Technische Universität München

M.Sc. Transportation Systems

Development of Data-driven Car-following Models

Master's Thesis of

Filippos Adamidis

Supervised by

Univ.-Prof. Dr. Constantinos Antoniou

M.Sc. Vishal Mahajan

February 5, 2021

Disclaimer

I declare that the following document has been written independently without the use of any external sources or auxiliary means other than those indicated and that everything stated is true to the best of my knowledge. This thesis has not been submitted elsewhere for purposes of assessment.

February 5, 2021

Filippos Adamidis

Acknowledgements

I would like to thank Univ.-Prof. Dr. Constantinos Antoniou for his guidance and support from the beginning until the end of this thesis. His ideas and comments have played a crucial role in shaping and steering this study. Further, I would like to thank my supervisor, M.Sc. Vishal Mahajan, for his practical hands-on comments and feedback throughout this work.

I am very grateful to the German Academic Exchange Service (DAAD) for its financial support and the multiple social events, which allowed me to focus on my master's studies and enjoy my stay in Munich.

Last but not least, special thanks go to my family and close friends for their lasting support, patience and understanding throughout the years, especially knowing that I am a difficult person to deal with. Their contribution will never be forgotten.

Abstract

Car-following models are essential parts of any traffic simulation package. In the past, researchers have successfully explained car-following behaviour mathematically. The recent availability of accurate datasets from drone videography has facilitated the development of data-driven car-following models. This thesis investigates if data-driven models are more precise than mathematical models, despite their inferior interpretability. To this aim, the highway drone dataset (highD) from Cologne, Germany is clustered into two traffic phases based on Gaussian mixture models clustering. Subsequently, car-following models with two leaders are developed in each phase using multiple linear regression, including penalisation techniques, random forest and gradient boosting machines regression. Gipps' and Krauss' models are calibrated with the same dataset and then compared with the trained data-driven models. Further, a validation procedure using unseen data with varying reaction times confirms that double data-driven methods can surpass the performance of conventional models. It is the first time that a sensitivity analysis of Krauss' car-following model takes place and the calibrated model is put to comparison with Gipps' and other models. The results of this study are meant to improve the results of microscopic traffic simulations.

Keywords: car-following, data-driven, reaction time, traffic phase, Gipps, Krauss, sensitivity

Table of Contents

List of Figures vi

List of Tables viii

Abbreviations ix

Part I: Introduction and Literature Review 1

1. Introduction 1
   - 1.1. Problem statement 2
   - 1.2. Research questions and objectives 3
   - 1.3. Contributions 3
   - 1.4. Thesis outline 3
2. Literature review 5
   - 2.1. Data from drone videography 5
   - 2.2. The highway drone dataset (highD) 7
   - 2.3. Mathematical car-following models 12
   - 2.4. Multiple car-following models 13
   - 2.5. Data-driven longitudinal and lateral movement models 14
   - 2.6. Traffic phase identification 16
   - 2.7. Summary and research gaps 17

Part II: Data Collection, Methodology and Analysis 19

3. Data collection 19
   - 3.1. General information about the dataset 19
   - 3.2. Track files 20
   - 3.3. Track Meta Information files 22
   - 3.4. Recording Meta Information files 23
4. Methodology 24
   - 4.1. Double car-following scenario 25
   - 4.2. Reaction time emulation procedure 25
   - 4.3. Traffic phase identification 28
   - 4.4. Model development 32
   - 4.5. Model validation 39

5. Data Analysis 42
5.1. Software and tools 42
5.2. Exploratory data analysis 43

Part III: Results 51

6. Traffic phase identification 51
6.1. Traffic phase by direction 51
6.2. Traffic phase by lane 56
6.3. Matching congested and free states to trajectories 58

7. Model development and validation 59
7.1. Congested phase 59
7.2. Free phase 70
7.3. Sensitivity analysis of mathematical car-following models 74
7.4. Sensitivity analysis of data-driven models 82
7.5. Validation of the data-driven and mathematical car-following models 87

Part IV: Further Proposals and Conclusions 92

8. Proposals for future work 92

9. Main findings and conclusions 96
9.1. Main findings 96
9.2. Conclusions 96

Bibliography 98

Appendix 108

List of Figures

Figure 2.1 Fundamental relations of traffic flow 17
Figure 3.1 A three-lane highway from highD (Track 37) 21
Figure 3.2 Coordinate system and lane numbering in highD 22
Figure 4.1 Overview of the methodology 24
Figure 4.2 Double car-following scenario 25
Figure 4.3 Workflow of traffic phase identification 28
Figure 4.4 Workflow of model development 32
Figure 4.5 Example of a random forest for regression with 600 trees 36
Figure 4.6 Workflow of car-following model calibration and validation 39
Figure 5.1 Virtual loop detectors in the middle of a highway section 44
Figure 5.2 Macroscopic properties of traffic by direction 45
Figure 5.3 Macroscopic properties of traffic by lane 46
Figure 5.4 Boxplots of velocity, distance headway and percentage of trucks per track 48
Figure 5.5 A random velocity trajectory from highD and the re-extracted NGSIM I-80 50
Figure 6.1 BIC of different parametrisation combinations and number of clusters in traffic phase by direction 52
Figure 6.2 Best three parametrisation combinations in traffic phase by direction 53
Figure 6.3 GMM clustering results of three-lane highway areas by direction 54
Figure 6.4 BIC of different parametrisation combinations and number of clusters in traffic phase by lane 56
Figure 6.5 Best three parametrisation combinations in traffic phase by lane 57
Figure 6.6 All trajectories of Track 12, lane 2 (top); trajectories in congested phase (bottom) 58
Figure 7.1 Velocity - preceding velocity - distance headway plots in congested phase 60
Figure 7.2 Correlation plot for reaction time $\tau = 1.2\ s$ in congested phase 61
Figure 7.3 Linear regression residuals plot in congested phase 64
Figure 7.4 Elastic net grid search of a and λ 66
Figure 7.5 Variable importance in random forest regression 68
Figure 7.6 Variable importance in gradient boosting machines regression 69
Figure 7.7 Velocity - preceding velocity - distance headway plots in free phase 70
Figure 7.8 Correlation plot for reaction time τ=1.2 s in free phase 71
Figure 7.9 Distribution of xVelocity in free phase 72
Figure 7.10 Linear regression residuals plot in free phase 74

Figure 7.11 Sensitivity analysis of Gipps' car-following model in congested phase 76
Figure 7.12 Sensitivity analysis of Gipps' car-following model in free phase 78
Figure 7.13 Sensitivity analysis of Krauss' car-following model in congested phase.....80
Figure 7.14 Sensitivity analysis of Krauss' car-following model in free phase.....81
Figure 7.15 Sensitivity analysis of data-driven and mathematical models in congested phase (test set) 84
Figure 7.16 Sensitivity analysis of data-driven and mathematical models in free phase (test set) 86
Figure 7.17 Distribution of longitudinal velocity in the re-extracted NGSIM dataset.....87
Figure 7.18 Validation results in congested phase 89
Figure 7.19 Distribution of longitudinal velocity in two-lane Tracks of highD.....90
Figure 7.20 Validation results in free phase 91
Figure 8.1 Correlation plot in congested phase including density per lane.....92
Figure 8.2 Boxplots of velocity, distance headway and time-to-collision for different vehicle types ..93
Figure 8.3 Boxplots of velocity, distance headway and time-to-collision for different vehicle pairs...94
Figure 9.1 Density - velocity diagram with two clusters (by direction).....108
Figure 9.2 Density - flow diagram with two clusters (by direction) 109
Figure 9.3 Flow - velocity diagram with two clusters (by direction) 110
Figure 9.4 Density - velocity diagram with two clusters (by lane) 111
Figure 9.5 Density - flow diagram with two clusters (by lane) 112
Figure 9.6 Flow - velocity diagram with two clusters (by lane).....113
Figure 9.7 Comparison of real and estimated velocity trajectories (data-driven and mathematical models) from Track 24, Vehicle IDs: 402 & 500 114

List of Tables

Table 2.1 Summary of studies related to highD 11
Table 3.1 Attributes in a Track file from highD 20
Table 3.2 Attributes in a Track Meta Information file from highD 22
Table 3.3 Attributes in a Recording Meta Information file from highD 23
Table 4.1 Time-lagged variables of the ego vehicle 26
Table 4.2 Time-lagged variables of the preceding vehicle 26
Table 4.3 Time-lagged variables of the leading vehicle 27
Table 5.1 Comparison of highD and NGSIM 49
Table 6.1 Summary statistics of the congested and free phases 55
Table 7.1 Predictors included in linear regression in congested phase 62
Table 7.2 Measures of fit of linear regression models in congested phase 62
Table 7.3 Coefficients of the linear regression LR1 in congested phase 63
Table 7.4 Coefficients of ridge regression in congested phase 65
Table 7.5 Coefficients of lasso regression in congested phase 66
Table 7.6 Coefficients of elastic net regression in congested phase 67
Table 7.7 Coefficients of the linear model including the velocity of the preceding in free phase 73
Table 7.8 Coefficients of the linear model excluding the velocity of the preceding in free phase 73
Table 7.9 Sensitivity analysis of the reaction time using Gipps' model in congested phase (test set) 77
Table 7.10 Sensitivity analysis of the reaction time using Gipps' model in free phase (test set) 79
Table 7.11 Sensitivity analysis of the reaction time using Krauss' model in congested phase (test set) 80
Table 7.12 Sensitivity analysis of the reaction time using Krauss' model in free phase (test set) 81
Table 7.13 Sensitivity analysis of the linear regression model in congested phase 82
Table 7.14 Sensitivity analysis of the elastic net regression model in congested phase 82
Table 7.15 Sensitivity analysis of the random forest model in congested phase 83
Table 7.16 Sensitivity analysis of the gradient boosting machines model in congested phase 83
Table 7.17 Sensitivity analysis of the linear regression model (including preceding velocity) in free phase 85
Table 7.18 Sensitivity analysis of the linear regression model (excluding preceding velocity) in free phase 85
Table 9.1 Gradient boosting machines parameter grid search 115

Abbreviations

BIC	Bayesian Information Criterion
DHW	Distance Headway
EM	Expectation Maximisation
FD	Fundamental Diagram
GBM	Gradient Boosting Machines
GMM	Gaussian Mixture Models
GNN	Graph Neural Network
IDM	Intelligent Driver Model
LASSO	Least Absolute Shrinkage and Selection Operator
LOESS	Locally Estimated Scatterplot Smoothing
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MCF	Multiple Car-Following
NGSIM	Next Generation Simulation dataset
NN	Neural Network
OBD	On-board Diagnostics
OLS	Ordinary Least Squares
OVM	Optimal Velocity Model
OVM2	Optimal Velocity Model with two predecessors
RF	Random Forest
RMSE	Root Mean Square Error
RMSN	Root Mean Square Normalised error
SSM	Safety Surrogate Measure
SUMO	Simulation of Urban Mobility
THW	Time headway
TTC	Time-to-collision
UAV	Unmanned Aerial Vehicle
VIF	Variance Inflation Factor

Part I: Introduction and Literature Review

1. Introduction

Private transport is connected to freedom of choice, flexibility, comfort and pleasure. On the other hand, it is also associated with environmental concerns, waste of resources and space and serious safety deficiencies. Despite the continuous efforts to emphasise the drawbacks of passenger cars, they still constitute the prevailing mode of inland transport in Europe (Eurostat, 2020). In Germany, around 75% of transportation takes place using motorised individual transport means and over half of the total freight measured in tonne-kilometres gets transported by trucks (Umweltbundesamt, 2020). A vast network of free-to-access highways with high standards and easy access from most metropolitan areas supports the use of private means of passenger transport not only for short-distances but also for longer trips.

Road transport plays a similar, if not principal, role in most countries around the world. Hence, research in individual transport has been a favourite subject in many universities, research institutions and industrial research entities. Yet, we still have not managed to develop complete understanding of all the processes involved when driving. This can be attributed to many reasons, such as the fact that the characteristics of traffic and the behaviour of drivers are highly dynamic processes. Especially with regards to the decisions and actions of individuals when driving, it can be said that they depend greatly on personal perceptions, experiences and on their way of thinking and acting.

The various subtasks involved in driving have been categorised earlier by Cumming (1963). Among them, car-following, the task of one vehicle following the preceding vehicle on a single lane of the road, is one of the simplest ones, for which we have developed a good understanding. Years of effort have been put in creating models that can estimate the movement of a follower based on the behaviour of the leader. These models are characterised as microscopic, because they reproduce the behaviour of individuals, in comparison to macroscopic models that confront individuals collectively as an aggregate entity.

Some of the car-following models are well known because of their applications in automotive technologies or in traffic simulation packages. The most prominent models can be categorised into several groups: the General Motors' model, based on the concept that the follower adapts their speed to that of the leader as an almost immediate stimulus-response action (Chandler et al., 1958); the Safe Distance models, such as Gipps' model, where the follower tries to maintain a safe distance to the leader so as to avoid collision (Gipps, 1981); the Action Points models, where the follower reacts to the speed change of the leader only when it surpasses a perceptual threshold (Evans & Rothery, 1977); the Optimal Velocity (OVM) models, where the distance headway and the difference between the current and the optimal velocity define the adjustment

of the acceleration (Bando et al., 1995) and finally; the Risk Homeostasis Theory model, where the driver adjust their driving behaviour to achieve the level of risk they are willing to accept (Wilde, 1982). All these models offer a physical explanation of the car-following behaviour.

1.1. Problem statement

Recently, researchers have increasingly proposed data-driven car-following models, which offer no physical explanation of the included parameters but can adapt well to new data. In this alternative approach, the availability of reliable data is very important. Of course, the emergence of new non-intrusive data sources like cameras and drones has enabled us to capture more reliable and unbiased observations.

At the same time, the widespread access to computers and the internet has sparked an unprecedent interest and exchange of ideas in traffic research, not only in new areas but also in long-standing domains of transportation. Nowadays, machine learning is a widespread topic in transportation research and real-life applications. Therefore, a logical question that arises is "How can we improve the existing knowledge in traffic research using the opportunity of new data sources and the capabilities of computer-aided learning?".

To answer this question, it is necessary to address many different subproblems. First, it is necessary to identify a dataset that contains high-quality trajectories. A trajectory is the path of an object through space and time. In this study, a "velocity trajectory" corresponds to the observations of speed through time for a single vehicle. The dataset needs not only to provide trajectories that minimise the error between the reality and the captured data, but also to provide variables that describe the vehicle and its relations to the surrounding vehicles or infrastructure.

Then, it is also necessary to determine the conditions under which different drivers interact with each other. Based on previous research and existing knowledge in traffic engineering, a data-driven, unsupervised methodology will help to group aggregation intervals with common characteristics together. This grouping is crucial in identifying which vehicles were driven under conditions of interaction between drivers and which took place with no interaction at all.

The development of data-driven models should follow two main principles. Firstly, it is important to create models with high accuracy, so as to improve the results of existing research. Secondly, these models have to be kept as simple as possible, in order to make use of necessary parameters only. On the other hand, interpretability and explainability are not the key in data-driven models; some algorithms like linear regression do provide both, others like random forests function as "black boxes" that can be logically explained only partially. Therefore, a results-oriented approach will be followed in this thesis.

In view of this result-oriented approach, there is a need to include a validation procedure, which will ensure the applicability of the models on further occasions. Therefore, new datasets have to be sought, which will be similar in form with the original dataset but will also feature differences with regards to the traffic scene, data collection methods, data resolution etc.

1.2. Research questions and objectives

The issued discussed above can be summarised into two main research questions:

- Is it **possible** to create data-driven car-following models using high-speed trajectories from drones?
- Do data-driven models perform **better** than mathematical car-following models on highway traffic?

To answer these research questions, a stepwise methodology will be introduced in the next chapter. The objectives of this methodology are to prove the suitability of the chosen dataset for modelling reactions between drivers, to achieve a scientifically justified segregation of the dataset in different traffic phases and to develop a procedure for training, testing and validating data-driven models. In order to determine the ability of those models to reproduce realistically the velocity of any following vehicle, they are compared with fine-tuned, established mathematical car-following models. Finally, an objective of this study is to propose methodological advances that could improve the data-driven models in the future.

1.3. Contributions

This study aims to contribute to the following topics of car-following models on highways:

- Unsupervised traffic phase identification using aggregated direction-based or lane-based macroscopic properties of traffic flow from drone videography. Emphasis is placed on the consistency of the results with regards to the existing knowledge on traffic phases.
- Development of precise data-driven models with one or two leaders using different algorithms. The modelling approach is mainly directed towards the achievement of higher precision rather than interpretability.
- Sensitivity analysis and comparison of data-driven and calibrated mathematical models using different reaction times. For the first time, Krauss' car-following model is calibrated and put to comparison with other models.
- Application of data-driven and calibrated mathematical models on unseen data and comparison of their results.

1.4. Thesis outline

This report is structured in four parts, each of them containing two or three chapters. A short summary of these chapters is presented below:

Part I: Introduction and Literature Review

- **Chapter 1** introduces the research problem, the research questions, the objectives and the contributions of this study.

- **Chapter 2** includes the literature review on subjects in immediate relation to this study and states any possible research gaps. These subjects concern car-following models, novel data-collection methods, studies related to specific datasets etc.

Part II: Data Collection, Methodology and Analysis

- **Chapter 3** introduces the dataset along with explanations of the included variables.
- **Chapter 4** presents the methodology that is used in this study during data analysis, model development and validation. A general flowchart of the methodology is given in the beginning of this chapter. Similarly, flowcharts in the beginning of each subchapter provide more information about the processes that will be used on each stage of the methodology.
- **Chapter 5** lists the software that will be used in analysis and model development and provides a concise exploratory data analysis.

Part III: Results

- **Chapter 6** presents the results of traffic phase identification as well as some analysis of the results after splitting the dataset into two subsets.
- **Chapter 7** includes variable selection, model development and testing and model validation of data-driven models. Additionally, a calibration procedure for mathematical car-following models is presented. Then, mathematical and data-driven models are compared.

Part IV: Further proposals and conclusions

- **Chapter 8** includes proposals and a preliminary analysis of some proposals for future work
- **Chapter 9** summarises the main finding of this study and states the main conclusions.

The references and an appendix can be found at the end of this document.

2. Literature review

In this chapter, scientific articles concerning the topics of this thesis are presented and analysed. The intention of this literature review is to discuss relevant and recent advances in data collection and analysis, in car-following models and generally in traffic engineering in order to provide the framework of this thesis and to identify possible gaps in methodology.

2.1. Data from drone videography

Modern applications in traffic engineering utilise a multitude of different data sources. Intrusive data collection methods include pneumatic road tubes, piezoelectric sensors and inductive loop detectors, which is the most common technology used in vehicle detection. Non-intrusive technologies rely on remote observations such as manual counting, microwave radars, ultrasonic and passive acoustic technologies, Bluetooth sensors and video image detection (Leduc, 2008). Lately, **unmanned aerial vehicles (UAVs) have been used increasingly as a means to capture traffic data in large volumes** (Barmpounakis et al., 2016).

An unmanned aerial vehicle, or commonly known as a drone, is any aircraft with no pilot on-board, but rather operated remotely from the ground. The classification of drones depends mainly on form, size and weight. Their form can either resemble an airplane, with fixed wings, or be based on propellers, usually like helicopters or multi-copters i.e. with more than one propeller. By size they are distinguished into micro, mini, medium or large drones. Weight is the main factor used as the unit of measurement for regulations by aviation authorities worldwide (Gattuso et al., 2021).

According to Salvo et al. (2014), UAVs have many advantages in comparison to traditional data collection methods, for example low operation altitudes which allow for better video resolution, reduced operation costs, as one drone can be used on multiple areas or even for different purposes, and more realistic data representations, as it is barely visible from within the vehicle and does not distract the driver. In their study they used a radio-controlled, vertical take-off and landing micro-drone in order to collect video footage for a traffic study in the city of Palermo, Italy, lasting ten minutes with a single charge. The collected dataset was used to calibrate a microscopic traffic simulation model.

Nevertheless, there are still **practical hurdles to overcome when deploying drones in traffic studies**. For example, the study area might be too large for a single drone and the duration of a single battery charge may not be sufficient to record a traffic event. An interesting question in research and development of new unmanned aerial vehicles is how to increase operational time with a single charge, either by optimising the materials in order to reduce weight, new energy-efficient aerodynamical forms, or by utilising new battery technologies. Barmpounakis et al. (2016) mentioned that the most of the new commercial rotorcraft drones come equipped with high-definition cameras and improved hovering abilities, which help researchers and practitioners acquire high quality traffic data. However, the authors raised **issues concerning privacy**, as the improved capabilities of UAVs enable them to be used as surveillance systems.

It has to be pointed out that the same concerns apply to most modern image-based traffic data acquisition systems.

Despite the fact that the idea to use drones in transportation is quite new, many researchers have already tried to investigate their ability to capture the traffic flow accurately and proposed a wide array of methodologies for new applications. Khan et al. (2017) presented an extensive review of transportation studies based on UAV data and then introduced a holistic methodological framework guiding the planning, conduction and completion of a drone-based traffic study. According to them, a complete drone-based traffic analysis comprises of six steps: scope definition, flight planning, flight implementation, data acquisition, data analysis and finally, data interpretation. This process leads to an optimised traffic application, for example the calibration of a simulation model. Further, they believe that flight telemetry data (altitude, orientation, speed of the drone) are very important for calibrating and stabilising the recorded video footage.

A similar review of applications of UAVs on traffic management can be found in Barmpounakis et al. (2016). The same authors investigated whether data gathered from drones are accurate enough (Barmpounakis et al., 2019). To answer their research question, they compared speed data collected from a drone to on-board diagnostics (OBD) speed data, which are directly measured on-vehicle. Results showed that **the accuracy of drone footage was high when an appropriate stabilisation and geo-reference procedure is followed**, whereas the error did not seem to increase upon increasing the cruising altitude of the drone. The authors achieved a speed estimation error of less than 1 km/h, which led them to their final observation that drone datasets are precise enough for traffic studies.

Aerial observations provide the necessary accuracy for measuring microscopic trajectories of vehicles in space and time. Kaufmann et al. (2018) observed moving synchronised patterns before a signalised intersection using a small quadcopter drone equipped with a 4K camera and a gimbal for image stabilisation. The drone hovered steadily 100 m above the intersection and the authors found that the accuracy diminishes at the end of the recorded section or, simply said, the accuracy was better close to the drone. One **problem they encountered was the lateral overlapping of different moving objects**, which caused the tracking algorithm to miss the movement of these objects. However, this was corrected by interpolating the position of one object between two known positions. Their research showed that moving synchronised patterns occur when traffic flow transitions from free to synchronised state. Barmpounakis et al. (2018) used a hexacopter to record traffic streams from a three-lane arterial in Athens, Greece. In total 3:30 hours were recorded during different working days, but under the same traffic and weather conditions. Their research modelled the decision of the driver of a powered two-wheeler (PTW) to overtake and their intention to overtake or undertake the preceding vehicle.

Lately, drone-based datasets have been increasing in volume and reach regarding the number of captured vehicles, road users and locations. To perform safety evaluation of automated driving systems, Krajewski et al. (2018) recorded 16.5 hours of traffic on different highways around Cologne, Germany. A quadcopter hovered directly above each highway for an average duration of 17 minutes during sunny and windless days, in order to reduce the need for

stabilisation. The highways drone dataset, also known as highD, includes around 110000 trajectories from six different highways, with the median duration of a trajectory being around 13.6 sec on a road segment of about 420 m in length. The accuracy of the trajectories is around 10 cm. Other notable works include the inD dataset containing trajectories of vehicles and vulnerable users (e.g. bicycles, pedestrians) on intersections (Bock et al., 2019), the roundD dataset with trajectories of vehicles and vulnerable users on roundabouts (Krajewski et al., 2020), the Stanford dataset containing almost exclusively pedestrians (Robicquet et al., 2016) and the Harpy dataset for developing vehicle tracking algorithms from video footage (Makrigiorgis et al., 2020).

The most extensive study to date utilised **a swarm of ten drones** hovering simultaneously over a central part of Athens, Greece (Barmpounakis & Geroliminis, 2020). The area included more than 100 km of road network and at least 100 signalised or non-signalised intersections. The authors based their study on an extended morning peak hour, from 08:00 to 10:30 on each working day, aiming to record the development and the unwinding of traffic phenomena. According to them, drone footage is able to record spatially wide and temporally lengthy disturbances, which can otherwise not be captured by inductive loop detectors or GPS sensors. The existence of multiple road users, such as passenger cars, powered two-wheelers, taxis, buses and trolleybuses will enable the research of inter-vehicle interactions. The authors concluded that **UAVs are able to replace human observers or conventional sensors**, while at the same time offering a high precision of measurement.

All in all, it can be said that drones have opened up many opportunities in traffic research, without lowering the quality of data. Most researchers agree that it is crucial to apply a stabilisation mechanism, be it physical during video recording or digital upon post-processing, and to ensure that traffic conditions allow to capture the phenomenon to be studied. However, almost all of them stress the fact that drones cannot be used under unfavourable weather conditions, which is also a decisive argument against the use of drones for online traffic management. For a review of UAV-based remote sensing applications see Xiang et al. (2019).

2.2. The highway drone dataset (highD)

The release of the Next Generation Simulation (NGSIM, 2006) dataset in 2006 sparked the interest of researchers and practitioners to develop more accurate microscopic traffic simulation models. Until then, limitations in data collection methods, technologies and devices were hindering further advances in their understanding on microscopic and, to a certain extent, mesoscopic and macroscopic phenomena (Li et al., 2020). Until today, **NGSIM** remains the most referenced dataset in traffic engineering, despite the fact that some researchers have underlined its **shortcomings in data accuracy**, stemming from technical and methodological limitations. These limitations mainly refer to camera equipment (e.g. lower resolution), post-processing methodologies (e.g. during trajectory extraction from video) and the lack of a reliable procedure to assess the quality of the extracted trajectories (Coifman & Li, 2017; Montanino & Punzo, 2015; Punzo et al., 2011). New data acquisition methods, such as drones as seen in the previous paragraphs, have helped to overcome these limitations.

HighD aims to improve the shortcomings of NGSIM and to open up new opportunities in safety evaluation of automated driving vehicles. It is the first open dataset **containing purely naturalistic highway trajectories** (Krajewski et al., 2018) and it was **recorded by utilising aerial measurements**, which has several advantages in terms of accuracy, effort effectiveness and privacy protection. Despite its relatively short lifetime, several researchers have already managed to publish significant work. Most applications focus around four themes; estimation and forecasting of fundamental macroscopic characteristic of traffic flow, trajectory and manoeuvre prediction based on environmental influences, calibration of car-following models and safety evaluation of (automated) driving. The following paragraphs summarise research based on the highD dataset. For a comparison between NGSIM and highD see paragraph 5.2.3.

Forecasting of density information can be used in numerous ways. When a driver receives a timely warning of a high-density area downstream, they can adapt the speed of their vehicle, increase their distance headway or change lane, in order to avoid the high-density area overall. Dülgar et al. (2019) studied the distance headways between the ego and the preceding vehicle in highD, in order to develop an **empirical method which estimated the local traffic density downstream** and to warn drivers in advance. Their methodology was based on unbiased time-exponential moving average, which enabled the transmission of density information upstream. As a result, the drivers received the high-density information before reaching the affected area and therefore, they were able take all necessary actions to minimise potential safety implications. The same authors concluded in Dülgar et al. (2019) that **spatiotemporal traffic dynamics differ in each lane** and therefore, future warning systems should consider lane-specific traffic state information.

The existence of multiple lanes (two or three per direction) and of some impeded traffic states in highD enables the investigation of inter-vehicle interactions. Diehl et al. (2019) created a model for vehicle interactions by representing the traffic scene as a graph of interacting vehicles. Comparing this graph neural network's (GNN) predictive performance to a calibrated Intelligent Driver Model (IDM), they achieved a 30% decrease in prediction error. In their research, Messaoud et al. (2019) tried to predict the future position of the ego vehicle based on its past positions and the respective positions of all surrounding vehicles. Their model used a long short-term memory (LSTM) encoder-decoder architecture able to model interactions to all surrounding vehicles. An innovative feature of this research was that it placed **emphasis on observations from influential vehicles**, in a sense it imitates human decision making, e.g. before and during a lane change it focused more on surrounding vehicles in the target lane than on preceding vehicles on the same lane. Their model performed better on the highD dataset in comparison to the NGSIM dataset, perhaps due to the trajectory inaccuracies mentioned earlier.

Another study that utilised information from the environment is that of Khakzar et al. (2020). The authors integrated a lane occupancy and risk map in an LSTM encoder-decoder architecture to predict the future trajectory of the ego vehicle based on the trajectories of adjacent vehicles. This model could prove useful in estimating the future trajectory of an automated vehicle under the coexistence of adjacent cars driven by humans. The authors believe that their study could be expanded to include more scenarios, such as driving on curves or roundabouts. Woo et al. (2020) state instead, that data-driven models for trajectory

prediction depend largely on the selected dataset. To overcome this limitation, they optimised the parameters of a General Motors' based car-following model in real time to predict the trajectory of a vehicle operated by a human driver, which is preceded by an autonomous vehicle. This real-time optimisation procedure using trajectories from highD was highly effective in predicting the future position of the following vehicles and took into account the operational characteristics of a driver as well.

