1282	1.1240	1.1214	1.1197
Max error	103.7000	103.8000	103.8667	103.7750	103.8000	103.8000
r^2	0.4186	0.4432	0.4512	0.4552	0.4578	0.4594
MedAE	0.4000	0.4500	0.4333	0.4250	0.4200	0.4333
test mean	0.9241	0.9241	0.9241	0.9241	0.9241	0.9241

estimators, the previous model obtained a better R^2 score (with both training and test data) and explained variance. Furthermore, all errors except the maximum error were higher in the model analyzed in Table 6.75 compared to the initial model using integer encoding for the day of the week. Therefore, the first data set overall obtained better results in predicting the volume of class A vehicles.

Table 6.76: The Random Forest model results regarding the volume of vehicles of class B using the Md7 data set (sample B).

estimators nº	10	20	30	40	50
Train R^2 score	0.9916	0.9932	0.9938	0.9940	0.9942
Test R^2 score	0.9555	0.9588	0.9598	0.9602	0.9606
MAE	14.3945	13.9832	13.8489	13.7794	13.7349
MSE	558.1015	516.8422	504.4104	498.2652	494.0864
RMSE	23.6242	22.7342	22.4591	22.3219	22.2281
Max error	472.2000	473.2500	408.5333	405.3250	415.7200
r^2	0.9555	0.9588	0.9598	0.9602	0.9606
MedAE	8.4000	8.2000	8.1333	8.1000	8.0800
test mean	123.3873	123.3873	123.3873	123.3873	123.3873

The table with the results of the Random Forest model regarding the volume of vehicles belonging to class B 6.76 does not show the results for 60 estimators because the machine did not have enough memory to finish the prediction. Since with the first data set it was possible to use 60 estimators, the results of the initial model will be compared with the results obtained with the second data set and minus 10 estimators. The results obtained by the R^2 score with the train data were very similar in the two models, although the value of the R^2 score of the initial model was better. The explained variance and R^2 score results using the test data were higher for the initial model, while all errors except for the maximum error were lower for the same model. In sum, as observed in the results regarding the prediction of the volume of class A vehicles, the models trained with the first data set overall obtained better results than the models trained with the second one.

Table 6.77: The Random Forest model results regarding the volume of vehicles of class C using the Md7 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9620	0.9685	0.9706	0.9717	0.9723	0.9727
Test R^2 score	0.7909	0.8030	0.8069	0.8087	0.8099	0.8106
MAE	1.8722	1.8257	1.8103	1.8027	1.7980	1.7950
MSE	7.9013	7.4446	7.2972	7.2263	7.1826	7.1547
RMSE	2.8109	2.7285	2.7013	2.6882	2.6800	2.6748
Max error	87.0000	123.6500	135.4333	142.0250	133.2400	132.2833
r^2	0.7910	0.8031	0.8070	0.8089	0.8100	0.8107
MedAE	1.2000	1.1500	1.1667	1.1500	1.1600	1.1500
test mean	4.8901	4.8901	4.8901	4.8901	4.8901	4.8901

The Random Forest model results regarding the volume of vehicles of class C using the second data set, which contains the mobility data set with seven fields for the day of the week, are presented in Table 6.77. As the number of estimators increased, all errors except the maximum error decreased. The explained variance and R^2 score results using both the train and test data were higher for the initial model trained with the first data set, while all errors, including the maximum error, were lower for the same model. As observed in the results regarding the prediction of the volume of class A and B vehicles, the models trained with the first data set overall obtained better results than the models trained with the second one regarding the class C vehicles.

Table 6.78: The Random Forest model results regarding the volume of vehicles of class D using the Md7 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.8750	0.8935	0.8996	0.9027	0.9046	0.9058
Test R^2 score	0.2912	0.3201	0.3299	0.3348	0.3382	0.3400
MAE	0.5969	0.5890	0.5863	0.5850	0.5840	0.5835
MSE	0.8946	0.8581	0.8458	0.8395	0.8353	0.8330
RMSE	0.9459	0.9263	0.9196	0.9163	0.9140	0.9127
Max error	29.1000	29.6000	30.9667	30.9500	31.4600	31.1667
r^2	0.2925	0.3215	0.3313	0.3362	0.3395	0.3414
MedAE	0.4000	0.3500	0.3667	0.3500	0.3600	0.3667
test mean	0.6557	0.6557	0.6557	0.6557	0.6557	0.6557

Tables 6.78 and 6.79 shows the results of the Random Forest model regarding the volume of vehicles of class D and 0, respectively, using the second pre-processed data set. The results obtained with seven fields to represent the day of the week instead of one were worse than the previously, regarding the prediction of class D and 0 vehicles. As opposed to many other models, the value of explained variance was different from the value of the R^2 score with the test sample.

The Random Forest model results regarding the total volume of vehicles using only the data set of VCI and seven fields for the day of the week are present in Table 6.80. The model was trained with a number of estimators multiple of 10 successively higher up to the machine's memory limit.

Table 6.79: The Random Forest model results regarding the volume of vehicles of class 0 using the Md7 data set (sample B).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9793	0.9830	0.9843	0.9848	0.9852	0.9855
Test R^2 score	0.8848	0.8916	0.8940	0.8951	0.8956	0.8960
MAE	0.8346	0.8182	0.8120	0.8089	0.8070	0.8059
MSE	7.3674	6.9338	6.7814	6.7061	6.6758	6.6538
RMSE	2.7143	2.6332	2.6041	2.5896	2.5837	2.5795
Max error	719.3000	713.7000	716.6667	715.1750	714.8000	716.1667
r^2	0.8848	0.8916	0.8940	0.8952	0.8956	0.8960
MedAE	0.2000	0.2000	0.2000	0.2000	0.2200	0.2167
test mean	1.7000	1.7000	1.7000	1.7000	1.7000	1.7000

Table 6.80: The Random Forest model results regarding the total volume of vehicles using the Md7 data set (sample B).

estimators nº	10	20	30	40	50
Train R^2 score	0.9920	0.9935	0.9940	0.9943	0.9944
Test R^2 score	0.9573	0.9606	0.9616	0.9620	0.9624
MAE	15.0246	14.5760	14.4274	14.3543	14.3043
MSE	598.5451	552.8397	538.4783	532.2430	527.3547
RMSE	24.4652	23.5125	23.2051	23.0704	22.9642
Max error	695.5000	693.0000	698.1333	682.4000	680.5400
r^2	0.9573	0.9606	0.9616	0.9620	0.9624
MedAE	8.9000	8.6500	8.6000	8.5750	8.5400
test mean	131.5573	131.5573	131.5573	131.5573	131.5573

Both the model trained with the first data set and the one trained with the second data set used a maximum of 50 estimators for the test sample B. For the model whose results are presented in Table 6.80, the R^2 score and the explained variance were lower than those obtained by the model in the previous subsection. At the same time, all errors except the maximum error were consistently higher, also with the maximum number of estimators. Therefore, it can be concluded that the change made from Integer encoding to One hot encoding worsened the overall results.

6.2.3.3 Using data sets MWwS and MW

In this subsection, the results of the models trained and tested using the third data set will be presented.

Random Forest model results regarding the volume of vehicles of class A using the mobility and the weather data set excluding and including the sky Level 1 Coverage field are present in Tables 6.81 and 6.82. Regarding the volume of vehicles of class A, the addition of meteorological data both with and without the sky Level 1 Coverage field worsened the results obtained by the Random Forest model compared with the model using the mobility data exclusively. The results with the addition of the sky Level 1 Coverage field were the worst of the three models mentioned.

Table 6.81: Random Forest model results regarding the volume of vehicles of class A using the MWwS data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9080	0.9217	0.9263	0.9286	0.9300	0.9309
Test R^2 score	0.4759	0.4984	0.5057	0.5094	0.5117	0.5133
MAE	0.6628	0.6527	0.6492	0.6475	0.6465	0.6458
MSE	1.1772	1.1265	1.1102	1.1019	1.0967	1.0932
RMSE	1.0850	1.0614	1.0536	1.0497	1.0472	1.0456
Max error	70.5000	70.1000	71.0000	69.6750	69.3000	69.3167
r^2	0.4765	0.4990	0.5063	0.5100	0.5123	0.5138
MedAE	0.4000	0.4000	0.4000	0.4000	0.3800	0.3833
test mean	0.9178	0.9178	0.9178	0.9178	0.9178	0.9178

Table 6.82: Random Forest model results regarding the volume of vehicles of class A using the MW data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9079	0.9216	0.9263	0.9286	0.9299	0.9309
Test R^2 score	0.4749	0.4977	0.5049	0.5086	0.5108	0.5125
MAE	0.6636	0.6533	0.6498	0.6481	0.6471	0.6463
MSE	1.1794	1.1283	1.1120	1.1037	1.0987	1.0951
RMSE	1.0860	1.0622	1.0545	1.0506	1.0482	1.0464
Max error	70.9000	70.4000	71.9667	74.8000	70.5000	70.3833
r^2	0.4755	0.4982	0.5054	0.5092	0.5114	0.5130
MedAE	0.4000	0.4000	0.4000	0.4000	0.4000	0.3833
test mean	0.9178	0.9178	0.9178	0.9178	0.9178	0.9178

Table 6.83: Random Forest model results regarding the volume of vehicles of class B using the MWwS data set (sample A).

estimators nº	10	20	30
Train R^2 score	0.9967	0.9973	0.9975
Test R^2 score	0.9822	0.9832	0.9835
MAE	9.4717	9.2252	9.1396
MSE	222.8657	210.1466	205.8617
RMSE	14.9287	14.4964	14.3479
Max error	531.0000	475.0000	491.8000
r^2	0.9822	0.9832	0.9835
MedAE	5.9000	5.7500	5.7000
test mean	124.0935	124.0935	124.0935

The results of the Random Forest model regarding the volume of vehicles of class B using the third pre-processed data set, excluding and including the sky Level 1 Coverage field are presented in Tables 6.83 and 6.84. While with all the features of the meteorological data and the mobility data, it was possible to use 40 estimators, without the sky Level 1 Coverage feature, it was only possible to use 30. With the same number of estimators (30), the error of the model trained with

Table 6.84: Random Forest model results regarding the volume of vehicles of class B using the MW data set (sample A).

estimators nº	10	20	30	40
Train R^2 score	0.9967	0.9973	0.9975	0.9976
Test R^2 score	0.9821	0.9832	0.9835	0.9837
MAE	9.4863	9.2393	9.1529	9.1110
MSE	223.2038	210.4918	206.1645	204.1245
RMSE	14.9400	14.5083	14.3584	14.2872
Max error	495.1000	470.8500	473.7000	478.8750
r^2	0.9821	0.9832	0.9835	0.9837
MedAE	5.9000	5.7500	5.7333	5.7000
test mean	124.0935	124.0935	124.0935	124.0935

all the meteorological data was smaller. However, the opposite is true when comparing the best results of the two models. Therefore, within the models trained with the third data set, the one that used the sky Level 1 Coverage field obtained better results than the without this feature. The results obtained with the third data set continued to be worse than those obtained with the first data set regarding the prediction of class B vehicles.

Table 6.85: Random Forest model results regarding the volume of vehicles of class C using the MWwS data set (sample A).

estimators nº	10	20	30	40
Train R^2 score	0.9736	0.9777	0.9791	0.9798
Test R^2 score	0.8517	0.8587	0.8611	0.8623
MAE	1.5383	1.5058	1.4945	1.4891
MSE	5.6429	5.3772	5.2838	5.2391
RMSE	2.3755	2.3189	2.2986	2.2889
Max error	223.0000	214.2000	199.1000	196.4000
r^2	0.8517	0.8587	0.8612	0.8623
MedAE	0.9000	0.9000	0.9000	0.9000
test mean	4.9245	4.9245	4.9245	4.9245

The results of the Random Forest model regarding the volume of vehicles of class C using the third pre-processed data set, excluding and including the sky Level 1 Coverage field are presented in Tables 6.85 and 6.86. In addition, the results of the Random Forest model regarding the volume of vehicles of class D using the third pre-processed data set, excluding and including the sky Level 1 Coverage field are presented in Tables 6.87 and 6.88. In both targets, the results obtained with and without the sky Level 1 Coverage feature were similar; however, without the sky Level 1 Coverage feature the results were slightly better. The results obtained were much better than those obtained by linear regression models and decision trees with the same data set. As with the other targets, the results obtained with the third data set without the sky Level 1 Coverage field continued to be worse than those obtained with the first data set regarding the forecast of class D vehicles.

Table 6.86: Random Forest model results regarding the volume of vehicles of class C using the MW data set (sample A).

estimators nº	10	20	30	40
Train R^2 score	0.9735	0.9776	0.9790	0.9797
Test R^2 score	0.8511	0.8581	0.8605	0.8617
MAE	1.5414	1.5094	1.4980	1.4923
MSE	5.6644	5.3999	5.3058	5.2601
RMSE	2.3800	2.3238	2.3034	2.2935
Max error	211.0000	201.8500	189.1333	188.9250
r^2	0.8511	0.8581	0.8606	0.8618
MedAE	0.9000	0.9000	0.9000	0.9000
test mean	4.9245	4.9245	4.9245	4.9245

Table 6.87: Random Forest model results regarding the volume of vehicles of class D using the MWwS data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.8983	0.9138	0.9190	0.9216	0.9232	0.9243
Test R^2 score	0.4292	0.4553	0.4637	0.4676	0.4700	0.4719
MAE	0.5232	0.5157	0.5132	0.5121	0.5113	0.5108
MSE	0.7224	0.6894	0.6787	0.6737	0.6707	0.6683
RMSE	0.8499	0.8303	0.8238	0.8208	0.8189	0.8175
Max error	48.7000	44.1500	40.3667	40.6500	40.7400	40.5000
r^2	0.4298	0.4559	0.4643	0.4682	0.4706	0.4725
MedAE	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
test mean	0.6596	0.6596	0.6596	0.6596	0.6596	0.6596

Table 6.88: Random Forest model results regarding the volume of vehicles of class D using the MW data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.8981	0.9136	0.9188	0.9214	0.9229	0.9240
Test R^2 score	0.4268	0.4531	0.4617	0.4657	0.4681	0.4700
MAE	0.5243	0.5167	0.5142	0.5131	0.5123	0.5117
MSE	0.7253	0.6920	0.6811	0.6762	0.6731	0.6707
RMSE	0.8517	0.8319	0.8253	0.8223	0.8204	0.8190
Max error	49.0000	44.3500	39.4667	39.9750	40.1800	40.0333
r^2	0.4274	0.4537	0.4623	0.4663	0.4687	0.4706
MedAE	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
test mean	0.6596	0.6596	0.6596	0.6596	0.6596	0.6596

Finally, class 0 vehicles were predicted, hence the prediction of vehicle detection failures in the sensors. Tables 6.89 and 6.90 show the results of the Random Forest model regarding the volume of vehicles of class 0 using the mobility data set and the weather data set, including the sky Level 1 Coverage field. The results obtained by the Random Forest models with the third data set, with and without the sky Level 1 Coverage features, had no significant differences in the

Table 6.89: Random Forest model results regarding the volume of vehicles of class 0 using the MWwS data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9831	0.9861	0.9871	0.9875	0.9877	0.9880
Test R^2 score	0.9039	0.9088	0.9106	0.9117	0.9121	0.9121
MAE	0.7176	0.7036	0.6989	0.6963	0.6948	0.6937
MSE	5.3412	5.0693	4.9725	4.9082	4.8853	4.8850
RMSE	2.3111	2.2515	2.2299	2.2154	2.2103	2.2102
Max error	643.8000	654.7000	641.5000	643.2250	651.6800	661.7500
r^2	0.9039	0.9088	0.9106	0.9117	0.9121	0.9121
MedAE	0.2000	0.1500	0.1667	0.1750	0.1800	0.1667
test mean	1.5838	1.5838	1.5838	1.5838	1.5838	1.5838

Table 6.90: Random Forest model results regarding the volume of vehicles of class 0 using the MW data set (sample A).

estimators nº	10	20	30	40	50	60
Train R^2 score	0.9831	0.9861	0.9871	0.9875	0.9877	0.9879
Test R^2 score	0.9036	0.9088	0.9106	0.9116	0.9120	0.9121
MAE	0.7185	0.7045	0.6997	0.6972	0.6956	0.6945
MSE	5.3609	5.0682	4.9725	4.9161	4.8931	4.8846
RMSE	2.3154	2.2513	2.2299	2.2172	2.2120	2.2101
Max error	654.5000	660.0500	654.8000	652.0250	658.7200	670.0500
r^2	0.9036	0.9089	0.9106	0.9116	0.9120	0.9122
MedAE	0.2000	0.1500	0.1667	0.1750	0.1800	0.1667
test mean	1.5838	1.5838	1.5838	1.5838	1.5838	1.5838

metrics presented. The results obtained with the third data set continued to be worse than those obtained with the first data set.