Another potential area for applications using highD is car-following models, which describe the longitudinal dynamics of single vehicles with respect to the preceding vehicle in the same lane. Using data from highD and NGSIM I-80, Kurtc (2020) investigated if **multivehicle anticipation** i.e. a car-following model with two predecessors in interaction in the same lane decreases the calibration error of the models. For this purpose, she **optimised the parameters** of the intelligent driver model (IDM) and the optimal velocity model with two predecessors (OVM2). The optimal velocity model (OVM2) resulted in lower calibration errors, which confirmed that drivers consider several leading vehicles in front. This study was based on **platoon calibration; all trajectories are considered simultaneously during calibration** and there is no distinction by driver pairs. Finally, **the author did not take into account any reaction time**, because some previous studies have shown that reaction time has an insignificant influence when incorporated into a model (Kesting & Treiber, 2008). More details about this issue will be given later.

Lane change trajectories are usually modelled as sinusoidal curves, splines or polynomials (Yao et al., 2012). In Krajewski et al. (2018), the authors formulated an optimisation problem in order to identify the most influential parameters of a lane change with regards to the surrounding vehicles. These were the minimal distance headway (DHW), the time headway (THW), the time-to-collision (TTC) safety surrogate measure (SSM) and the gap size between the preceding and the following vehicle in the target lane. Specialised neural network architectures have been used to train models that generated manoeuvre trajectories. Using highly accurate trajectories from highD as input, Krajewski et al. (2018) showed that the trained networks were able to generate synthetic manoeuvre trajectories that were realistic and more accurate than fitted polynomial curves. The longitudinal and lateral velocity and acceleration of a vehicle are crucial in predicting a lane change. Mahajan et al. (2020) trained an LSTM model that **predicts a lane change up to three seconds earlier**, which allowed to **warn adjacent vehicles of a potential safety risk** resulting from a lane changing manoeuvre.

Moreover, naturalistic trajectories can be used in driving risk estimation. Each vehicle can theoretically emit a dynamic characteristic potential field around it, which drifts forward, when the speed of the preceding vehicle is higher than that of the follower, and backward in the opposite case. This potential field has advantages in comparison to other SSMs, such as the TTC, which suffer under limitations when the speed of the preceding and the following vehicles are equal. It is confirmed that the dynamic characteristic potential field is an important feature when estimating the driving style of the following vehicle (Woo, et al., 2018). A comprehensive methodology to estimate risk under different macroscopic and microscopic conditions can be found in Mahajan (2019), where **the role of traffic congestion on the severity of rear-end crashes is investigated**.

On the whole, the literature review of highD revealed that it is a versatile dataset, which can be used in many domains of traffic engineering and research. Among them, the most prevalent are forecasting of macroscopic characteristics of traffic flow, trajectory and manoeuvre generation based on neural networks, safety evaluation of vehicles driving on highways and calibration of parameters in car-following models. However, it is evident that the **paradigm is shifting from mathematical models** using physical properties of movement **towards a more data-oriented** approach, which is promoted by the ample availability of accurate data.

Table 2.1 Summary of studies related to highD

Study	Dataset	Objective	Method	Results
Krajewski, Bock, Kloecker, & Eckstein (2018)	highD	Identify the parameters that describe a lane-change trajectory	Solution of an optimisation problem	-
Krajewski, Moers, Neger, & Eckstein (2018)	highD	Generation of manoeuvre trajectories	Generative Adversarial Networks	Realistic lane change manoeuvre trajectories
Woo et al. (2018)	Simulator	Novel feature to determine driving style	Dynamic characteristic potential model	The proposed measure is more effective than TTC
Dülgar et al. (2019)	highD	Local downstream density estimation	Unbiased time-exponential moving average	Drivers receive a warning of upcoming high density traffic areas which improves driving safety
Dülgar et al. (2019)	highD	Lane-based local downstream density estimation	Unbiased time-exponential moving average	Drivers receive recommendation for lane-change
Diehl et al. (2019)	highD	Predict future traffic participant actions	Graph Neural Networks	-
Mahajan (2019)	highD	Prediction of lane-change intentions and crash risk estimation in different traffic states	Model-based clustering, LSTM	Traffic congestion and lane-changing in free flow associated with severity of rear-end crashes.
Mahajan, Katrakazas, & Antoniou (2020)	highD	Prediction of lane-change manoeuvres	Density-based clustering, LSTM	Lane changes can be predicted in real-time, 3 seconds in advance
Khakzar et al. (2020)	NGSIM highD	Predicting the trajectory of a vehicle under the influence of other vehicles	LSTM for time-series prediction	Independence of the methodology from a specific environment
Woo et al. (2020)	highD	Predict the future positions of the ego vehicle	Real-time optimisation of General Motors' model based on operational characteristics of driver	Improvement of trajectory prediction with respect to the change of locations
Kurtc (2020)	highD	Calibration of car-following models	Optimisation of IDM and OVM2 models	Drivers take into account more than one vehicle in front

2.3. Mathematical car-following models

The car-following model of Gipps (Gipps, 1981) is probably the most commonly used model in research and practice (Ciuffo et al., 2012). It belongs to the safe distance models, where the driver tries to maintain a distance to the preceding vehicle; if the leading vehicle stops, then the driver will be able to safely stop his or her vehicle after a time delay τ. The velocity of the following vehicle (n) is subject to three conditions: (1) the velocity never exceeds the driver's desired velocity, (2) the acceleration decreases as the velocity increases, until it becomes zero at the desired velocity, and (3) when intervehicle distances are short, the deceleration behaviour of the follower is largely determined by the safe distance to the leader $(n-1)$. The first two conditions are described by the first equation, whereas the last condition is given in the second equation, which have been adapted from Papathanasopoulou and Antoniou (2015):

$$u_n(t+\tau) = min \begin{cases} u_n(t) + 2.5 \cdot a_n \cdot \tau \cdot \left[1 - \frac{u_n(t)}{V_n}\right] \cdot \sqrt{0.025 + \frac{u_n(t)}{V_n}} \\ b_n \cdot \tau + \sqrt{b_n^2 \cdot \tau^2 - b_n \cdot [2 \cdot [x_{n-1}(t) - S_{n-1} - x_n(t)] - u_n(t) \cdot \tau - \frac{u_{n-1}^2(t)}{\hat{b}}]} \end{cases} \tag{2.1}$$

where

$u_n(t+\tau)$ is the velocity of the follower (n) after time delay τ (m/s),

τ is the reaction time (uniform for all vehicles) (s),

V_n is the desired velocity of the follower (n) (m/s),

a_n is the maximum desired acceleration of the driver in vehicle n (m/s^2),

b_n is the maximum desired deceleration of driver (n) for avoiding a crash (m/s^2),

$\hat{b}$ is the maximum breaking deceleration that the leader $(n-1)$ is willing to apply (m/s^2),

S_{n-1} is the length of the leading vehicle $(n-1)$ plus the safety distance that the driver in vehicle (n) is not willing to compromise (m),

$x_{n-1}(t)$ and $x_n(t)$ are the locations of the leader $(n-1)$ or the follower (n) respectively at time t (m) and finally,

$u_{n-1}(t)$ and $u_n(t)$ are the velocities of the leader $(n-1)$ or the follower (n) respectively at time t (m/s).

The parameter ranges can be retrieved from similar studies in the bibliography. Papathanasopoulou and Antoniou (2015) performed an extensive search and summarised the ranges of those parameters: reaction time $\tau = 0.4 - 2.8\ s$, maximum desired velocity of the follower $V_n = 10.4 - 29.6\ m/s$, maximum desired acceleration of the follower $a_n = 0.8 -$

$2.6\ m/s^2$, maximum desired deceleration $b_n = 1.6 - 5.2\ m/s^2$, maximum estimated deceleration of the leader $\hat{b} = 3.0 - 4.5\ m/s^2$ and distance $S_{n-1} = 5.6 - 7.5\ m$.

Another model that belongs to the category of safe distance models is that of Krauss (1998). An important application of Krauss' model can be found in SUMO (Simulation of Urban Mobility), which is an open-source package for microscopic traffic simulation (Alvarez Lopez, et al., 2018). The model formulation has been adapted from Kanagaraj et al. (2013) and Song et al. (2014); the safe speed is given by the formula:

$$v_{safe}(t) = v_l(t) + \frac{g(t) - v_l(t) \cdot T}{\frac{v_l(t) + v_f(t)}{2b} + T} \tag{2.2}$$

where

$v_l(t)$ is the velocity of the leading vehicle at time t (m/s),

$g(t)$ is the distance gap between the leading and the following vehicles at time t (m),

b is the maximum desired deceleration of the vehicle (m/s^2) and

T is the reaction time of the driver (s).

The desired velocity v_{des} is calculated as the minimum of the maximum allowed velocity v_{max}, the reachable velocity until the next time step due to the dynamic capabilities of the vehicle $v_f + a_{max} \cdot T$ and the safe speed:

$$v_{des}(t) = \min\left[v_{max},\ v_f(t) + a_{max} \cdot T, v_{safe}(t)\right] \tag{2.3}$$

where

a_{max} is the maximum acceleration of the following vehicle (m/s^2).

Due to the imperfection of the driver, the desired velocity is reduced by a random error ε (Krajzewicz et al., 2004). Therefore, the speed of the follower after reaction time T is given by:

$$v_f(t + T) = \max\left[0, rand[v_{des}(t) - \varepsilon \cdot a_{max}, v_{des}(t)]\right] \tag{2.4}$$

Krauss' car-following model is relatively new and less commonly applied than Gipps' model and a literature review does not provide sufficient information about the ranges of its parameters.

2.4. Multiple car-following models

Car-following models are characterised as "multiple" (or multivehicle or multi-anticipation) when the driver does not strictly follow the preceding vehicle only, but also **observes two or more vehicles ahead**. The concept of multiple car-following (MCF) models is not new; General Motors examined the influence of two or more vehicles on car-following behaviour as early as 1965, although they concluded that a driver considers only a single vehicle in front

(Bexelius, 1968). Their findings contradicted those of Buckley (1962), whose research supported that **drivers on highways generally observe at least two vehicles ahead**. Since then, some researchers have tried to explain interaction between more than two vehicles during car following.

A lot of focus has been put on maintaining traffic stability when modelling double car following. The optimal velocity model of Bando et al. (1995) was extended to include interaction between multiple vehicles (Lenz et al., 1999) and, later on, to consider the desired following distance and the effect of reaction time (Chen et al., 2016). Their numerical simulation results showed that MCF models can improve the stability of traffic flow and avoid the unrealistic modelling problem of negative velocity. Based on the full velocity difference model (Jiang et al., 2001), an improved MCF model was proposed that takes as input information from multiple preceding vehicles on the same lane (Guang-Han & Di-Hua, 2009). The results confirmed the influence of multiple preceding vehicles on traffic dynamics.

Recently, multiple car-following models have been **used in conjunction with driver-assistance systems**. Guo et al. (2017) used car-following data from an urban signalised main arterial to investigate the connection between the velocity fluctuations of many preceding vehicles and the ego vehicle and to create a cooperative adaptive cruise control strategy that optimises fuel consumption and emissions. Similarly, a multiple car-following model for simulation was proposed by Kuang et al. (2019), who believed that vehicle-to-vehicle communication is important for stability in traffic flow.

The literature review of multiple car-following models has led to three main observations. First, **research on MCF models is quite limited** in comparison to mathematical car-following models with one preceding vehicle, despite the fact that both concepts appeared around the same time. Second, existing double models consider exclusively vehicles on the same lane and ignore other vehicles or other vehicle types on adjacent lanes. Last but not least, to the best knowledge of the author of this study, **no attempt has been made in modelling multiple car-following behaviour using data-driven methods**. Therefore, there are many open opportunities, should the necessary data become available.

2.5. Data-driven longitudinal and lateral movement models

Car-following models have been studied for decades. Mathematical car-following models have been very successful in emulating the following behaviour under different circumstances and for different purposes each (Zheng & He, 2015). Recently, the availability of mass data from novel sources and the growing popularity of statistical learning has enabled data-driven approaches in car-following (Zhu et al., 2018).

Some researchers argue that even simple data-driven models can provide more reliable results than physical car-following models. Papathanasopoulou and Antoniou (2015) proposed a model suitable for integration into a microscopic simulation package based on locally estimated scatterplot smoothing (LOESS) regression. LOESS is a flexible, non-parametric regression technique that adapts a curve or a surface to a subset of data around one point, which is determined by a nearest neighbours algorithms. By optimising the parameters of LOESS (span

and degree), the authors were able to **achieve lower errors in comparison to a calibrated Gipps' model**. For this purpose, they used data collected from instrumented vehicle in real traffic conditions from Naples, Italy. A significant conclusion of this study is that **data-driven models allow the inclusion of new variables without the need to reformulate the functional forms of models** – although this might not promote our understanding of traffic flow theory. Later, the authors enriched their car-following model by adding information about the density in adjacent lanes (Papathanasopoulou & Antoniou, 2016). As trajectories from multilane roads were needed, their research utilised the enhanced NGSIM I-80 dataset (Montanino & Punzo, 2013). The updated model performed better than the previous one, with the authors noting that **higher speeds and longer distances between vehicles could imply a consistent driving behaviour leading to a lower fit error**. According to them, similar additions about adjacent vehicles to car-following models could result in robust models for autonomous driving.

Autonomous vehicles need to provide similar riding comfort and driving behaviour resembling manually driven cars, enabling natural interaction with surrounding drivers (Wang et al., 2016; Wei et al., 2010). Zhu et al. (2018) researched human-like autonomous car-following models based on deep reinforcement learning. The agents created in such networks combine reinforcement learning algorithms and neural networks, and the training of those agents happens by rewarding them when the gap between the real and the estimated speed or distance headway closes. The authors incorporated **elements of driving styles** and trained the agents using two thousand car-following periods from the Shanghai Naturalistic Driving Study (Zhu, Wang, Tarko, & Fang, 2018). The authors stated that their research provides better results, especially in comparison to studies based on regression techniques, because it adapts to different driving styles. Similarly, Fu et al. (2019) investigated driver response to cut-in events, when adjacent vehicles suddenly change lane and move closely in front of a vehicle, based on real driving experiments. Their intention was to incorporate the reaction of drivers during cut-ins into an autonomous vehicle car-following model. An important finding of this study is that drivers dislike cut-ins and therefore try to maintain lower gaps to the preceding vehicle, in order to discourage them.

Data-driven car-following models usually consider velocities and headways. Hao et al. (2018) considered more factors such as vehicle types, the number of vehicles in front of the preceding vehicle, their velocities and headways etc., based on rough set theory. Their model was able to simulate car-following from NGSIM I-80 more accurately than previous ones by utilising each time the most important rules from the decision set, while removing redundant variables. The **leader's vehicle type was also considered** in Wu et al. (2019), with the proposed model delivering better results than calibrated collision avoidance and Intelligent Driver (IDM) models. Finally, Aghabayk et al. (2013) considered the heterogeneity in the leader's vehicle type (passenger or truck) with encouraging results during passenger – truck vehicles interaction pairs.

Neural networks (NN) based microscopic car-following models were generally found to be accurate in predicting traffic dynamics (Zhou et al., 2017). However, the use of neural networks has not been limited to car-following; Liu et al. (2019) proposed a recurrent neural network, a class of artificial neural networks usually applied in speech recognition, which replicates the

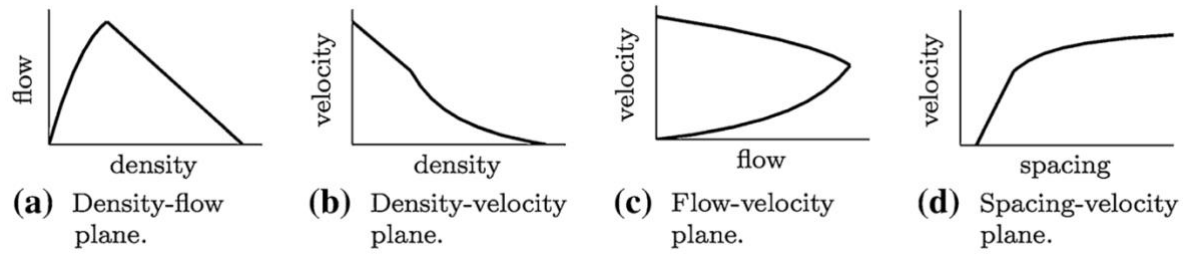
situation assessment and lane changing decision. The model looks back to the position of adjacent vehicles for two seconds before making the decision to change lane, effectively mimicking the memory of the driver, and this was a first-of-a-kind application in traffic research. Further, **neural networks can be combined with mathematical car-following** models in order to achieve synergies, as showcased by Yang et al. (2019). The authors collected data from a freeway with a drone, trained models based on neural networks, calibrated Gipps' car-following model and finally, they combined the predictions of the models using a linear function. The proposed Gipps-NN framework can offer results balanced between real and safe driving, which could prove useful in automated vehicles.

A current question in car-following modelling is how to model traffic when there is no clear lane separation. **Weak lane-discipline or even lane-less traffic** is prevalent in many parts of the developing world, especially in countries with high percentage of powered two-wheelers, and it probably is a future-proof concept for increasing existing roadway capacity even in the developing world (Papageorgiou et al., 2019). Swarm-like models and modified pedestrian simulation can be used to model weak lane-discipline. Papathanasopoulou and Antoniou (2018) used the innovative concept of virtual lanes with variable width to simplify the problem of lane-less traffic. Their study was based on mixed traffic trajectory data collected from Chennai, India (Kanagaraj et al., 2015).

Last but not least, a wide range of methodologies have been trialled when car-following models are concerned. Dabiri and Abbas (2018) applied the gradient boosting of regression tree algorithm to evaluate vehicle trajectory datasets. (Zhou, Fu, Wang, & Zhang, 2020) split the drivers into conservative and aggressive by k-means and then trained the car-following models, whereas Li C. et al. (2016) transferred characteristic potential fields, used for navigation in robotics, into car-following. **Unfortunately, the vast majority of traffic flow studies in this chapter use the original NGSIM dataset** i.e. with highly inaccurate trajectories, before being processed and evaluated by Punzo et al. (2011). A review of trajectory-based traffic flow studies can be found in Li L. et al. (2020).

2.6. Traffic phase identification

Classical traffic flow theories recognise the existence of two phases: *free* and *congested* flow. In the free flow phase, traffic levels are far below capacity, which means drivers can choose the speed they want and change lanes easily. In congested phase, it is not possible to choose one's desired speed, drivers are forced to drive as part of a traffic stream and lane-changes are less frequent (Sun & Zhou, 2005). Edie (1961) was the first to propose a two-phase model, whereas numerous research papers suggest the existence of more than two phases e.g. additionally the synchronized congested phase (Kerner & Rehborn, 1996; Kerner, 2004), or even more. No matter the number of phases, a common observation of most traffic flow studies is that vehicles in congested phase impede one another and therefore, the movement of one vehicle depends on preceding, adjacent or following vehicles on the same road.



(a) Density-flow plane. **(b)** Density-velocity plane. **(c)** Flow-velocity plane. **(d)** Spacing-velocity plane.

Figure 2.1 Fundamental relations of traffic flow - adapted from van Wageningen-Kessels et al. (2015)

Multiple phases facilitate more accurate modelling of traffic flow under different conditions. As a consequence, numerous efforts have been made to approximate macroscopic models with multiple phases. **A major difficulty associated with this task is determining the breakpoints of the phases**. Sun and Zhou (2005) proposed the use of k-means cluster analysis for solving this segmentation problem. In Antoniou et al. (2013), the authors clustered density, speed and flow data from inductive loop detectors using density-based models. They observed that **a smaller number of clusters resulted into parsimonious traffic phases** i.e. they have better interpretability. Later, the authors developed a traffic phase estimation approach for newly acquired measurements. Mahajan (2019) clustered aggregated traffic flow properties from highD using Gaussian mixture models (GMM) into free and congested phases, aiming to estimate crash risk during lane-changes on highways, and identified 70 km/h as the borderline speed between free-flowing and congested traffic.

Thus far, the approaches for estimating the breakpoint determination problem have been very diverse. Earlier approaches focused on estimating analytically one point, where the transition between two linear models happens, with the first model representing the congested and the second the free phase (Quandt, 1958). More complicated studies indicate that driver-vehicle combinations play a significant role in traffic flow – density relations. Kockelman (2001) combined measurements from loop detectors and data from a household survey to model traffic flow – density relations for different groups of vehicle types and genders. Finally, Koutsopoulos and Farah (2012) looked at vehicle pairs separately, in order to identify their regime (e.g. car-following, free-flow or emergency braking) and ultimately, to model the acceleration of the following vehicle in conjunction to the leader.

2.7. Summary and research gaps

The above literature review focused on the main topics that will be addressed in this study. Unmanned aerial vehicles, a novelty in transportation research, seem to open up many opportunities for accurate and cost-effective data acquisitions. The highD dataset is a recent example, which was captured through aerial video recording and has been used already extensively in research. Among those research topics are also car-following models, which have been a favourite subject for many years. Data-driven car-following models are a relatively new field, which is currently in "growth mode". Another topic that will be addressed in this study is model-based traffic phase identification.

All in all, car-following behaviour is a well-studied field in transportation engineering. However, some gaps have already been identified in this literature review:

1. In order to model car-following dynamics given a dataset of naturalistic trajectories, it is necessary to find the traffic states that intensify interaction between drivers.
2. The car-following dynamics in highD have only been studied through mathematical car-following models. It has to be investigated if data-driven models are able to deliver better results.
3. Two of the models applied most often are Gipps' model and the intelligent driver model with some adjustments. Newer models, such as Krauss' car-following model, can be analysed and put in comparison with more traditional and data-driven models.
4. It is necessary to perform a thorough investigation of the kinematic characteristics that can improve the performance of car-following models and if alterations in assumed reaction time are able to affect their performance.

Part II: Data Collection, Methodology and Analysis

3. Data collection

This chapter gives some general information about the highD dataset and then explains the structure of the included files. In its initial form, highD comprises of several files in text format, which are presented separately in data collection but were combined in data analysis and model development.

3.1. General information about the dataset

This study utilises the highD naturalistic trajectory dataset (Krajewski et al., 2018). HighD is a work of the Institute of Automotive Engineering of RWTH Aachen University and can be downloaded freely for non-commercial purposes. Its format facilitates research in validation of safety features for automated or autonomous vehicles, but it can be used from a wider perspective in other traffic engineering applications as well.

The use of drones equipped with high-resolution cameras makes it possible to measure every vehicle's movement with very high accuracy. The dataset includes 110000 vehicle trajectories from 60 recordings in six different locations around Cologne, Germany. Each recording has an average duration of 17 minutes, which results in a total duration of 16.5 hours of vehicle trajectories at a frame frequency of 25 Hz. The observed highway segments have a length of around 420 m and each vehicle is visible for a median duration of 13.6 s. The quadcopter drone hovered directly next to the highway at an altitude of around 100 m during sunny and windless days, from around 08:00 in the morning until 17:00 in the afternoon. The drone was equipped with motor-based flight stabilisation and gimbal-based camera stabilisation mechanisms. Thereafter, the videos were processed by software for stabilisation and the frames were rotated, in order to bring the lane markings on a horizontal position (Krajewski et al., 2018).

Each one of the 60 recordings resulted in three files, which means that the highD dataset consists of 180 files in total (excluding the aerial photographs of each highway area). The first file, also known as a Track, contains the analytical trajectories of the vehicles in one recording and some metadata at a frame frequency of 25 Hz. The second file is called Track Meta Information file and contains summary statistics per vehicle in each recording. The third one, called Recording Meta Information file, contains a general overview and some relevant statistics per recording.

3.2. Track files

The Track files contain detailed attributes about the movement of individual vehicles and the traffic scene surrounding a vehicle at each frame. Due to the vast number of individual vehicles, Krajewski et al. (2018) used state-of-the-art computer vision for trajectory extraction. Later on, track post-processing algorithms were applied in order to reconstruct smooth time series of positions, velocities and accelerations in both x and y directions. Their trajectory reconstruction algorithms resulted in smooth trajectories that require no further corrections, as opposed to the NGSIM dataset, which has been often criticised for its inferior accuracy. Further manoeuvre classification algorithms were used to determine if interaction between adjacent vehicles exists and, in that case, to calculate distance headway (DHW), time headway (THW) and time-to-collision (TTC). The list of attributes in a Track file is shown on Table 3.1.

Table 3.1 Attributes in a Track file from highD

Name	Description	Unit
frame	current frame	-
id	vehicle's id	-
x	x position of the upper left corner of the vehicle's bounding box	m
y	y position of the upper left corner of the vehicle's bounding box	m
width	width of the bounding box of the vehicle (vehicle's length)	m
height	height of the bounding box of the vehicle (vehicle's width)	m
xVelocity	longitudinal velocity in the image coordinate system	m/s
yVelocity	lateral velocity in the image coordinate system	m/s
xAcceleration	longitudinal acceleration in the image coordinate system	m/s
yAcceleration	lateral acceleration in the image coordinate system	m/s
frontSightDistance	distance to the end of the recorded highway section in driving direction from the vehicle's centre	m
backSightDistance	distance to the end of the recorded highway section in the opposite driving direction from the vehicle's centre	m
dhw	distance headway (equal to 0 when no preceding vehicle exists)	m
thw	time headway (equal to 0 when no preceding vehicle exists)	s
ttc	time-to-collision (equal to 0 when no preceding vehicle exists)	s
precedingXVelocity	longitudinal velocity of the preceding in the image coordinate system (equal to 0 when no preceding vehicle exists)	m/s
precedingId	id of the preceding vehicle in the same lane (equal to 0 when no preceding vehicle exists)	-
followingId	id of the following vehicle in the same lane (equal to 0 when no following vehicle exists)	-

leftPrecedingId	id of the preceding vehicle on the adjacent lane on the left in the direction of travel (equal to 0 when no vehicle exists preceding on the left)	-
leftAlongsideId	id of the adjacent vehicle on the adjacent lane on the left in the direction of travel (equal to 0 when no vehicle exists alongside on the left)	-
leftFollowingId	id of the following vehicle on the adjacent lane on the left in the direction of travel (equal to 0 when no vehicle exists following on the left)	-
rightPrecedingId	id of the preceding vehicle on the adjacent lane on the right in the direction of travel (equal to 0 when no vehicle exists preceding on the right)	-
rightAlongsideId	id of the adjacent vehicle on the adjacent lane on the right in the direction of travel (equal to 0 when no vehicle exists alongside on the right)	-
rightFollowingId	id of the following vehicle on the adjacent lane on the right in the direction of travel (equal to 0 when no vehicle exists following on the right)	-
laneId	id of the lane, starts at 1 and is assigned in ascending order (see figure X)	-

Additionally, an aerial photograph of each highway area is provided as part of the dataset (Figure 3.1). Three out of the six areas are highways with three lanes per direction, two areas are two-lane highways and one area has four lanes on the one direction and three lanes on the other (Mahajan, 2019). Three-lane highways will be used during training and testing of the car-following models, whereas the two-lane areas will be used for validation. The area with four lanes on the one side and three on the other will be treated as a three-lane highway.

Figure 3.1 A three-lane highway from highD (Track 37) - Krajewski et al. (2018)

The coordinate system corresponds to the system of the video recordings. The origin of the system is situated in the upper left corner of each aerial photograph (Figure 3.2). The x-axis is horizontal and grows to the right, which means that vehicles travelling on lanes 2, 3 and 4 (upper) on three-lane highways (lanes 2 and 3 on two-lane highways) have negative velocities. Vehicles travelling on lanes 6, 7 and 8 (lower) on three-lane highways (lanes 5 and 6 on two-lane highways) have positive velocities. Lanes 1, 5 and 9 corresponds to the hard-shoulders and were not recorded (lanes 1, 4 and 7 on two-lane highways). The y-axis grows downwards, which means that the y-coordinate increases when the lane id of a vehicle increases too. Vehicles travelling on the hard shoulder were not considered in this study. However, as the direction of travel was not considered as input in any of the models, signs of velocities and

accelerations of vehicle travelling to the left (lanes 2, 3 and 4 on three-lane highways – lanes 2 and 3 on two-lane highways) were inverted, so that all velocities become positive and all accelerations or decelerations positive or negative accordingly.

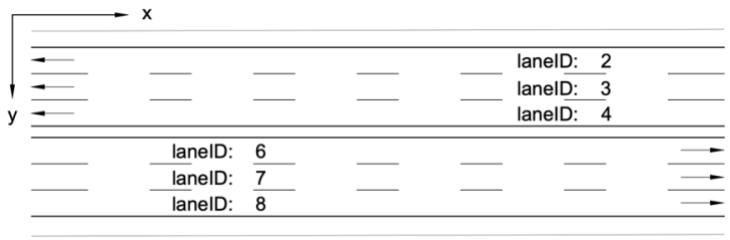


Figure 3.2 Coordinate system and lane numbering in highD

3.3. Track Meta Information files

This file type contains summarised information about each vehicle in one Track. This information includes the id of the vehicle, its physical properties such as the length, the width and its class, as well as basic statistics such as the minimum, mean and maximum velocity and number of lane changes. The purpose of these meta files is to allow filtering by class, direction of travel, average velocity etc. The list of attributes in a Track Meta Information file can be seen on Table 3.2.

Table 3.2 Attributes in a Track Meta Information file from highD

Name	Description	Unit
id	vehicle's id	-
width	width of the bounding box of the vehicle (vehicle's length)	m
height	height of the bounding box of the vehicle (vehicle's width)	m
initialFrame	current frame in which the vehicle trajectory starts	-
finalFrame	current frame in which the vehicle trajectory ends	-
numFrames	total lifetime of the track as number of frames	-
class	vehicle class of the tracked vehicle (car or truck)	-
drivingDirection	driving direction of the vehicle, either 1 for the left direction (upper lanes) or 2 for the right direction (lower lanes)	-
traveledDistance	distance covered by the vehicle	m
minXVelocity	minimum velocity in driving direction (longitudinal)	m/s
maxXVelocity	maximum velocity in driving direction (longitudinal)	m/s
meanXVelocity	mean velocity in driving direction (longitudinal)	m/s
minDHW	minimum distance headway: this value is set to -1 if no preceding vehicle exists	m
minTHW	minimum time headway: this value is set to -1 if no preceding vehicle exists	s
minTTC	minimum time-to-collision: his value is set to -1, if no preceding vehicle or valid TTC exists	s
numLaneChanges	number of lane changes detected by changing laneId	-

3.4. Recording Meta Information files

This file type contains general information about each recording. That information includes, among others, data about the location of each recording, its duration, the time, day and month it took place, the number of cars and trucks in the respective Track file and others. Although these files were not part of this study, they are however mentioned for the sake of completeness.

Table 3.3 Attributes in a Recording Meta Information file from highD

Name	Description	Unit
id	id of the recording. A unique id corresponds to each recording	-
frameRate	frame rate of the video recording	Hz
locationId	id of the recording location. Six different locations exist in the dataset	-
speedLimit	speed limit of the driving lanes	m/s
month	month of the recording	-
weekDay	weekday of the recording	-
startTime	time at which the recording started	hh:mm
duration	duration of the recording	s
totalDrivenDistance	total driven distance of all vehicles in the Track	m
totalDrivenTime	total driven time of all vehicles in the Track	s
numVehicles	number of vehicles tracked including cars and trucks	-
numCars	number of cars tracked	-
numTrucks	number of trucks tracked	-
upperLaneMarkings	exact y positions of the upper lane markings	m
lowerLaneMarkings	exact y positions of the lower lane markings	m

Regarding the speed limits, two locations were found to have speed limits at 120 km/h, one location at 130 km/h and three locations had no speed limit at all. In all cases, the highway areas were selected such that all lanes have the same speed limit when applicable (Mahajan, 2019).

4. Methodology

This chapter explains briefly the methodology that is implemented in this study. The methodology can be divided roughly into three stages: traffic phase identification, model development and model validation. Flow charts at the beginning of each section provide a graphical explanation of the procedures. The general framework of the methodology is shown on Figure 4.1.

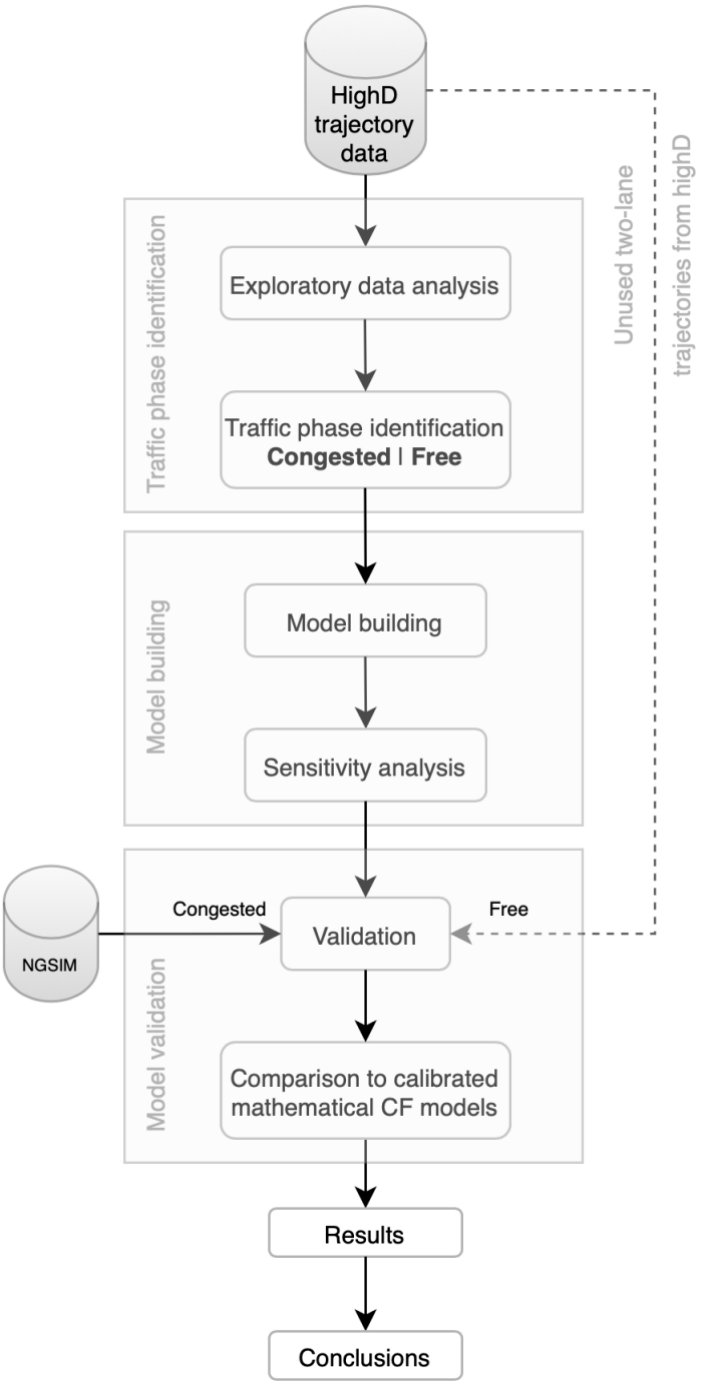


Figure 4.1 Overview of the methodology

4.1. Double car-following scenario

In mathematical car-following models, the driver of the ego (following) vehicle adjusts the velocity of his or her vehicle according to the movement of the preceding vehicle, so as to keep sufficient spacing to the preceding vehicle. In a double car-following scenario, the driver of the ego vehicle pays attention not only to the preceding vehicle, but also to the movement of the vehicle in front of the preceding (leading). Consequently, in order to train a new double car-following model, it is essential to know the dynamic properties (e.g., velocity, acceleration, distance headway etc.) of three successive vehicles: **the ego, the preceding and the leading** (Figure 4.2).

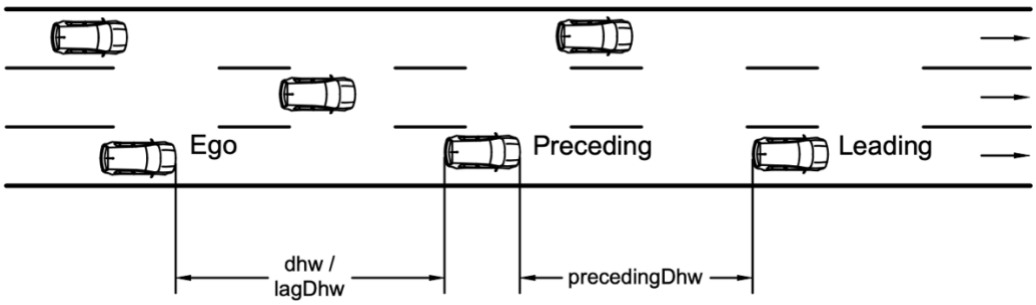


Figure 4.2 Double car-following scenario

The goal is to estimate the velocity of the ego vehicle in the present, based on past data from the preceding and the leading vehicles. Hence, some of the variables of the ego vehicle refer to the present and some to the past, while the variables of the preceding and the leading vehicles refer all to the past. This concept emulates the delay in the actions of the driver due to their reaction time and will be explained further in the following section.

4.2. Reaction time emulation procedure

Drivers do not react to a traffic event instantaneously, but rather they need time to perceive the event, process the stimulus, prepare a response and finally execute the necessary actions (Basak et al., 2013). In microscopic traffic simulation models, these delays are modelled cumulatively as a reaction time between the actions of the leader and the respective, time-delayed actions of the follower. The reaction time plays a major role in the output of car-following models and most often gets treated as a calibration parameter, in order to find the value that minimises the error of the simulated data in comparison to the observed data. Although reaction times differ from driver to driver according to their age, experience, physical and perceptual capabilities, they are usually regarded uniformly, thus introducing additional errors to a car-following model. However, as the ground truth about personalised reaction times is unknown, especially in a dataset recorded by drones with the drivers being unaware of the recording, generalising and trialling different reaction times is probably still the best strategy.

In the Track files of highD each line refers to a frame, with 25 frames making a second (25 Hz). Let t be a given frame and τ or tau be the uniform reaction time of all drivers. In order

to find the time-lagged dynamic properties of each vehicle at frame t, each Track is merged with itself using the following conditions:

$$[track, id, frame\ t - \tau] = [track, id, frame\ t'] \quad (4.1)$$

which means that the observation of one vehicle at frame $t - \tau$ is matched with its observations at frame t'. For example, a vehicle rides on a highway at frame $t = 68$. Assuming a reaction time of 1.2 s, equal to 30 frames, its velocity was decided based on the stimulus from its environment at frame $t - \tau = 68 - 30 = 38$. Therefore, to find its time-lagged dynamic properties, its actual dynamic properties have to be merged with those from frame $t' = 38$. As expected, this procedure results in data loss, because negative frames have to be removed from the merging process. The time-lagged metavariables that arise from this process are shown on Table 4.1.

Table 4.1 Time-lagged variables of the ego vehicle

Name	Description	Unit
precedingFrame	t - τ frame	-
lagXVelocity	longitudinal velocity of the ego vehicle at t - τ	m/s
lagXAcceleration	longitudinal acceleration of the ego vehicle at t - τ	m/s
lagDhw	distance headway at $t - \tau$ (between the ego and the preceding vehicle)	m
lagTtc	time-to-collision at $t - \tau$ (between the ego and the preceding vehicle)	s
class	vehicle class of the ego vehicle (car or truck)	

The same procedure applies also when finding the time-lagged properties of the preceding vehicle. In this case, the merging conditions become as following:

$$[track, precedingId, frame\ t - \tau] = [track, id, frame\ t'] \quad (4.2)$$

where the dynamic properties of the preceding vehicle $(precedingId)$ are found by searching on the original Tracks data as if the preceding vehicle was a standalone vehicle (id). The time-lagged variables of the preceding vehicle are included on Table 4.2.

Table 4.2 Time-lagged variables of the preceding vehicle

Name	Description	Unit
precedingId	preceding vehicle's id	-
precedingXVelocity	longitudinal velocity of the preceding vehicle at t - τ	m/s
precedingXAcceleration	longitudinal acceleration of the preceding vehicle at t - τ	m/s
precedingDhw	distance headway at $t - \tau$ (between the preceding and the leading vehicle)	m

precedingTtc	time-to-collision at $t - \tau$ (between the preceding and the leading vehicle)	s
precedingClass	vehicle class of the preceding vehicle (car or truck)	-

The preceding of the preceding is the leading vehicle. Therefore, the $precedingId$ column of the preceding vehicle is renamed into leadingId and the merging equality is now the following:

$$[track, leadingId, frame\ t - \tau] = [track, id, frame\ t'] \tag{4.3}$$

In this way, the dynamic properties of the ego, preceding and leading vehicles at frame $t - \tau$ are found. Thereafter, a data reduction technique is applied, which keeps only the observations that are integer multiples of the reaction time in the dataset. The whole process will be referred to as "reaction time emulation procedure" in this study. As expected, larger reaction times result in datasets of smaller size and vice versa.

Table 4.3 Time-lagged variables of the leading vehicle

Name	**Description**	**Unit**
leadingId	leading vehicle's id	-
leadingXVelocity	longitudinal velocity of the leading vehicle at t - τ	m/s
leadingXAcceleration	longitudinal acceleration of the leading vehicle at t - τ	m/s

Two additional meta-variables are calculated using the time-lagged velocities of ego, preceding and leading vehicles. The first one, $deltaPrecedingXVelocity$, is the difference of velocity of the preceding minus the velocity of the ego vehicle. The second one, $deltaLeadingXVelocity$, is the difference of the velocity of the leader minus the velocity of the ego vehicle.

The final dataset can be used for training and testing machine learning models, including the reaction time a priori as dataset formatting criterion, rather than as an independent variable. The same procedure is followed when formatting the re-extracted NGSIM dataset of Coifman and Li (2017), although the frame rate of the video extraction was 10 Hz in this case.

4.3. Traffic phase identification

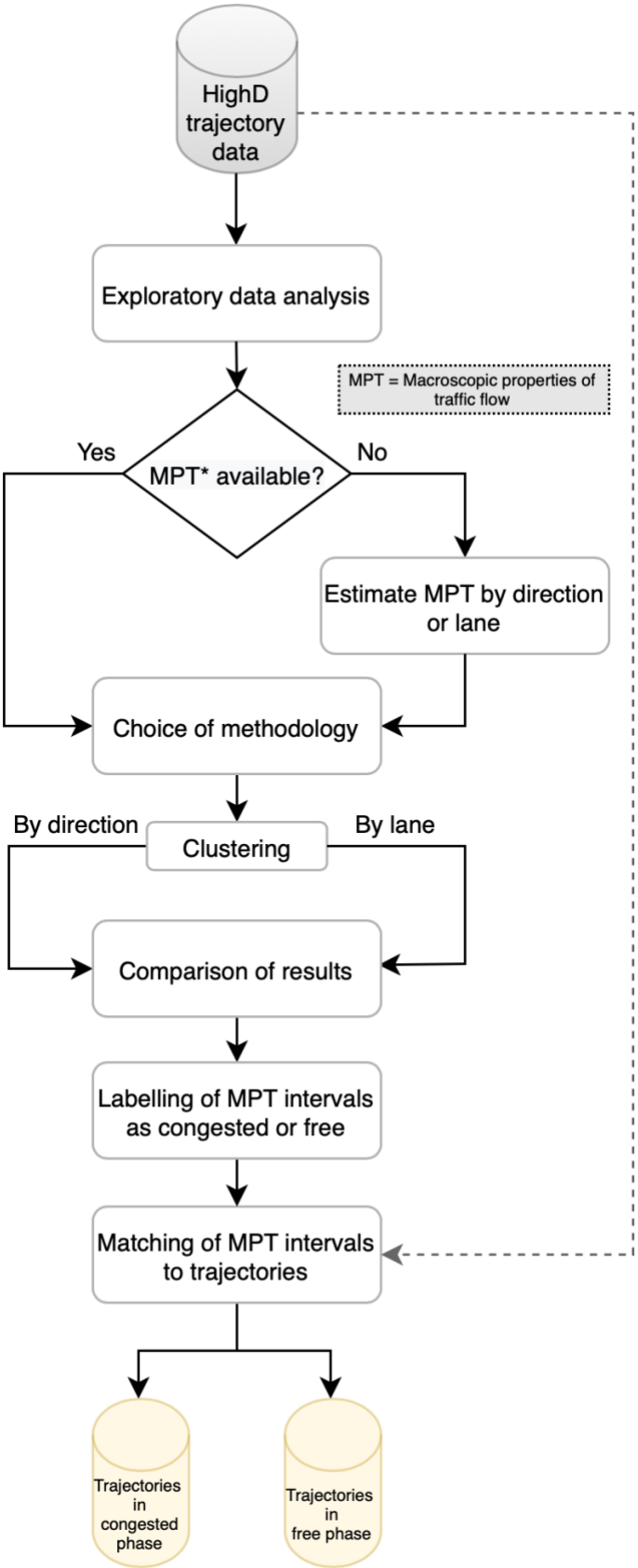


Figure 4.3 Workflow of traffic phase identification

4.3.1. Data aggregation

Traffic flow, density and average velocity, also known as the macroscopic properties of traffic flow, are interconnected to each other. In real world, the most common way to measure these properties is via inductive loop detectors. In this study, *virtual* detectors (Figure 5.1) have been placed in the middle of each section, in order to observe the macroscopic properties by aggregating and averaging spatially and temporally disaggregate data, as they have been extracted from video footage in the form of unique trajectories. Although the following definitions refer to per section macroscopic properties, they can also be applied per lane.

The number of vehicles passing a cross-section of a road or a lane during a time interval is the traffic flow. Road or lane traffic flow is given by the following equation:

$$q = \frac{N}{T} \tag{4.4}$$

where N is the number of cars passing the road or lane cross-section and T is the respective time interval. Traffic flow is usually given in vehicles per hour (veh/h). Similarly, traffic density is the number of vehicles on a particular lane or road at a given time instant t divided by the length of the lane or the road.

$$k_t = \frac{n}{L} \tag{4.5}$$

where n is the number of vehicles on a road or lane section and L is the length of this section. Density is usually given in vehicles per kilometre (veh/km). Over a time, interval, the average traffic density can be calculated using equation (4.6), where T is the duration of the time interval in seconds or frames and k_t is the traffic density on second or frame t.

$$\bar{k} = \frac{\sum_{t=1}^{T} k_t}{T} \tag{4.6}$$

Finally, the average velocity on a road or a lane is calculated by dividing the sum of the velocities of all vehicles on that road or lane by the number of vehicles on that section or lane on time instant t.

$$v_t = \frac{\sum_{i=1}^{n} v_i}{n} \tag{4.7}$$

where v_i is the velocity of vehicle i and n is the total number of vehicles on the lane or road at time instant t. Velocity can in given in metres per second (m/s) or kilometres per hour (km/h). For a time interval, the above formula becomes:

$$\bar{v} = \frac{\sum_{t=1}^{T} v_t}{T} \tag{4.8}$$

4.3.2. Clustering of traffic states

During a time interval, traffic can be described by the macroscopic properties of traffic flow, which constitute a unique state L_T.

$$(\bar{v}, \bar{k}\,, q)_T \rightarrow L_T$$

Plotting different traffic states on a common density – velocity chart results in the fundamental diagram of traffic flow (FD). Traffic states can be grouped into homogeneous clusters with similar properties. Literature suggests that the most effective approach is a low number of different clusters, also known as traffic phases (see literature review); clustering the traffic states into a free phase and a congested phase solves the speed breakpoint determination problem.

Unsupervised learning refers to the extraction of undetected patterns from data with no pre-existing labelling, or else, from data where the outcome is unknown (Bruce & Bruce, 2017). One of the goals of unsupervised learning is clustering, which aims to classify data into meaningful groups with similar properties. Based on velocity, density and flow, Gaussian mixture models clustering aims to distribute individual observations into a predetermined number of Gaussian distributions.

4.3.3. Gaussian mixture models clustering

Based on content adapted from Reynolds (2009), McGonagle et al. (n.d.) and Maklin (2019).

Gaussian mixture models (GMM) are probabilistic models of M weighted Gaussian distributions of subpopulations within an overall population, given by the equation:

$$p(x|\lambda) = \sum_{i=1}^{M} w_i g(\mathrm{x}|\mu_i, \Sigma_i) \tag{4.9}$$

where x is a d-dimensional continuous vector (e.g. containing density, velocity and flow data at different time intervals), $g(\mathrm{x}|\mu_i, \Sigma_i), i = 1, \dots, M$ are the density functions of the Gaussian subpopulations and $w_i, i = 1, \dots, M$ are the mixture weights with $\sum_{i=1}^{M} w_i = 1$. Then, the Gaussian density function of each subpopulation is given by:

$$g(\mathrm{x}|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathrm{x} - \mu_i)^T \Sigma_i^{-1} (\mathrm{x} - \mu_i)\right) \tag{4.10}$$

where μ_i is the mean vector and Σ_i the covariance matrix of each subpopulation. These parameters can be summarised as:

$$\lambda = \{w_i, \mu_i, \Sigma_i\}, i = 1, \dots, M$$

The parameters of λ are estimated using maximum likelihood (ML) estimation, which aims to maximise the likelihood of the GMM given the training data. Assuming independence between a sequence of T training vectors $X = \{x_1, \dots, x_T\}$, the GMM likelihood can be written as:

$$p(X|\lambda) = \prod_{t=1}^{T} p(x_t|\lambda) \tag{4.11}$$

which is a non-linear function of λ and therefore, direct optimisation is not possible. Using a special case of expectation-maximisation (EM) algorithm, the parameters of the ML can be estimated (Dempster et al., 1977).

The main idea of EM is, beginning with a randomly assigned initial model λ, to estimate a new model $\bar{\lambda}$, so that $p(X|\bar{\lambda}) \geq p(X|\lambda)$, which then becomes the benchmark model for the next iteration, and so on. On each iteration, the mixture weights, means and variances are calculated respectively as:

$$\bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} p(i|x_t, \lambda) \tag{4.12}$$

$$\bar{\mu}_i = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) x_t}{\sum_{t=1}^{T} p(i|x_t, \lambda)} \tag{4.13}$$

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i|x_t, \lambda) x_t^2}{\sum_{t=1}^{T} p(i|x_t, \lambda)} - \bar{\mu}_i^2 \tag{4.14}$$

The a posteriori probability of subpopulation i is given by:

$$p(i|x_t, \lambda) = \frac{w_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^{M} w_k g(x_t|\mu_k, \Sigma_k)} \tag{4.15}$$

The dimensions of x and μ_i can be one or bigger than one.

GMMs are a soft clustering algorithm, which means that each observation belongs to every cluster with varying levels of membership. Different models can be compared using the Bayesian Information Criterion (BIC). According to formula that is used each time to calculate the BIC, the metric either has to be minimised or maximised. The **mclust** package in R (Scrucca et al., 2016) uses the following formula:

$$BIC = 2 \cdot \ln\big(L(\lambda|x)\big) + k \cdot \ln(n) \tag{4.16}$$

which means that the metric has to be maximised in this case. The likelihood $L(\theta|x)$ is the probability that a particular outcome x is observed, given that the true values of the parameters are λ. Finally, k is the number of clusters in the model and n is the number of observations.

4.4. Model development

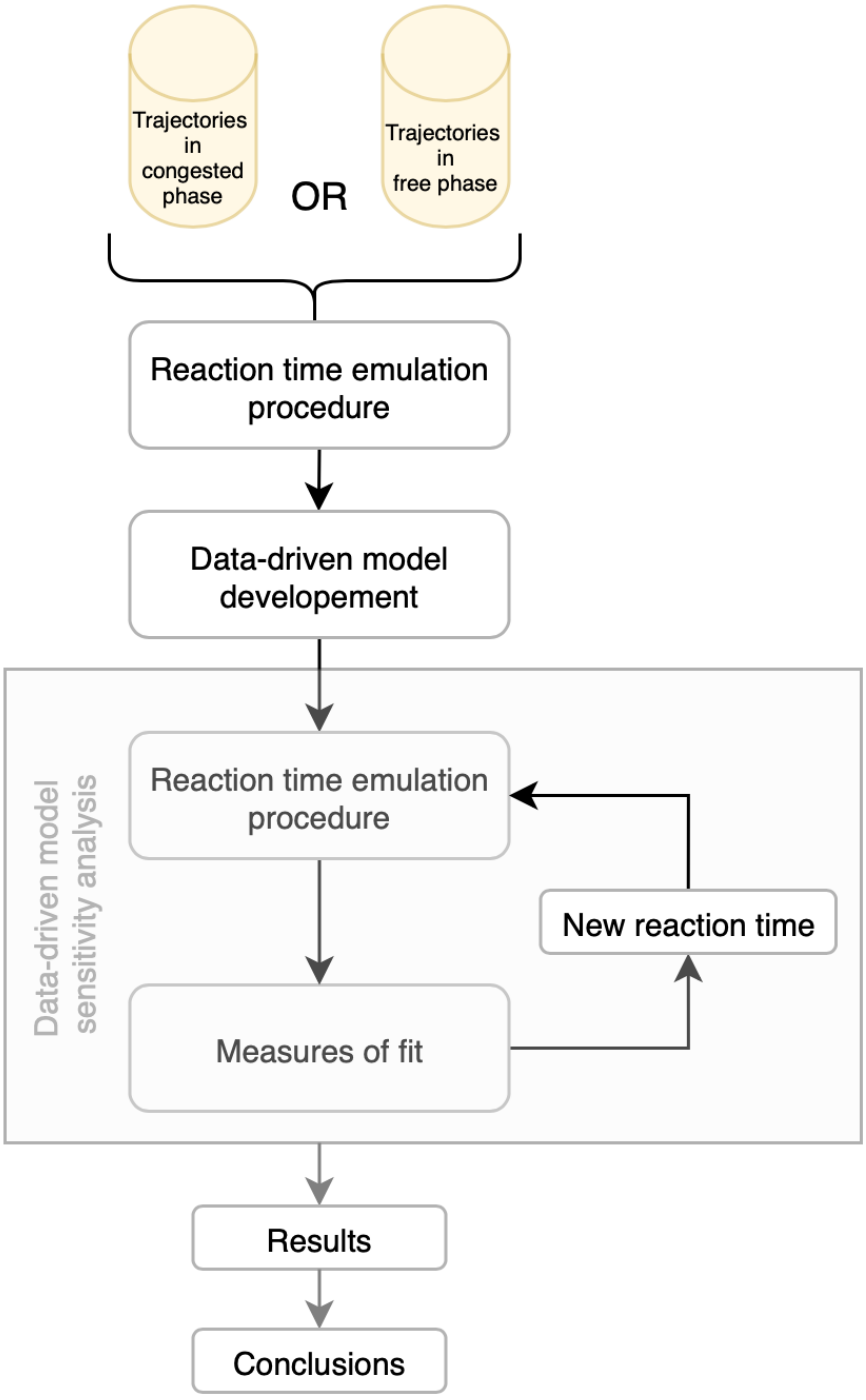


Figure 4.4 Workflow of model development

4.4.1. Linear regression

Based on content adapted from Van de Geer (2005), Washington et al. (2011), and Hastie and Tibshirani (2020).

Linear regression is a linear approach in modelling relationships between a continuous dependent variable and one or more independent variables. A simple algebraic expression of a multiple linear regression model is given by:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_P X_{pi} + \varepsilon_i \tag{4.17}$$

In this expression, Y_i is the dependent variable, β_0 is the intercept (constant) term, $\beta_1, \beta_2, \dots, \beta_p$ are the regression coefficients, $X_1, X_2, \dots, X_p$ are the independent variables and ε_i are the disturbance terms. The subscript i corresponds to the observation, with $i = 1,2, \dots, n$.

A regression coefficient $\beta_j, j = 1,2, \dots, p$ estimates the expected change in Y per unit change in X_j, should all other independent variables be held constant. Given the estimates $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$, then predictions can be made using:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \hat{\beta}_2 X_{2i} + \cdots + \hat{\beta}_P X_{pi} \tag{4.18}$$

Two popular functions for optimising the above function are ordinary least squares (OLS) and maximum likelihood estimation (MLE). In OLS, the estimates of $\beta_0, \beta_1, \dots, \beta_p$ are the values that minimise the residual sum of squares:

$$L_{OLS}(\hat{\beta}) = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 = \sum_{i=1}^{n} (Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1i} - \hat{\beta}_2 X_{2i} - \cdots - \hat{\beta}_P X_{pi})^2 \tag{4.19}$$

The main assumptions of regression analysis are the following:

i. The regression model is linear in parameters
ii. The mean of disturbances is zero
iii. Normality of disturbances
iv. Homoscedasticity or equal variance of disturbances
v. Nonautocorrelation of disturbances
vi. No correlation between the disturbances and the independent variables
vii. No multicollinearity between independent variables

Multicollinearity exists when independent variables are found to be highly correlated with each other. This phenomenon can be detected through correlation plots of all independent variables or through the Variance Inflation Factor (VIF):

$$VIF_i = \frac{1}{1 - R_i^2} \tag{4.20}$$

where R_i^2 is the pairwise coefficient of determination between two independent variables. Empirical studies in applied statistics suggest that VIF values higher than 4 suggest further investigation and values higher than 10 indicate the existence of multicollinearity (Simon et al., 2018). Strategies for avoiding multicollinearity include the removal of one or more collinear variables, Ridge and Lasso regression and finally, even doing nothing, as sometimes

it is necessary to measure the effects of the collinear variables in a model or the inclusion of those variables contributes to the interpretation of a model (Washington et al., 2011).

4.4.2. Regression goodness-of-fit statistics

The sum of square errors (disturbances) depicts the variation of the dependent variable that cannot be explained by the model and is given by:

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{4.21}$$

The regression sum of squares describes how well a regression model represents the modelled data and is given by:

$$SSR = \sum_{i=1}^{n} \left(\hat{Y}_i - \bar{Y}\right)^2 \tag{4.22}$$

The total sum of squares is the variation of the values of a dependent variable from the sample mean of the dependent variable:

$$SST = \sum_{i=1}^{n} (Y_i - \bar{Y})^2 \tag{4.23}$$

It can be shown algebraically that $SST = SSR + SSE$. Then, the coefficient of determination *R-squared* is defined as:

$$R^2 = \frac{SST - SSE}{SST} = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} \tag{4.24}$$

The R^2 is the proportion of total variance explained by X. If $R^2 = 1$, all of variance is explained, whereas if $R^2 = 0$, then the dependent and independent variables are uncorrelated. To penalise the inclusion of more variables in a linear model, the adjusted R^2 can be used, which adjusts for the number of independent variables p in a model relative to the number of data points n (Theil, 1961).

$$R^2_{adjusted} = 1 - (1 - R^2)\frac{n-1}{n-p-1} \tag{4.25}$$

4.4.3. Ridge, Lasso and Elastic Net regression

Generally, linear regression models should be easily interpretable and avoid overfitting. This can be achieved with various techniques, including ridge, lasso and elastic net regression, which focus on reducing the weight of individual variables or totally excluding some variables. In ridge regression (Hoerl & Kennard, 1970), the OLS function (4.19) is supplemented in a

way, so that the residual sum of squares and the regression coefficients are minimised simultaneously:

$$L_{OLS}(\hat{\beta}) = \sum_{i=1}^{n}\left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1i} - \hat{\beta}_2 X_{2i} - \cdots - \hat{\beta}_P X_{pi}\right)^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2 \tag{4.26}$$

The OLS function in ridge regression penalises the sum of squared regression coefficients (L_2 penalty). It aims to decrease model complexity, while keeping all variables in the model. On the other hand, Least Absolute Shrinkage and Selection Operator (lasso) regression penalises the sum of the absolute values of the regression coefficients (L_1 penalty). The aim of lasso is to force the coefficients to adjust to zero i.e. to minimise the number of independent variables in the linear model (Tibshirani, 1996). The OLS function in Lasso is given by:

$$L_{OLS}(\hat{\beta}) = \sum_{i=1}^{n}\left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1i} - \hat{\beta}_2 X_{2i} - \cdots - \hat{\beta}_P X_{pi}\right)^2 + \lambda \sum_{j=1}^{p} \left|\hat{\beta}_j\right| \tag{4.27}$$

The regularisation parameter (λ) can be chosen by performing k-fold cross-validation to find the λ that minimises the residual sum of squares. Another method, which was not applied in this study, includes choosing the λ hat minimises the Akaike (AIC) or Bayesian (BIC) Information Criterion (Oleszak, 2019). The main difference between lasso and ridge is that the former tries to set some coefficients to zero, thus performing variable selection, while the latter tries to reduce the size of the coefficients but keeps all variables in a model. In other words, both methods try to solve the issue of collinearity, albeit in a different way.