Table 6.91: Random Forest model results regarding the total volume of vehicles using the MWwS data set (sample A).

estimators nº	10	20	30	40
Train R^2 score	0.9971	0.9975	0.9977	0.9978
Test R^2 score	0.9838	0.9847	0.9850	0.9852
MAE	9.7546	9.5009	9.4143	9.3710
MSE	226.2310	213.6501	209.3173	207.2611
RMSE	15.0410	14.6168	14.4678	14.3966
Max error	686.6000	686.2500	674.3000	661.9000
r^2	0.9838	0.9847	0.9850	0.9852
MedAE	6.2000	6.0500	6.0000	5.9750
test mean	132.1792	132.1792	132.1792	132.1792

Both the model regarding the total volume of vehicles trained with the first data set and the one training with the third data set, without the sky Level 1 Coverage field, used a maximum of 40

Table 6.92: Random Forest model results regarding the total volume of vehicles using the MW data set (sample A).

estimators nº	10	20	30
Train R^2 score	0.9970	0.9975	0.9977
Test R^2 score	0.9838	0.9847	0.9850
MAE	9.7684	9.5149	9.4281
MSE	226.6468	214.1297	209.7412
RMSE	15.0548	14.6332	14.4824
Max error	656.0000	670.9000	658.9667
r^2	0.9838	0.9847	0.9850
MedAE	6.2000	6.0500	6.0000
test mean	132.1792	132.1792	132.1792

estimators for the test sample A. The results of the latter model are presented in Table 6.91. For the model whose results are present in this table, the R^2 score and the explained variance were lower than those obtained by the previous subsection model. Meanwhile, all errors except the maximum error were consistently higher with the maximum number of estimators. With the addition of the sky Level 1 Coverage field, the results obtained regarding the total volume of vehicles were also worse in comparison to the results achieved with the same data set without this feature; in this way, the results with the third data set without feature engineering remained worse than the results obtained with the mobility data, even with the same number of estimators.

In all tests performed, none of the models was able to use 70 estimators. Specifically, the Random Forest algorithms used a maximum of 60 estimators in the predictions of a single y due to the machine's memory, which cannot allocate 939524096 bytes. In addition, as the number of estimators increased, the time needed to train the model also increased. Also, as the number of estimators increased, the R^2 score increased, and the errors decreased. The results obtained with the first data set were the best for all targets.

6.2.4 Ensemble of the previous models

The ensemble model will only use the models of a single output that obtained the best results. Therefore, Linear Regression and the neural networks Models will not be part of the ensemble model because of their below-average results. The Multi-Output Regressor and the Chain Regressor will not be used as they are multi-output models. Given the limited power of the machine, many algorithms had to use less than optimal parameters.

6.2.4.1 Simple averaging

The combination of the XGB Regressor using 1000 estimators with the Decision Tree Regressor with equal weights was tested with the first data set. The results of the ensemble model with the mobility data, present in Table 6.93, were worse than those obtained with the initial Decision Tree

Regressor model and better than the XGB Regressor estimator with 1000 estimators. The same test was done with the XGB Regressor with 10000 estimators instead of 1000 estimators.

Table 6.93: Results of the simple averaging model using the Md1 data set and with a validation sample.

	XGB Regressor (estimators: 1000)	Decision Tree	Ensemble
Train R^2 score	0.9714	1.0000	0.9974
Validation R^2 score			0.9974
MAE validation			3.6891
MSE validation			36.1940
RMSE validation			6.0161
Max error validation			191.0484
explained variance validation			0.9974
MedAE validation			2.3314
validation mean			132.1551
Test R^2 score	0.9711	0.9800	0.9840
MAE test	12.3259	8.4724	8.8088
MSE test	403.1959	278.5584	223.8374
RMSE test	20.0797	16.6901	14.9612
Max error test	735.2572	727.0000	622.3772
explained variance test	0.9711	0.9800	0.9840
MedAE test	7.7940	3.0000	4.6739
test mean	132.1792	132.1792	132.1792

The combination of the XGB Regressor using 10000 estimators with the Decision Tree Regressor with equal weights was tested with the first and third data sets. The results of these simple averaging models are present in Tables 6.94 and 6.95. The results of the ensemble model with the mobility data, present in Table 6.94, were worse than those obtained with the initial Decision Tree Regressor model and better than the XGB Regressor estimator with 10000 estimators.

Finally, the results obtained with the third data set by the simple averaging model are present in Table 6.95 and show that the ensemble model improved the results compared to the initial estimators' results. However, the results obtained by the ensemble model were worse than those obtained by decision trees only with mobility data.

All tests, both successful and unsuccessful, performed with the simple averaging models are shown in Table 6.96. Due to lack of time, several relevant tests were not performed. The missing tests could have gotten better results than the Decision Trees with the mobility data. Then the ensemble models used belong to the weighted averaging type.

6.2.4.2 Weighted averaging

The results obtained with the same weights for all models are the same as those obtained by Simple averaging with the same models previously. The only tests performed used mobility data and the XGB Regressor estimators with 1000 estimators and the Decision Tree Regressor and are present

Table 6.94: Results of the simple averaging model using the Md1 data set and with a validation sample.

	XGB Regressor (estimators: 10000)	Decision Tree	Ensemble
Train R^2 score	0.9714	1.0000	0.9976
Validation R^2 score			0.9976
MAE validation			3.6209
MSE validation			33.6806
RMSE validation			5.8035
Max error validation			207.4820
explained variance validation			0.9976
MedAE validation			2.3090
validation mean			132.1551
Test R^2 score	0.9711	0.9800	0.9849
MAE test	12.3259	8.4724	8.5845
MSE test	403.1959	278.5584	211.0035
RMSE test	20.0797	16.6901	14.5260
Max error test	735.2572	727.0000	628.6761
explained variance test	0.9711	0.9800	0.9849
MedAE test	7.7940	3.0000	4.5940
test mean	132.1792	132.1792	132.1792

Table 6.95: Results of the simple averaging model using the MW data set and with a validation sample.

	XGB Regressor (estimators: 10000)	Decision Tree	Ensemble
Train R^2 score	0.9714	1.0000	0.9912
Validation R^2 score			0.9912
MAE validation			7.1320
MSE validation			122.8932
RMSE validation			11.0857
Max error validation			409.2491
explained_variance validation			0.9912
MedAE validation			4.6480
validation mean			132.1551
Test R^2 score	0.9711	0.9736	0.9827
MAE test	12.3259	11.5756	10.1153
MSE test	403.1959	368.0480	241.8535
RMSE test	20.0797	19.1846	15.5516
Max error test	735.2572	863.0000	680.7308
explained_variance test	0.9711	0.9736	0.9827
MedAE test	7.7940	7.0000	6.5078
test mean	132.1792	132.1792	132.1792

Table 6.96: Results of simple averaging models tested.

Data set	estimators	final estimator	MAE
MW	XGB Regressor (estimators: 10000)	Decision Tree Regressor	10.1152885
Md1	XGB Regressor (estimators: 10000)	Decision Tree Regressor	8.5845386
Md1	XGB Regressor (estimators: 1000)	Decision Tree Regressor	8.8088015
Md1	Random Forest Regressor (estimators: 50)	XGB Regressor (estimators: 10000)	Memory Error
Md1	Random Forest Regressor (estimators: 50)	XGB Regressor (estimators: 5000)	Memory Error
Md1	Random Forest Regressor (estimators: 50)	XGB Regressor (estimators: 1000)	Memory Error
Md1	Random Forest Regressor (estimators: 40)	XGB Regressor (estimators: 1000)	Memory Error
Md1	Random Forest Regressor (estimators: 30)	XGB Regressor (estimators: 1000)	Memory Error

in Table 6.97. The best result for the validation and test sample was obtained by the model with XGB Regressor with a weight of 0.3 and Decision Trees with a weight of 0.7. Although the MAE with the test sample was lower for the decision trees, the remaining metrics obtained equal or better values than the best ensemble model mentioned.

6.2.4.3 Stacking

Since the best models obtained better results with the data set with only mobility data, the Stacking algorithms used this data set exclusively. Tables 6.98, 6.99, and 6.99 presents all the tests that were carried out, including the models that were unsuccessfully tried to create because a memory error was obtained.

Table 6.98 presents the results of the Stacking models using the Random Forest as the final estimator. This table presents the results obtained using as estimators the XGB Regressor model with 1000 estimators and the LGM Regressor with 131072 leaves, and, as a final estimator, the Random Forest model with 20 estimators. When trying to increase the number of Random Forest estimators from 20 to 30, the machine could not train the final model due to memory issues. To increase the number of estimators used in the Random Forest final estimator to 30, the number of leaves of the LGB Regressor was reduced to 65536. However, this increase was limited to 30, as the same memory problem was gotten with 40 estimators in Random Forest final estimator (and with the XGB Regressor with 1000 estimators and the LGM Regressor with 65536 leaves).

Then tests were performed with the XGB Regressor as the final estimator. The Random Forest estimators with 30 estimators and the LGBM Regressor with 65536 leaves were used with the XGB Regressor with 100 and 1000 estimators. Given that both got a memory error, the number of Random Forest estimators was reduced to 20 and then 10. Finally, with the 2 estimators: Random Forest (with 10 estimators) and the LGBM Regressor (with 65536 leaves) and the XGB Regressor with 100 estimators as the final estimator, it was possible to obtain results. Then the

Table 6.97: Results of the ensemble of the Decision Tree Regressor model with the XGB Regressor model with 1000 estimators using the Md1 data set.

	XGB	DT						
XGB weight	1	0	0.4	0.6	0.55	0.45	0.7	0.3
Decision tree weight	0	1	0.6	0.4	0.45	0.55	0.3	0.7
Train R^2 score	0.97	1.00	1.00	0.99	0.99	0.99	0.99	1.00
Validation								
val R^2 score			1.00	0.99	0.99	0.99	0.99	1.00
MAE val	12.36	9.95	4.92	7.38	6.76	5.53	8.61	3.69
MSE val			64.33	144.71	121.60	81.41	196.96	36.19
RMSE val			8.02	12.03	11.03	9.02	14.03	6.02
Max error val			254.73	382.10	350.26	286.57	445.78	191.05
r^2 val			1.00	0.99	0.99	0.99	0.99	1.00
MedAE val			3.11	4.66	4.27	3.50	5.44	2.33
val mean			132.16	132.16	132.16	132.16	132.16	132.16
Test								
Test R^2 score	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98
MAE test	12.33	8.47	9.07	9.83	9.61	9.23	10.33	8.81
MSE test	403.20	278.56	222.87	248.05	238.36	225.83	273.68	223.84
RMSE test	20.08	16.69	14.93	15.75	15.44	15.03	16.54	14.96
Max error test	735.26	727.00	638.50	731.95	662.69	646.57	732.78	622.38
r^2 test	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98
MedAE test	7.79	3.00	5.21	6.10	5.89	5.45	6.51	4.67
test mean	132.18	132.18	132.18	132.18	132.18	132.18	132.18	132.18

Table 6.98: Results of the Stacking models using the Random Forest as the final estimator.

estimators	final estimator	MAE
XGB Regressor (estimators: 1000) & LGBM Regressor (num leaves: 131072)	Random Forest (estimators: 20)	9.4738
XGB Regressor (estimators: 1000) & LGBM Regressor (num leaves: 131072)	Random Forest (estimators: 30)	Memory Error
XGB Regressor (estimators: 1000) & LGBM Regressor (num leaves: 65536)	Random Forest (estimators: 30)	9.6257
XGB Regressor (estimators: 1000) & LGBM Regressor (num leaves: 65536)	Random Forest (estimators: 40)	Memory Error

same estimators were tested with the final estimator the XGB Regressor with 500 estimators. Then the Random Forest and Decision Tree estimators were used, using the XGB Regressor as the final estimator. Only one combination of the tests performed was successful, the use of the two Random Forest estimators (with 10 estimators) and the Decision Tree Regressor.

Then tests were performed with the LGBM Regressor as the final estimator. This final estimator was combined with both Random Forest with Decision Trees and Random Forest with the

Table 6.99: Results of the Stacking models using the XGB Regressor as the final estimator.

estimators	final estimator	MAE
Random Forest (estimators: 30) & LGBM Regressor (num leaves: 65536)	XGB Regressor (estimators: 1000)	Memory Error
Random Forest (estimators: 30) & LGBM Regressor (num leaves: 65536)	XGB Regressor (estimators: 100)	Memory Error
Random Forest (estimators: 20) & LGBM Regressor (num leaves: 65536)	XGB Regressor (estimators: 100)	Memory Error
Random Forest (estimators: 10) & LGBM Regressor (num leaves: 65536)	XGB Regressor (estimators: 100)	8.2646
Random Forest (estimators: 10) & LGBM Regressor (num leaves: 65536)	XGB Regressor (estimators: 500)	8.2698
Random Forest (estimators: 10) & Decision Tree Regressor	XGB Regressor (estimators: 1000)	7.8715
Random Forest (estimators: 40) & Decision Tree Regressor	XGB Regressor (estimators: 1000)	Memory Error
Random Forest (estimators: 30) & Decision Tree Regressor	XGB Regressor (estimators: 1000)	Memory Error
Random Forest (estimators: 20) & Decision Tree Regressor	XGB Regressor (estimators: 1000)	Memory Error

XGB Regressor. The best result obtained with the Stacking models having the XGB Regressor as the final estimator was obtained with the Random Forest and Decision Tree Regressor estimators.

Table 6.100: Results of the Stacking models using the LGBM Regressor as the final estimator.

estimators	final estimator	MAE
Random Forest (estimators: 30) & Decision Tree Regressor	LGBM Regressor (num leaves: 65536)	Memory Error
Random Forest (estimators: 20) & Decision Tree Regressor	LGBM Regressor (num leaves: 65536)	7.6504
Random Forest (estimators: 10) & Decision Tree Regressor	LGBM Regressor (num leaves: 131072)	7.8277
Random Forest (estimators: 20) & Decision Tree Regressor	LGBM Regressor (num leaves: 131072)	Memory Error
Random Forest (estimators: 20) & XGB Regressor (estimators: 100)	LGBM Regressor (num leaves: 131072)	Memory Error
Random Forest (estimators: 20) & XGB Regressor (estimators: 100)	LGBM Regressor (num leaves: 65536)	Memory Error
Random Forest (estimators: 10) & XGB Regressor (estimators: 100)	LGBM Regressor (num leaves: 65536)	8.3798

Finally, the decision trees were tested as the final estimator, however, some tests are missing due to lack of time and the results obtained were much worse than those obtained previously.

After presenting all the tests performed, the results of the ensemble model obtained were compared with the initial estimators. The results obtained by the Stacking model in Table 6.102

Table 6.101: Results of the Stacking models using the Decision Tree as the final estimator.

estimators	final estimator	MAE
Random Forest (estimators: 10) & XGB Regressor (estimators: 100)	Decision Tree Regressor	12.7852
Random Forest (estimators: 20) & XGB Regressor (estimators: 100)	Decision Tree Regressor	12.4433
Random Forest (estimators: 30) & XGB Regressor (estimators: 100)	Decision Tree Regressor	Memory Error

and 6.103 were worse than the results of the Random Forest model with 20 and 30 estimators, respectively.