Finally, elastic net regularisation is a method combining the penalties of lasso (L_1) and ridge (L_2) regression (Hastie & Tibshirani, 2020). The OLS function in elastic net is given by:

$$L_{OLS}(\hat{\beta}) = \sum_{i=1}^{n}\left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{1i} - \hat{\beta}_2 X_{2i} - \cdots - \hat{\beta}_P X_{pi}\right)^2 + \lambda\left(\frac{1-\alpha}{2}\sum_{j=1}^{p} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{p} \left|\hat{\beta}_j\right|\right) \tag{4.28}$$

where α is the mixing parameter between ridge ($\alpha = 0$) and lasso ($\alpha = 1$). As with λ, the mixing parameter α can be estimated as the one that minimises the residual sum of squares through k-fold cross-validation or through grid search.

4.4.4. Random forest

Ensemble learning is a machine learning technique that combines more than one learning algorithms or models in order to achieve better predicting performance than a single algorithm. In random forests, ensemble learning is realised through bootstrap aggregation. Bootstrap aggregation or *bagging* is a procedure which can be used to reduce variance in those algorithms that have high variance, such as classification and regression trees. In bootstrapping, samples are randomly generated from the original dataset with replacement. Then, a classification or

regression tree is trained in parallel based on each sample. Finally, in bagging the trees are aggregated into some kind of an averaging process aiming to get an ensemble model with lower variance and less overfitting (Breiman, 1996).

This paragraph summarises *bagging*; let $X = x_1, x_2, \dots, x_n$ and $Y = y_1, y_2, \dots, y_n$ be the independent and dependent variables of a training set respectively. By bagging repeatedly M times the training set, a random sample with replacement is selected and M regression trees are created. Therefore, for $m = 1,2, \dots M$:

i. Sample M training subsets X_m, Y_m from X, Y with replacement.
ii. Train a regression tree f_m using X_m, Y_m.
iii. To find the prediction for an unseen sample x', average the predictions from all individual trees on x':

$$\hat{f} = \frac{1}{M} \sum_{m=1}^{M} f_m(x') \tag{4.29}$$

The first person to create an algorithm for random forests was Ho (1995), who noticed that complex decision trees lose their generalisation accuracy when confronted with new data. Random forest regression is a supervised ensemble learning algorithm that trains a multitude of decisions trees in parallel based on *bagging*, in order to output the mean prediction of the individual trees. In comparison to decision trees, random forests introduce one extension; in addition to bootstrapping of the original dataset as discussed earlier, a random forest performs bootstrap sampling of variables at each node i.e. it selects a random subset of variables instead of selecting the optimal variables that minimise the node impurity (Bruce & Bruce, 2017). The number of variables sampled at each node, as well as the number of trees, are parameters that can be set in most programming packages (Liaw & Wiener, 2002).

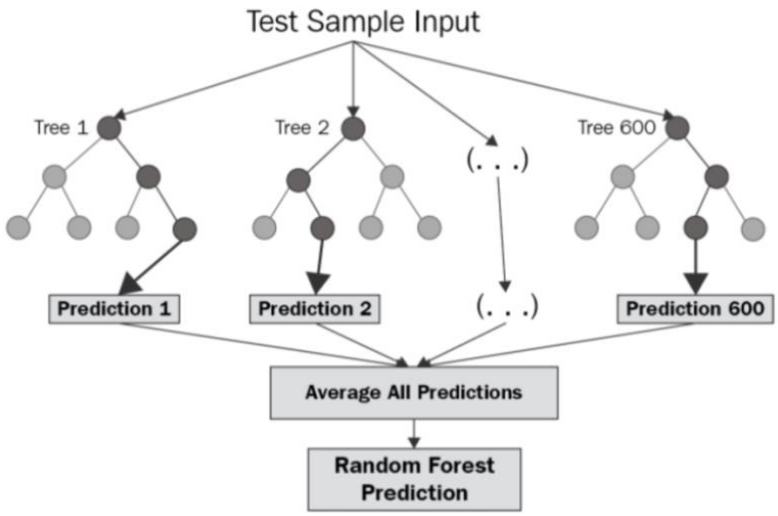


Figure 4.5 Example of a random forest for regression with 600 trees (Bakshi, 2020)

Variable selection in random forests can be performed by removing one variable at a time and observing the change in evaluation statistics. If accuracy decreases significantly when

removing a variable (usually measured as the increase in mean squared error), then this variable should probably be kept in the model, although there are no predetermined cut-off limits. Another methodology to determine the inclusion of a variable in a random forest is to measure how much impurity increases, when a variable is not included in node splits. The higher the increase in impurity, the more important this variable is. In regression trees, node impurity corresponds to the variance of the node i.e. the magnitude of difference of outputs between the optimal and a sub-optimal subset of variables (Strobl & Zeileis, 2008).

As with all machine learning methodologies, random forests have both advantages and disadvantages. One of the biggest advantages of random forests is their application in both regression and classification problems. Furthermore, they require almost no prior manipulation of data, such as normalisation or scaling, and they work well on large datasets. Another major advantage is their ability to deal with collinear variables and to handle non-linear variables. On the other hand, they lack in interpretability, which means that a trained random forest cannot be easily explained and in contrast to linear regression, they are unable to handle values of independent variables outside of the observed range and finally, their outcome depends highly on the tuned parameters.

4.4.5. Gradient boosting machines regression

Another technique to realise ensemble learning is *boosting*. The general idea behind *boosting* is to train models sequentially, which focus on learning from the mistakes of the previous model and so on. This is achieved by placing higher weight on instances with higher errors and lower weight on lower errors, rather than randomly selecting subsets of data and variables at each node as done in bagging.

Gradient boosting machines (GBM) is a special case, based on the way that the shortcomings of each model are identified. Instead of adjusting weights, gradient boosting fits new models sequentially on the residuals of the previous model (Bruce & Bruce, 2017). The residuals of each round are used to calculate the gradient, which is the partial derivative of the loss function. Then, the gradient is used in order to find the direction in which the model parameters maximise the reduction in the residuals. Gradient boosting relies heavily on training parameters such as the learning rate (step of descent), shrinkage (reduction of the learning rate) and the differentiable loss function. Finally, other quantitative parameters like the number of iterations and the number of observations sampled in each round also play a significant role in finding the minimum of the loss function (Elsinghorst, 2018).

Let $\{(X_{1i}, X_{2i}, \dots, X_{ji}, Y_i)\}_{i=1}^{n}$ be the original dataset, a differentiable function $L(Y, F(X))$ and the number of iterations M. Gradient boosting regression can be summarised in the following pseudocode from Friedman (2001) and Hastie et al. (2009):

First, initialise the model with a constant value:

$$F_0(X) = \arg min_\gamma \sum_{i=1}^{n} L(Y_i, \gamma) \tag{4.30}$$

For $m = 1, \dots, M$:

i. Compute the residuals of each iteration for $n = 1, \dots, n$

$$r_{im} = -\left[\frac{\partial L(Y_i, F(X_{1i}, \dots, X_{ji}))}{\partial F(X_{1i}, \dots, X_{ji})}\right]_{F(X)=F_{m-1}(X)} \tag{4.31}$$

ii. Fit a new model $h_m(X)$ on the residuals using the dataset $\{(X_{1i}, X_{2i}, \dots, X_{ji}, r_i)\}_{i=1}^{n}$

iii. Compute multiplier γ_m by solving the following optimisation problem

$$\gamma_m = \arg min_\gamma \sum_{i=1}^{n} L(Y_i, F_{m-1}(X_{1i}, \dots, X_{ji}) + \gamma h_m(X_{1i}, \dots, X_{ji})) \tag{4.32}$$

iv. Update the model

$$F_m(X) = F_{m-1}(X) + \gamma_m h_m(X) \tag{4.33}$$

v. Repeat the iterative process until $m = M$.

Gradient boosting is a very flexible algorithm, which can be applied on categorical and continuous data. Its multiple tuning options enable it to optimise on different functions and, due to this, it possesses very high predictive accuracy. However, large search grids are needed to tune all parameters, which also influence the results in a very definitive way, and this process is computationally heavy. Further, the fact that GBM tries to remedy the errors sequentially may result in overfitting and emphasise outliers. Finally, models are less interpretable in comparison to simple decision trees and linear regression models (Boehmke, n.d.).

4.5. Model validation

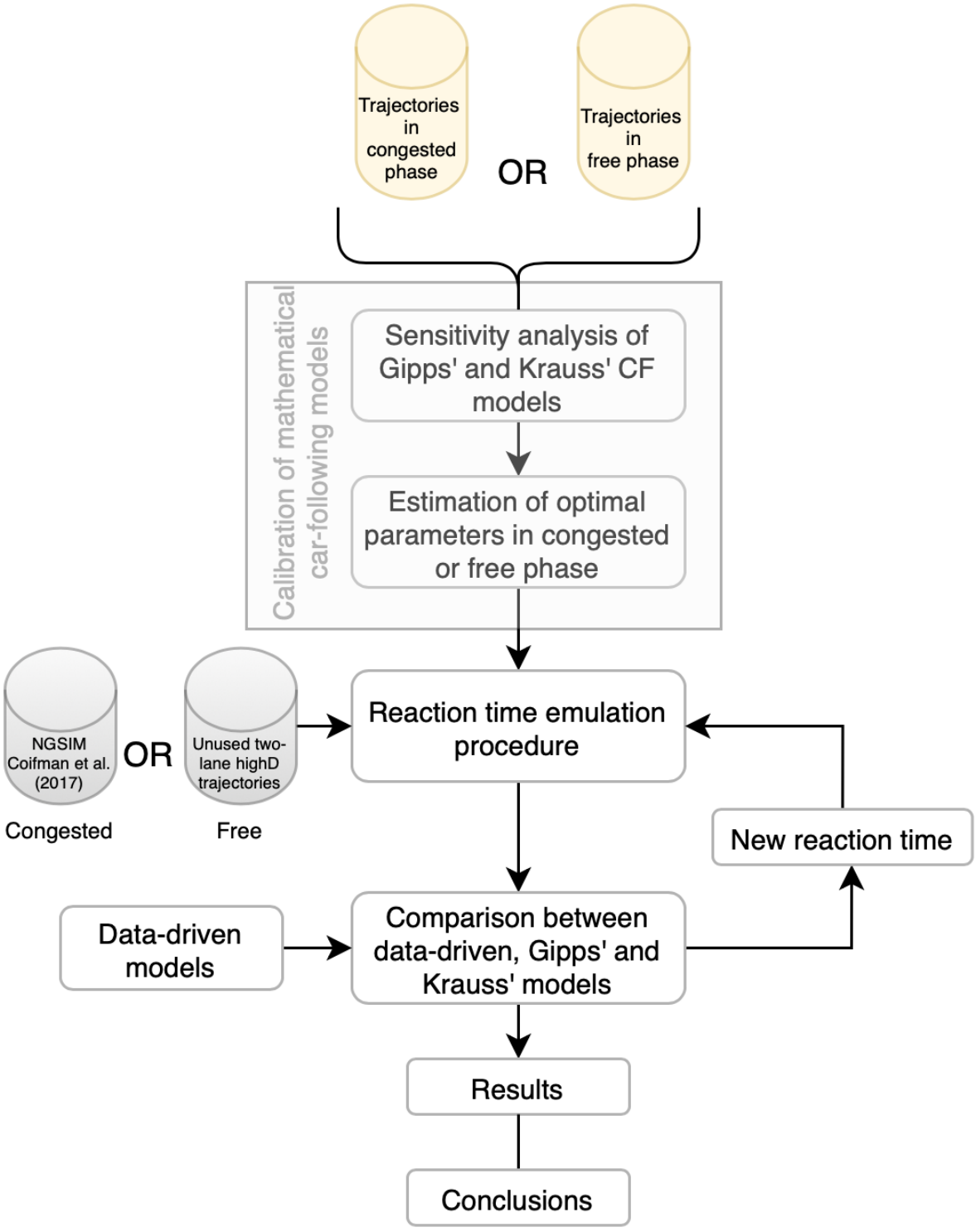


Figure 4.6 Workflow of car-following model calibration and validation

4.5.1 Sensitivity analysis of car-following models

A sensitivity analysis will be conducted in order to find the optimal values of the parameters of Gipps' and Krauss' car-following models. The first step includes a sequential sensitivity analysis of all parameters aiming to find the optimal subset of values that minimise the error. In the second step, these optimal parameters are applied on Gipps' and Krauss' models and a further sensitivity analysis is performed by adjusting only the reaction time τ. A major difference between the first and the second steps is that the former utilises the training set (75% of the total observations), while the latter utilises the test set (25% of the total observations).

In the beginning, an average value of each parameters is chosen. Then, the effect of each parameter on the performance of the model, measured by the RMSN, is examined separately and the combination of parameters that minimise this measure of fit is chosen. Based on the optimal subsets of parameters, the effect on the reaction time on the performance of the model is studied with time steps of 0.4 s.

However, Krauss' car-following model is relatively new and less commonly applied than Gipps' model and a literature review does not provide sufficient information about the parameter ranges. Hence, it was decided to apply the same ranges as in the sensitivity analysis of Gipps' model: in detail, maximum acceleration $a_{max} = 0.8 - 2.6\ m/s^2$, maximum desired deceleration $b = 1.6 - 5.2\ m/s^2$ and reaction time $T = 0.4 - 2.8\ s$ with increments of 0.4 s. The random error ε can take continuous values from 0 to 1 and the maximum velocity of the highways was $v_{max} = 36.11\ m/s\ (= 130\ km/h)$, which is a common speed limit on German highways (and highD). Thereafter, the same two-step optimisation procedure was followed.

The comparison between data-driven and mathematical models happens on two occasions. The first occasion is during the sensitivity analysis of τ, where different reaction times are applied on all models and their performance is compared based on RMSN, RMSE, MAE and R^2. A second comparison takes places when validating the models using unseen datasets, where the models are compared again by changing the reaction time and calculating the usual measures of fit.

4.5.2. Measures of fit

Except of the coefficient of determination R-squared, which was defined earlier under linear regression, other frequently used evaluation metrics include the mean absolute error (MAE), the mean squared error (MSE), the root mean square error (RMSE) and the normalised root mean square error (RMSN). All these evaluation statistics are pairwise measures of error between predictions and actual observations (Alessandrini & Sperati, 2017). Using different evaluation statistics, different aspects of the obtained predictions can be analysed (Antoniou et al., 2013).

The mean absolute error is a scale-dependent measure calculated by the arithmetic average of the absolute errors:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{4.34}$$

The mean squared error is a scale-dependent measure, which incorporates the variance of the estimator i.e. it shows how spread the actual observations are around the regression line. The MSE is the default minimisation metric in many regression algorithms, as it penalises larger deviations of the predictions from the actual observations heavily.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.35}$$

The root mean square error, as its name denotes, is the square root of the previous measure. The advantage of this scale-dependent measure is its direct comparability with the actual observations, while at the same time it penalises large deviations of the regression model from the actual observations more heavily than smaller deviations. Hence, the $RMSE$ depends on the order of magnitude of the observed values:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{4.36}$$

The normalised root mean square error is a percentage-based variation of the root mean square error, that allows direct comparison of errors of different magnitude and is given by the following formula:

$$RMSN = \frac{\sqrt{n\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}}{\sum_{i=1}^{n} Y_i} \tag{4.37}$$

In this study, the above statistics refer to the errors between the velocities extracted from video footage and the velocities estimated by the respective model each time.

5. Data Analysis

This chapter starts by listing and citing the software packages that were used in this study. Then, a brief analysis focussing on the macroscopic properties of traffic is presented, including different aggregation methods. Finally, a comparison of highD with NGSIM is included, as the latter is the benchmark dataset in similar studies.

5.1. Software and tools

All analyses in this study were conducted in R programming language (R Core Team, 2019) in the integrated development environment RStudio (RStudio Team, 2018). Basic functions in R can be expanded through 'packages', which introduce new functions, machine learning models, parametrisation options or new types of graphical representations. Some of the most important packages that were used in this study are:

- **readr** is used to read comma-separated values files (Wickham et al., 2018)
- **dplyr** is complete package for data manipulation e.g. summarising, ordering, filtering etc. (Wickham et al., 2020)
- **ggplot2**, an advanced data visualisation package (Wickham, 2016)
- **plotly**, for the creation of interactive graphs from 'ggplot2' (Plotly Technologies Inc., 2015)

All packages above are collectively part of the '**tidyverse'** package, which follows consistent rules of grammar and data structures. Further packages include:

- **scales**, which provides functions for determining breaks and labels for axes and legends in ggplot2 automatically (Yan Q. , 2019)
- **corrplot**, which produces graphical displays of correlation matrices (Wei & Simko, 2017)
- **ggpubr**, a package for modifying and arranging multiple ggplot2 graphs (Kassambara, 2020)
- **gridExtra**, for arranging multiple graphs on a single plot (Auguie, 2015)
- **pdp**, a package for plotting partial dependence from various machine learning algorithms (Greenwell, 2017)
- **vip**, a package for creating variable importance plots for different machine learning algorithms (Greenwell & Boehmke, 2020)
- **rpart**, which contains functions for recursive partitioning of classification, regression and survival trees (Therneau & Atkinson, 2019)
- **cluster**, a package for cluster analysis (Maechler et al., 2019)
- **dbscan**, which implements density-based algorithms for clustering in R (Hahsler et al., 2019)
- **mclust** is a package for applying model-based clustering and density estimation with a variety of different covariance structures (Scrucca et al., 2016)

- **caret**, contains functions for resampling and cross-validation during regression model training and validation (Kuhn, 2020)
- **MLmetrics**, a package of evaluation metrics such as loss and score functions for machine learning (Yan, 2016)
- **Metrics**, an implementation of machine learning evaluation metrics with zero dependencies (Hamner & Frasco, 2018)
- **performance**, a collection of functions for assessing model quality in regression, such as the R-squared, correlation coefficients, root mean square error etc. (Lüdecke et al., 2020)
- **forecast**, which contains functions for analysing and plotting univariate time series predictions (Hyndman & Khandakar, 2008)
- **MASS**, which includes, among other, an algorithm for stepwise regression (Venables & Ripley, 2002)
- **tree**, a package for training, plotting and validating classification and regression trees (Ripley, 2019)
- **randomForest** is used to train classification and regression models based on forests of trees (Liaw & Wiener, 2002)
- **gbm**, a package for training gradient boosting machine regression models (Greenwell et al., 2019)

The hardware used for this study is a 2018 MacBook Air with an Intel Core i5 processor at 1.6 GHz and 8 GB of RAM.

5.2. Exploratory data analysis

5.2.1. Analysis of fundamental diagrams

In traffic flow theory, a fundamental diagram (FD) is a scatterplot between two macroscopic properties of traffic – flow, density and average velocity. Although this term most commonly refers to the flow - density relationship, it is also used interchangeably for the average velocity – density and average velocity – flow relationships. However, it is important to note that there is no causality among flow, density and speed, but rather a more complex mechanism that can be traced down to the movement of individual vehicles. A single point on an FD indicates a traffic state i.e. a quantitative description of traffic at a given time point. As flow, density and velocity are continuous measures, an infinite number of traffic states can exist (Antoniou et al., 2013). However, traffic states with similar characteristics can be grouped together and form clusters, which are called traffic phases.

The fundamental diagram is highly important in many applications of traffic engineering. The main areas of application include the determination of capacity of an arterial, as a decision aid in traffic management and control, in traffic assignment as approximation of travel time in macroscopic simulation models, as the main determinant of Level of Service and other (Hoogendoorn, 2016). In this study, the FD was extracted from the trajectories in order to perform a visual evaluation of the macroscopic properties of traffic flow in the highD dataset

and, most importantly, in order to define which trajectories are in congested and which in free phase. The states in the FD were extracted both by lane and by section in an aggregated manner.

Traffic flow is calculated approximately at the middle of each road lane or section using the concept of the virtual loop detectors (Figure 5.1). It is assumed that such detectors are placed at x = 200 m in each direction of travel, which corresponds to at least two detectors per Track. The vehicles crossing the detectors are aggregated every 30 s for each lane or section and this value is extrapolated to hourly values by multiplying by 3600/30. The average density per lane or section is calculated by averaging the number of vehicles in each aggregation interval and multiplying by 1000/400, where 400 m is the approximate length of one track, in order to extrapolate the result to vehicles per kilometre (veh/km). Similarly, the average velocity is calculated by taking the average of the velocities of all vehicles on the section or lane per direction in each aggregation interval (30 s) and then multiplying the results by 3600/1000 in order to get the average velocity in km/h. For more details see subsection 4.3.1.

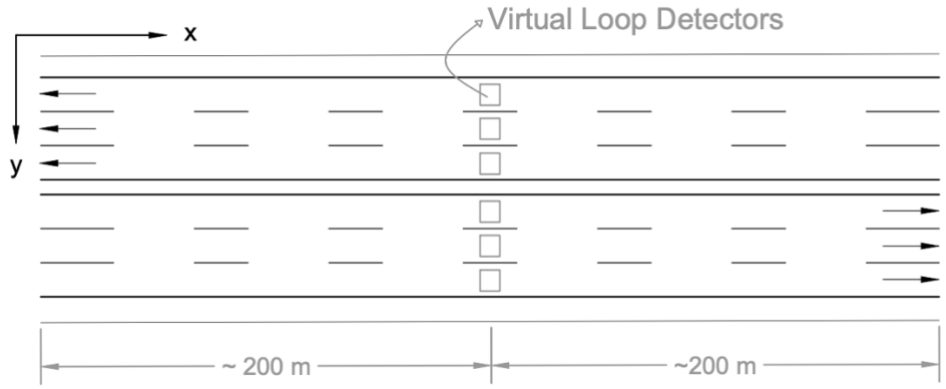


Figure 5.1 Virtual loop detectors in the middle of a highway section

The fundamental diagrams of traffic flow for each direction of three-lane highway areas can be seen on Figure 5.2. As expected, the fundamental diagrams resemble those of van Wageningen-Kessels et al. (2015), which were provided in the literature review part of this study. It can be seen that the overwhelming majority of traffic states refer to low vehicle densities (< 80 veh/km) and speeds close to the speed limit (> 100 km/h), resembling free-flowing traffic phases. The traffic flow per direction ranges between 0-7000 veh/h in the free-flowing phase, with average values closer to 3500 veh/h, which is far below the maximum achieved value of 7000 veh/h. Similarly, average vehicle density is around 40 veh/km, which is much lower than the maximum density of 150 veh/km, according to the diagonally placed density plots. The correlation coefficients of velocity against density and flow are negative, which means that an increase in traffic density or flow results in a drop of the average vehicle speed and vice versa, although the causality is unknown. It is important to note that this figure observes all vehicles regardless of vehicle type.

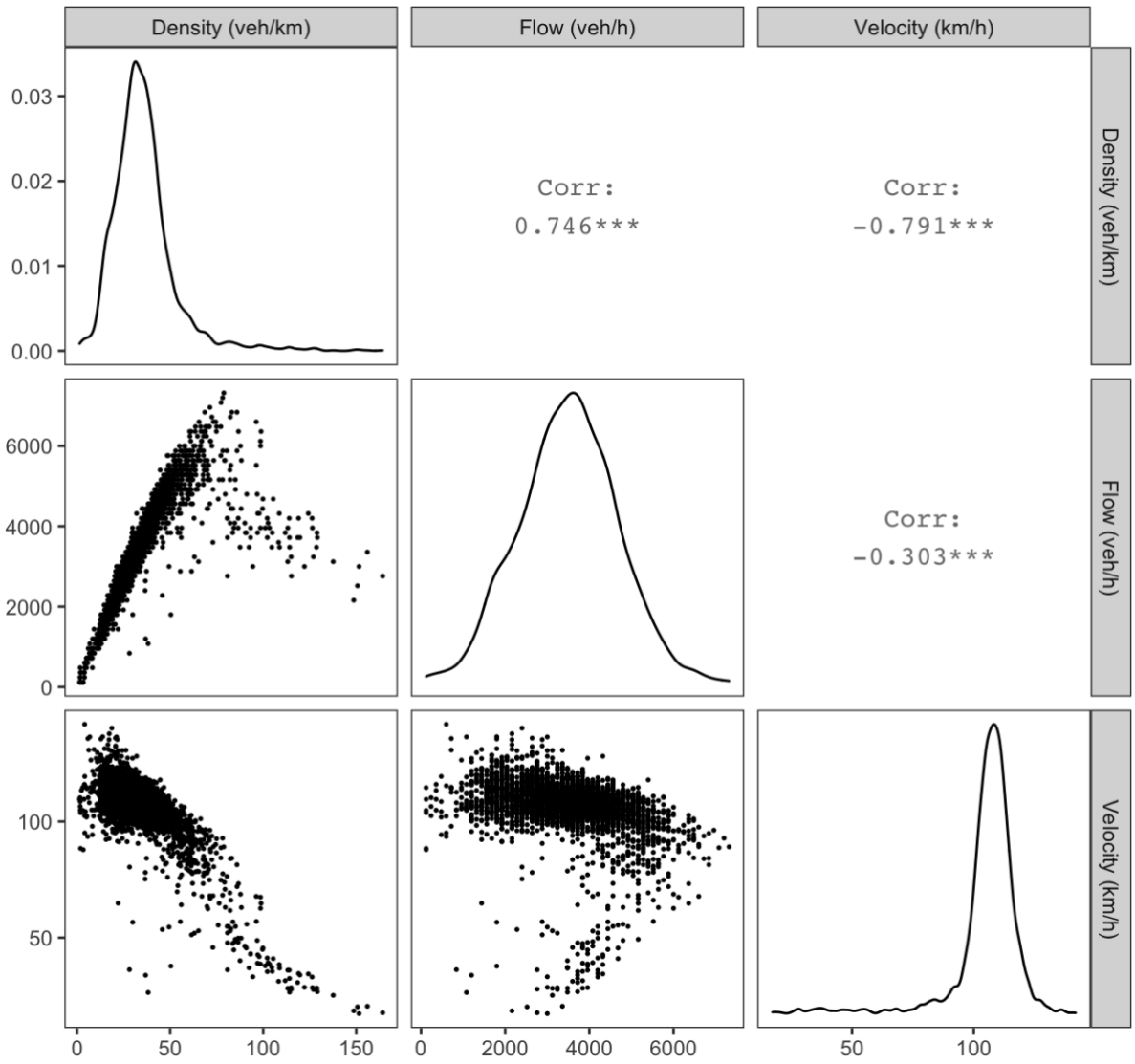


Figure 5.2 Macroscopic properties of traffic by direction

Similar observations can be made on the lane-based fundamental diagrams of traffic (Figure 5.3). It can be seen that vehicle density, flow and average velocity values are more dispersed, which is reasonable when averaging and then extrapolating on less data. However, it is obvious that even the lane-based diagrams follow the general pattern of an FD. A possible anomaly in this plot is the lax velocity – flow correlation, which reveals that lane-based flow states are more evenly distributed on the y axis than the respective flow states from section-based FD. In this case, the aggregation interval (30 s) seems to have played a crucial role.