Table 6.102: Comparison of the models used in the ensemble with the final ensemble model (Random Forest using 20 estimators).

	estimator 1	estimator 2	final estimator	Ensemble
	XGB Regressor (estimators: 1000)	light GBM (num leaves: 131072)	Random Forest (estimators: 20)	Stacking
train R^2 score	0.9714	0.9938	0.9980	0.9912
test R^2 score	0.9711	0.9880	0.9876	0.9857
MAE	12.3259	8.5981	7.9600	9.4738
MSE	403.1959	167.2301	172.6777	200.0130
Root MSE	20.0797	12.9317	13.1407	14.1426
Max error	735.2572	707.4645	695.8500	706.6000
r^2	0.9711	0.9880	0.9876	0.9857
MedAE	7.7940	5.6638	4.5000	6.2500
test mean	132.1792	132.1792	132.1792	132.1792

Table 6.103: Comparison of the models used in the ensemble with the final ensemble model (Random Forest using 30 estimators).

	estimator 1	estimator 2	final estimator	Ensemble
	light GBM (num leaves: 65536)	XGB Regressor (estimators: 1000)	Random Forest (estimators: 30)	Stacking
Train R^2 score	0.9920	0.9714	0.9981	0.9898
Test R^2 score	0.9876	0.9711	0.9879	0.9855
MAE	8.8361	12.3259	7.8761	9.6257
MSE	173.1877	403.1959	168.7344	202.8826
Root MSE	13.1601	20.0797	12.9898	14.2437
Max error	719.7618	735.2572	692.3333	722.7333
r^2	0.9876	0.9711	0.9879	0.9855
MedAE	5.8941	7.7940	4.4333	6.4333
test mean	132.1792	132.1792	132.1792	132.1792

Tables 6.104, 6.105 and 6.106 present the results of the Stacking models with the XGB Regressor as the final estimator. When using Random Forest with 10 estimators and light GBM with

65536 leaves as initial estimators and XGB Regressor with 100 and 500 estimators as final estimator, some errors were smaller than the models used without ensemble, namely the MSE and the Root MSE. Finally, when using Random Forest with 10 estimators and Decision Tree Regressor as initial estimators and XGB Regressor with 1000 estimators as final estimator, some errors were smaller than the models used without ensemble, namely the Mean Absolute Error, the Mean Squared Error, and the Root MSE.

Table 6.104: Comparison of the models used in the ensemble with the final ensemble model (XGB Regressor using 100 estimators).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 10)	light GBM (num leaves: 65536)	XGB Regressor (estimators: 100)	Stacking
Train R^2 score	0.9976	0.9920	0.9570	0.9948
Test R^2 score	0.9868	0.9876	0.9570	0.9883
MAE	8.1963	8.8361	15.4406	8.2646
MSE	184.0324	173.1877	600.8939	163.8700
Root MSE	13.5659	13.1601	24.5131	12.8012
Max error	721.7000	719.7618	751.0862	720.0037
explained variance	0.9868	0.9876	0.9570	0.9883
MedAE	4.6000	5.8941	9.7776	5.2186
test mean	132.1792	132.1792	132.1792	132.1792

Table 6.105: Comparison of the models used in the ensemble with the final ensemble model (XGB Regressor using 500 estimators).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 10)	light GBM (num leaves: 65536)	XGB Regressor (estimators: 500)	Stacking
Train R^2 score	0.9976	0.9920	0.9684	0.9948
Test R^2 score	0.9868	0.9876	0.9682	0.9882
MAE	8.1963	8.8361	12.9617	8.2698
MSE	184.0324	173.1877	443.4438	164.3755
Root MSE	13.5659	13.1601	21.0581	12.8209
Max error	721.7000	719.7618	739.7271	720.5920
explained variance	0.9868	0.9876	0.9682	0.9882
MedAE	4.6000	5.8941	8.1792	5.2327
test mean	132.1792	132.1792	132.1792	132.1792

In Tables 6.107 and 6.108, the ensemble model obtained the smallest errors compared to initially used models. The next model to obtain better results was the Random Forest in both tables. Table 6.109 shows the results of Stacking using Random Forest with ten estimators and the XGB Regressor with 100 estimators as initial estimators, and light GBM with 65536 leaves. In this Table, the results obtained by Random Forest with ten estimators were better than those obtained by the Ensemble model.

Table 6.106: Comparison of the models used in the ensemble with the final ensemble model (XGB Regressor using 1000 estimators).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 10)	Decision Tree Regressor	XGB Regressor (estimators: 1000)	Stacking
Train R^2 score	0.9976	1.0000	0.9714	0.9982
Test R^2 score	0.9868	0.9800	0.9711	0.9869
Mean Absolute Error	8.1963	8.4724	12.3259	7.8715
Mean Squared Error	184.0324	278.5584	403.1959	183.1384
Root MSE	13.5659	16.6901	20.0797	13.5329
Max error	721.7000	727.0000	735.2572	723.9507
explained variance	0.9868	0.9800	0.9711	0.9869
MedAE	4.6000	3.0000	7.7940	4.0011
test mean	132.1792	132.1792	132.1792	132.1792

Table 6.107: Comparison of the models used in the ensemble with the final ensemble model (light GBM using 65536 leaves).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 20)	Decision Tree Regressor	light GBM (num leaves: 65536)	Stacking
Train R^2 score	0.9980	1.0000	0.9920	0.9986
Test R^2 score	0.9876	0.9800	0.9876	0.9877
MAE	7.9600	8.4724	8.8361	7.6504
MSE	172.6777	278.5584	173.1877	171.9903
Root MSE	13.1407	16.6901	13.1601	13.1145
Max error	695.8500	727.0000	719.7618	700.1720
explained variance	0.9876	0.9800	0.9876	0.9877
MedAE	4.5000	3.0000	5.8941	3.9066
test mean	132.1792	132.1792	132.1792	132.1792

Table 6.108: Comparison of the models used in the ensemble with the final ensemble model (light GBM using 131072 leaves).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 10)	Decision Tree Regressor	light GBM (num leaves: 131072)	Stacking
Train R^2 score	0.9976	1.0000	0.9938	0.9984
Test R^2 score	0.9868	0.9800	0.9880	0.9871
Mean Absolute Error	8.1963	8.4724	8.5981	7.8277
Mean Squared Error	184.0324	278.5584	167.2301	180.4616
Root MSE	13.5659	16.6901	12.9317	13.4336
Max error	721.7000	727.0000	707.4645	710.7648
explained variance	0.9868	0.9800	0.9880	0.9871
MedAE	4.6000	3.0000	5.6638	3.9924
test mean	132.1792	132.1792	132.1792	132.1792

Table 6.109: Comparison of the models used in the ensemble with the final ensemble model (light GBM using 65536 leaves).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 10)	XGB Regressor (estimators: 100)	light GBM (num leaves: 65536)	Stacking
Train R^2 score	0.9976	0.9570	0.9920	0.9967
Test R^2 score	0.9868	0.9570	0.9876	0.9865
Mean Absolute Error	8.1963	15.4406	8.8361	8.3798
Mean Squared Error	184.0324	600.8939	173.1877	187.8952
Root MSE	13.5659	24.5131	13.1601	13.7075
Max error	721.7000	751.0862	719.7618	724.8339
explained variance	0.9868	0.9570	0.9876	0.9865
MedAE	4.6000	9.7776	5.8941	4.8436
test mean	132.1792	132.1792	132.1792	132.1792

Finally, the Decision Tree Regressor model was used as the final estimator. The results are presented in Tables 6.110 and 6.111. The same estimators used in table 6.110 were used in table 6.111, however, the number of estimators in the Random Forest model was increased to 20. In both tables, the result of the ensemble model was much worse than the results of the Random Forest models with 10 and 20 estimators and the Decision Tree Regressor models used in the Ensemble.

Table 6.110: Comparison of the models used in the ensemble with the final ensemble model (Decision Tree Regressor).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 10)	XGB Regressor (estimators: 100)	Decision Tree Regressor	Stacking
Train R^2 score	0.9976	0.9570	1.0000	0.9812
Test R^2 score	0.9868	0.9570	0.9800	0.9712
Mean Absolute Error	8.1963	15.4406	8.4724	12.7852
Mean Squared Error	184.0324	600.8939	278.5584	401.4658
Root MSE	13.5659	24.5131	16.6901	20.0366
Max error	721.7000	751.0862	727.0000	854.0000
explained variance	0.9868	0.9570	0.9800	0.9712
MedAE	4.6000	9.7776	3.0000	8.0000
test mean	132.1792	132.1792	132.1792	132.1792

6.3 Auto ML algorithms

In opposition to the previously presented machine learning and ensemble models, auto ML models perform feature selection; therefore, models only need to be tested with the data set with all fields. Since the Linux machine did not have enough memory resources, the use of weather data was not possible. Therefore, the tests performed used meteorological data exclusively. The only library used was the Auto-Sklearn library. In this library, the parameter time_left_for_this_task defines a

Table 6.111: Comparison of the models used in the ensemble with the final ensemble model (Decision Tree Regressor).

	estimator 1	estimator 2	final estimator	Ensemble
	Random Forest (estimators: 20)	XGB Regressor (estimators: 100)	Decision Tree Regressor	Stacking
Train R^2 score	0.9980	0.9570	1.0000	0.9826
Test R^2 score	0.9876	0.9570	0.9800	0.9728
Mean Absolute Error	7.9600	15.4406	8.4724	12.4433
Mean Squared Error	172.6777	600.8939	278.5584	380.4034
Root MSE	13.1407	24.5131	16.6901	19.5039
Max error	695.8500	751.0862	727.0000	705.0000
explained variance	0.9876	0.9570	0.9800	0.9728
MedAE	4.5000	9.7776	3.0000	8.0000
test mean	132.1792	132.1792	132.1792	132.1792

time limit in seconds to search for the most suitable models. The chance of finding better models rises with the increase of this parameter value. The parameter per_run_time_limit is an integer and optional parameter that by default corresponds to 1/10 of the time_left_for_this_task parameter. The first parameter is the time limit for a single call to the machine learning model. Therefore, the model fitting will be terminated if the ML algorithm runs over the time limit. It is recommended to set this value high enough in order that typical machine learning algorithms can fit the provided training data.

Table 6.112: Results of Auto-Sklearn model using the mobility data exclusively and with the parameter per_run_time_limit (run time limit) with value 100 and parameter time_left_for_this_task (time for task) with value 1000.

	class A	class B	class C	class D	class 0	total volume
run time limit	100	100	100	100	100	100
time for task	1000	1000	1000	1000	1000	1000
Train R^2 score	0,0697	0,0193	0,1669	0,0660	0,1332	0,1844
Test R^2 score	0,0698	0,0193	0,1669	0,0661	0,1328	0,1845
MAE	1,0060	90,6626	4,2426	0,7933	2,3708	87,3689
MSE	2,1296	12270,9429	31,4867	1,1830	55,7324	11411,7053
RMSE	1,4593	110,7743	5,6113	1,0877	7,4654	106,8256
Max error	113,1657	564,7437	143,6584	39,2833	621,4224	612,7251
r^2	0,0698	0,0193	0,1669	0,0661	0,1328	0,1845
MedAE	0,8492	86,6955	3,5716	0,6076	1,5776	82,6093
test mean	0,9241	123,4426	4,8963	0,6567	1,7086	131,6283

The results of Auto-Sklearn models using the mobility data exclusively and with the parameter per_run_time_limit with value 100 and the parameter time_left_for_this_task with value 1000 are presented in Table 6.112. Regarding the prediction of all outputs and using the mobility data exclusively and with a run time limit (per_run_time_limit) of 100 and a time for task

(time_left_for_this_task) of 1000, the model with a higher rank was of the type gradient boosting model. The final ensemble model was only composed of the gradient boosting model; therefore, this model had an ensemble weight of 1.0. While in the prediction of class B vehicles, the model id is 14, in the prediction of the remaining outputs, the model id is 12. The metric to evaluate the results for all outputs was the R^2 score. Regarding the prediction of vehicles of class A and with the previously mentioned parameters, the number of target algorithms run was 18. Of these algorithms, the number of successful target algorithms was 2, the number of crashed target algorithms was 6, the number of target algorithms that exceeded the time limit was 7, and the number of target algorithms that exceeded the memory limit was 3. Finally, the best validation score (R^2 score) was 0.069461. Regarding the prediction of vehicles of class B and with the previously mentioned parameters, the number of target algorithms run was 20; of each, only one algorithm was successful. Of the remaining run algorithms, 10 crashed, 6 exceeded the time limit, and 3 exceeded the memory limit. The best validation score was 0.019316. Regarding the prediction of vehicles of classes C, D, and 0, the number of target algorithm runs was 18. Of the 18 algorithms, 2 were successful, 5 crashed, 8 exceeded the time limit, and 3 exceeded the memory limit. The best validation score of the model predicting the vehicles class C, D, and 0, was 0.166867, 0.066087, and 0.133744, respectively. Regarding the prediction of the total volume of vehicles, the best validation score was 0.184404, and the number of target algorithms run was the lowest, only 17. Of the 17, the number of successful target algorithm runs was 2. Of the remaining 15, the number of crashed target algorithm runs was 5, the number of target algorithms that exceeded the time limit was 7, and the number of target algorithms that exceeded the memory limit was 3.

Since the time limits to run the model were low, the results obtained were worse than those obtained by all other models. In sum, the number of successful target algorithm runs for all outputs was one or two. The hyper-parameters of the best and only prediction model used by the ensemble model for each of the outputs are specified in Table 6.113. The ideal hyper-parameters were the same for classes A, C, D, 0 and for the total volume target.

Table 6.113: Hyper-parameters of the only and best prediction model for each of the outputs using the mobility data exclusively.

	class A / C / D / 0 / total volume	class B
run time limit	100	100
time for task	1000	1000
l2 regularization	6,00E-10	3,00E-06
learning rate	0,1229	0,0157
max iter	2	2
max leaf nodes	26	7
min samples leaf	8	23
n iter no change	0	0
random state	1	1
validation fraction	None	None
warm start	True	True

To obtain better models, the value of the parameters per_run_time_limit and time_left_for_this_task was changed to higher values, to identify the most appropriate model within those available. The results of this models are presented in Table 6.114.

Table 6.114: Results of Auto-Sklearn model using the Md1 data and with the parameter per_run_time_limit (run time limit) with value 10000 and parameter time_left_for_this_task (time for task) with value 100000.

	class A	class B	class C	total volume
run time limit	10000	10000	10000	10000
time for task	100000	100000	100000	100000
Train R^2 score	0.60345	0.97896	0.85099	0.98554
Test R^2 score	0.53310	0.97764	0.84830	0.98262
Mean Absolute Error	0.65278	10.28082	1.58474	10.09181
Mean Squared Error	1.06889	279.77563	5.73382	243.21936
Root Mean Squared Error	1.03387	16.72649	2.39454	15.59549
Max error	80.90900	443.75402	134.81543	570.40573
explained variance	0.53310	0.97764	0.84830	0.98262
MedAE	0.40728	6.36644	0.97695	6.48118
test mean	0.92413	123.44258	4.89626	131.62826

Table 6.114 presents the results obtained with the hyper-parameters run time limit with a value of 10000 and the time for task of 100000. The time for task of 100000 seconds corresponds to approximately 27.78 hours. Since this experiment lasted more than one day of execution, only classes A, B, C, and total volume were tested. The value of the R^2 score using the test and training data was very close for all targets except the prediction of the volume of class A vehicles. The maximum errors were relatively high compared to the mean and median errors. The errors in the prediction of the total volume of vehicles were smaller than the errors in the prediction of the volume of class B vehicles.

Table 6.115: Models regarding the volume of class A vehicles with run time limit parameter of 10000 and the time task with value 100000. All models belong to the gradient boosting type.

model id	rank	ensemble weight	cost	duration	l2 regularization	learning rate	max leaf nodes
80	1	0.24	0.472780	1510.86775	0.25490	0.08256	2016
108	2	0.06	0.474642	1456.52065	0.08443	0.08256	512
102	3	0.22	0.477671	912.74219	4.69625e-06	0.17630	429
82	4	0.08	0.480437	1824.82168	0.28316	0.04026	1583
87	5	0.22	0.492781	1047.67295	0.03375	0.40315	2012
97	6	0.10	0.499834	1069.05728	0.26939	0.33355	2012
58	7	0.08	0.556569	757.24505	0.30195	0.97664	336

The ensemble model used seven gradient boosting models for the class A prediction, with ids 80, 108, 102, 82, 87, 97, and 58 identified in Table 6.115. In all models used in the ensemble,

the optimal model used the parameter max iteration with value of 512, the random state of 1, the validation fraction with value of None, and the warm start set to True. In addition, all models belong to the gradient boosting type. The best validation score for this target, using the metric R^2, was 0.527220. Of the 122 target algorithm ran, only 28 ran successfully. Of the reaming 94 algorithms, 76 crashed, 6 exceeded the time limit, and 12 exceeded the memory limit.

Table 6.116: Models regarding the volume of class B vehicles with run time limit parameter of 10000 and the time task with value 100000. All models belong to the gradient boosting type.

model id	rank	ensemble weight	cost	duration	l2 regularization	learning rate	max leaf nodes
106	1	0.74	0.022377	2369.135194	3.02751e-05	0.049541	1799
53	2	0.26	0.022711	2243.950212	5.73607e-06	0.045536	1295

The ensemble model used two gradient boosting models for the class B prediction, with ids 106 and 53 identified in Table 6.116. The best validation score for this target, using the metric R^2, was 0.977623, very close to the maximum. Of the 143 target algorithm ran, only 25 ran successfully. Of the reaming 118 algorithms, 106 crashed, 5 exceeded the time limit, and 7 exceeded the memory limit.

Table 6.117: Models regarding the volume of class C vehicles with run time limit parameter of 10000 and the time task with value 100000. All models belong to the gradient boosting type.

model id	rank	ensemble weight	cost	duration	l2 regularization	learning rate	max leaf nodes
90	1	0.40	0.154256	705.494255	0.015437	0.6129321	57
71	2	0.14	0.154337	1006.974079	0.096630	0.0975348	144
92	3	0.16	0.154422	2053.564050	1.161892e-08	0.0234118	471
112	4	0.30	0.155777	695.853620	0.013526	0.7783744	57

The ensemble model used four gradient boosting models for the class B prediction, with ids 90, 71, 92 and 112 identified in Table 6.116. The best validation score for this target, using the metric R^2, was 0.845744 Of the 288 target algorithm ran, only 31 ran successfully. Of the reaming 257 algorithms, 240 crashed, 5 exceeded the time limit, and 12 exceeded the memory limit.

The ensemble model used four gradient boosting models for the total volume prediction, with ids 81, 86, 62, and 90 identified in Table 6.118. The best validation score for this target, using the metric R^2, was 0.981260, very close to the maximum and the higher value obtained. Of the 97 target algorithm ran, only 37 ran successfully. Of the reaming 60 algorithms, 51 crashed, 3 exceeded the time limit, and 6 exceeded the memory limit.

Table 6.118: Models regarding the total volume with run time limit parameter of 10000 and the time task with value 100000. All models belong to the gradient boosting type.

model id	rank	ensemble weight	cost	duration	l2 regularization	learning rate	max leaf nodes
81	1	0.36	0.018740	1905.303388	1.007982e-09	0.085688	1851
86	2	0.10	0.018886	1844.203236	4.785917e-07	0.107773	1913
62	3	0.24	0.019747	995.955914	0.000188	0.637655	786
90	4	0.30	0.020955	1282.844183	8.743260e-06	0.902720	1613

6.4 Microscopic simulation using OD matrix created

The results regarding the evaluation of the microscopic simulation using the OD matrix created by comparing the results of the real radars with the virtual radars positioned on the SUMO simulation are presented in this section.

The exact model created to get OD matrices manages to get a null error for some days of the week and some hours. However, it may be impossible to obtain an exact solution or a minimally distributed matrix. However, when the model error is not null, this error will contribute to the simulation error. It can sometimes be advantageous to have a matrix with a higher error if it implies that the OD matrix contains fewer null cells and lower values in each filled cell, as this makes the simulation more realistic.

Throughout implementing the OD matrix in the SUMO simulator, some difficulties were faced using the existing VCI network model provided. After running the simulation, some obvious discrepancies were noticeable, namely some excessive traffic clogging throughout the whole route. Upon further observation, this can be accredited to some questionable ceding of passage in most entryways and exits of the VCI; what was observed was the main route coming to a complete stop in order to allow incoming traffic to merge into the lane. After the analysis, the reason for this phenomenon was narrowed down to a problem in the network file; specifically, this file does not define the correct right-of-way priorities and, as such, entryways had priority over the main route.

As far as the simulation goes, the results measured on the virtual radars are as shown in table 6.119, where the values from the radar readings in the simulation are compared with those observed in VCI. The results shown are at 12 noon on a regular Monday. The values observed in the simulation were quite different from the observed values, both in absolute and relative values. The relative error varies between 55% and 75%, which is calculated by calculating the difference between the predicted/real value and the value observed in the virtual sensor, divided by the predicted value. The discrepancy in the values may be explain by hardware limitations not allowing the simulation to create as many instances as it would need, or the right-of-way anomaly described before. In short, a realistic simulation was not obtained.

Table 6.119: Comparison between observed and actual radar values regarding the simulation of a typical Monday at 12:00.

Sensor id	Lane direction	Hour	observed value	actual value	Absolute error	Relative error
8	C	12:00	1555	3753	2198	59%
8	D	12:00	1899	4268	2369	56%
9	C	12:00	1793	4268	2475	58%
9	D	12:00	1579	4037	2458	61%
10	C	12:00	456	1831	1375	75%
10	D	12:00	1340	2946	1606	55%
11	C	12:00	1065	2864	1799	63%
11	D	12:00	1223	3383	2160	64%
12	C	12:00	1490	3879	2389	62%
12	D	12:00	1436	3981	2545	64%
19	C	12:00	529	1464	935	64%
19	D	12:00	740	1754	1014	58%
20	C	12:00	447	1295	848	65%
20	D	12:00	535	1247	712	57%
21	C	12:00	503	1188	685	58%
21	D	12:00	544	1310	766	58%

6.5 Discussion of the results

In this section, the results obtained by all analyzed models are compared. Since it was impossible to perform sufficient tests on the neural networks due to the high number of hyper-parameters, the results are not in the comparison tables. Tables 6.120 and 6.121 compare the best multi-output models obtained regarding the prediction of all the classes and outputs, respectively. The best results were obtained in both tables by the Decision Tree model, with the best R^2 score and explained variance and the smallest errors. In both tables, the explained variance values were equal to the values obtained as the R^2 score metric evaluating the test data. As with the results of all multi-output predictions, the max error is not shown as the metric implemented by auto-Sklearn only supports single output.

The errors obtained by Linear Regression, Multi-Output Regressor, and Chain Regressor regarding the prediction of all the classes for sample A were approximately 13.7087, 1328.940, 36.4546, and 10.458 for the Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and the Median Absolute Error. For these three models, the R^2 score and the explained variance with the test and training data was approximately 0.2268. In contrast, the decision trees in which the R^2 score was much higher for the training and test data and the difference between the R^2 score with the training and test data was very noticeable.

The errors obtained by Linear Regression, Multi-Output Regressor, and Chain Regressor regarding the prediction of all the outputs for sample A were approximately 22.1237, 2316.771, 48.1328, and 17.019 for the Mean Absolute Error, Mean Squared Error, Root Mean Squared Er-

Table 6.120: Comparison between the best multi-output models obtained regarding the prediction of all the classes (sample A).

Type of model	Linear Regression	Multi-Output Regressor	Regressor Chain	Decision Tree Regressor
Data set	Mw	Mw	Mw	Md1
Train R^2 score	0.2268	0.2268	0.2268	1.0000
Test R^2 score	0.2268	0.2268	0.2268	0.6693
MAE	13.7087	13.7087	13.7087	2.2090
MSE	1328.9402	1328.9403	1328.9403	58.3435
RMSE	36.4546	36.4546	36.4546	7.6383
r^2	0.2268	0.2268	0.2268	0.6693
MedAE	10.4580	10.4581	10.4581	0.4000
test mean	26.4358	26.4358	26.4358	26.4358

Table 6.121: Comparison between the best multi-output models obtained regarding the prediction of all the outputs (sample A).

Type of model	Linear Regression	Multi-Output Regressor	Regressor Chain	Decision Tree Regressor
Data set	Mw	Mw	Mw	Md1
Train R^2 score	0.2692	0.2692	0.2692	1.0000
Test R^2 score	0.2690	0.2690	0.2690	0.7211
MAE	22.1237	22.1237	22.1237	3.2477
MSE	2316.7708	2316.7710	2316.7710	94.5064
RMSE	48.1328	48.1328	48.1328	9.7214
r^2	0.2690	0.2690	0.2690	0.7211
MedAE	17.0185	17.0186	17.0186	0.6667
test mean	44.0597	44.0597	44.0597	44.0597

ror, and the median absolute error. For these three models, the R^2 score and the explained variance with the test and training data was approximately 0.269. In contrast, in the decision trees, the R^2 score was much higher for the training and test data, and the difference between the R^2 score with the training and test data was very noticeable.

The results regarding the forecast of the volume of vehicles of class A with sample A are presented in Table 6.122. In this table, the smallest MAE, the MedAE and the Max error was obtained by the decision trees; however, the lowest MSE and Root MSE was obtained by the Random Forest. The R^2 score with test data was higher for Random Forest. In the Decision Tree algorithm, the value of the explained variance was different from the R^2 score with the test data.

The results regarding the forecast of the volume of vehicles of class B with sample A are presented in Table 6.123. On the one hand, the Random Forest obtained the lower MAE, MSE, and Root MSE and the higher explained variance. On the other hand, the lower Max error and MedAE were obtained by the light GBM. The highest R^2 score with the training data was obtained with the Decision Tree model because these models tend to over-fit the data. The value of the R^2 score with the test data was invariably equal to the explained variance.

Table 6.122: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class A (sample A).

Type of model	Linear Regression	Decision Tree	light GBM	XGB Regressor	Random Forest
Data set	Mw	Md1	Md1	Mw	Md1
Train R^2 score	0.2045	1.0000	0.7950	0.4986	0.9493
Test R^2 score	0.2062	0.3997	0.6042	0.4912	0.6444
MAE	0.8935	0.5116	0.5902	0.6750	0.5145
MSE	1.7829	1.3482	0.8889	1.1427	0.7987
Root MSE	1.3353	1.1611	0.9428	1.0690	0.8937
Max error	115.1919	69.0000	74.4686	80.4873	69.3167
r^2	0.2062	0.4001	0.6042	0.4912	0.6449
MedAE	0.6504	0.0000	0.3544	0.4218	0.2667
y test mean	0.9178	0.9178	0.9178	0.9178	0.9178

Table 6.123: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class B (sample A).

Type of model	Linear Regression	Decision Tree	light GBM	XGB Regressor	Random Forest
Data set	Mw	Md1	Md1	Mw	Md1
Train R^2 score	0.4762	1.0000	0.9933	0.9673	0.9980
Test R^2 score	0.4751	0.9781	0.9867	0.9668	0.9868
MAE	60.8327	8.2083	8.3833	12.3542	7.6106
MSE	6562.2532	273.8407	166.0109	415.0029	165.0363
Root MSE	81.0077	16.5481	12.8845	20.3716	12.8466
Max error	533.0754	594.0000	356.0174	515.8823	453.9250
r^2	0.4751	0.9781	0.9867	0.9668	0.9868
MedAE	46.8492	2.000	5.4179	7.7261	4.2000
y test mean	124.0935	124.0935	124.0935	124.0935	124.0935

Table 6.124: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class C (sample A).

Type of model	Linear Regression	Decision Tree	light GBM	XGB Regressor	Random Forest
Data set	Mw	Md1	Md1	Md1	Md1
Train R^2 score	0.2952	1.0000	0.9381	0.8444	0.9853
Test R^2 score	0.2949	0.8255	0.8828	0.8432	0.8979
MAE	3.8065	1.2796	1.3954	1.6121	1.1982
MSE	26.8256	6.6385	4.4592	5.9663	3.8853
Root MSE	5.1793	2.5765	2.1117	2.4426	1.9711
Max error	300.1313	156.0000	189.9283	301.4577	69.4167
r^2	0.2949	0.8255	0.8828	0.8432	0.8979
MedAE	2.9489	0.0000	0.8649	0.9980	0.6500
y test mean	4.9245	4.9245	4.9245	4.9245	4.9245

The results regarding the forecast of the volume of vehicles of class C with sample A are presented in Table 6.124. The Random Forest model obtained lower MAE, MSE, Root MSE, and Max error and higher R^2 score and r^2. On the contrary, the lower MedAE was obtained by the Decision Tree model.

Table 6.125: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class D (sample A).

Type of model	Linear Regression	Decision Tree	light GBM	XGB Regressor	Random Forest
Data set	Mw	Md1	Md1	Md1	Md1
Train R^2 score	0.1061	1.0000	0.7652	0.4376	0.9432
Test R^2 score	0.1057	0.3214	0.5556	0.4347	0.6057
MAE	0.7372	0.4031	0.4750	0.5459	0.4140
MSE	1.1317	0.8588	0.5624	0.7153	0.4990
Root MSE	1.0638	0.9267	0.7499	0.8458	0.7064
Max error	50.4368	32.0000	38.2792	50.6621	28.5333
r^2	0.1057	0.3219	0.5556	0.4347	0.6064
MedAE	0.5757	0.0000	0.2728	0.3298	0.2167
y test mean	0.65955	0.6596	0.6596	0.6596	0.6596

The results regarding the forecast of the volume of vehicles of class D with sample A are presented in Table 6.125. While the lower MAE and MedAE were obtained by the Decision Tree algorithm; the lower MSE, Root MSE, and Max error were obtained by the Random Forest. The latter also obtained the higher test R^2 score.

Table 6.126: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class 0 (sample A).

Type of model	Linear Regression	Decision Tree	light GBM	XGB Regressor	Random Forest
Data set	Mw	Md1	Md1	Md1	Md1
Train R^2 score	0.05203	1.0000	0.9713	0.7249	0.9899
Test R^2 score	0.05188	0.8817	0.9167	0.7155	0.9269
MAE	2.27341	0.5807	0.6880	1.1716	0.5736
MSE	52.7077	6.5756	4.6319	15.8174	4.0664
Root MSE	7.26001	2.5643	2.1522	3.9771	2.0165
Max error	770.382	866.0000	718.1866	749.8588	681.0333
r^2	0.05188	0.8817	0.9167	0.7155	0.9269
MedAE	1.26551	0.0000	0.1739	0.3733	0.1333
y test mean	1.58382	1.5838	1.5838	1.5838	1.5838

The results regarding the forecast of the volume of vehicles of class 0 with sample A are presented in Table 6.126. The Random Forest algorithm obtained the lower MAE, MSE, Root MSE, and Max error and the best R^2 score with the test data. Finally, the Decision Tree presents a 0 value for the median absolute error; therefore, this model displayed the lower MedAE.

Table 6.127: Comparison between the best ML single-output models obtained regarding the prediction of the total volume of vehicles (sample A).

Type of model	Linear Regression	Decision Tree	light GBM	XGB Regressor	Random Forest
Data set	Mw	Md1	Md1	Md1	Md1
Train R^2 score	0.4812	1.0000	0.9938	0.9714	0.9982
Test R^2 score	0.4802	0.9800	0.9880	0.9711	0.9880
MAE	64.1990	8.4724	8.5981	12.3259	7.8357
MSE	7255.9238	278.5584	167.2301	403.1959	166.9070
Root MSE	85.1817	16.6901	12.9317	20.0797	12.9193
Max error	733.2564	727.0000	707.4645	735.2572	682.3500
r^2	0.4802	0.9800	0.9880	0.9711	0.9880
MedAE	49.8209	3.0000	5.6638	7.7940	4.4250
y test mean	132.1792	132.1792	132.1792	132.1792	132.1792

The results regarding the forecast of the total volume of vehicles with sample A are presented in Table 6.127. While the Random Forest algorithm obtained the lower MAE, MSE, Root MSE, and Max error; the Decision Tree demonstrated the lower MedAE. Both light GBM and Random Forest had the highest R^2 score and r^2 with the test data of 0.9880. Then the results were compared with sample B.

Table 6.128: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class A (sample B).

Type of model	XGB Regressor	light GBM	Auto ML	Random Forest
Data set	Md1	Md1	Md1	Md1
Train R^2 score	0.5337	0.7876	0.6035	0.9264
Test R^2 score	0.5055	0.5226	0.5331	0.4854
Mean Absolute Error	0.6706	0.6542	0.6528	0.6786
Mean Squared Error	1.1449	1.1053	1.0689	1.1914
Root Mean Squared Error	1.0700	1.0513	1.0339	1.0915
Max error	104.7760	104.3790	80.9090	104.0667
explained variance	0.5055	0.5226	0.5331	0.4863
median absolute error	0.4152	0.3881	0.4073	0.4167
y test mean	0.9241	0.9241	0.9241	0.9241

The results regarding the forecast of the volume of vehicles of class A with sample B are presented in Table 6.128. The results obtained by the four models were quite close. The auto ML model achieved the best results on almost all metrics (Mean Absolute Error, Mean Squared Error, Root Mean Squared Error, and Max error).

The results regarding the forecast of the volume of vehicles of class B with sample B are presented in Table 6.129. Again, the results obtained by the four models were quite close. Regarding the prediction of the volume of class B vehicles, the light GBM model obtained the best results for all metrics.

Table 6.129: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class B (sample B).