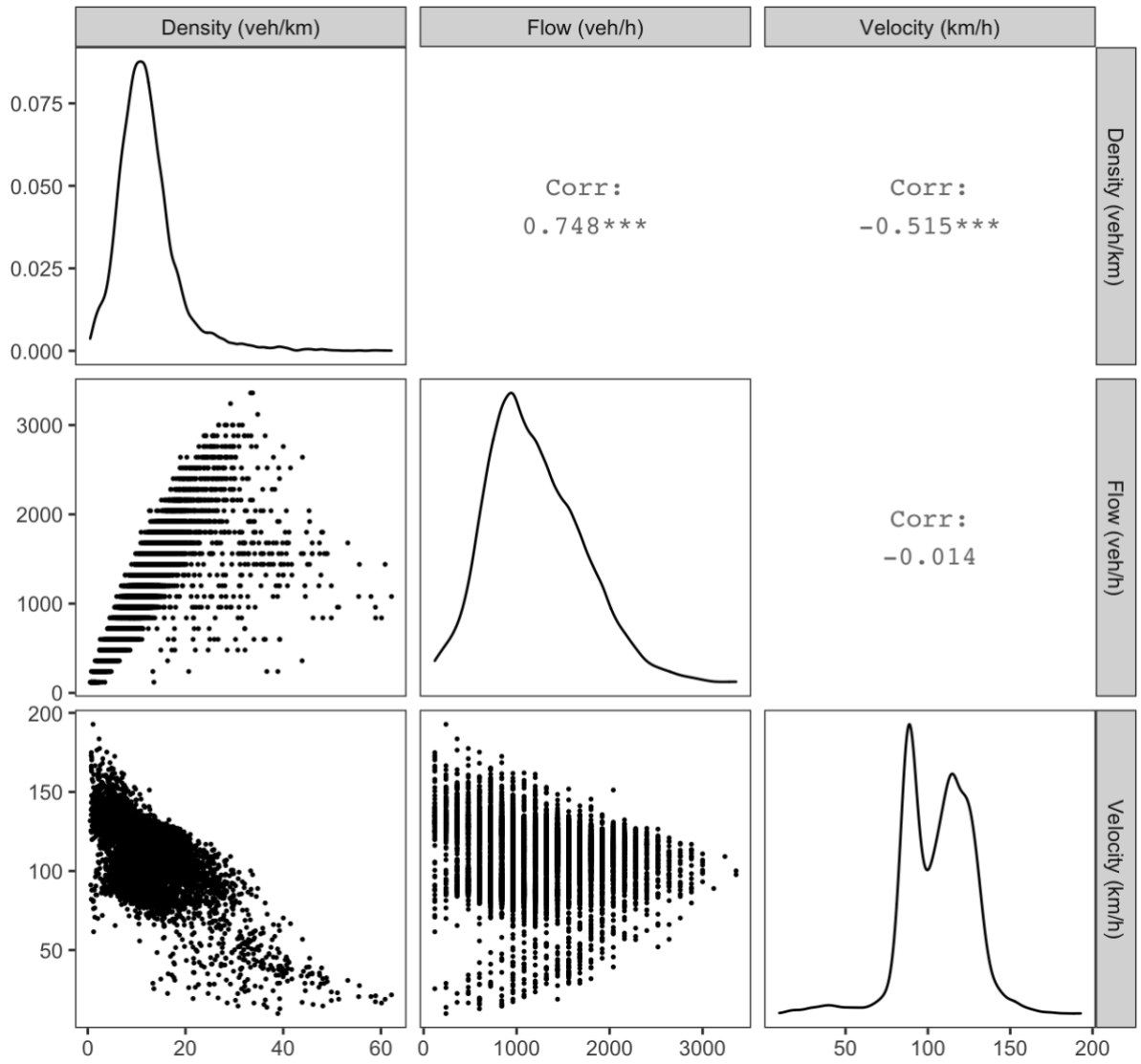


Figure 5.3 Macroscopic properties of traffic by lane

5.2.2. Analysis of aggregated values per track

Other data aggregation methods include boxplots and bar charts per track. Figure 5.4 shows boxplots of velocities and distance headways and bar plots of the percentage of trucks among all vehicles in each track. In most cases, the average speed is above 100 km/h, which indicates that drivers are free to choose their own speed and therefore, they drive closer to the speed limit. Only tracks 11, 12, 13, 25 and 26 show lower average velocities and lower 25th and 75th quartiles, which means that drivers on those tracks have not been able to choose freely the speed of their vehicles. On two occasions, tracks 25 and 26 in western direction, average velocities fall under 50 km/h. Further, as can be seen on Figure 5.4 (middle), distance headway boxplots of tracks 11, 12, 13, 25 and 26 have shorter interquartile ranges than the rest. A possible explanation for this could be that that traffic has become so dense that it is no longer possible for a driver to choose their own distance headway and, as such, vehicles are almost more equidistantly spread on the highway in comparison to other tracks. On these occasions,

the distance to the preceding vehicle starts playing an important role in the determination of the follower's velocity.

A probable cause of lower average velocities and denser syntheses of vehicles is high demand or, better formulated, demand closer to capacity of the highway. However, trucks could also turn out important in such cases, especially when considering that highD (and generally highways in Germany) carry a high percentage of trucks. Heavy vehicles could be accused of creating congestion-like conditions due to inferior physical capabilities (lower velocity, acceleration, deceleration etc.) or because of restrictions imposed by the rule of the road, like a speed limit as low as 80 km/h based on the weight of the truck. An optical evaluation of the percentage of trucks among all vehicles in each track Figure 5.4 (bottom) does not reveal a connection between trucks and lower average velocities or distance headways; the percentage of trucks in tracks 11, 12, 13, 25 and 26 is lower than or in accordance with their percentage in all other tracks. Hence, the most possible explanation behind low average velocity is high demand.

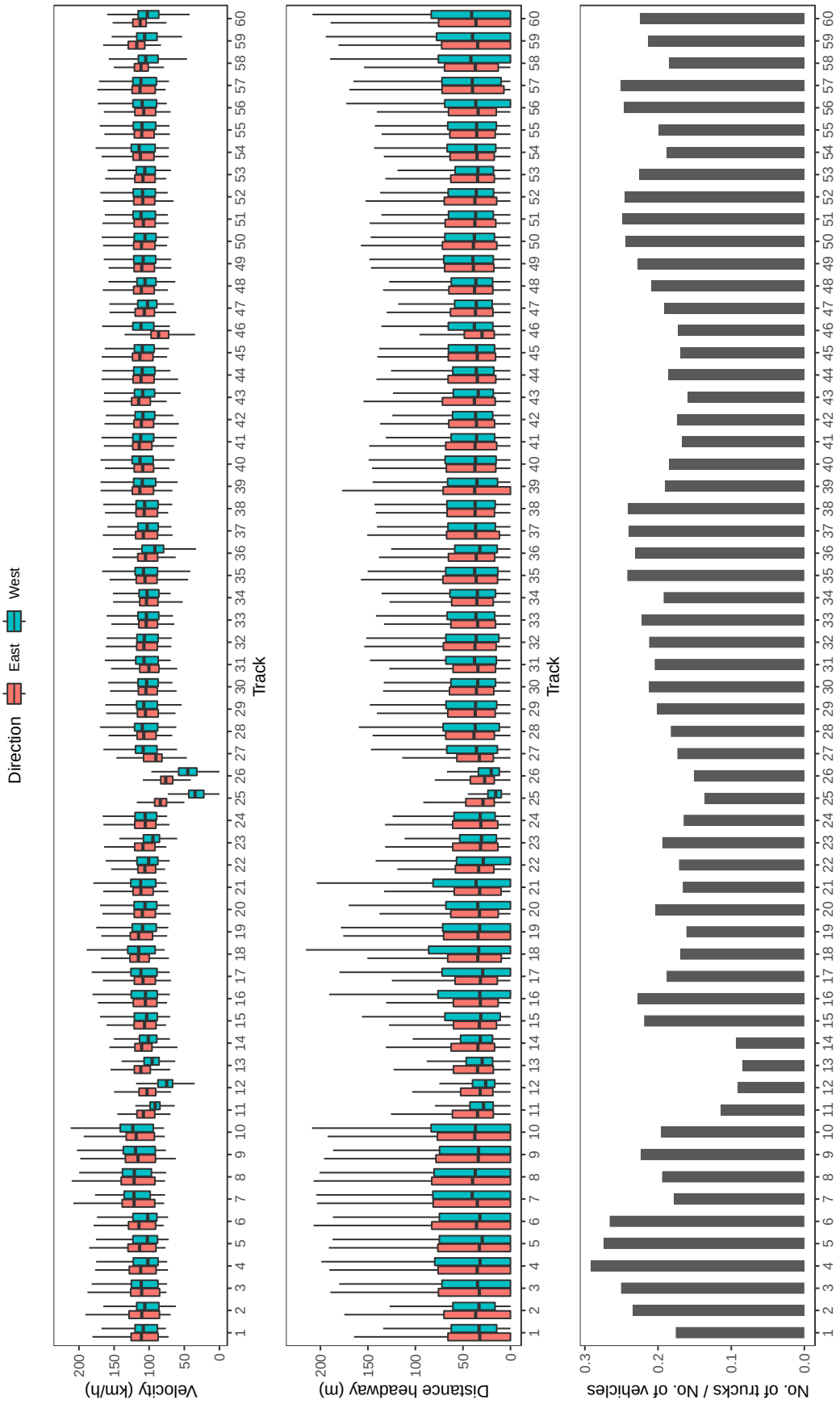


Figure 5.4 Boxplots of velocity (top), distance headway (middle) and percentage of trucks (bottom) per track

5.2.3. Macroscopic comparison of highD and NGSIM

At first sight, the highD and NGSIM datasets seem to be similar in many ways. However, the original study of Krajewski et al. (2018) has underlined that many differences exist between the two datasets, not only regarding data acquisition methods, size and accuracy, but also with regards to the dynamic properties of the captured vehicles and the composition of traffic. First and foremost, video recordings in highD were captured by drones, whereas in NGSIM cameras mounted on buildings were used. Therefore, the average duration per recording in highD is much shorter at around 15 minutes due to battery life constraints. However, highD has a longer total duration of almost 16.5 hours, in comparison to the 1.5 hour of NGSIM. In highD, the average percentage of trucks in traffic is higher (23%) than in NGSIM (3%) and the video recordings were able to capture many more unique vehicles. The fact that average speeds for cars stand at 120 km/h in highD and less than 75 km/h in NGSIM, in combination with the shorter highways areas in highD (around 420 m in highD and 500-1000 m in NGSIM), has resulted in temporally shorter trajectories in the former. However, a major difference is the frame rate of each dataset, 25 Hz in highD and 10 Hz in NGSIM, which means a vehicle travelling the same distance at the same speed would give 1.5 times more data points in highD in comparison to the NGSIM dataset (Table 5.1).

Table 5.1 Comparison of highD and NGSIM (Krajewski et al, 2018; Mahajan, 2019)

Attribute	**highD**	**NGSIM**	**Unit**
Total duration	16.5	1.5	hours
Average duration per recording	15 - 20	45	minutes
Recording frequency	25	10	Hz
Length of highway area	420	500 – 1000	m
Lanes per direction	2-3	5-6	-
Vehicles (total)	110000	9206	-
Truck percentage per recording	0 - 50%	< 10%	-
Mean velocity	Cars - 120 Trucks - 80	< 75	km/h
Traffic density	low	high	-

As it has already been mentioned in literature review, NGSIM is a very agile dataset for traffic modelling studies but comes with major inaccuracies. A lot of research has been done on the NGSIM trajectories, but only a few researchers have tried to study and, later, to reconstruct the dataset. Two notable efforts are those of Montanino and Punzo (2013), and Montanino and Punzo (2015), who initially applied a multistep filtering procedure to eliminate outliers and remove random residual disturbances from the I-80 dataset. Then, the authors reconstructed the trajectories of individual vehicles, while at the same time paying attention to platoon integrity of trajectories as a means of validation, underlining the fact that the ground truth is still unknown. Looking into previous studies, they came to the conclusion that trajectory inaccuracies impact mainly lane-changing models, while they have little impact on car-following.

On the other hand, Coifman and Li (2017) state exactly the opposite, although they agree that the trajectories of NGSIM are beyond repairing. Contradicting the findings of Montanino and Punzo (2015), the authors state that positioning errors have a major impact on car-following applications too. They found that errors occur more often on certain areas on the highway and for certain vehicle sizes, probably due to camera-positioning inefficiencies. The re-extracted trajectories showed more reasonable velocity-spacing relationships and corrected the extremely high accelerations found in the original dataset. A comparison of a random velocity trajectory before and after re-extraction can be seen on Figure 5.5 (right). The re-extracted dataset of Coifman and Li contains vehicle trajectories from the I-80 recording in the morning 04:00 – 04:15 period and has become freely available to the research community (Coifman & Li, 2017). Due to its simple format and better quality than the original NGSIM dataset, it has been chosen as one of the validation datasets in this study.

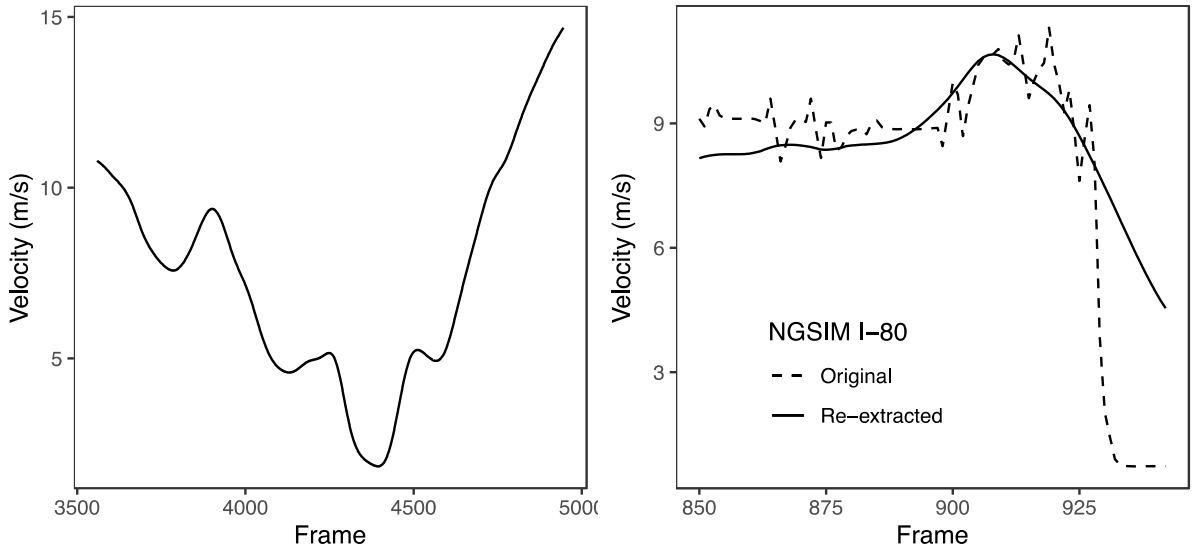


Figure 5.5 A random velocity trajectory from highD (left) and the re-extracted NGSIM I-80 (right)

Finally, the time-to-collision (TTC) safety surrogate measure is not included in the dataset of Coifman and Li and It is calculated using formula (5.1).

$$TTC = \frac{Dhw}{xVelocity - precedingXVelocity} \tag{5.1}$$

The time-lagged TTC is calculated by using the respective time-lagged variables. In the special case that the follower and the leader are travelling at the same speed exactly, then the TTC is equal to zero.

Part III: Results

6. Traffic phase identification

This chapter includes the results of traffic phase identification, after following the methodology that was described in the previous chapters. Macroscopic states of traffic are clustered either by direction of travel or by lane and the results are compared based on established model selection techniques and visual evaluation.

6.1. Traffic phase by direction

Model-based clustering selects the number of components and the covariance parametrisation using the Bayesian Information Criterion (BIC). The number of components corresponds to the optimal number of clusters that maximise the BIC. Parametrisation refers to the volume, shape and orientation of the isodensity ellipses in the multidimensional space, or in other words they can be thought of loosely as the "bounding boxes" of a number of points belonging to the same distribution. For example, a "VEV" parametrisation can be interpreted as Volume: **V**ariable, Shape: **E**qual and Orientation: **V**ariable between the various isodensity ellipses and is an input parameter of the GMM training algorithm. According to the parametrisation, the shape of the "bounding" box can be defined as spherical, diagonal or ellipsoidal (Celeux & Govaert, 1995).

In this study, GMM clustering is performed on the aggregated density, flow and average velocity features on three-lane highway areas, either by direction or by lane. On Figure 6.1 (by direction), it can be seen that fourteen different parametrisation combinations are trialled in order to find those that maximise the BIC. Out of those, at least eight models, namely "VVV", "EVV", "VEV", "EEV", "VVE", "VEE", "EVE" and "EEE", produce almost identical results with regards to the BIC.

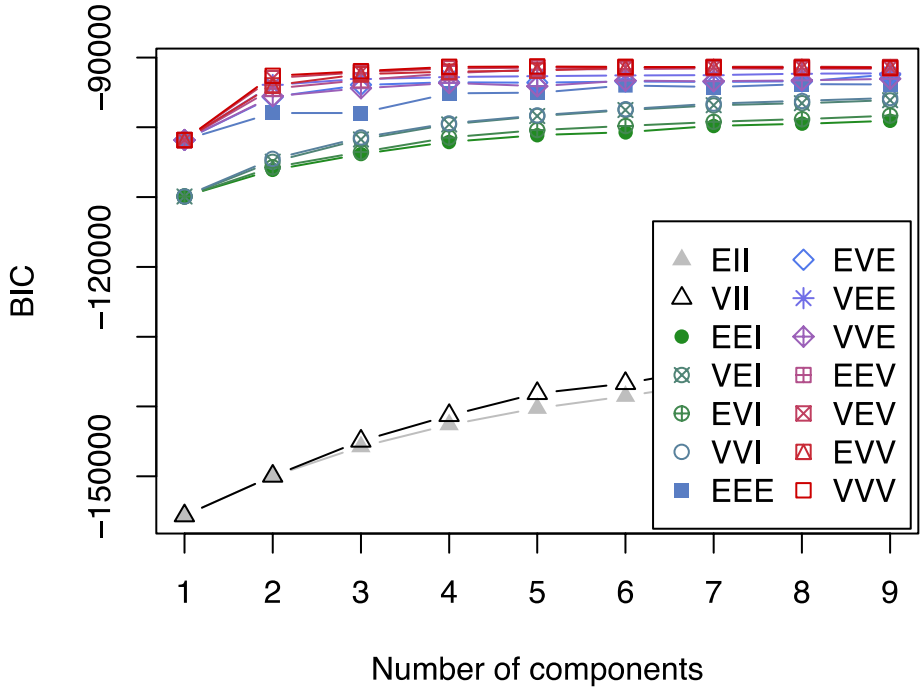


Figure 6.1 BIC of different parametrisation combinations and number of clusters in traffic phase by direction

According to the results of the model-based clustering (Figure 6.1), the best three parametrisations are “VVV” with five clusters, “EVV” with eight clusters and “VVV” with four clusters (Figure 6.2). In those graphs, intervals with high density and low average velocity, also known as congested states, were either put in one cluster (“VVV,5” and “VVV,4”) or two neighbouring clusters (“EVV,8”). Since the goal of this procedure is to separate the congested from the free states, all other clusters except of the aforementioned should be regarded as the free phase. This means that the actual total number of clusters is either two or three.

Further evidence of the existence of two or three clusters comes from the “shoulder” method. It shows that the maximisation of BIC slows down significantly after the first two clusters (Figure 6.1). Therefore, limiting the number of clusters down to two leads in a slight loss of fit but in return results in a more parsimonious model, which can be easily interpreted into recognisable traffic phases.

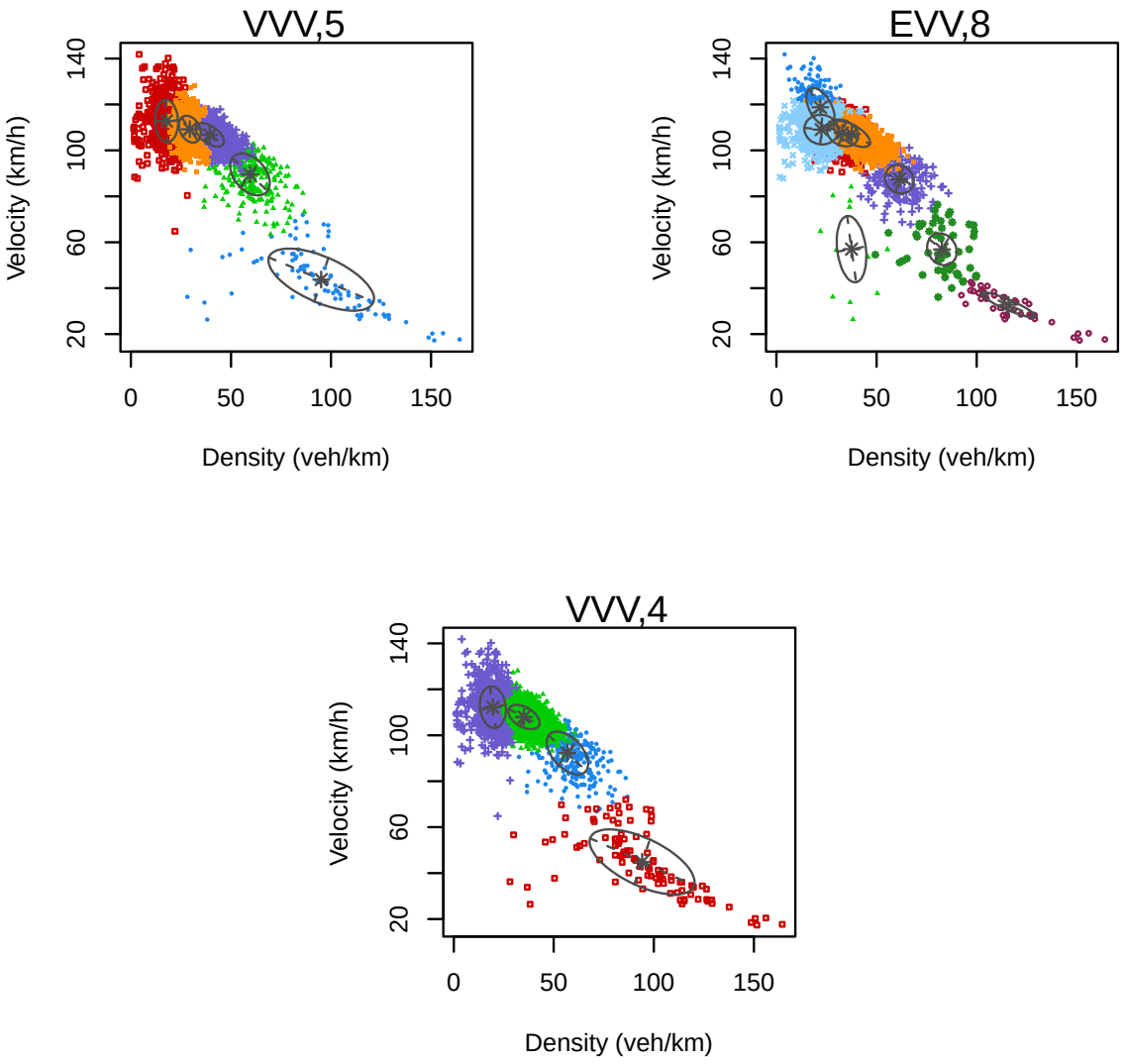


Figure 6.2 Best three parametrisation combinations in traffic phase by direction

After that, the selection can be narrowed down to models with two clusters. In theory, a model with variable volume, shape and orientation should be able to separate the aggregated dataset better than a model with equal parametrisation. However, a visual evaluation of all models with two clusters (Appendix: Figure 9.1) shows that only in “EVV,2”, “EEV,2”, “EVE,2” and “EEE,2” there is a clear separation of the Gaussian distributions, as there exists no overlapping of the distributions of the congested and the free phases. Then, “EVV,2”, “EEV,2”, “EVE,2” are found to suffer under one disadvantage; some observations clustered in the congested phase seem to belong to the free phase in reality, as they have rather low density and flow (Appendix: Figure 9.2). Hence, only the “EEE,2” model remains, with ellipsoidal distribution and equal volume, shape and orientation, which seems to provide the least overlapping of distributions and a clear threshold between the subspaces of congested and free phase. Overall, it has to be stressed, that the results of all models with two clusters are very similar.

The results of the “EEE” clustering model per highway direction can be found on Figure 6.3. On the lower part, it can be seen which states were clustered into the congested and which into the free phase. It is obvious that the overwhelming majority of states were clustered into the free phase, as expected due to the high average velocities in highD. The density function of traffic density, flow and average velocity can be found in the middle. At around 75 veh/km density and 70 km/h velocity the transition from congested to free phase takes place - on the other hand, flow is not a good indicator of the actual traffic phase, as same flow levels can be achieved both in congested and free phases. On the top of this figure, the correlation coeffiecients between the features and the clusters can be found, with all correlations being statistically significant. Summary statistics of the two different phases are shown on Table 6.1, where count refers to the number of aggregation intervals (states) classified in each cluster (phase).

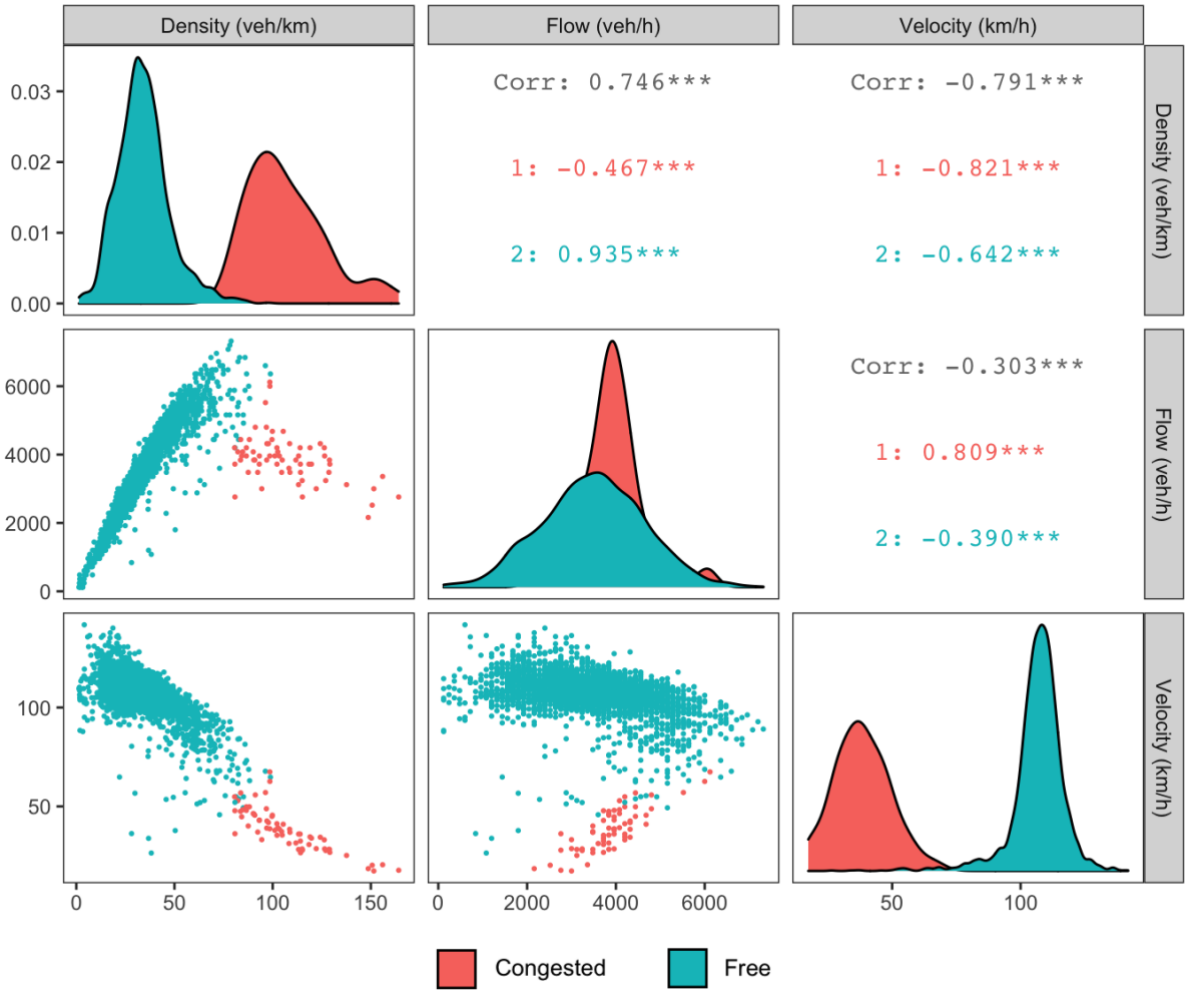


Figure 6.3 GMM clustering results of three-lane highway areas by direction

Table 6.1 Summary statistics of the congested and free phases

	Density (veh/km)	Flow (veh/h)	Velocity (km/h)
		Congested	
Count	66	66	66
Min	60.63	2160	17.34
Median	102.40	3840	37.58
Mean	107.08	3876	38.32
Max	164.18	6120	67.45
Standard deviation	19.73	702.55	10.79
		Free	
Count	3212	3212	3212
Min	1.32	120	26.41
Median	33.15	3480	107.76
Mean	34.17	3486	106.82
Max	98.85	7320	141.84
Standard deviation	13.25	1149.28	10.04

6.2. Traffic phase by lane

The same procedure is followed in model-based clustering by lane. On Figure 6.4, the optimal number of clusters and the parametrisation results can be seen. The three models that maximised the BIC were “VVV” with seven, eight or nine clusters (Figure 6.5). They suggest that observations in the lower right part of each graph are indeed placed in one cluster, whereas all other observations in the upper left part are split into more clusters. Therefore, the former could be considered as the congested phase and all the latter as the free phase. The BIC plot (Figure 6.4) provides further justification, because it is obvious that the maximisation of this metric almost stops after three clusters. To achieve better parsimony and interpretability, it is better to keep the lowest possible number of distributions, as the drop in the fit can be easily offset by the more comprehensible results according to the established knowledge in traffic engineering.