Type of model	XGB Regressor	light GBM	Auto ML	Random Forest
Data set	Md1	Md1	Md1	Md1
Train R^2 score	0.9750	0.9927	0.9790	0.9972
Test R^2 score	0.9738	0.9832	0.9776	0.9805
Mean Absolute Error	10.9285	9.3136	10.2808	9.9728
Mean Squared Error	328.4855	210.5634	279.7756	244.9987
Root Mean Squared Error	18.1242	14.5108	16.7265	15.6524
Max error	482.6334	352.7842	443.7540	495.8000
explained variance	0.9738	0.9832	0.9776	0.98045
median absolute error	6.7766	5.8313	6.3664	6.2833
y test mean	123.3873	123.3873	123.4426	123.3873

Table 6.130: Comparison between the best ML single-output models obtained regarding the prediction of the volume of vehicles of class C (sample B).

Type of model	XGB Regressor	light GBM	Auto ML	Random Forest
Data set	Md1	Md1	Md1	Md1
Train R^2 score	0.8443	0.9341	0.8510	0.9781
Test R^2 score	0.8428	0.8536	0.8483	0.8469
Mean Absolute Error	1.6083	1.5455	1.5847	1.5808
Mean Squared Error	5.9388	5.5329	5.7338	5.7844
Root Mean Squared Error	2.4370	2.3522	2.3945	2.4051
Max error	195.2000	164.7019	134.8154	120.0167
explained variance	0.8428	0.8536	0.8483	0.8470
median absolute error	0.9947	0.9313	0.9769	0.9667
y test mean	4.8901	4.8901	4.8963	4.8901

The results regarding the forecast of the volume of vehicles of class C with sample B are presented in Table 6.130. The results obtained were close, however, the light GBM model obtained better results in most metrics.

Finally, the results related to the total volume of vehicles were compared. The results regarding the forecast of the total volume of vehicles with sample B are presented in Table 6.131. While Random Forest had the highest R^2 score with training data, the light GBM had the highest R^2 score with test data. All errors except Max error were lower for light GBM. Regarding the total volume, while for sample A, the best algorithm was Random Forest, which obtained an MAE of 7.8357 and a Root MSE of 12.9193; for sample B, the best algorithm was light GBM which had a Mean Absolute Error of 9.4910 and a Root Mean Squared Error of 14.3910. Therefore, the results obtained were worse with sample B.

In the linear regression, Multi-Output Regressor and Chain Regressor models, the MW data set with the *skyc1* field was the one that obtained the best results. In the Decision Tree, light

Table 6.131: Comparison between the best ML single-output models obtained regarding the prediction of the total volume of vehicles (sample B).

Type of model	XGB Regressor	light GBM	Auto ML	Random Forest
Data set	Md1	Md1	Md1	Md1
Train R^2 score	0.9813	0.9935	0.9855	0.9974
Test R^2 score	0.9794	0.9852	0.9826	0.9822
Mean Absolute Error	10.7115	9.4910	10.0918	10.2875
Mean Squared Error	288.3156	207.0997	243.2194	249.2838
Root Mean Squared Error	16.9799	14.3910	15.5955	15.7887
Max error	736.8875	695.6560	570.4057	695.9200
explained variance	0.9794	0.9852	0.9826	0.9822
median absolute error	6.8318	6.1082	6.4812	6.6200
y test mean	131.5573	131.5573	131.6283	131.5573

GBM, and Random Forest models, the Md1 data set obtained the best results for all targets. For the XGB Regressor model, the best results were obtained with both the MW and the Md1 data sets depending on the target. The best models depend both on the target and on the metrics that are most valued, since no model was always better for all targets. However, the models that obtained the best results used the mobility data.

The outliers in the volume forecasts of some classes sometimes arise from the lack of identification of the vehicle passing the sensor. Some entries have an expected total volume; however, most vehicles belong to class 0 because they have not been identified. The training time of models with all data is much higher than when using only one segment and one-way sensor data for all model families. As mentioned previously, another paper has applied Linear Regression (LR), Sequential Minimal Optimisation (SMO) Regression, and Regression Tree (M5Base classifier) on the same traffic flow data. The results of that paper [6] are compared in this section with the results obtained in this thesis.

Table 6.132: Comparison of the results of this thesis with another paper using the same data set regarding the prediction of the volume of vehicles class C (sample A).

Type of model	Simple LR	LR	SMO Regr	M5P Tree	LR	Decision Tree	light gbm	XGB Regr	RF
MAE	2.064	1.988	1.845	1.246	3.8065	1.2796	1.3954	1.6121	1.1982
RMSE	3.127	3.026	3.226	2.048	5.1793	2.5765	2.1117	2.4426	1.9711

Table 6.133: Comparison of the results of this thesis with another paper using the same data set regarding the prediction of the total volume (sample A).

Type of model	Simple LR	LR	SMO Regr	M5P Tree	LR	DT	light gbm	XGB Regr	RF
MAE	14.47	12.29	8.675	4.652	64.199	8.472	8.598	12.3259	7.8357
RMSE	22.73	20.24	27.4	8.211	85.182	16.690	12.932	20.0797	12.9193

On the left side of the Tables 6.132 and 6.133, in the first four columns, are the results of the regression models related to the forecast of the total volume and volume of vehicles class C implemented by [6]. The machine learning models in the previous paper performed better than all the machine and deep learning models implemented in this dissertation.

Table 6.134: Comparison of the results of this thesis with another paper using the same data set regarding the prediction of the volume of vehicles class C (sample B).

Type of model	Simple LR	LR	SMO Regr	M5P Tree	XGB Regr	light GBM	Auto ML	Random Forest
MAE	2.064	1.988	1.845	1.246	0.6706	0.6542	0.6528	1.5808
RMSE	3.127	3.026	3.226	2.048	1.0700	1.0513	1.0339	2.4051

Table 6.135: Comparison of the results of this thesis with another paper using the same data set regarding the prediction of the total volume (sample B).

Type of model	Simple LR	LR	SMO Regr	M5P Tree	XGB Regr	light GBM	Auto ML	Random Forest
MAE	14.47	12.29	8.675	4.652	10.7115	9.4910	10.0918	10.2875
RMSE	22.73	20.24	27.4	8.211	16.9799	14.3910	15.5955	15.7887

As in the tables in sample A, on the left side of Tables 6.134 and 6.135, in the first four columns, are the results of the regression models related to the forecast of the volume of vehicles class C and the total volume, respectively, implemented by [6]. In opposition to sample B, the results obtained in this thesis were significantly worse than those obtained previously in the paper [6]. However, to compare the models precisely, one would have to know the sample of tests used in this previous paper.

Table 6.136: Comparison of the results of the simple averaging algorithms studied in this thesis with another paper using the same data set regarding predicting the total volume (sample A)

Type of model	Simple LR	LR	SMO Regr	M5P Tree	XGB Regr (est: 10000) & DT	XGB Regr (est: 10000) & DT	XGB Regr (est: 1000) & DT
Data set					Mw	Md1	Md1
MAE	14.47	12.29	8.675	4.652	10.1153	8.5845	8.8088
RMSE	22.73	20.24	27.4	8.211	15.5516	14.5260	14.9612

Finally, the Ensemble models were compared with the models obtained previously. The three simple averaging models implemented are shown in Table 6.136. Of the implemented weighted averaging models, the ones that obtained the best results are shown in Table 6.137. Of the 11 models using Stacking tested, the best three are shown in Table 6.138. The three best models obtained similar results, and the best model used the Random Forest estimators with 20 estimators and decision trees and as the final estimator the light GBM with 65536 leaves. Of the three types of ensemble used, the ones that obtained the best results were the Stacking algorithms.

Table 6.137: Comparison of the results of the weighted averaging algorithms studied in this thesis with another paper using the same data set regarding predicting the total volume (sample A).

Type of model	Simple LR	LR	SMO Regression	M5P Tree	DT	Weighted averaging	Weighted averaging
XGB weight					0	0.4	0.3
DT weight					1	0.6	0.7
MAE	14.47	12.29	8.675	4.652	8.47	9.07	8.81
RMSE	22.73	20.24	27.4	8.211	16.69	14.93	14.96

Table 6.138: Comparison of the results of the Stacking algorithms studied in this thesis with another paper using the same data set regarding predicting the total volume (sample A).

Type of model	Simple LR	LR	SMO Regr	M5P Tree	RF (est: 10) & DT | f: XGB Regr (est: 1000)	RF (est: 20) & DT | f: light GBM (leaves: 65536)	RF (est: 10) & DT | f: light GBM (leaves: 131072)
MAE	14.47	12.29	8.675	4.652	7.8264	7.6504	7.8277
RMSE	22.73	20.24	27.4	8.211	13.4266	13.1145	13.4336

Regarding the total volume and using sample A, while the best ML algorithm was the Random Forest, which obtained a MAE of 7.8357 and a Root MSE of 12.9193; the best Ensemble algorithm obtained a MAE of 7.6504 and a Root MSE of 13.1145. Therefore, the results obtained by simple and weighted averaging and Stacking models with sample A were worse than those obtained previously in the other paper.

6.6 Summary

In sum, the models that obtained the best results used the mobility data without the meteorological information. Of the three types of ensemble used, the ones that obtained the best results were the Stacking algorithms. Although the ensemble generally improves the results, since this process requires a high computational power, it ends up having to use less powerful algorithms in the process. With less powerful algorithms being used, the results turn out to be close to those obtained with ML models. Finally, the results obtained by ensemble models were worse than those obtained previously in the other paper.

Chapter 7

Conclusions and Future Work

The problem addressed in this dissertation belongs to the traffic flow prediction type that corresponds to a governmental perspective and; more specifically corresponds to the network-based type. This type of problem uses loop detector sensors to collect data about traffic in the specific segment they are situated, such as the type and the number of vehicles passing through. This work used a case study regarding the mobility segment; specifically, the data used comes from a series of inductive-loop sensors throughout the highway VCI in Porto, Portugal.

In this dissertation, prediction models based not only on heterogeneous data related to mobility gathered from a series of inductive-loop sensors but also on meteorological data were studied and implemented. Predictive models used include machine and deep learning, auto-machine learning, and ensemble models. Models are trained and evaluated either in predicting values in all sensors of the urban network under analysis, using the complete data set with inputs from all sensors, or in predicting only one sensor in the urban network, using only data from that sensor. The results obtained show that meteorological data do not improve forecasts with the best models. After having different models, the best models were ensemble to obtain better results. With the information about the traffic, regardless of using the historical data set or the predicted data, it is possible to build a simulation of the entire VCI, extrapolating from the incomplete data given by different sensors on the road. Therefore, an exact method is proposed for creating OD matrices on which the entry and exit of vehicles on a highway are extrapolated using only information relating to some highway segments. Based on the OD matrices created, this work proposes a method to implement a digital twin of a highway by applying meta-modelling approaches to calibrate the a simulation model. Unfortunately, the micro-simulation in SUMO did not obtain the desired results given the high absolute and relative error.

The main contribution of this work is the ensemble of prediction models for urban mobility considering heterogeneous data. In short, the dissertation includes the study and implementation of prediction models applied to traffic conditions in urban networks, as well as an ensemble and empirical evaluation of models. Furthermore, an exact method is proposed for creating OD matrices using information related to some segments of a highway. Finally, this OD matrix created was used by a simulator to create a digital twin of a highway.

Future Work

Although an analysis was made for a random sensor in the VCI, it would be relevant in the future to analyze all sensors individually and aggregate those with a similar distribution in a single model. In the future, it would also be interesting to extrapolate the meteorological data, given that the sensors have very distant geographic locations. In addition, another data set with data on mobility related to Porto was found, and it is available on [62]. Therefore, it would be interesting to complement the data provided to the model with the data from this data set. Both the traffic forecasting process and the generation of OD matrices could be improved with the use of routes. In the future, it would be interesting to have a simulation that includes the smoke released by cars passing through the VCI. In addition, other more powerful simulators could be used to provide a more realistic digital twin.

In addition, the volume of class 0 vehicles corresponds to unidentified vehicles. Therefore, an analysis should be carried out in which the targets (except the total volume) are analyzed simultaneously and the error is evaluated as the sum of the predicted vehicle volumes of each class in comparison with the total volume and not class by class. Therefore, it would be considered that the forecast was sometimes even correct, but the vehicles were not identified.

An interesting next step would be to provide the traffic prediction of the network based on historical data with user-friendly metrics. This step involves researching the most meaningful metrics for an ordinary driver. The best known is travel time; however, the step solution could also create a scale that ranks traffic congestion levels, among other intuitive metrics. For example, it would be interesting to obtain the route travel times using the SUMO simulator or another more suitable one. Finally, given that SUMO supports customized vehicle types, it would be interesting to enhance the simulation by creating a more accurate digital twin of the VCI.

Appendix A

Additional data set information

This annex contains additional information regarding the position of the sensors used in the data set of the case study. Table A.1 contains a detailed description of the position of the sensors, including their id on the map in figure 4.1, the road they are on, their coordinates (latitude and longitude), and a description of the sensor's position in terms of regions it connects and direction.

Table A.1: Sensors location description.

Id	Road	Latitude	Longitude	Description
1	A20	4,107,731	-8,577,477	Carvalhos - Canelas 165
2	A20	4,108,205	-8,575,104	Carvalhos - Canelas 165
3	A20	4,109,138	-8,574,559	Canelas - Avintes(N222) 166
4	A20	4,112,471	-857,349	Avintes(N222) - Oliveira do Douro(A44) 167
5	A20	4,115,455	-858,269	Freixo(A43) - Mercado Abastecedor 169
6	A20	4,116,678	-8,585,455	Mercado Abastecedor - Antas 170
7	A20	4,117,043	-8,592,237	Antas - A3 171
8	A20	4,117,136	-8,599,315	A3 - Paranhos 172
9	A20	411,724	-8,606,449	Paranhos - Amial 173
10	A20	411,721	-8,632,161	N14/Circunvalação - Boavista 175
11	A28	4,115,733	-8,646,378	Campo Alegre - Boavista 192
12	A28	4,116,396	-8,648,117	Boavista - Rotunda AEP 193
13	A28	4,118,264	-8,661,078	Rotunda AEP - Senhora da Hora 194
14	A43	4,115,208	-8,574,777	Campanhã - Mercado Abastecedor 186
15	A43	4,115,827	-8,567,406	Mercado Abastecedor - Areosa 187
16	A43	4,115,524	-8,556,818	Areosa - Valbom 188
17	A43	4,115,524	-8,556,818	Areosa - Valbom 188
18	A43	411,352	-8,537,397	Valbom - Gondomar 189
19	A44	4,112,202	-8,615,691	Coimbrões - Av. Da República 182
20	A44	4,112,549	-860,651	Av. Da República - Gaia (centro) 183
21	A44	411,312	-8,589,719	Gaia (centro) - Pte. Freixo (A20) 184
22	N14	4,117,749	-862,232	Circunvalação (A20) - Areosa/Monte Burgos 163
23	N14	4,118,872	-8,625,984	Areosa/Monte Burgos - Leça do Balio 161
24	A1	4,112,975	-8,635,617	Canidelo - Afurada 179
25	A1	4,112,538	-8,635,809	A44/A29 - Canidelo 178
26	A1	4,111,113	-8,608,549	Santo Ovideo - A44/A29 177

Appendix B

Results

In this chapter, the experiments carried out that proved to be of lesser relevance will be presented.

B.1 Neural Networks

In this section, the results obtained with the neural networks with a single hidden layer and two hidden layers will be presented.

B.1.1 Single hidden layer

In the initial tests only one hidden layer was used. The results obtained with neural networks with only one layer will be presented in this subsection.

Table B.1: Neural Networks model results regarding the volume of vehicles of class A using the Md1 data set (sample B).

max iterations	5000	5000	5000
hidden layer size	2, 4, 5, 6, 7, 8, 10, 15, 19	3, 9, 11, 12, 13, 14, 16, 17	18, 20
Train R^2 score	0.00	0.00	0.00
Test R^2 score	0.00	0.00	0.00
Mean Absolute Error	1.05	1.05	1.05
Mean Squared Error	2.32	2.32	2.32
RMSE	1.52	1.52	1.52
Max error	114.07	114.08	114.06
explained variance (r^2)	0.00	0.00	0.00
MedAE	0.93	0.92	0.94
test mean	0.92	0.92	0.92

Table B.1 presents the neural Networks model results regarding class A's vehicle volume using only the mobility data set with a single field for the day of the week, specifically sample B. The tests in the table continuously used 5000 maximum iterations, and the hidden layer size varies from 2 to 20. Since many results are the same, the results have been aggregated. The results obtained

with the single hidden layer with a size value of 2, 4 through 8, 10, 15, and 19 were the same when rounded to two decimal places, as were the results obtained with the single hidden layer with a size value of 3, 9, 11 through 14, and 16 to 17. The R^2 score (with training and test data) and the explained variance were always 0 with the neural networks with a single hidden layer varying from size 2 to 20. Furthermore, the mean absolute error was invariably 1.05, the mean squared error was always 2.32, and the root mean squared errors were 1.52 constantly. The only metrics that did not always get the same value were the max and median absolute errors. The results obtained with only one hidden layer, several sizes and 5000 maximum iterations were close and unpromising, as they obtained worse results than the linear regression model and decision trees with the same data set.