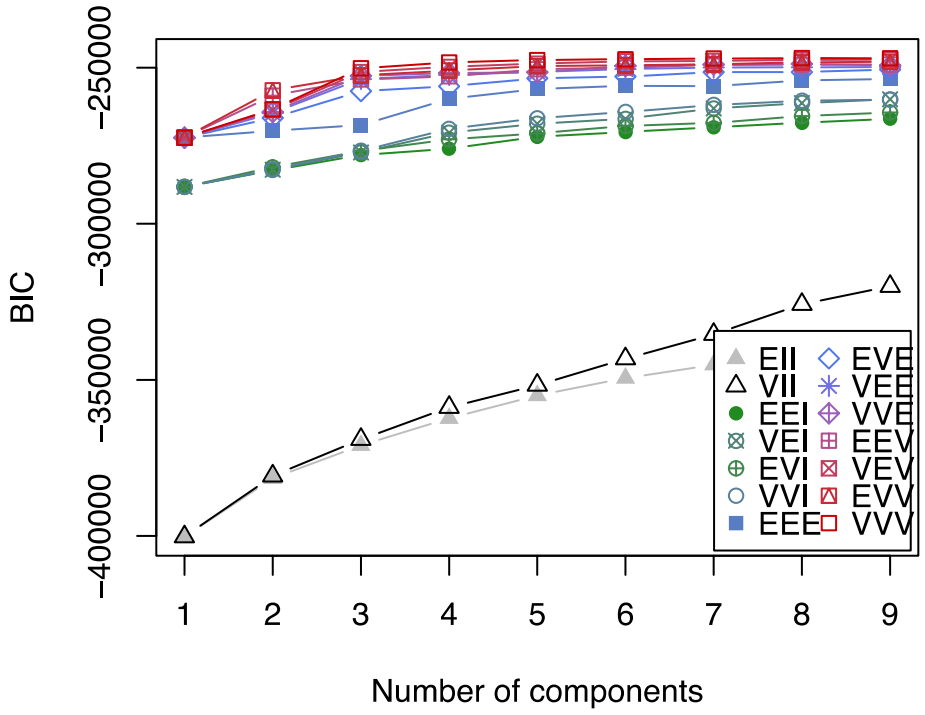


Figure 6.4 BIC of different parametrisation combinations and number of clusters in traffic phase by lane

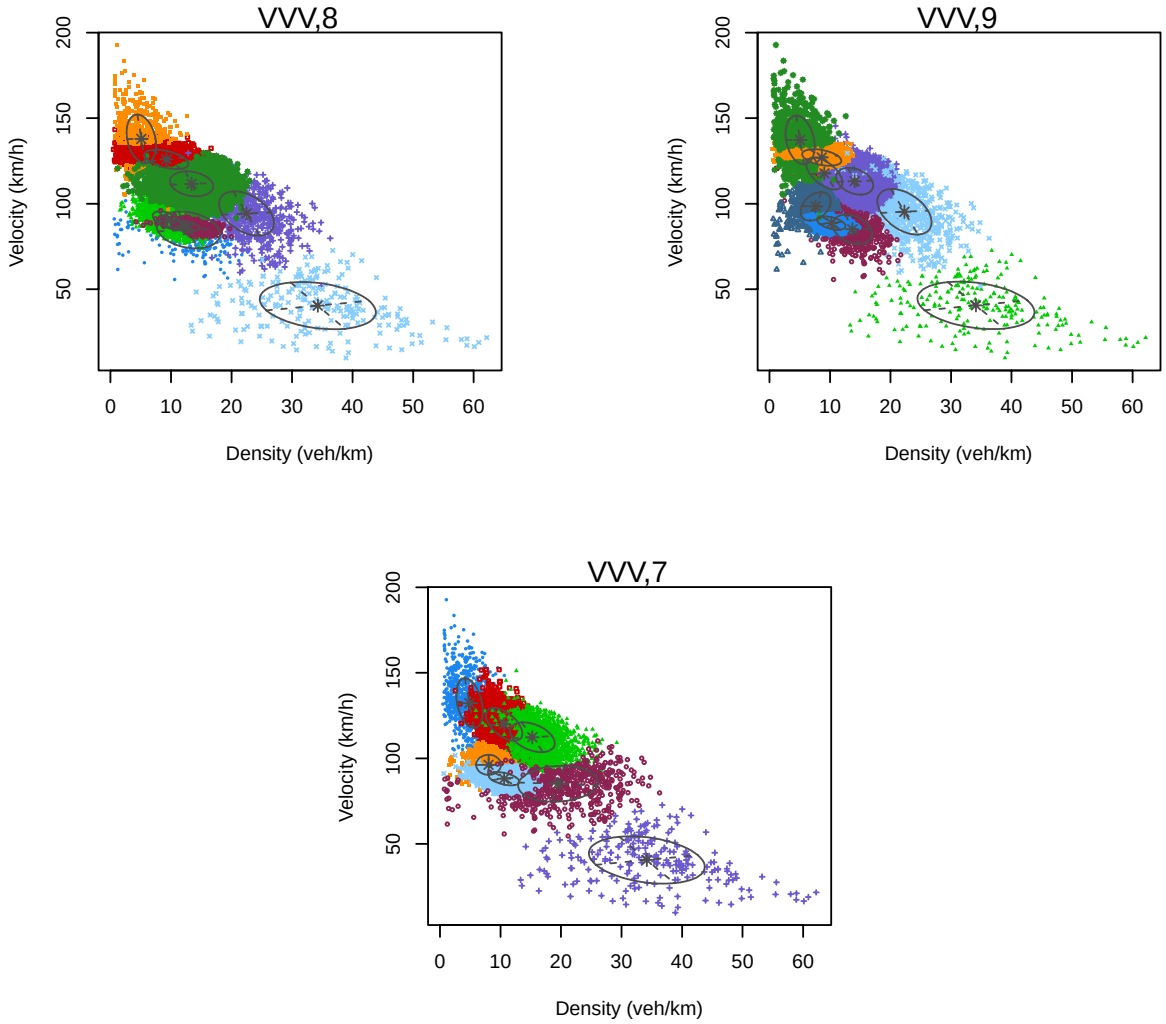


Figure 6.5 Best three parametrisation combinations in traffic phase by lane

To choose among all models with two clusters, the results of "VVV", "EVV", "VEV", "EEV", "VVE", "VEE", "EVE" and "EEE" models were plotted on density – velocity – flow diagrams (Appendix: Figure 9.4 - Figure 9.6). On the density – velocity diagrams, it is obvious that six out of eight models resulted in irrational results; they either show that the congested phase extends from the lowest to the highest traffic density, or that the free phase is a very small subspace surrounded by congested states. These patterns are not consistent with previous experiences regarding traffic phase, with the only notable exception being the models "EVV,2" and "EEV,2".

Regarding the last two models, a look into the density – flow (Appendix: Figure 9.5) and the flow – velocity (Appendix: Figure 9.6) diagrams reveals two weak spots. First, it seems that traffic states are very close to one another, probably because of the projection of the aggregation intervals into hourly values making the differentiation between different intervals almost non-existent, and there does not exist a clear limit between the congested and the free phases. Second, there seem to be quite a few misclassified states, for example some with low density and high average velocity were classified into the congested state and vice versa. Overall, the

visual evaluation of the macroscopic diagrams showed that the direction-based clustering produced better traffic phase separation than the lane-based, although with less observations, but also with fewer uncertainties due to extrapolation of the aggregation interval.

6.3. Matching congested and free states to trajectories

The ultimate goal of traffic phase identification is to find out which trajectories belong to the congested and which to free traffic phase. In theory, more interactions between vehicles happen during the congested phase, where car-following dynamics are different in comparison to the free phase. A process of disaggregation is followed thereafter, by matching the number of the Tracks, the frame and the direction at each interval between the clustered and the original trajectories. An example of trajectories in congested phase separated from the original trajectories is shown in Figure 6.6 ,which depicts all trajectories of lane 2 from Track 12 (top) and the subset of the trajectories that have been put in the congested cluster (bottom).

This matching procedure concludes the traffic phase identification process. The outputs of this process are two datasets: the first corresponds to the congested phase and the second corresponds to the free phase of highD three-lane Tracks. These datasets will be subsequently used to develop car-following models in congested and in free phase separately.

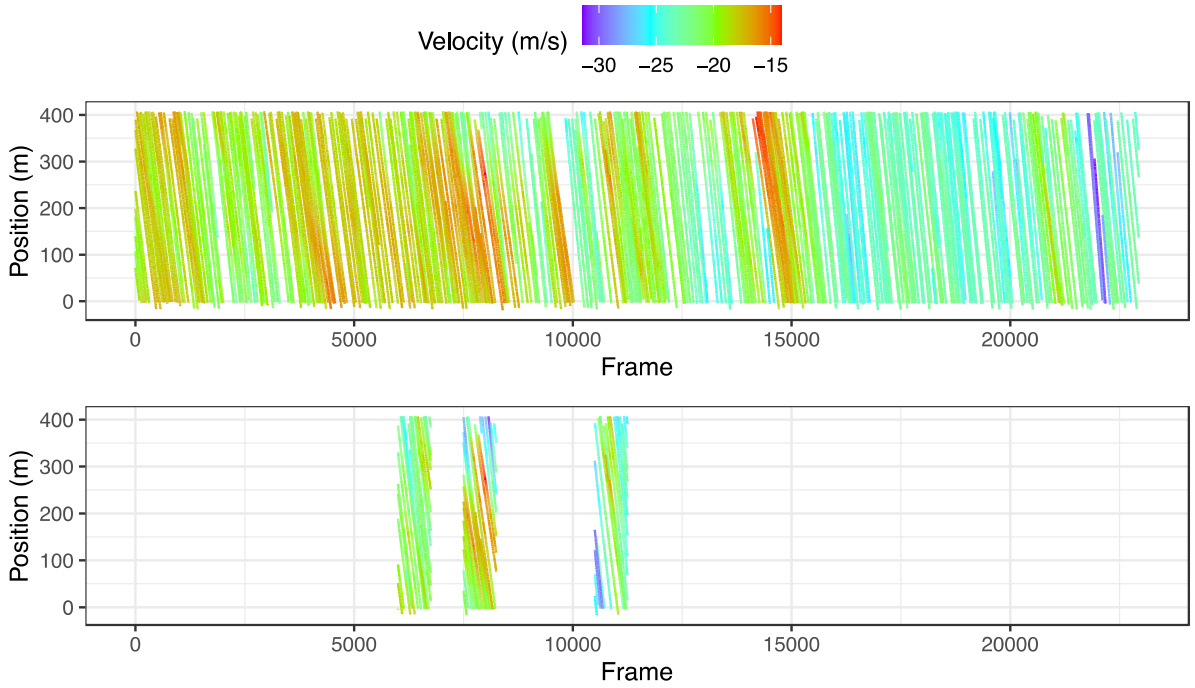


Figure 6.6 All trajectories of Track 12, lane 2 (top); trajectories in congested phase (bottom)

7. Model development and validation

This chapter presents the results of model development in congested and free phases, including variable selection and parametrisation of the data-driven models. Additionally, the optimal parameters of Gipps' and Krauss' mathematical models are estimated by sensitivity analyses and the models are compared. Finally, the calibrated mathematical and data-driven models are applied on unseen datasets, in order to estimate their performance and to investigate the role of reaction time.

7.1. Congested phase

As it has already been described in methodology, the goal of a car-following model is to estimate the velocity of the ego vehicle in the next time step based on some stimulus from its environment. The movement of the ego vehicle depends highly on the movement of the preceding vehicle, which functions as a physical barrier to the ego vehicle when no overtaking is possible. Given a larger than the minimum distance between the ego and the preceding vehicle, the driver has the option of accelerating and achieving a higher speed in a future time step in comparison to their current speed. Therefore, the distance headway is also an important factor to consider in car-following models. Other important factors may be the acceleration of the preceding vehicle, the difference in the velocities of the preceding and the following vehicles, the time-to-collision surrogate measure etc.

Figure 7.1 depicts the dependencies between velocity and distance headway. On the left, it is clear that the velocity of the ego vehicle increases given a larger distance to the preceding vehicle. At higher values of distance headways, they stop being very important and thus, there is almost a free choice of velocity, as can be seen by the increasing width of the velocity observations when distance headways increase (widening cone of observations). On the right, it can be seen that the velocities of the ego and the preceding vehicle are almost perfectly correlated, because a unit increase in preceding velocity means a unit increase in ego velocity as well. Therefore, linear regression should play a crucial role when modelling these dependencies.

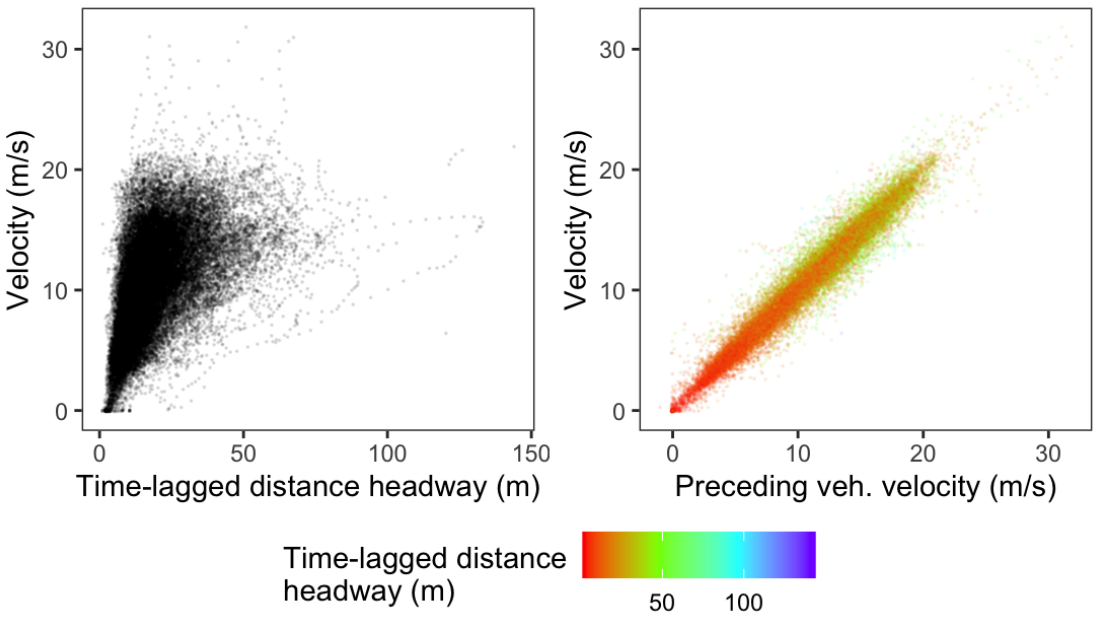


Figure 7.1 Velocity - preceding velocity - distance headway plots in congested phase

However, at higher speeds such as when driving on a highway, visibility conditions are usually better and it is possible for a driver to look even two vehicles ahead. In double car-following models, a driver adjusts the dynamics of their vehicle by taking into account the dynamics of two vehicles in front. The following paragraphs describe the development of a double model, which can be summarised in four steps: choice of independent variables, tuning of model parameters, model training based on k-fold cross-validation and finally, model testing.

7.1.1. Multiple linear regression models

Variable selection

The correlation plot on Figure 7.2 depicts the strength of the correlation between all variables. It can be seen that $xVelocity$ is positively correlated (in blue) with $lagXVelocity$, $precedingXVelocity$ and $leadingXVelocity$, which are also correlated between each other. The distance headways between the ego and the preceding vehicles, $lagDhw$, and between the preceding and the leading vehicle, $precedingDhw$, show weaker positive correlation with the velocity of the ego vehicle, but the correlations are still statistically significant. The acceleration of the preceding vehicle $precedingXAcceleration$, of the leader $leadingXAcceleration$ and the speed differences ($deltaPrecedingXVelocity$ and ($deltaLeadingXVelocity$) are negatively correlated (in red) with the ego velocity.

On the other hand, the correlation coefficients of both time-to-collision measures (ttc and $lagTtc$) are not statistically significant at significance level $a = 0.05$. Their p-values are given on the correlation plot instead. Hence, ttc and $lagTtc$ are the first independent variables to be removed from the regression model. At this point, it is important to recall that all variables

except the first four refer to the previous time step i.e. to the current time minus the reaction time $\tau = 1.2\ s$.

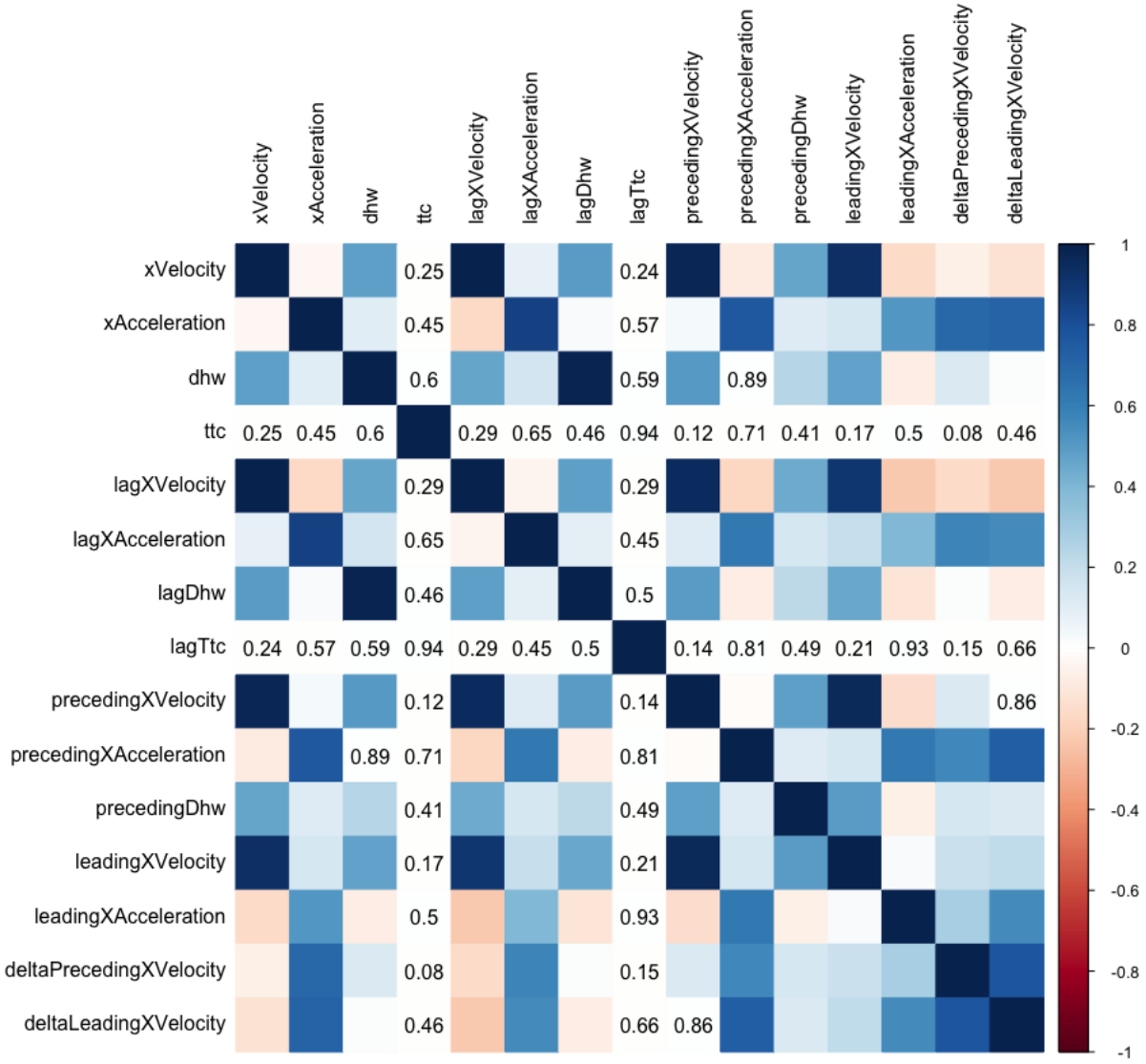


Figure 7.2 Correlation plot for reaction time $\tau = 1.2\ s$ in congested phase

Additionally, as $deltaPrecedingXVelocity$ and $deltaLeadingXVelocity$ are linear combinations of the velocity variables of the ego, preceding and leading vehicles, they are removed from linear regression model. Linear combinations of variables have to be avoided in regression, because they violate the assumption of non-multicollinearity between the predictors. Further on, a multicollinearity check of the remaining variables reveals that both $precedingXVelocity$ and $lagDhw$ are highly collinear with $lagXVelocity$ and $leadingXVelocity$ with $VIF > 10$. Multicollinearity between the speeds of three following vehicles is expected, as each vehicle presents a physical barrier to the next and so on. A common response against multicollinearity is doing nothing, when improvements in interpretation are expected (Washington et al., 2011, p.111).

Model development

A conscious decision was made to develop and compare three linear models, after dropping $lagXVelocity$. The first model includes $leadingXVelocity$, the second one excludes $leadingXVelocity$, while the third one excludes the leading vehicle completely. A summary of the included predictors as well as some measures of fit (test set) can be seen on Table 7.1 and Table 7.2, after training the linear regression models using 10-fold cross-validation with $\tau = 1.2\ s$. An automated test detected no influential observations in all three models.

Table 7.1 Predictors included in linear regression in congested phase

Model	Response	lagXAcceleration	lagDhw	precedingXVelocity	precedingXAcceleration	precedingDhw	leadingXVelocity	leadingXAcceleration
LR1		•	•	•	•	•	•	•
LR2	xVelocity	•	•	•	•	•		•
LR3		•	•	•	•	•		

Table 7.2 Measures of fit of linear regression models in congested phase

Model	**Precision metrics (training set)**	**Precision metrics (test set)**				**Running time (s)**
	R^2	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**	
LR1	0.96	0.08	0.88	0.62	0.96	3.06
LR2	0.96	0.08	0.89	0.62	0.96	2.73
LR3	0.95	0.08	0.89	0.63	0.95	2.52

Based on the table above, the model with $leadingXVelocity$ outperforms very slightly the other linear models. As the purpose of this study is to include more information about the preceding and the leading vehicles than mathematical car-following model, a decision was made to proceed with the LR1 model.

The regression coefficients of LR1 rounded to three decimals can be seen on Table 7.3. All independent variables are statistically significant at significance level $a = 0.05$. The magnitudes of the coefficients are different, as is the magnitude of each independent variable. However, not all coefficients can be logically explained; for example, the coefficient of $precedingXAcceleration$ shows that the following driver decelerates when the preceding vehicle accelerates, which is the opposite of the expected. The same can be said about

$precedingDhw$. Later in this report, a sensitivity analysis of the reaction time for values of $\tau = 0.4 - 2.8\ s$ will estimate its precision on another dataset.

Table 7.3 Coefficients of the linear regression LR1 in congested phase

Response	(Constant)	lagXAcceleration	lagDhw	precedingXVelocity	precedingXAcceleration	precedingDhw	leadingXVelocity	leadingXAcceleration
xVelocity	0.222	0.183	0.005	0.850	-0.899	-0.002	0.122	0.153

A disaggregated analysis of the residuals helps to check if the main assumptions of linear regression are met (Figure 7.3). Homoscedasticity and linearity can be detected through the residuals vs. fitted plot (top left) and though the histogram of the residuals (top right). The assumption of linearity is supported by the apparent existence of equal number of points above and below the horizontal line on the first plot and by the almost symmetric distribution of residuals around zero on the second plot. Residuals seem to be heteroscedastic with higher variance around the middle of the first graph; however, a Breusch-Pagan test rejects the null hypothesis of homoskedasticity in the model. The assumption of normality seems to be sufficient based on the Q-Q (quantile-quantile) plot on the bottom left of Figure 7.3, where deviations from the normal line are present only at the ends of the plot. Further, the P-P (probability-probability) plot shows two cumulative distribution function against each other and also points that the distribution of residuals may be normal (bottom right).

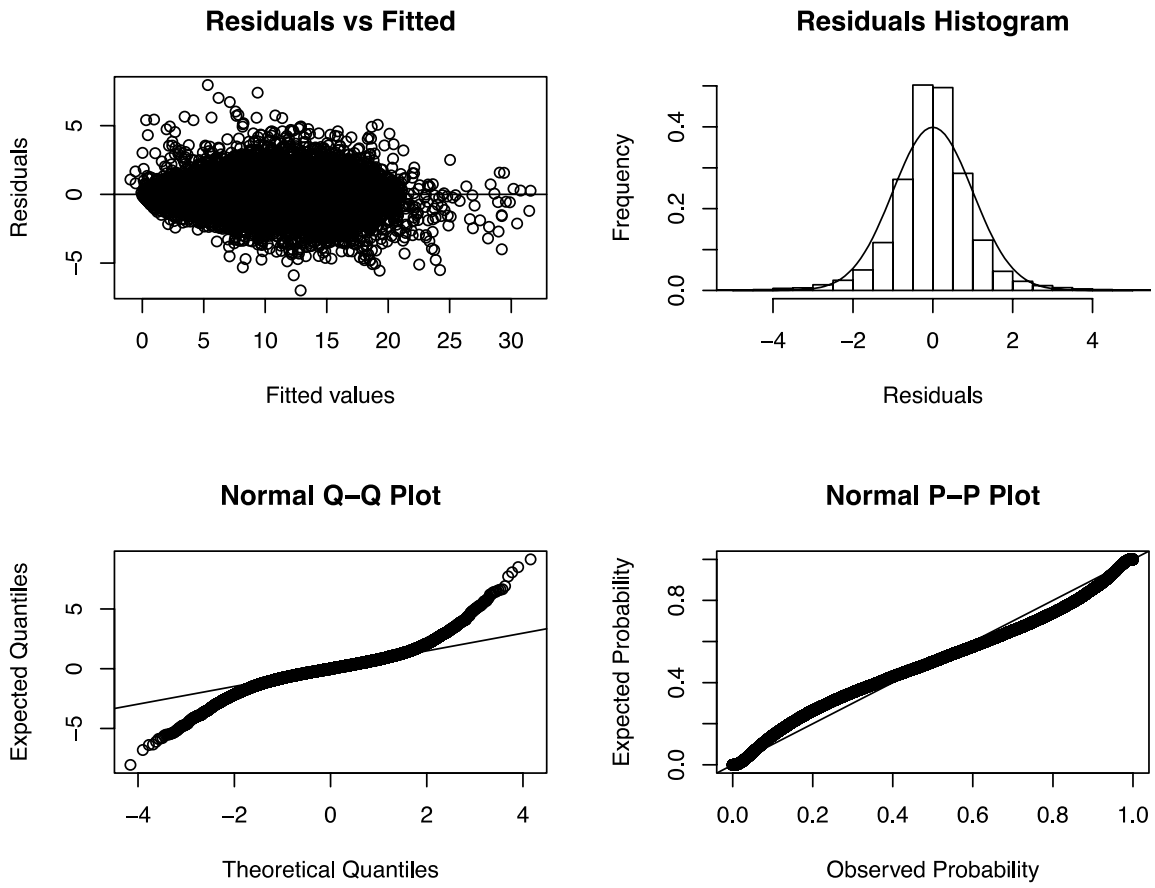


Figure 7.3 Linear regression residuals plot in congested phase

Automatic variable selection procedures choose the variables to be included in a regression model based on a predefined criterion. Stepwise regression adds or removes independent variables one at a time according to their statistical significance, although it does not account for multicollinearity. An attempt to train a linear model using stepwise regression resulted in an almost perfect fit with $R^2 \approx 1.00$. Best subsets regression, unlike stepwise, tries all combinations of the given independent variables and returns the model that optimises the precisions metrics. With nine predictors, the best subsets algorithm creates and tests $2^9 = 512$ combinations of independent variables. Still, the algorithm does not detect multicollinearity, with the best model returning a perfect fit $R^2 \approx 1.00$. Both models are discarded in favour of LR1, despite the slightly worse performance.

7.1.2. Ridge, Lasso and Elastic Net regression

To reduce the number of variables and avoid overfitting, a model can be developed based on ridge (L_2 penalty), lasso (L_1 penalty) and elastic net (L_1 and L_2 penalty) regression. The λ that minimises the residual sum of squares is found by performing a grid search. Values of λ very close to zero mean that the output is similar to linear regression, while for high λ values the coefficients of the independent variables are forced to become zero.

In ridge regression, the mixing parameter from formula (4.28) is $\alpha = 0$. The model is trained multiple times for all different values of λ with 10-fold cross-validation using all variables except $lagTtc$. The optimal value that results in a minimum error is $\lambda = 0.404$. As expected, the ridge regression algorithm does not remove any independent variables and therefore, the coefficients are different than those of linear regression. The absolute values of all coefficients are lower than those of the previous linear regression model, with the exception of $lagXAcceleration$ and $leadingXVelocity$, which means that the regularisation procedure was indeed able to reduce the coefficients. The coefficient of determination is $R^2 = 0.99$ using the training set, whereas the respective precision metrics using the test set are $RMSN = 0.08$, $RMSE = 0.49\ m/s$, $MAE = 0.36\ m/s$ and $R^2 = 0.99$. The running time using the hardware configuration mentioned above is 4.22 s.

Table 7.4 Coefficients of ridge regression in congested phase (statistically significant at significance level α = 0.05)

Response	(Constant)	lagXVelocity	lagXAcceleration	lagDhw	precedingXVelocity	precedingXAcceleration	precedingDhw	leadingXVelocity	leadingXAcceleration
xVelocity	0.364	0.474	0.519	0.001	0.274	-0.390	-0.003	0.197	-0.084

In lasso regression, the mixing parameter α is equal to 1. The optimal value of λ is equal to 0.02 with 10-fold cross-validation using all variables except $lagTtc$. The L_1 regularisation procedure keeps the variables $lagXVelocity$, $lagXAcceleration$, $precedingXVelocity$, $precedingXAcceleration$ and $leadingXVelocity$ in the model, where the former two variables have by far the largest coefficients. The R^2 using the training set is almost equal to 1 and using the test set, the precision metrics are $RMSN = 0.08$, $RMSE = 0.13\ m/s$, $MAE = 0.10\ m/s$ and $R^2 = 0.99$. The running time is almost identical to ridge regression at 4.36 s.

Table 7.5 Coefficients of lasso regression in congested phase (statistically significant at significance level α = 0.05)

Response	(Constant)	lagXVelocity	lagXAcceleration	lagDhw	precedingXVelocity	precedingXAcceleration	precedingDhw	leadingXVelocity	leadingXAcceleration
xVelocity	0.093	0.917	0.946	-	0.047	0.061	-	0.027	-

As lasso returned better precision metrics than ridge regression, elastic net regression should tend to lasso i.e. values of a should be close to 1. The results of a grid search of a and λ are shown on Figure 7.4. The $RMSE$ is minimised when $\alpha = 1$, i.e. identical to lasso regression, and $\lambda = 0.024$, only marginally larger than the respective λ of lasso regression. As such small differences may owe to the algorithm that optimise the OLS function in R, the results can be regarded as similar to the previous paragraph. The coefficients of the model are shown on Table 7.6, the precision metrics of the test set are $RMSN = 0.08, RMSE = 0.13\ m/s$, $MAE = 0.10\ m/s$ and $R^2 = 0.99$ and the running time is 12.88 s, larger than both lasso and ridge regression due to the two-dimensional grid search (a and λ).

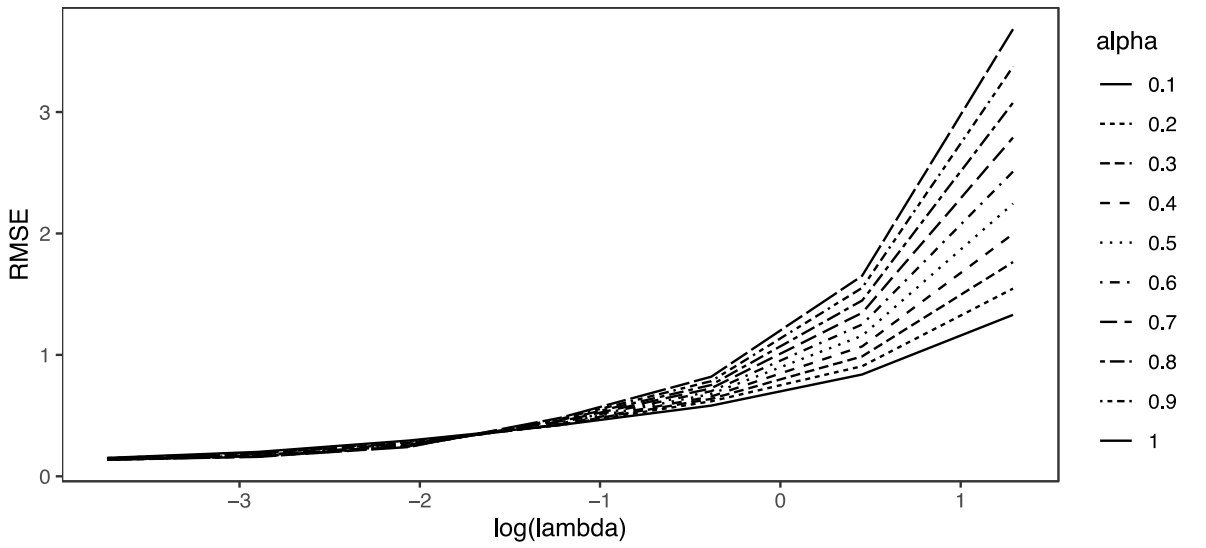


Figure 7.4 Elastic net grid search of a and λ

Table 7.6 Coefficients of elastic net regression in congested phase (statistically significant at significance level $\alpha = 0.05$)

Response	(Constant)	lagXVelocity	lagXAcceleration	lagDhw	precedingXVelocity	precedingXAcceleration	precedingDhw	leadingXVelocity	leadingXAcceleration
xVelocity	0.106	0.911	0.937	-	0.049	0.044	-	0.030	-

All things considered, L_1 regularisation (lasso) achieves better results than L_2 regularisation (ridge), which is also confirmed by the grid search in elastic net regression. However, the magnitude of improvement over the simple linear regression model LR1 is not large, mainly due to two reasons; the simple linear regression models already achieved high performance in modelling the car-following behaviour after two leading vehicles and second, even on the congested state, the trajectories of velocity are largely linear combinations of the trajectories from the previous vehicle. Despite that, lasso simplified the model by keeping just five variables in total, which concern the velocities and accelerations of the preceding and the following vehicles.

7.1.3. Random Forest

Initially, a random forest with all variables is trained, in order to select the variables to be included in the final model. To achieve this, 500 samples with replacement are randomly generated from the training dataset and an equal number of trees are grown. At each node three out of the nine variables are randomly sampled to create the split conditions. Figure 7.5 (left) shows the increase in the MSE when a variable is removed from the trees, while the figure on the right shows the increase in node impurity when a variable is not included when splitting at the nodes. It is important to note that the rank of importance differs between the left and the right plots.

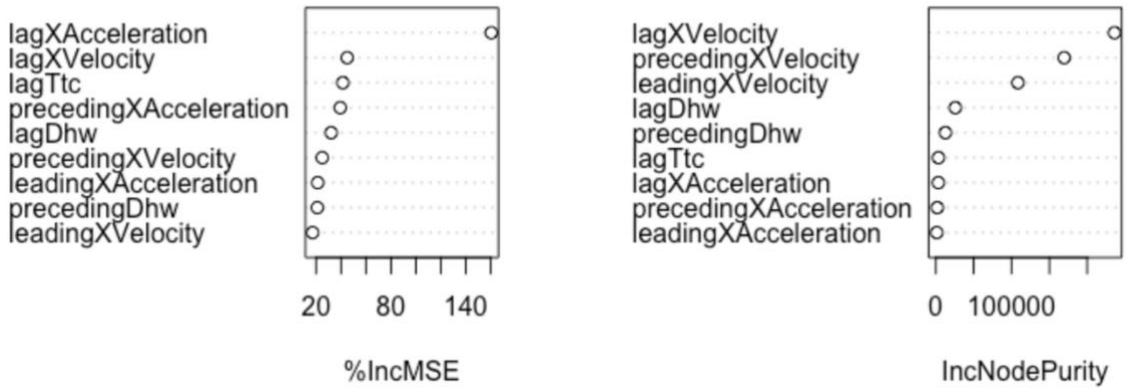


Figure 7.5 Variable importance in random forest regression

A major advantage of random forests is their ability to combine multiple collinear variables. Therefore, a method for selecting variables includes trial and error between different sets of variables based on their importance. Almost perfect results were achieved when including the predictors $lagXVelocity, lagXAcceleration, lagTtc, lagDhw, precedingXVelocity$ and $leadingXVelocity$, which are ranked high in at least one variable importance plot, with three variables splitting at each node. The training set produced a coefficient of determination $R^2 \approx 1$, $RMSE = 0.07\ m/s$ and $MAE = 0.05\ m/s$. Similarly, the precision metrics of the test set are $RMSN = 0.02$, $RMSE = 0.17\ m/s$, $MAE = 0.11\ m/s$ and $R^2 = 1.00$, whereas the running time was much higher than the previous models at 297.88 s. Cross-validation does not apply on algorithms with bootstrapping, because these algorithms create subsets of the original dataset with replacement by default.

7.1.4. Gradient boosting machine regression

An initial GBM model including all variables is trained, in order to estimate the importance of each variable (Figure 7.6). This model was trained based on 10-fold cross-validation, at 10,000 trees, 0.01 reduction of learning rate and maximum 5 nodes per tree. In GBM regression, variable importance refers to the reduction in the loss function given a set of features and is usually termed as relative variable influence. Although a predetermined cut-off point does not exist, just four variables should be kept based on their importance values and the "shoulder" method: $lagXVelocity$ (76.24), $precedingXVelocity$ (21.73), $leadingXVelocity$ (1.14) and $lagXAcceleration$ (0.73). It is important to note that these values do not refer to the percentage of variance explained by each predictor.

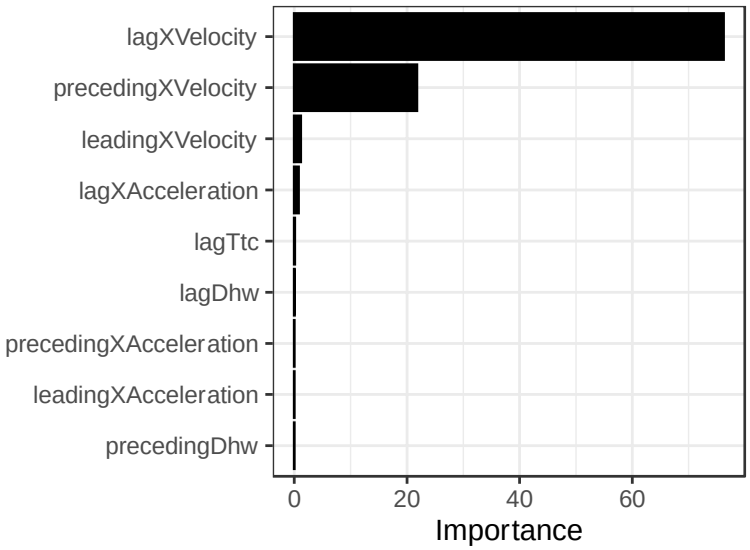


Figure 7.6 Variable importance in gradient boosting machines regression

In comparison to random forests, GBMs require a considerable effort to fine-tune their parameters, which play a crucial role in the minimisation of the loss function. The implementation of GBM in R (package **gbm**) allows to tweak the number of trees, the maximum number of nodes per tree, the reduction of the learning rate (shrinkage) and the minimum number of observations in the terminal nodes of each tree. As always, a usual way to approximate the optimum values of those parameters is to perform a grid search, although in the case of GBM this operation is very time consuming and resource intensive.

A first attempt to estimate the number of trees that suffice for finding the local optimum of the loss function is made. The results indicated that roughly 5000 trees would suffice, if all other parameters were kept as above. However, to study the influence of each parameter and to estimate the best subset of parameters, a grid search with 5-fold cross-validation using the training set is performed. The parameters of this process are: number of trees (3000, 5000, 7000), reduction of the learning rate (0.01, 0.05, 0.10), number of splits at each node (1, 3, 5) and minimum number of observations in the terminal nodes of each tree (5, 10, 15). Using the hardware configuration specified previously, this process took around ten hours to complete.

The results of the grid search (Appendix: Table 9.1) indicate that many models performed almost equally well. The best model with 7000 trees, 0.01 reduction, 5 splits and 5 terminal observations returned an $RMSE = 0.135\ m/s$. However, even simpler models, such as the one with 3000 trees, 0.05 reduction, 3 splits and 5 terminal observations performed almost equally well with $RMSE = 0.139\ m/s$. Therefore, the latter is chosen as the outcome of this grid search, due to its "simplicity", shorter training time and good enough performance.

The last model is trained again within a running time of 156 s. The training set measures of fit are $R^2 \approx 1$, $RMSE = 0.13\ m/s$ and $MAE = 0.10\ m/s$ and those of the test set are $RMSN = 0.01$, $RMSE = 0.14\ m/s$, $MAE = 0.10\ m/s$ and $R^2 \approx 1.00$.

7.2. Free phase

An important early observation from the free phase trajectories indicated that only 0.43% of the data points include a preceding vehicle. This percentage falls even to 0.07% when preceding and leading vehicles are considered. This observation justifies the results of traffic phase clustering and confirms that traffic density is low during the free phase. Hence, there are not many interactions between drivers and much less interactions between three sequential vehicles. This led to the decision to develop a data-driven model taking into account just one preceding vehicle.

The interactions between the velocity of the ego and the preceding vehicle and the time-lagged distance headway between them are shown on Figure 7.7. On the left, it can be observed that distance headway stops being significant in determining the speed of the follower just after 70-80 m, where there is a higher concentration of observations only around 30-40 m/s. On the right, one important observation regarding the speed differences of ego and preceding can be made; when the preceding and the following vehicles are travelling at the same speed (on the 45° diagonal), the distance between them is up to 100 m. When the preceding vehicle has a greater speed than the following and vice versa (not on the 45° diagonal), the distance headway between them grows over 100 m. Expectedly, this means that the follower tries to keep a steady speed and a steady distance to the leader.

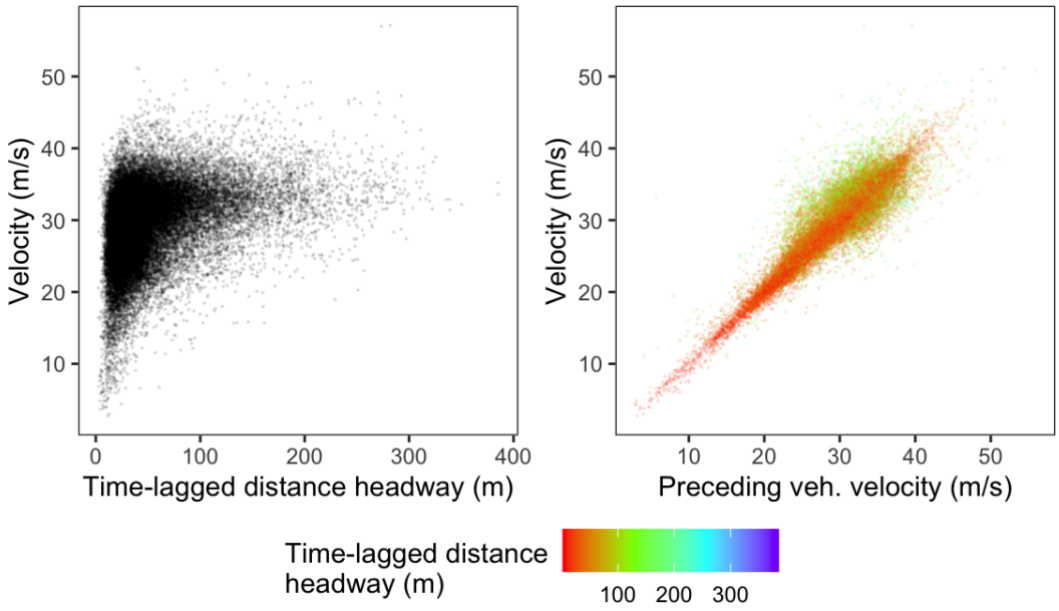


Figure 7.7 Velocity - preceding velocity - distance headway plots in free phase

7.2.1. Multiple linear regression model

Variable Selection

A correlation plot of the variables with time lag $\tau = 1.2\ s$ is shown on Figure 7.8. More or less the same observations as in the congested phase apply here as well. $xVelocity$ is positively

and highly correlated with $lagXVelocity$ and $precedingXVelocity$. The correlation of the distance headway between the ego and the preceding vehicle $lagDhw$ is weakly positive, but still statistically significant. The speed difference $deltaPrecedingXVelocity$ is unexpectedly negatively correlated with $xVelocity$, which means that the velocity of the ego vehicle decreases, when the preceding vehicle drives faster than the ego. The time-to-collision and the acceleration of the preceding do not show statistically significant correlation with $xVelocity$, as their p-values are larger than the significance level $a = 0.05$. It is important to remember that all variables on Figure 7.8 refer to the previous time step i.e. to the current time minus the reaction time $\tau = 1.2\ s$, except of the first four variables that refer to the current time step.

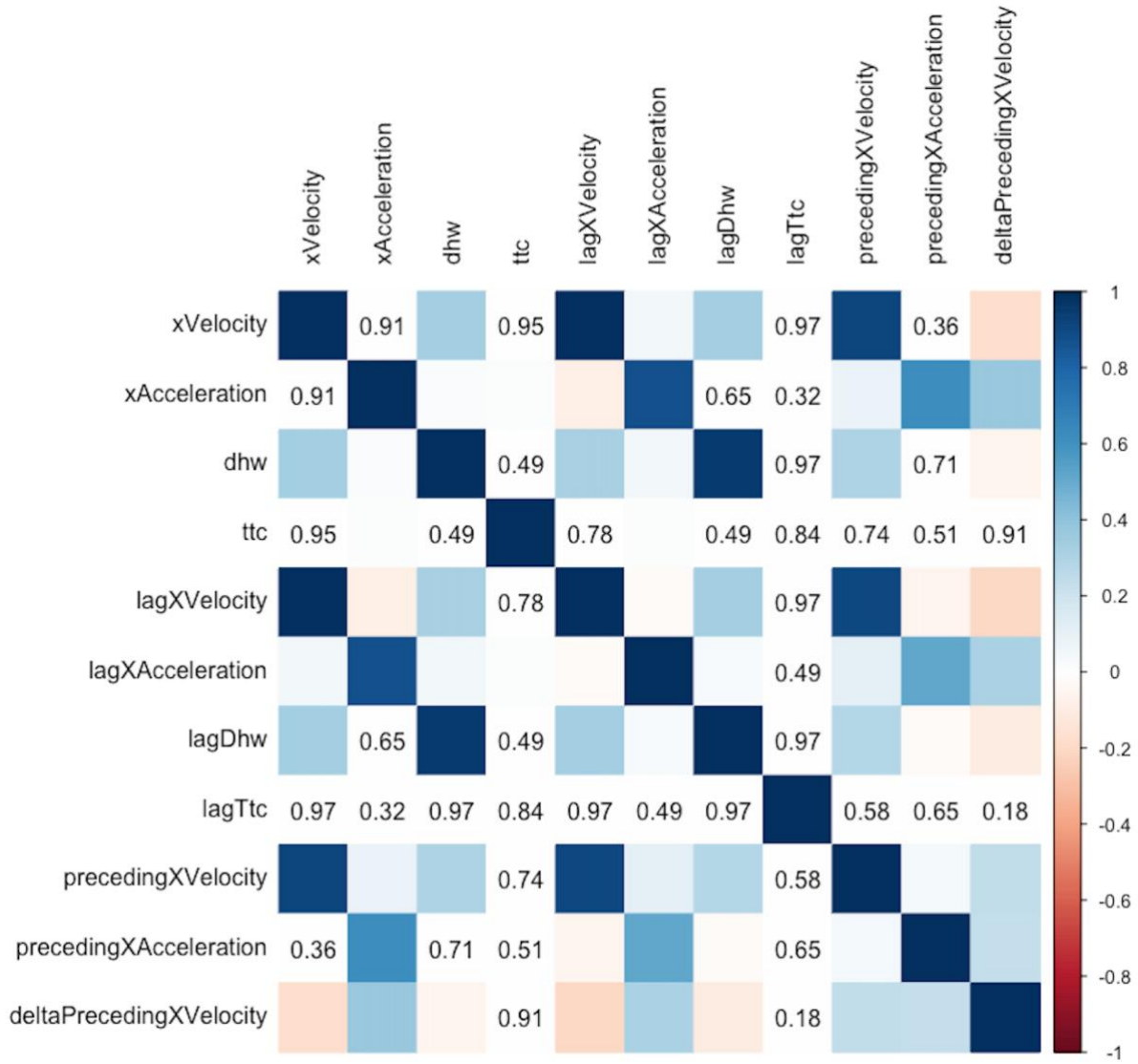


Figure 7.8 Correlation plot for reaction time τ=1.2 s in free phase

The first time-lagged variable to be removed from the dataset is $lagTtc$, which is not correlated with $xVelocity$. Then, $deltaPrecedingXVelocity$ was also removed, because linear combinations of variables cannot be included multiple times in a model. The dataset is split in

training (75%) and test (25%) sets, which have the same distribution of velocities, as shown in Figure 7.9.

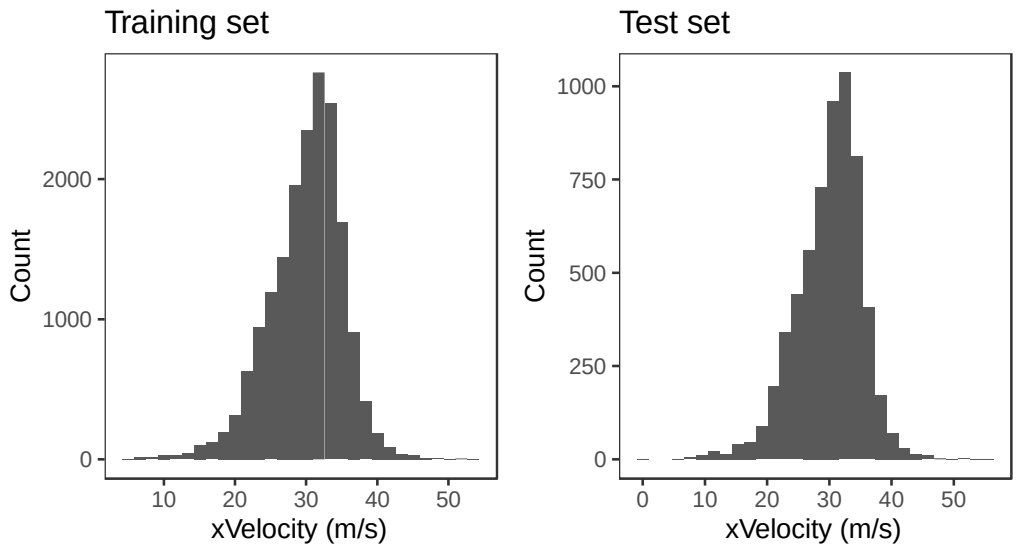


Figure 7.9 Distribution of xVelocity in free phase

Model development

To check the assumption of non-collinearity, a linear model is trained using the remaining variables. Emphasis was placed on $precedingXVelocity$ and the time-lagged distance headway $lagDhw$, as they constitute key variables in mathematical car-following models and probably also in their data-driven counterparts. The $precedingXVelocity$ is found to be multicollinear with $lagXVelocity$, albeit with $VIF \approx 6$, which is situated on the borderline between existence and non-existence of multicollinearity. The same applies to $lagDhw$ with $lagXVelocity$ and $precedingXVelocity$, with VIF values close to 6. By taking into account the limitations above and by comparing different combinations of variables, it is determined that the optimal model of $xVelocity$ should include the variables $lagXVelocity$, $lagXAcceleration$, $precedingXVelocity$ and $precedingXAcceleration$, meaning that the distance headway is not included at all. The precision metrics of the training set indicate an almost perfect fit with $RMSN \approx 0.00$, $RMSE = 0.10\,m/s$, $MAE = 0.07\,m/s$ and $R^2 \approx 1$,. Respectively, they were, $RMSN \approx 0.00$, $RMSE = 0.10\,m/s$, $MAE = 0.07\,m/s$ and $R^2 \approx 1$ using the test set. The regression coefficients of the linear model are shown on Table 7.7 and the residual plots on Figure X. All regression coefficients are statistically significant at significance level $a = 0.05$.

Table 7.7 Coefficients of the linear model including the velocity of the preceding in free phase

Response	(Constant)	lagXVelocity	lagXAcceleration	precedingXVelocity	precedingXAcceleration
xVelocity	0.045	0.989	1.084	0.010	0.115

The regression coefficients show that the velocity of the ego vehicle depends mostly on its time-lagged velocity $lagXVelocity$, rather than the velocity of the preceding vehicle $precedingXVelocity$. A model without precedingXVelocity, which shows multicollinearity with $precedingXVelocity$, returned almost the same results as the complete one. However, it is questionable how this model would perform given unseen data, because the preceding vehicle is essentially missing. Therefore, it is kept for further evaluation using another dataset. All regression coefficients (Table 7.8) are statistically significant at significance level $a = 0.05$.

Table 7.8 Coefficients of the linear model excluding the velocity of the preceding in free phase

Response	(Constant)	lagXVelocity	lagXAcceleration	precedingXVelocity	precedingXAcceleration
xVelocity	0.068	0.998	1.100	-	0.118

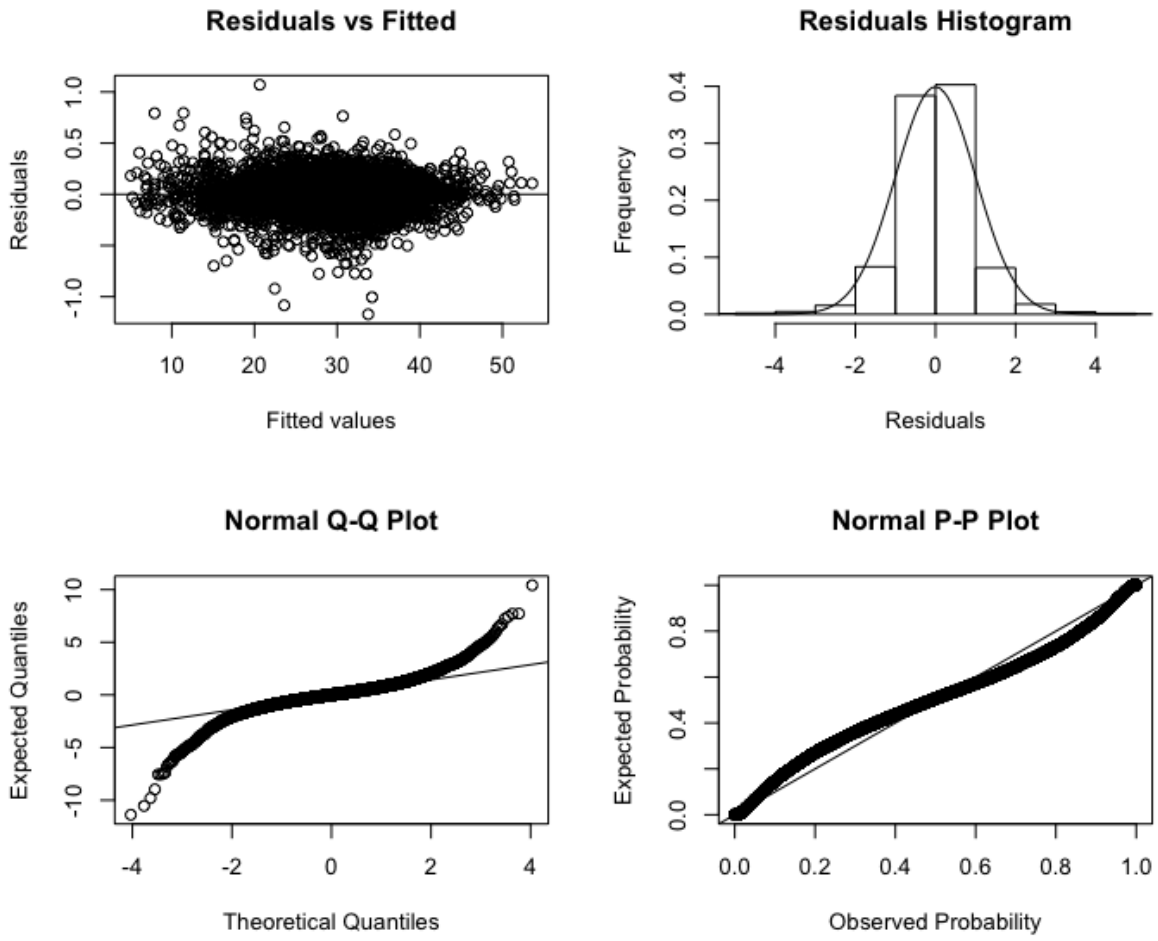


Figure 7.10 Linear regression residuals plot in free phase

A closer look into the residual plots helps to determine if the assumptions of linear regression are satisfied (Figure 7.10). The residuals vs fitted (top left) and residuals histogram (top right) plots seem to confirm the assumption of linearity and homoscedasticity due to an apparently equal distribution of residuals above and below the horizontal zero line and a symmetric distribution around zero on the histogram. However, a Breusch-Pagan test rejects the null hypothesis of homoscedasticity in the model. Based on the Q-Q (quantile-quantile) plot on the bottom left of Figure 7.10, the assumption of normality seems to be valid with deviations from the normal line only at the ends of the plot. Finally, the P-P (probability-probability) plot shows two cumulative distribution functions against each other and also points to the normality of the residuals (bottom right).

7.3. Sensitivity analysis of mathematical car-following models

7.3.1. General information about sensitivity analyses of car-following models

The sensitivity analyses of Gipps' and Krauss' car-following models are performed in two steps. The first step includes a sequential analysis of all model parameters, in order to find the

subset of parameters that minimise the RMSN (Figure 7.11-Figure 7.14), based on the parameter ranges provided in Papathanasopoulou & Antoniou (2015) and using the training sets (75% of the dataset). In the second step, all optimal parameters from the previous step are held constant and a further sensitivity analysis of the reaction time τ is performed by sequentially trialling reaction times between 0.4 and 2.8 s at intervals of 0.4 s using the test sets (25% of the dataset). The results of the second step include the RMSE, RMSN, MAE and R^2 and can be found on Tables 7.9-7.12. This two-step process has been named platoon calibration (Kurtc, 2020), because all trajectories are observed in an aggregated manner and without considering unique driver pairs or driver-vehicle combinations.

7.3.2. Gipps' car-following model on congested trajectories

The evaluation statistics are optimised when the maximum desired speed $V_n = 16\ m/s$, the safety distance between the ego and the preceding vehicle $S_{n-1} = 5.6\ m$, the maximum desired acceleration $a_n = 0.8\ m/s^2$, the maximum desired deceleration $b_n = -5\ m/s^2$, the maximum estimated deceleration $\hat{b} = -3\ m/s^2$ and the reaction time is $\tau = 0.4\ s$. This parameter subset results in $RMSN = 0.04$, $RMSE = 0.45\ m/s$, $MAE = 0.27\ m/s$ and $R^2 = 0.99$ when applied on the test set.