Table B.2: Neural Networks model results regarding the volume of class A vehicles using the MW data set (sample A)

max iterations	10000	10000	10000	100000	100000	100000	100000
hidden layer size	2	3, 5	4	10, 17, 20	11, 16, 18, 19	15	21
Train R^2 score	0.00	0.00	0.00	0.00	0.00	0.21	0.17
Test R^2 score	0.00	0.00	0.00	0.00	0.00	0.21	0.17
MAE	1.04	1.04	1.04	1.04	1.04	0.86	0.84
MSE	2.25	2.25	2.25	2.25	2.25	1.78	1.87
Root MSE	1.50	1.50	1.50	1.50	1.50	1.33	1.37
Max error	113.08	113.07	113.08	113.09	113.08	113.76	113.76
r^2	0.00	0.00	0.00	0.00	0.00	0.21	0.20
MedAE	0.92	0.93	0.92	0.91	0.92	0.54	0.47
test mean	0.92	0.92	0.92	0.92	0.92	0.92	0.92

Given the bad results obtained with only 5000 maximum iterations, the number of iterations was changed to the doubled (10000) and further increased to 100000 iterations. These neural networks model results regarding the volume of class A vehicles using the mobility and the weather data set, including the sky Level 1 Coverage are presented in table B.2. With 10000 iterations and a network size of 2 to 5, the results continued to be worse than linear regression. Increasing the maximum number of iterations to 100000, regardless of the network size, obtained better results than all previous neural network models. The neural networks that obtained the best results were those with the hidden network size of 15 and 21. The results obtained by the linear regression model were worse than those obtained by the two aforementioned neural networks.

The results of the neural networks model regarding the total volume forecasts can be found in the Tables B.3, B.4, and B.5.

With 3000 iterations, the results of the R^2 score with the training data were similar to the R^2 score with the test data, demonstrating that all the neural networks tested did not over-fit the data. The best results using 3000 iterations were obtained with a hidden layer of size 12. Even the best model of neural networks with 3000 iterations did not obtain better results than the linear regression for the same data set. Following, the maximum number of iterations has been increased

Table B.3: Results of the Neural Networks model with single output regarding the total volume of vehicles using the Md1 data set (sample B). The max iterations (max iter) parameter is always 3000.

max iter	3000	3000	3000	3000	3000	3000	3000
hidden layer	2	3	4	5	6	7	8
train R^2 score	0.497	0.498	0.495	0.496	0.499	0.501	0.496
test R^2 score	0.497	0.498	0.496	0.496	0.499	0.501	0.496
MAE	60.347	61.859	60.360	60.350	60.492	61.142	60.547
MSE	7045.972	7034.725	7071.315	7065.812	7027.027	6993.975	7065.742
RMSE	83.940	83.873	84.091	84.058	83.827	83.630	84.058
Max error	657.728	643.232	660.305	659.620	657.299	648.152	660.790
r^2	0.501	0.501	0.500	0.500	0.501	0.501	0.499
MedAE	41.580	45.008	41.485	41.479	42.055	43.719	41.980
y mean	131.557	131.557	131.557	131.557	131.557	131.557	131.557

Table B.4: Results of the Neural Networks model with single output regarding the total volume of vehicles using the Md1 data set (sample B).

max iterations	3000	3000	3000	3000	3000	3000
hidden layer size	9	10	11	12	13	14
Train R^2 score	0.497	0.574	0.501	0.576	0.495	0.498
Test R^2 score	0.497	0.575	0.501	0.577	0.496	0.499
MAE	60.389	56.240	60.628	55.646	62.273	60.511
MSE	7052.563	5961.577	6998.389	5933.248	7067.094	7028.466
RMSE	83.980	77.211	83.656	77.028	84.066	83.836
Max error	658.839	630.863	652.672	637.051	641.363	657.480
r^2	0.500	0.576	0.501	0.577	0.500	0.501
MedAE	41.682	39.891	42.507	38.802	45.783	42.104
test mean	131.557	131.557	131.557	131.557	131.557	131.557

from 3000 to 5000 in order to improve the results obtained for forecasting the total volume, and the data set used was the third.

The results of networks of the same size compared to the previous networks with another data set and a number of smaller iterations were both better and worse. In this way, the results obtained with neural networks with a hidden layer still do not stand out. As the results were not promising, only class A and total volume targets were predicted with a hidden layer.

B.1.2 Two hidden layers

Since the amount of data is very high, the size of the network has been increased for better results. This subsection presents the results of models with two layers. Mobility data has 15 input features,

Table B.5: Neural Networks model results with single output regarding the total volume of vehicles using the MW data set (sample A).

max iterations	5000	5000	5000	5000	5000	5000
hidden layer	1	2	3	4	5	6
Train R^2 score	0.500	0.500	0.500	0.495	0.500	0.582
Test R^2 score	0.499	0.499	0.499	0.494	0.499	0.581
MAE	61.205	61.009	61.035	60.549	60.820	56.973
MSE	6995.479	6995.207	6994.655	7058.364	6998.185	5845.195
Root MSE	83.639	83.637	83.634	84.014	83.655	76.454
Max error	739.830	741.934	741.435	750.438	743.529	692.613
explained variance	0.499	0.499	0.499	0.498	0.499	0.581
MedAE	43.758	43.275	43.354	41.938	42.804	40.749
test mean	132.179	132.179	132.179	132.179	132.179	132.179

so tests were done with a hidden layer 1 of size ranging from 3 to 14 and a hidden layer 2 of size ranging from 3 to hidden layer 1 size. In all tests, the mobility data set and sample B were used.

Table B.6: Neural Networks model with two hidden layers results regarding the volume of vehicles of class A using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

max iter	5000	5000	5000	5000	5000	5000	5000
hidden layers	(3, 2); (4, 3); (6, 5); (7, 4/6); (8, 2); (9, 5)	(4, 2); (5, 4); (6, 2); (7, 3); (7, 5); (8, 4/7)	(5, 2)	(5, 3); (10, 2)	(6, 3); (8, 3/6); (9, 2); (9, 4/8)	(6, 4)	(7, 2)
Train R^2 score	0.00	0.00	0.20	0.00	0.00	0.18	0.00
Test R^2 score	0.00	0.00	0.20	0.00	0.00	0.18	0.00
MAE	1.05	1.05	0.89	1.05	1.05	0.95	1.05
MSE	2.32	2.32	1.86	2.32	2.32	1.90	2.32
Root MSE	1.52	1.52	1.36	1.52	1.52	1.38	1.52
Max error	114.08	114.07	114.82	114.08	114.09	114.82	114.06
r^2	0.00	0.00	0.20	0.00	0.00	0.19	0.00
MedAE	0.92	0.93	0.61	0.92	0.91	0.73	0.94
test mean	0.92	0.92	0.92	0.92	0.92	0.92	0.92

Of the models presented in Table B.6, the one with the best results was the one with hidden layers of sizes 5 and 2. This last network obtained an R^2 score higher than the linear regression. With the network of size 6 in the first layer and 4 in the second layer, the explained variance was different from the R^2 score with the test data.

Table B.7 presents the neural networks model with two hidden layers results regarding the volume of vehicles of class A using only the mobility data set with a single field for the day of the week, specifically, sample B. Although the size network (9, 3) did not had the higher R^2 score, it

Table B.7: Neural Networks model with two hidden layers results regarding the volume of vehicles of class A using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

max iter	5000	5000	5000	5000	5000	5000
hidden layers	(8, 5)	(9, 3)	(9, 6/7)	(10, 3)	(10, 4/5)	(10, 6)
Train R^2 score	0.00	0.18	0.00	0.20	0.00	0.00
Test R^2 score	0.00	0.18	0.00	0.19	0.00	0.00
MAE	1.05	0.85	1.05	0.91	1.05	1.05
MSE	2.32	1.90	2.32	1.87	2.32	2.32
Root MSE	1.52	1.38	1.52	1.37	1.52	1.52
Max error	114.05	114.80	114.07	114.81	114.08	114.09
r^2	0.00	0.19	0.00	0.20	0.00	0.00
MedAE	0.95	0.52	0.93	0.67	0.92	0.91
test mean	0.92	0.92	0.92	0.92	0.92	0.92

had lower mean and median errors. This neural network had an error and a lower R^2 score than the linear regression model trained with the same mobility data and tested with the same sample B.

Table B.8: Neural Networks model with two hidden layers results regarding the volume of vehicles of class A using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(10, 7/8/9); (11, 3/4); (11, 8/9)	(11, 2); (11, 6)	(11, 5/7); (12, 2/5/6)	(11, 10); (12, 7); (13, 4)	(12, 3)	(12, 4)	(12, 8); (14, 2)
max iter	5000	5000	5000	5000	5000	5000	5000
Train R^2 score	0.00	0.00	0.00	0.00	0.00	0.18	0.00
Test R^2 score	0.00	0.00	0.00	0.00	0.00	0.18	0.00
MAE	1.05	1.05	1.05	1.05	1.05	0.85	1.05
MSE	2.32	2.32	2.32	2.32	2.32	1.91	2.32
Root MSE	1.52	1.52	1.52	1.52	1.52	1.38	1.52
Max error	114.08	114.09	114.07	114.08	114.06	115.56	114.09
r^2	0.00	0.00	0.00	0.00	0.00	0.19	0.00
MedAE	0.92	0.91	0.93	0.92	0.94	0.51	0.91
test mean	0.92	0.92	0.92	0.92	0.92	0.92	0.92

Excluding the size network (12, 4), all results in tables B.8 and B.9 have an R^2 score and explained variance of 0 and a mean absolute error of 1.05, an MSE of 2.32 and a Root MSE of 1.52. The error obtained with these neural networks was worse than that obtained previously with linear regression. After predicting the volume of class A vehicles, the target that was then predicted was the volume of class B vehicles. The prediction of the volume of class B vehicles also used 5000 maximum iterations and the results are presented in Tables B.10, B.11, and B.12.

Table B.9: Neural Networks model with two hidden layers results regarding the volume of vehicles of class A using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(12, 9/10); (12, 11); (13, 3/6); (13, 9/12); (14, 3/12)	(13, 2/5); (13, 11)	(13, 7/8); (13, 10); (14, 4/5)	(14, 6/7); (14, 8/9); (14, 10)	(14, 11)	(14, 13)
max iter	5000	5000	5000	5000	5000	5000
Train R^2 score	0.00	0.00	0.00	0.00	0.00	0.00
Test R^2 score	0.00	0.00	0.00	0.00	0.00	0.00
MAE	1.05	1.05	1.05	1.05	1.05	1.05
MSE	2.32	2.32	2.32	2.32	2.32	2.32
Root MSE	1.52	1.52	1.52	1.52	1.52	1.52
Max error	114.07	114.06	114.08	114.08	114.09	114.09
r^2	0.00	0.00	0.00	0.00	0.00	0.00
MedAE	0.93	0.94	0.92	0.92	0.91	0.91
test mean	0.92	0.92	0.92	0.92	0.92	0.92

Table B.10: Neural Networks model with two hidden layers results regarding the volume of vehicles of class B using Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(3, 2)	(4, 2)	(4, 3)	(5, 2)	(5, 3)	(5, 4)	(6, 2)
max iter	5000	5000	5000	5000	5000	5000	5000
Train R^2 score	0.00	0.49	0.50	0.49	0.49	0.50	0.00
Test R^2 score	0.00	0.49	0.50	0.49	0.49	0.50	0.00
MAE	91.67	58.54	57.52	58.44	58.56	57.61	91.68
MSE	12534.01	6361.11	6313.02	6336.17	6365.99	6302.18	12534.01
Root MSE	111.96	79.76	79.45	79.60	79.79	79.39	111.96
Max error	583.64	543.44	558.70	552.93	543.11	549.37	583.61
r^2	0.00	0.49	0.50	0.50	0.49	0.50	0.00
MedAE	87.36	42.46	40.82	42.63	42.43	40.17	87.39
test mean	123.39	123.39	123.39	123.39	123.39	123.39	123.39

In table B.10, the neural networks with sizes (3, 2), (6, 2) had the worst results, with an R^2 score of 0 and a median absolute error greater than 90. In the same table, the network that obtained the best results was the one with size (4, 3).

All the results presented in Table B.10 were worse than the best result obtained in Table B.11 with the neural network with size (4, 3).

Table B.12 shows the results of the neural networks of size (7,6) to (8,7). The best result was obtained by the largest neural network of size 8 in the first hidden layer and 7 in the second hidden layer.

Table B.11: Neural Networks model with two hidden layers results regarding the volume of vehicles of class B using Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(6, 3)	(6, 4)	(6, 5)	(7, 2)	(7, 3)	(7, 4)	(7, 5)
max iter	5000	5000	5000	5000	5000	5000	5000
Train R^2 score	0.50	0.49	0.49	0.00	0.49	0.49	0.50
Test R^2 score	0.50	0.49	0.49	0.00	0.49	0.49	0.50
MAE	57.69	59.09	59.13	91.70	58.60	58.42	57.87
MSE	6313.42	6401.74	6383.96	12534.02	6369.40	6336.38	6315.00
Root MSE	79.46	80.01	79.90	111.96	79.81	79.60	79.47
Max error	560.00	540.40	549.76	583.54	543.82	538.73	556.60
r^2	0.50	0.49	0.50	0.00	0.49	0.50	0.50
MedAE	41.24	43.45	43.72	87.46	42.52	42.20	41.47
test mean	123.39	123.39	123.39	123.39	123.39	123.39	123.39

Table B.12: Neural Networks model with two hidden layers results regarding the volume of vehicles of class B using Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(7, 6)	(8, 2)	(8, 3)	(8, 4)	(8, 5)	(8, 6)	(8, 7)
max iter	5000	5000	5000	5000	5000	5000	5000
Train R^2 score	0.00	0.49	0.50	0.49	0.50	0.49	0.57
Test R^2 score	0.00	0.49	0.50	0.49	0.50	0.49	0.57
MAE	91.64	58.05	57.64	57.45	57.75	57.05	53.35
MSE	12534.05	6341.14	6308.63	6360.45	6298.97	6364.56	5418.61
Root MSE	111.96	79.63	79.43	79.75	79.37	79.78	73.61
Max error	583.81	546.49	546.56	553.99	554.96	566.71	494.26
r^2	0.00	0.49	0.50	0.49	0.50	0.50	0.58
MedAE	87.19	41.44	40.61	39.98	41.32	39.35	36.69
test mean	123.39	123.39	123.39	123.39	123.39	123.39	123.39

The largest network tested, with a size of 10 in the first hidden layer and 2 in the second hidden layer, was the one with the lowest mean absolute error of 51.72. Since the last result was the best, it is possible that the results would improve with the increase in the network. Due to lack of time, all tests were not run-up to the size network (14, 13) as in the prediction of the class A target. Then the same tests were done for the vehicles of class C.

Table B.14 shows the results of networks of size (3, 2) to (6, 2). The mean absolute error of networks, except size 5 and 2 networks in hidden layers 1 and 2, respectively, was 4.68. Contrary to the other networks, the network (5, 2) obtained a non-zero R^2 score. Then, the size of the network was sequentially increased up to (7, 5). The results are presented in Table B.15.

While the lowest Mean Absolute Error in Table B.14 was 3.52, the lowest Mean Absolute Error in Table B.15 was 3.50. Therefore, the best network analyse for class C was of size (6, 3). If larger neural networks had been tested, it is possible that better results would have been obtained. No tests were carried out for classes D and 0 due to lack of time. Tables B.16, B.17, B.18, and

Table B.13: Neural Networks model with three hidden layers results with single output regarding the volume of vehicles of class B using the Md1 data set (sample A). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(9, 2)	(9, 3)	(9, 4)	(9, 5)	(9, 6)	(9, 7)	(9, 8)	(10, 2)
max iter	5000	5000	5000	5000	5000	5000	5000	5000
train R^2 score	0.49	0.49	0.57	0.49	0.49	0.49	0.49	0.60
test R^2 score	0.49	0.49	0.57	0.49	0.49	0.49	0.49	0.60
MAE	57.55	58.91	53.51	57.34	58.66	59.47	58.31	51.72
MSE	6343.6	6360.68	5395.99	6332.81	6349.69	6409.1	6333.66	5044.77
RMSE	79.65	79.75	73.46	79.58	79.68	80.06	79.58	71.03
Max error	551.54	549.44	528.23	560.91	552.58	547.68	553.35	534.07
r^2	0.49	0.50	0.57	0.50	0.50	0.50	0.50	0.60
MedAE	40.27	43.30	37.34	40.14	43.11	44.37	42.41	34.45
y mean	123.39	123.39	123.39	123.39	123.39	123.39	123.39	123.39

Table B.14: Neural Networks model with two hidden layers results regarding the volume of vehicles of class C using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(3, 2)	(4, 2)	(4, 3)	(5, 2)	(5, 3)	(5, 4)	(6, 2)
max iter	5000	5000	5000	5000	5000	5000	5000
Train R^2 score	0.00	0.00	0.00	0.32	0.00	0.00	0.00
Test R^2 score	0.00	0.00	0.00	0.32	0.00	0.00	0.00
Mean Absolute Error	4.68	4.68	4.68	3.52	4.68	4.68	4.68
Mean Squared Error	37.78	37.78	37.79	25.78	37.79	37.78	37.79
Root MSE	6.15	6.15	6.15	5.08	6.15	6.15	6.15
Max error	193.09	193.11	193.08	193.76	193.08	193.09	193.08
r^2	0.00	0.00	0.00	0.32	0.00	0.00	0.00
MedAE	3.91	3.89	3.92	2.31	3.92	3.91	3.92
test mean	4.89	4.89	4.89	4.89	4.89	4.89	4.89

B.19 present the results concerning the forecast of the total volume of vehicles.

Table B.16 shows the results of networks of size (3, 2) to (6, 2), regarding the total volume of vehicles. Of all the networks present in the table, the network of size (4, 3) was the one that obtained the best results, with the highest R^2 score and the smallest errors. The error obtained with this neural network was lower than the obtained with the linear Regression model.

All the results presented in tables B.17, B.18, and B.19 were worse than those obtained by the size network (4, 3). The greatest Mean Absolute Error regarding the prediction of the total volume was obtained with the neural network (7,6). In opposition to the Mean Absolute Error close to 60 obtained with the other neural networks, the networks of size (3,2) and (7,6) obtained an error close to 100.

Table B.15: Neural Networks model with two hidden layers results regarding the volume of vehicles of class C using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(6, 3)	(6, 4)	(6, 5)	(7, 2)	(7, 3)	(7, 4)	(7, 5)
max iter	5000	5000	5000	5000	5000	5000	5000
Train R^2 score	0.31	0.32	0.32	0.00	0.32	0.32	0.31
Test R^2 score	0.31	0.32	0.32	0.00	0.32	0.32	0.31
Mean Absolute Error	3.50	3.67	3.60	4.68	3.66	3.66	3.52
Mean Squared Error	25.98	25.83	25.62	37.78	25.76	25.81	26.11
Root MSE	5.10	5.08	5.06	6.15	5.08	5.08	5.11
Max error	194.04	192.78	193.31	193.11	192.96	192.82	193.85
r^2	0.32	0.32	0.32	0.00	0.32	0.32	0.32
MedAE	2.25	2.56	2.45	3.89	2.59	2.56	2.17
test mean	4.89	4.89	4.89	4.89	4.89	4.89	4.89

Table B.16: Neural Networks model with two hidden layers results regarding the total volume of vehicles using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

hidden layers	(3,2)	(4,2)	(4,3)	(5,2)	(5,3)	(5,4)	(6,2)
max iter	3000	3000	3000	3000	3000	3000	3000
train R^2 score	0.00	0.50	0.56	0.50	0.50	0.50	0.50
test R^2 score	0.00	0.50	0.56	0.50	0.50	0.50	0.50
MAE	97.03	62.02	57.48	61.59	61.54	61.18	61.15
MSE	14016.93	7065.11	6161.23	6993.90	7016.58	6971.40	6996.27
RMSE	118.39	84.05	78.49	83.63	83.77	83.49	83.64
Max error	745.46	639.01	604.92	642.73	644.12	645.48	647.31
r^2	0.00	0.50	0.59	0.50	0.50	0.50	0.50
MedAE	92.54	45.09	39.87	45.01	44.41	43.33	43.70
test mean	131.56	131.56	131.56	131.56	131.56	131.56	131.56

B.2 XGB Regressor

This section presents the results of the XGB Regressor models lesser relevance. The results obtained by the XGB Regressor models regarding the volume of class A vehicles and using only the mobility data set with a single field for the day of the week and with a variable number of estimators are presented in Tables B.20 and B.21. Specifically, Table B.20 presents the results obtained by the XGB Regressor models by varying the parameter of the number of estimators from 100 to 250, and Table B.21 presents the results obtained by the same models by varying the number of estimators from 300 to 10000. Since the R^2 score metric values with the test sample are the same as those obtained by the explained variance, the row with this metric does not appear in the tables.

In Table B.20, with the increase in the estimators, the R^2 score and the explained variance obtained higher values, and the errors were consistently lower. Since the variation in the number of estimators in this table was low, the improvement was also negligible. As the number of estimators

Table B.17: Neural Networks model with two hidden layers results regarding the total volume of vehicles using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

max iter	3000	3000	3000	3000	3000	3000	3000
hidden layer 1 size	6	6	6	7	7	7	7
hidden layer 2 size	3	4	5	2	3	4	5
Train R^2 score	0.50	0.50	0.50	0.50	0.48	0.50	0.50
Test R^2 score	0.50	0.50	0.50	0.50	0.48	0.50	0.50
MAE	61.40	62.01	61.02	61.29	60.39	61.39	60.86
MSE	6974.02	7048.59	6954.70	6998.77	7256.72	6980.57	6963.18
RMSE	83.51	83.96	83.39	83.66	85.19	83.55	83.45
Max error	641.81	641.91	640.52	648.14	668.36	641.55	645.99
r^2	0.50	0.50	0.50	0.50	0.50	0.50	0.50
MedAE	44.57	45.27	43.91	44.03	39.87	44.11	43.62
test mean	131.56	131.56	131.56	131.56	131.56	131.56	131.56

Table B.18: Neural Networks model with two hidden layers results regarding the total volume of vehicles using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

max iter	3000	3000	3000	3000	3000	3000	3000
hidden layer 1 size	7	8	8	8	8	8	8
hidden layer 2 size	6	2	3	4	5	6	7
Train R^2 score	0.00	0.50	0.50	0.50	0.50	0.50	0.49
Test R^2 score	0.00	0.50	0.50	0.50	0.50	0.50	0.49
MAE	97.08	61.67	60.68	60.67	60.14	60.46	63.20
MSE	14016.98	7019.61	6953.23	7001.20	7045.52	7017.95	7155.54
RMSE	118.39	83.78	83.39	83.67	83.94	83.77	84.59
Max error	745.23	644.70	649.52	654.22	662.57	655.22	629.91
r^2	0.00	0.50	0.50	0.50	0.50	0.50	0.50
MedAE	92.77	44.74	42.74	42.68	41.52	41.94	45.92
test mean	131.56	131.56	131.56	131.56	131.56	131.56	131.56

increased, the difference between the R^2 score using the training and test data increased, ranging from 0.0037 to 0.0046, with the smallest and higher number of estimators, respectively.

Table B.21 shows that the increase in the number of estimators causes an increase in the explained variance and the R^2 score with both test and training data. Except for the maximum error, all errors were also smaller as the number of estimators increased. As in the previous table, with the increase in the number of estimators the difference between the R^2 score with the training and test data increased, ranging from 0.0048 to 0.0281, with the smallest and largest number of estimators present in the table respectively.

Table B.19: Neural Networks model with two hidden layers results regarding the total volume of vehicles using the Md1 data set (sample B). The size of hidden layers is displayed as (layer 1 size, layer 2 size).

max iter	3000	3000	3000	3000	3000	3000	3000
hidden layer 1	9	9	9	9	9	9	9
hidden layer 2	2	3	4	5	6	7	8
Train R^2 score	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Test R^2 score	0.50	0.50	0.50	0.50	0.50	0.50	0.50
MAE	60.52	60.07	60.67	60.30	61.45	62.19	61.19
MSE	7004.49	7059.85	6939.71	6989.70	6982.95	7040.24	6974.79
Root MSE	83.69	84.02	83.30	83.60	83.56	83.91	83.52
Max error	653.32	661.71	647.59	656.43	641.57	638.80	643.78
r^2	0.50	0.50	0.50	0.50	0.50	0.50	0.50
MedAE	42.18	41.40	43.23	42.23	44.74	46.02	44.26
test mean	131.56	131.56	131.56	131.56	131.56	131.56	131.56

Table B.20: Results of the XGB Regressor model with single output regarding the volume of vehicles of class A using the Md1 data set (sample B).

n estimators	100	120	140	150	200	250
Train R^2 score	0.4472	0.4519	0.4551	0.4561	0.4623	0.4661
Test R^2 score	0.4436	0.4481	0.4512	0.4522	0.4580	0.4615
MAE	0.7086	0.7064	0.7045	0.7037	0.7001	0.6975
MSE	1.2883	1.2779	1.2706	1.2683	1.2548	1.2468
Root MSE	1.1350	1.1305	1.1272	1.1262	1.1202	1.1166
Max error	107.0347	106.2047	105.8958	105.8742	104.2340	104.2597
MedAE	0.4463	0.4443	0.4433	0.4423	0.4398	0.4378
test mean	0.9241	0.9241	0.9241	0.9241	0.9241	0.9241

Table B.21: Results of the XGB Regressor model with single output regarding the volume of vehicles of class A using the Md1 data set (sample B).

n estimators	300	350	400	450	600	10000
Train R^2 score	0.4697	0.4727	0.4751	0.4773	0.4823	0.5337
Test R^2 score	0.4649	0.4677	0.4699	0.4717	0.4760	0.5055
MAE	0.6952	0.6934	0.6919	0.6907	0.6877	0.6706
MSE	1.2389	1.2324	1.2274	1.2233	1.2131	1.1449
Root MSE	1.1130	1.1101	1.1079	1.1060	1.1014	1.0700
Max error	104.2351	104.2418	104.2060	104.2150	104.2787	104.7760
MedAE	0.4361	0.4356	0.4342	0.4332	0.4306	0.4152
test mean	0.9241	0.9241	0.9241	0.9241	0.9241	0.9241

Then it was analyzed how the metrics vary with the increase in the number of estimators for the "Volume class B" target. The results are presented in Table B.22 and B.23. While the number of estimators in Table B.23 ranges from 100 to 10000, the number of estimators in Table B.22 ranges from 600 to 5000. The error obtained with just 100 estimators was much smaller than that

Table B.22: Results of the XGB Regressor model with single output regarding the volume of vehicles of class B using the Md1 data set (sample B).

n estimators	100	110	120	130	10000
Train R^2 score	0.9519	0.9528	0.9535	0.9542	0.9750
Test R^2 score	0.9519	0.9529	0.9536	0.9542	0.9738
Mean Absolute Error	15.0948	14.9244	14.7969	14.6766	10.9285
Mean Squared Error	602.3149	590.8299	581.7760	573.9292	328.4855
Root Mean Squared Error	24.5421	24.3070	24.1200	23.9568	18.1242
Max error	550.7170	549.6469	547.2021	547.5559	482.6334
explained variance	0.9519	0.9529	0.9536	0.9542	0.9738
median absolute error	9.3399	9.2300	9.1709	9.0981	6.7766
test mean	123.3873	123.3873	123.3873	123.3873	123.3873

Table B.23: XGB Regressor model results regarding the total volume of vehicles using the Md1 data set (sample B).

n estimators	600	1000	3000	4000	5000
Train R^2 score	0.9696	0.9717	0.9752	0.9761	0.9767
Test R^2 score	0.9694	0.9714	0.9747	0.9755	0.9760
Mean Absolute Error	12.7326	12.2903	11.5770	11.4325	11.3319
Mean Squared Error	428.2190	400.3747	354.1222	343.8686	336.3165
Root Mean Squared Error	20.6935	20.0094	18.8181	18.5437	18.3389
Max error	736.8936	736.4468	735.9923	735.4429	736.0355
explained variance	0.9694	0.9714	0.9747	0.9755	0.9760
median absolute error	8.0453	7.7622	7.3177	7.2321	7.1780
test mean	131.5573	131.5573	131.5573	131.5573	131.5573

obtained with linear regression models with the same data set. The variation from 100 to 10000 translates into a change in a mean absolute error from 15.0948 to 10.9285. The higher the number of estimators, the less the results improve as the number of estimators increases. This conclusion can be drawn from the fact that the increase from 600 estimators to 5000 translates into a difference in the Mean Absolute Error from 12.7326 to 11.3319. In Table B.23, the results obtained by the R^2 score with the training and test data were very close, and there were even situations in which the results with the test data set were better. In the results of Table B.22, the R^2 score with the training data was superior to the R^2 with the test data, suggesting that over-fit was starting to happen.

Finally, in Table B.24, the R^2 score with the test data was equal to the explained variance and was constantly lower than the R^2 score with the training data. The minimum Mean Absolute Error obtained by the XGB Regressor model was 10.7115.

Table B.24: XGB Regressor model results regarding the total volume of vehicles using the Md1 data set (sample B). The number of estimators of the model varies between 7000 and 30000.

n estimators	7000	8000	9000	20000	30000
Train R^2 score	0.9776	0.9780	0.9783	0.9803	0.9813
Test R^2 score	0.9768	0.9770	0.9773	0.9788	0.9794
Mean Absolute Error	11.1894	11.1352	11.0893	10.8225	10.7115
Mean Squared Error	325.8058	321.8385	318.3783	297.3514	288.3156
Root MSE	18.0501	17.9399	17.8432	17.2439	16.9799
Max error	737.6897	737.1721	738.5753	737.1618	736.8875
explained variance	0.9768	0.9770	0.9773	0.9788	0.9794
MedAE	7.0946	7.0624	7.0354	6.8893	6.8318
test mean	131.5573	131.5573	131.5573	131.5573	131.5573

Appendix C

Methodology to create the Sumo simulation

The process described in this chapter is a generic process to design simulations using OD matrices created by the exact model created. This methodology assumes the existence of a file with the network in the SUMO format. If the network does not exist, it must be created. In this project, the *net.net.xml* file represents Oporto's geographic zone containing the VCI network configuration. After the comprehension of the network, the first step to build a simulation consisted on creating a traffic assignment zone file (*taz.xml*) containing small areas where the traffic can start or end on the network. All files with a Taz suffix store squares on the map where traffic can start or end, which correspond to the entrances and exits of the VCI in this case study. The SUMO Netedit editor can be used tp drew this zones manually, copying where the real exits and entrances are located (in both directions) in the network representation. The plain-xml (*InnerOuterRingTaz.xml*) file encodes the areas where there are entrances and exits on the inner and outer ring of the VCI.

After having the entire network environment configured, the second part of the simulation generation process consisted of the parse of the OD matrix to the SUMO OD matrix file type. This was the last step to start the simulation generation using Sumo commands. To conclude and acquire the last simulation files in order to get the simulation running, first the *od2trips* [28] command was ran inside the simulation folder to get the *trips_day_hour.xml* file, giving as input the *taz.xml* file created before and the file containing the OD matrix converted to the XML notation, and run as follows:

```
od2trips -n InnerOuterRingTaz.xml --tazrelation-files
od-matrixes-xml/matrixOD_cd_{day}_{hour}.xml
--ignore-vehicle-type -o trips_{day}_{hour}.xml
```

The *od2trips* [28] command imports the OD matrix and splits it into single vehicle trips. In the command, the day corresponds to the day being simulated and hour to the time of day to be simulated. The *–ignore-vehicle-type* flag allowed the simulation to consider only one vehicle type, as the total volume of vehicles was used when constructing the OD matrix. Instead of using

the total volume of vehicles in the simulation, the forecast of vehicles by class can be used. In addition, a test with the *–different-source-sink* flag set to true to avoid loops on the trips generated was made. However, this flag was dropped as it did not significantly impact the final results. Afterward, routes are generated using *duarouter* [27] command also inside simulation folder as follows:

```
duarouter -n net.net.xml -r trips_{day}_{hour}.xml
--ignore-errors --write-trips
-o routes_{day}_{hour}.xml
```

The *duarouter* [27] command imports different demand definitions and builds vehicle routes from these definitions. In addition, it computes vehicle routes during the user assignment, which Sumo can use by employing the shortest path computation. If it is called iteratively, the duarouter executes dynamic user assignments. Finally, even though it is unnecessary for the project, this command can also repair any connectivity problems in existing route files. The preserved *net.net.xml* file containing the network configuration was provided as input along with the *trips_day_hour.xml* file generated. Errors are ignored on this generation with the *–ignore-errors* argument. In addition, two other flags were tested: the **–remove-loops** and *–with-taz* flags to remove loops and to force the vehicles to move only through entrances and exits, respectively. However, these flags had no success, especially the *–with-taz* that made the simulation impossible to run. Finally, the simulation may be ran. To visualize the simulation it is only necessary to provide to the *planner.sumo.cfg* the network and the routes generated file (based on the OD matrix generated) as input files as follows:

```
<input>
    <net-file value="net.net.xml"/>
    <route-files value="routes_{day}_{hour}.xml"/>
    <additional-files
         value="sumoSaveData.xml"/>
</input>
```

The microscopic simulation was evaluated by comparing the results of the actual radars with the virtual radars positioned on the simulation mirroring the real ones. The first step to evaluate the simulation is editing the *sumoSaveData.xml* file containing additions to the network by adding induction loops [29]. Induction loops count how many cars would pass through specific lanes on the highway during that hour (since it is the period of analysis selected). The induction loops were created for each lane in each direction with a frequency of 3600 to store the whole hour period on a *radar_out.xml* file. This file was parsed to condense all lane numbers into one to later compare it with the results captured in real life. The induction loops corresponding to the existing radars were added to the *sumoSaveData.xml* simulation with the following XML code:

```
<inductionLoop id="1063682_5_E0" lane="1063682_0"
```

```
            pos="-50" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063682_5_E1" lane="1063682_1"
            pos="-50" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063682_5_E2" lane="1063682_2"
            pos="-50" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063682_5_E3" lane="1063682_3"
            pos="-50" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063512_5_I0" lane="1063512_0"
            pos="0" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063512_5_I1" lane="1063512_1"
            pos="0" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063512_5_I2" lane="1063512_2"
            pos="0" freq="3600" file="radar_out.xml"/>
 <inductionLoop id="1063512_5_I3" lane="1063512_3"
            pos="0" freq="3600" file="radar_out.xml"/>
<inductionLoop id="121756_21_E0" lane="1181566_0"
            pos="0" freq="3600" file="radar_out.xml"/>
<inductionLoop id="121756_21_E1" lane="1181566_1"
            pos="0" freq="3600" file="radar_out.xml"/>
<inductionLoop id="121756_21_I0" lane="2027240_0"
            pos="200" freq="3600" file="radar_out.xml"/>
<inductionLoop id="121756_21_I1" lane="2027240_0"
            pos="200" freq="3600" file="radar_out.xml"/>
```

References

[1] Ali R. Abdellah, Omar Abdul Kareem Mahmood, Alexander Paramonov, and Andrey Koucheryavy. Iot traffic prediction using multi-step ahead prediction with neural network. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 1–4, 2019.

[2] Manish Agarwal, Thomas Maze, and Reginald Souleyrette. Impact of weather on urban freeway traffic low characteristics and facility capacity. In *Proceedings of the 2005 Mid-Continent Transportation Research Symposium*, 09 2005.

[3] Irfan Ahmed. Travel time prediction and explanation with spatio-temporalfeatures: A comparative study, December 2021. Available online at https://zenodo.org/record/5806069#.Ypfk3HbMK3C (last access: June 2022).

[4] Irfan Ahmed, Indika Kumara, Vahideh Reshadat, A. S. M. Kayes, Willem-Jan van den Heuvel, and Damian A. Tamburri. Travel time prediction and explanation with spatio-temporal features: A comparative study. *Electronics*, 11(1), 2022.

[5] Ranwa Al Mallah, Alejandro Quintero, and Bilal Farooq. Prediction of traffic flow via connected vehicles. *IEEE Transactions on Mobile Computing*, PP:1–1, 07 2020.

[6] Ishteaque Alam, Mohammad Fuad Ahmed, Mohaiminul Alam, João Ulisses, Dewan Md. Farid, Swakkhar Shatabda, and Rosaldo J. F. Rossetti. Pattern mining from historical traffic big data. In *2017 IEEE Region 10 Symposium (TENSYMP)*, pages 1–5, 2017.

[7] Ahmad Ali, Yanmin Zhu, and Muhammad Zakarya. Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. *Neural Networks*, 145:233–247, 2022.

[8] Manal Almutairi, Frederic Stahl, and Max Bramer. Reg-rules: An explainable rule-based ensemble learner for classification. *IEEE Access*, PP:1–1, 02 2021.

[9] Javad Artin, Amin Valizadeh, Mohsen Ahmadi, Sathish A. P. Kumar, and Abbas Sharifi. Presentation of a novel method for prediction of traffic with climate condition based on ensemble learning of neural architecture search (nas) and linear regression. *Complexity*, 2021:8500572, 2021.

[10] Romain Billot, Nour-Eddin El Faouzi, and Florian De Vuyst. Multilevel assessment of the impact of rain on drivers' behavior: Standardized methodology and empirical analysis. *Transportation Research Record*, 2107(1):134–142, 2009.

[11] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[12] Jason Brownlee. Why one-hot encode data in machine learning?, July 2017. Available online at https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ (last access: March 2022).

[13] Caltrans. Performance measurement system (pems) data source. Available online at https://dot.ca.gov/programs/traffic-operations/mpr/pems-source (last access: June 2022).

[14] Salvatore M. Carta, Sergio Consoli, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Maria Madalina Stanciu. Ensembling and dynamic asset selection for risk-controlled statistical arbitrage. *IEEE Access*, 9:29942–29959, 2021.

[15] Alex Cazañas-Gordón, Esther Parra-Mora, and Luís A. Da Silva Cruz. Ensemble learning approach to retinal thickness assessment in optical coherence tomography. *IEEE Access*, 9:67349–67363, 2021.

[16] Chen Chen, Lei Liu, Shaohua Wan, Xiaozhe Hui, and Qingqi Pei. Data dissemination for industry 4.0 applications in internet of vehicles based on short-term traffic prediction. *ACM Trans. Internet Technol.*, 22(1), oct 2021.

[17] David Chin. Empirical evaluation of user models and user-adapted systems. *User Model. User-Adapt. Interact.*, 11:181–194, 03 2001.

[18] Lígia Conceição, Gonçalo Homem de Almeida Correia, and José Pedro Tavares. The reversible lane network design problem (rl-ndp) for smart cities with automated traffic. *Sustainability*, 12(3), 2020.

[19] Lígia Conceição, Gonçalo Homem de Almeida Correia, and José Tavares. The future of smart cities with automated vehicles: benefits from reversible lanes in a connected traffic control system. 09 2019.

[20] Microsoft Corporation. Lightgbm - features, 2022. Available online at https://lightgbm.readthedocs.io/en/latest/Features.html#leaf-wise-best-first-tree-growth (last access: May 2022).

[21] Sagar Deep Deb. Ensemble dcnn structure for detection of covid-19 from chest x-ray images. Available online at https://github.com/sagardeepdeb/ensemble-model-for-COVID-detection (last access: July 2022).

[22] Sagar Deep Deb, Rajib Kumar Jha, Kamlesh Jha, and Prem S Tripathi. A multi model ensemble based deep convolution neural network structure for detection of covid19. *Biomedical Signal Processing and Control*, 71:103126, 2022.

[23] K. Deepthi. Eldma. Available online at https://github.com/Deepthi-K523/ELDMA (last access: June 2022).

[24] K. Deepthi and A. S. Jereesh. An ensemble approach based on multi-source information to predict drug-mirna associations via convolutional neural networks. *IEEE Access*, 9:38331–38341, 2021.

[25] Zulong Diao, Xin Wang, Dafang Zhang, Yingru Liu, Kun Xie, and Shaoyao He. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):890–897, Jul. 2019.

[26] George Dimitrakopoulos and Panagiotis Demestichas. Intelligent transportation systems. *IEEE Vehicular Technology Magazine*, 5(1):77–84, 2010.

[27] German Aerospace Center (DLR) et al. duarouter, 2022. Available online at https://sumo.dlr.de/docs/duarouter.html (last access: May 2022).

[28] German Aerospace Center (DLR) et al. od2trips, 2022. Available online at https://sumo.dlr.de/docs/od2trips.html (last access: May 2022).

[29] German Aerospace Center (DLR) and others. Induction loops detectors (e1), May 2022. Available online at https://sumo.dlr.de/docs/Simulation/Output/Induction_Loops_Detectors_%28E1%29.html (last access: May 2022).

[30] N Dodd and JW Vanloon. Applications of the traffic prediction model. *ECONOMETRICA*, 31(3):570, 1963.

[31] Hanxuan Dong, Fan Ding, Huachun Tan, and Hailong Zhang. Laplacian integration of graph convolutional network with tensor completion for traffic prediction with missing data in intercity highway network. *Physica A: Statistical Mechanics and its Applications*, 586:126474, 2022.

[32] Jorge Miguel Marques dos Reis. Previsão em tempo real de condições de tráfego em redes veiculares. Master's thesis, Faculdade de Engenharia da Universidade do Porto, R. Dr. Roberto Frias, 4200-465 Porto, 10 2016.

[33] Mesfer Al Duhayyim, Amani Abdulrahman Albraikan, Fahd N. Al-Wesabi, Hiba M. Burbur, Mohammad Alamgeer, Anwer Mustafa Hilal, Manar Ahmed Hamza, and Mohammed Rizwanullah. Modeling of artificial intelligence based traffic flow prediction with weather conditions. *Computers, Materials & Continua*, 71(2):3953–3968, 2022.

[34] Rohit Dwivedi. What is lightgbm algorithm, how to use it?, June 2020. Available online at https://www.analyticssteps.com/blogs/what-light-gbm-algorithm-how-use-it (last access: May 2022).

[35] edureka! Ai vs machine learning vs deep learning, January 2022. Available online at https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/ (last access: March 2022).

[36] Aniekan Essien, Ilias Petrounias, and Sandra Sampaio. *The Impact of Rainfall and Temperature on Peak and Off-Peak Urban Traffic*, pages 399–407. Springer International Publishing, 01 2018.

[37] Siyun Feng, Jiashuang Huang, Qinqin Shen, Quan Shi, and Zhenquan Shi. A hybrid model integrating local and global spatial correlation for traffic prediction. *IEEE Access*, 10:2170–2181, 2022.

[38] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[39] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. Supplementary material for efficient and robust automated

machine learning. Available online at https://ml.informatik.uni-freiburg.de/wp-content/uploads/papers/15-NIPS-auto-sklearn-supplementary.pdf (last access: March 2022).

[40] Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29, 11 2000.

[41] P Hogberg. Estimation of parameters in models for traffic prediction – nonlinear-regression approach. *Transportation Research*, 10(4):263–265, 1976.

[42] Faliang Huang, Guoqing Xie, and Ruliang Xiao. Research on ensemble learning. In *2009 International Conference on Artificial Intelligence and Computational Intelligence*, volume 3, pages 249–252, 2009.

[43] Saeid Iranmanesh, Forough Shirin Abkenar, Abbas Jamalipour, and Raad Raad. A heuristic distributed scheme to detect falsification of mobility patterns in internet of vehicles. *IEEE Internet of Things Journal*, 9(1):719–727, 2022.

[44] jbdj star. st. Available online at https://github.com/jbdj-star/st (last access: June 2022).

[45] Anirudh Ameya Kashyap, Shravan Raviraj, Ananya Devarakonda, Shamanth R Nayak K, Santhosh K V, and Soumya J Bhat. Traffic flow prediction models – a review of deep learning techniques. *Cogent Engineering*, 9(1):2010510, 2022.

[46] Michael Kyte, Zaher Khatib, Patrick Shannon, and Fred Kitchener. Effect of weather on free-flow speed. *Transportation Research Record*, 1776(1):60–68, 2001.

[47] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20), 2019.

[48] Microsoft. What is automated machine learning (automl)?, February 2022. Available online at https://docs.microsoft.com/en-us/azure/machine-learning/concept-automated-ml (last access: March 2022).

[49] Fábio Oliveira and Ana Paula Rocha. Filling missing values in spatial-temporal data collected from traffic sensors. In *2020 IEEE International Smart Cities Conference (ISC2)*, pages 1–7, 2020.

[50] Morteza Pakdaman, Iman Babaeian, Zohreh Javanshiri, and Yashar Falamarzi. European multi model ensemble (emme): A new approach for monthly forecast of precipitation. *Water Resources Management*, 36(2):611–623, 2022.

[51] Morteza Pakdaman, Yashar Falamarzi, Iman Babaeian, and Zohreh Javanshiri. Post-processing of the north american multi-model ensemble for monthly forecast of precipitation based on neural network models. *Theoretical and Applied Climatology*, 2020, 07 2020.

[52] Sriram Parthasarathy. Top 5 predictive analytics models and algorithms, February 2021. Available online at https://www.logianalytics.com/predictive-analytics/predictive-algorithms-and-models/ (last access: March 2022).

[53] Viral Patel, Manish Chaturvedi, and Sanjay Srivastava. Comparison of sumo and simtram for indian traffic scenario representation. *Transportation Research Procedia*, 17:400–407, 2016.

International Conference on Transportation Planning and Implementation Methodologies for Developing Countries (12th TPMDC) Selected Proceedings, IIT Bombay, Mumbai, India, 10-12 December 2014.

[54] scikit-learn developers (BSD License). 1.10. decision trees. Available online at https://scikit-learn.org/stable/modules/tree.html (last access: May 2022).

[55] scikit-learn developers (BSD License). 1.17. neural network models (supervised). Available online at https://scikit-learn.org/stable/modules/neural_networks_supervised.html (last access: May 2022).

[56] scikit-learn developers (BSD License). Decision tree regression. Available online at https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#sphx-glr-auto-examples-tree-plot-tree-regression-py (last access: May 2022).

[57] scikit-learn developers (BSD License). sklearn.metrics.r2_score. Available online at https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html (last access: May 2022).

[58] Aishwarya Singh. A comprehensive guide to ensemble learning (with python codes), June 2018. Available online at https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/ (last access: February 2022).

[59] Brian L. Smith, Kristin Byrne, Rachel B. Copperman, Susan M. Hennessy, and Noah Goodall. An investigation into the impact of rainfall on freeway traffic flow. In *Proceedings of the 83rd Annual Meeting of the Transportation Research Board*, 2003.

[60] Miao Tian, Chen Sun, and Shaozhi Wu. An emd and arma-based network traffic prediction approach in sdn-based internet of vehicles. *Wireless Networks*, 08 2021.

[61] Ioannis Tsapakis, Tao Cheng, and Adel Bolbol. Impact of weather conditions on macroscopic urban travel times. *Journal of Transport Geography*, 28:204–211, 2013.

[62] uci machine learning repository. Taxi service trajectory - prediction challenge, ecml pkdd 2015 data set. Available online at https://archive.ics.uci.edu/ml/datasets/Taxi+Service+Trajectory+-+Prediction+Challenge,+ECML+PKDD+2015 (last access: June 2022).

[63] Iowa State University. Iowa environmental mesonet. Available online at https://mesonet.agron.iastate.edu/request/download.phtml (last access: March 2022).

[64] Louise Wright and Stuart Davidson. How to tell the difference between a model and a digital twin. *Advanced Modeling and Simulation in Engineering Sciences*, 7(1):13, 2020.

[65] Haitao Yuan and Guoliang Li. A survey of traffic prediction: from spatio-temporal data to intelligent transportation. *Data Science and Engineering*, 6(1):63–85, 2021.

[66] Lu Zhang, Jianjun Tan, Dan Han, and Hao Zhu. From machine learning to deep learning: progress in machine intelligence for rational drug discovery. *Drug Discovery Today*, 22(11):1680–1685, 2017.

[67] Xiyue Zhang, Chao Huang, Yong Xu, Lianghao Xia, Peng Dai, Liefeng Bo, Junbo Zhang, and Yu Zheng. Traffic flow forecasting with spatial-temporal graph diffusion network. *ArXiv*, abs/2110.04038, 2021.

[68] Yi Zhang, Jun Xiao, Zhou Zhang, and Hua Dong. Intelligent design of robotic welding process parameters using learning-based methods. *Ieee Access*, page 9, 1 2022.

[69] Liang Zheng, Huimin Huang, Chuang Zhu, and Kunpeng Zhang. A tensor-based k-nearest neighbors method for traffic speed prediction under data missing. *Transportmetrica B: Transport Dynamics*, 8(1):182–199, 2020.

[70] Zhi-Hua Zhou. *Ensemble Learning*, pages 181–210. Springer Singapore, Singapore, 2021.