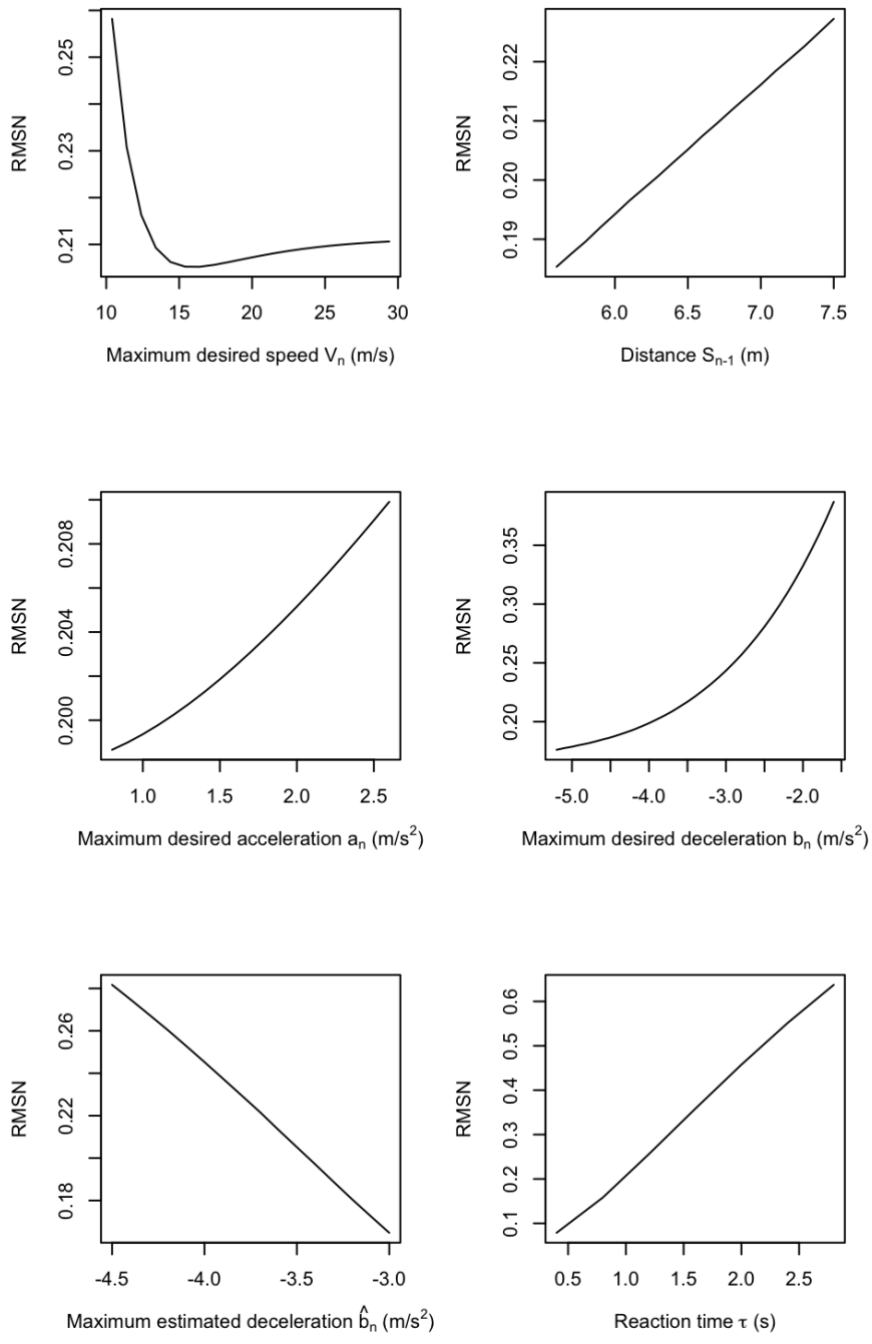


Figure 7.11 Sensitivity analysis of Gipps' car-following model in congested phase

Table 7.9 Sensitivity analysis of the reaction time using Gipps' model in congested phase (test set)

τ (s)	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**
0.4	0.04	0.45	0.27	0.99
0.8	0.09	0.93	0.59	0.96
1.2	0.15	1.61	1.14	0.92
1.6	0.24	2.56	2.04	0.87
2.0	0.34	3.58	3.08	0.82
2.4	0.43	4.59	4.11	0.77
2.8	0.53	5.62	5.15	0.71

7.3.3. Gipps' car-following model on free trajectories

The respective optimal parameter values in this model are $V_n = 29.6\ m/s, S_{n-1} = 5.6\ m, a_n = 0.8\ m/s^2$, $b_n = -5\ m/s^2$, $\hat{b} = -3\ m/s^2$ and the reaction time is $\tau = 0.4\ s$. This parameter subset results in $RMSN = 0.01, RMSE = 0.20\ m/s$, $MAE = 0.15\ m/s$ and $R^2 = 1.00$ when applied on the test set.

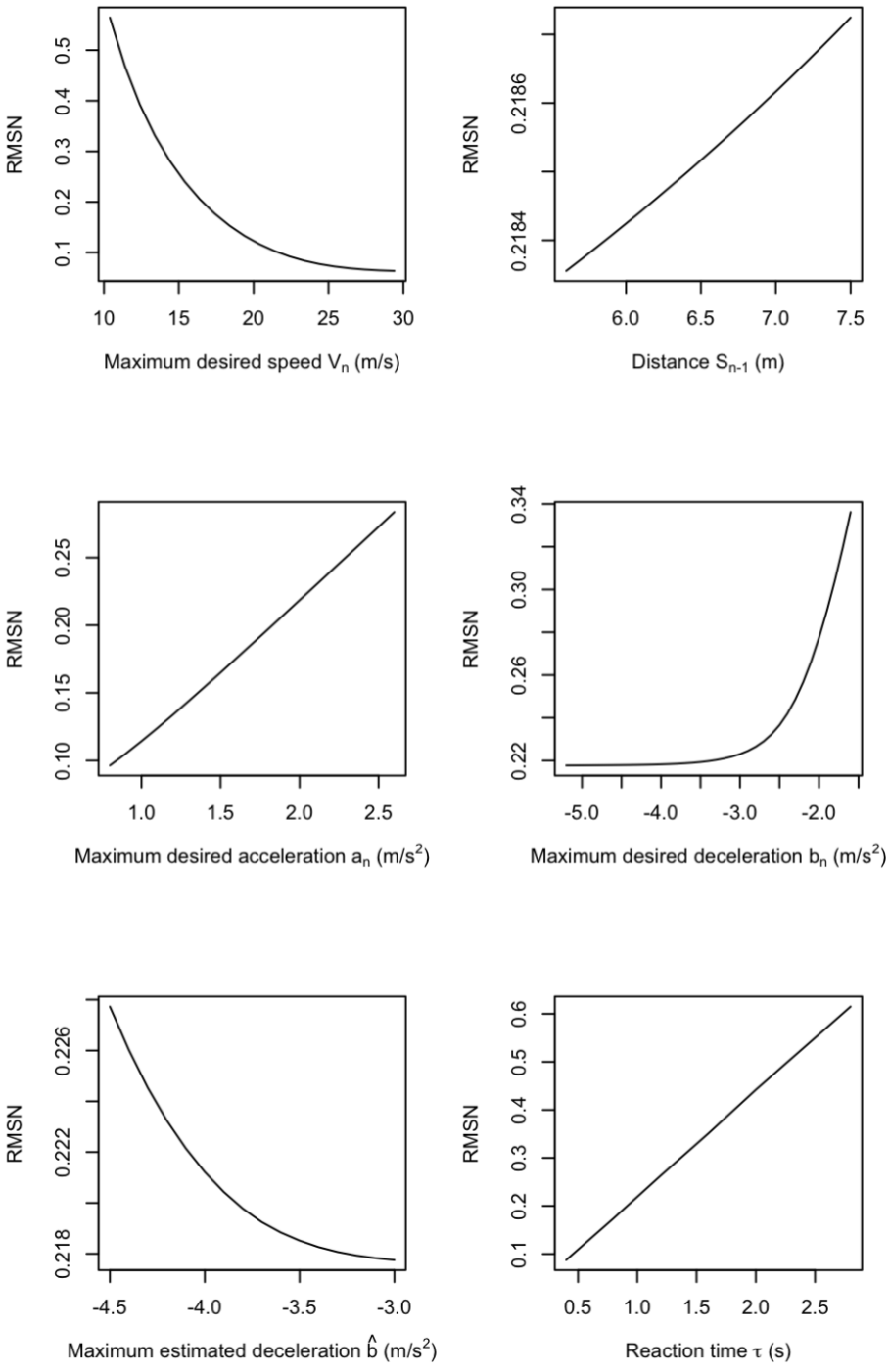


Figure 7.12 Sensitivity analysis of Gipps' car-following model in free phase

Table 7.10 Sensitivity analysis of the reaction time using Gipps' model in free phase (test set)

τ (s)	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**
0.4	0.03	1.03	0.94	1.00
0.8	0.07	2.06	1.90	0.99
1.2	0.10	3.08	2.82	0.99
1.6	0.14	4.11	3.82	0.97
2.0	0.18	5.15	4.84	0.94
2.4	0.22	6.43	6.09	0.90
2.8	0.26	7.64	7.35	0.86

7.3.4. Krauss' car-following model on congested trajectories

The evaluation statistics in this case are optimised for maximum acceleration $\alpha_{max} = 0.8\ m/s^2$, desired deceleration $b = -1.6\ m/s^2$, with the minimum noise parameter $\varepsilon = 0.1$ and at reaction time $T = 1.2\ s$. This parameter subset results in $RMSN = 0.17$, $RMSE = 1.79\ m/s$, $MAE = 1.17\ m/s$ and $R^2 = 0.87$.

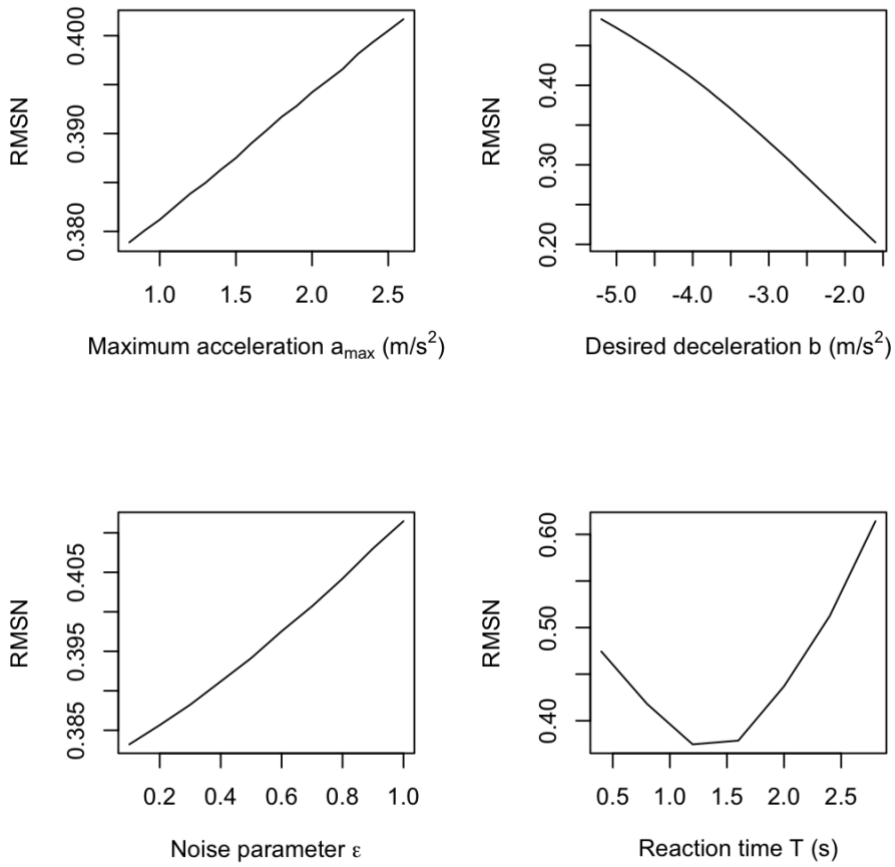


Figure 7.13 Sensitivity analysis of Krauss' car-following model in congested phase

Table 7.11 Sensitivity analysis of the reaction time using Krauss' model in congested phase (test set)

τ (s)	RMSN	RMSE (m/s)	MAE (m/s)	R^2
0.4	0.22	2.38	1.76	0.87
0.8	0.19	2.04	1.36	0.88
1.2	0.17	1.80	1.18	0.87
1.6	0.16	1.71	1.23	0.87
2.0	0.18	1.86	1.49	0.86
2.4	0.21	2.22	1.84	0.83
2.8	0.24	2.59	2.21	0.82

7.3.5. Krauss' car-following model on free trajectories

The optimal parameters of Krauss car-following model on high-speed highway traffic are $\alpha_{max} = 0.8\, m/s^2$, $b = -1.6\, m/s^2$, $\varepsilon = 0.1$ and reaction time $T = 2.0\, s$. The evaluation statistics equal $RMSN = 0.08, RMSE = 2.28$ m/s, $MAE = 1.63$ m/s and $R^2 = 0.78$.

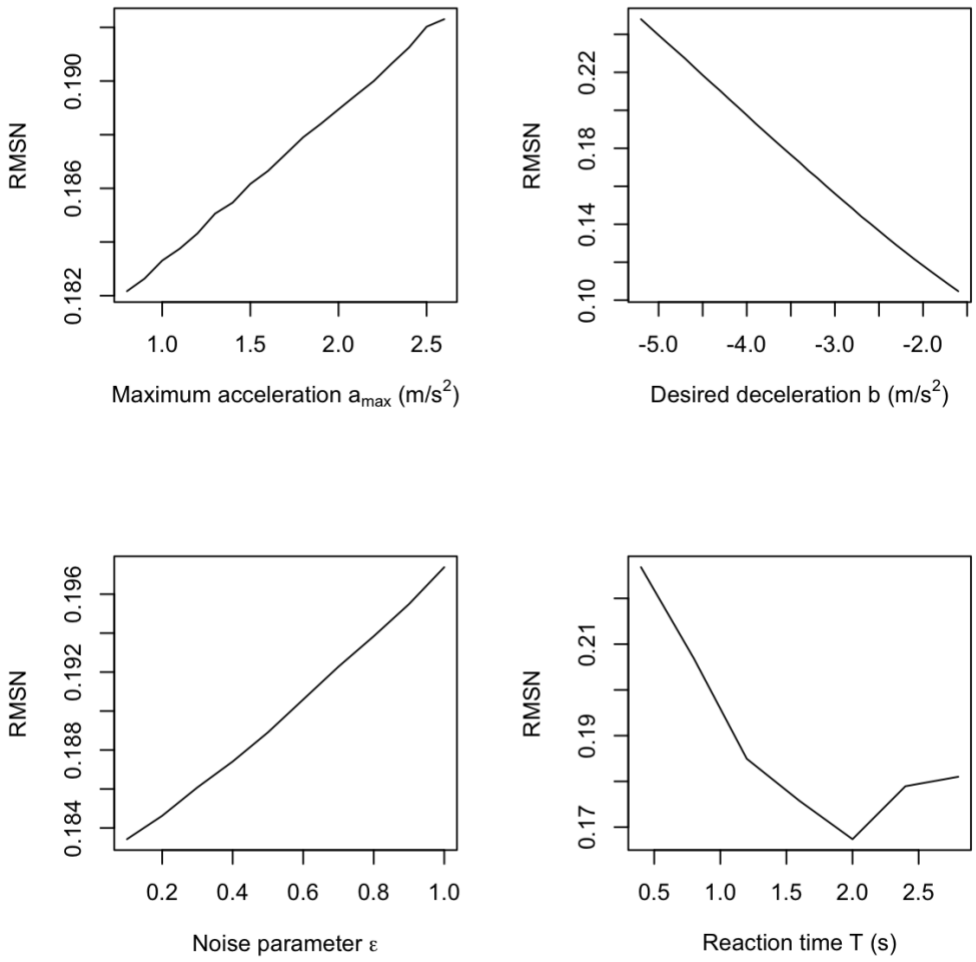


Figure 7.14 Sensitivity analysis of Krauss' car-following model in free phase

Table 7.12 Sensitivity analysis of the reaction time using Krauss' model in free phase (test set)

τ (s)	RMSN	RMSE (m/s)	MAE (m/s)	R^2
0.4	0.11	3.37	2.08	0.75
0.8	0.10	3.03	1.81	0.74
1.2	0.09	2.68	1.62	0.77
1.6	0.09	2.61	1.67	0.75
2.0	0.09	2.52	1.75	0.77
2.4	0.09	2.55	1.93	0.79
2.8	0.09	2.72	2.18	0.77

7.3.6. Observations about the sensitivity analysis of mathematical models

Both models seem to have advantages and disadvantages. With regards to the congested phase, Gipps' model seems to perform better for shorter reaction times, while the car-following model of Krauss seems to have almost constantly good evaluation metrics and depends less on the reaction time. During the free-flowing phase, Gipps' model performs better than the car-following models of Krauss up to a reaction time of $\tau = 2.0\ s$, but then its performance deteriorates significantly. On the other hand, Krauss' model has an inferior performance, which does not depend heavily on the reaction time. Therefore, reaction times τ or T are a major source of uncertainty, especially when considering the fact that the true reaction time is unknown (and different for every driver-vehicle combination) and further investigation into that matter is necessary in the future. Finally, a somewhat less important advantage of Krauss' model is that it contains just four parameters, while Gipps' model involves six parameters and hence, the calibration of its parameters is computationally more laborious.

7.4. Sensitivity analysis of data-driven models

7.4.1. Sensitivity analysis in congested phase

The sensitivity analysis of the data-driven car-following models is similar to the second step of the sensitivity analysis of the mathematical models. The accuracy of the trained models is tested by plugging in test sets of different reaction times τ from 0.4 to 2.8 s. It is important to note that the models are not trained again using the new reaction time, but the previously trained models with $\tau = 1.2\ s$ are used. The results of the sensitivity analysis are shown on Tables 7.13-7.16.

Table 7.13 Sensitivity analysis of the linear regression model in congested phase

τ (s)	RMSN	RMSE (m/s)	MAE (m/s)	R^2
0.4	0.09	1.00	0.71	0.94
0.8	0.09	0.92	0.64	0.95
1.2	0.08	0.86	0.60	0.96
1.6	0.08	0.87	0.61	0.96
2.0	0.09	0.90	0.65	0.95
2.4	0.09	0.96	0.71	0.95
2.8	0.10	1.05	0.78	0.94

Table 7.14 Sensitivity analysis of the elastic net regression model in congested phase

τ (s)	RMSN	RMSE (m/s)	MAE (m/s)	R^2
0.4	0.03	0.37	0.28	0.99
0.8	0.02	0.19	0.14	1.00
1.2	0.01	0.14	0.10	1.00

1.6	0.03	0.29	0.22	1.00
2.0	0.04	0.48	0.36	0.99
2.4	0.06	0.67	0.51	0.97
2.8	0.08	0.85	0.65	0.96

Table 7.15 Sensitivity analysis of the random forest model in congested phase

τ (s)	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**
0.4	0.03	0.36	0.28	0.99
0.8	0.02	0.19	0.14	1.00
1.2	0.01	0.11	0.06	1.00
1.6	0.03	0.30	0.21	1.00
2.0	0.04	0.47	0.35	0.99
2.4	0.06	0.64	0.49	0.98
2.8	0.08	0.85	0.64	0.96

Table 7.16 Sensitivity analysis of the gradient boosting machines model in congested phase

τ (s)	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**
0.4	0.04	0.36	0.30	0.99
0.8	0.02	0.19	0.16	1.00
1.2	0.01	0.11	0.10	1.00
1.6	0.03	0.30	0.21	1.00
2.0	0.04	0.47	0.35	0.99
2.4	0.06	0.64	0.50	0.98
2.8	0.08	0.85	0.64	0.96

The comparison of different car-following models on Figure 7.15. shows that the double data-driven models generally overperform the mathematical models, with their advantages being prevalent on two different levels. First, it can be seen that the data-driven models return more stable precision metrics; regardless of the reaction time, the RMSN does not go over 0.10, the values of RMSE and MAE almost never go above 1 and the R^2 does not fall bellow 0.95 respectively. On the other hand, the mathematical models depend heavily on the reaction time, with Krauss' model showing a more stable performance over time than Gipps' car-following model. Second, the precision metrics of data-driven models are better in almost all cases, the only notable exception being the linear model at $\tau = 0.4\ s$ in comparison to Gipps'. The differences become more pronounced as the reaction time increases. Figure 7.15 shows the results of the sensitivity analysis, including the mathematical car-following models (Krauss and Gipps) from Tables 7.9-7.12.

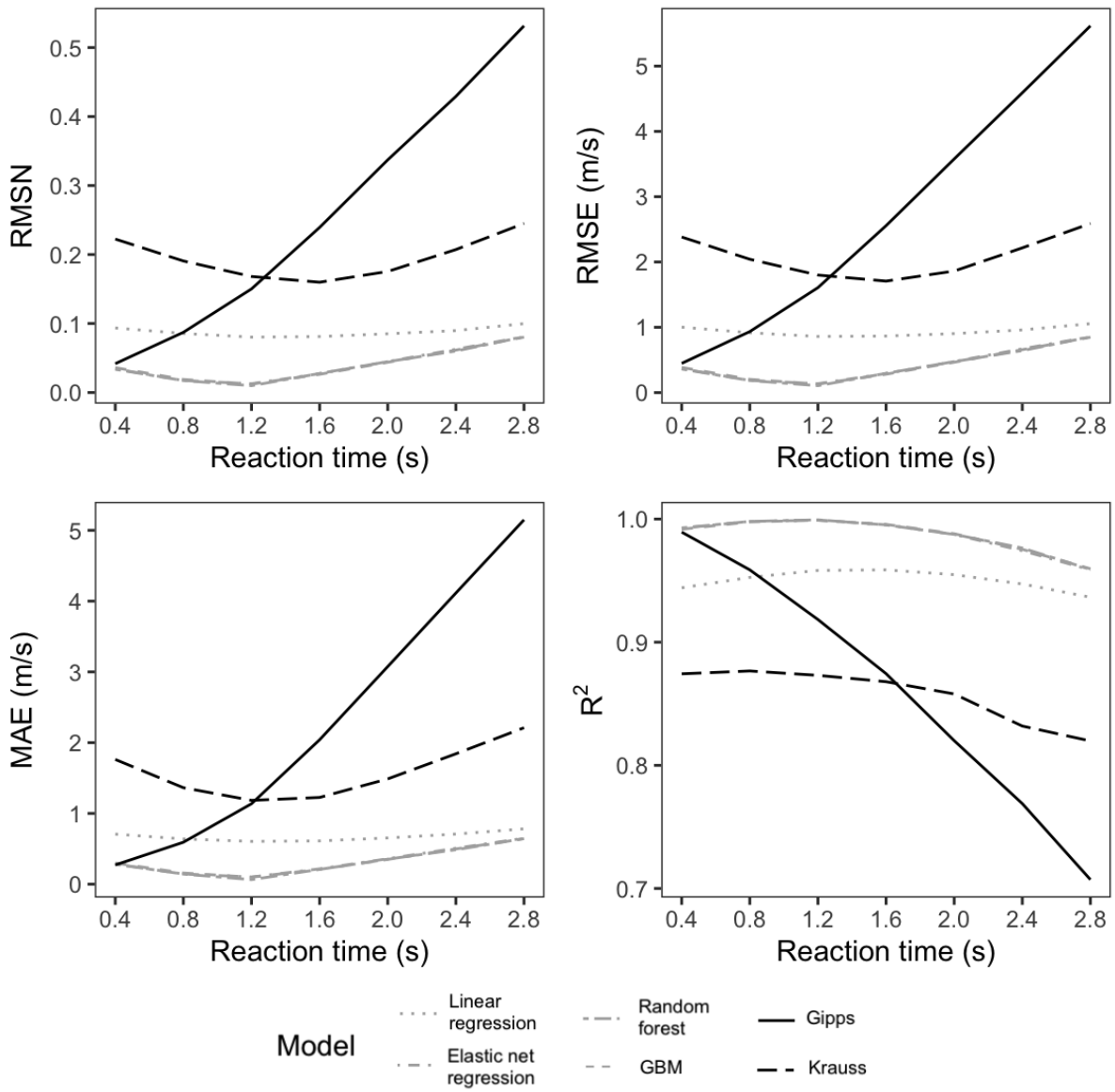


Figure 7.15 Sensitivity analysis of data-driven and mathematical models in congested phase (test set)

7.4.2. Sensitivity analysis in free phase

Later on, different reaction times are applied on the linear regression car-following model without re-training it, in order to test its accuracy when reaction time changes. Different test sets of reaction times τ from 0.4 to 2.8 s at increments of 0.4 s are trialled and the results of this sensitivity analysis are shown on Tables 7.17-7.18. As expected, the evaluation metrics are optimal around $\tau = 1.2\ s$, which was used to train the linear model. A comparison with Gipps' and Krauss' car-following models shows that both are being outperformed by the first and the second linear models. In free phase, Krauss' model seems to have a stable performance, which is worse than Gipps' for the first two reaction times (0.4 and 0.8 s). On the other hand, Gipps' performance is initially almost on par with the data-driven models but deteriorates sharply soon after (Figure 7.16).

Table 7.17 Sensitivity analysis of the linear regression model (including preceding velocity) in free phase

τ (s)	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**
0.4	0.01	0.30	0.22	1.00
0.8	0.01	0.16	0.11	1.00
1.2	0.00	0.11	0.07	1.00
1.6	0.01	0.24	0.17	1.00
2.0	0.01	0.39	0.27	0.99
2.4	0.02	0.51	0.38	0.99
2.8	0.02	0.65	0.48	0.98

Table 7.18 Sensitivity analysis of the linear regression model (excluding preceding velocity) in free phase

τ (s)	**RMSN**	**RMSE (m/s)**	**MAE (m/s)**	**R^2**
0.4	0.01	0.30	0.22	1.00
0.8	0.01	0.16	0.11	1.00
1.2	0.00	0.11	0.07	1.00
1.6	0.01	0.24	0.17	1.00
2.0	0.01	0.39	0.27	0.99
2.4	0.02	0.51	0.38	0.99
2.8	0.02	0.65	0.48	0.98

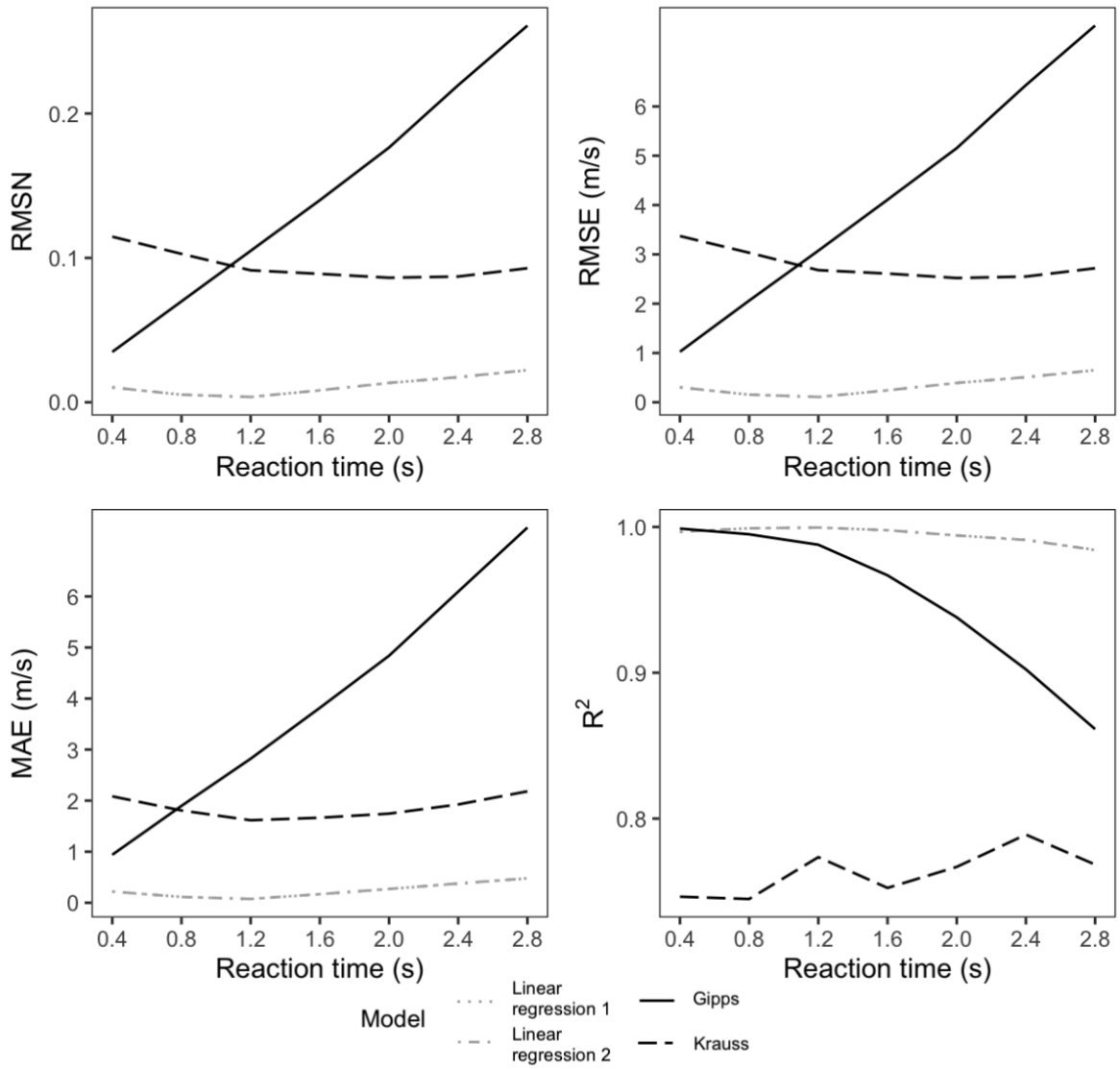


Figure 7.16 Sensitivity analysis of data-driven and mathematical models in free phase (test set)

Nevertheless, data-driven models in most cases are not interpretable, which is an important disadvantage compared to mathematical models. Except of the linear and elastic net regression models, random forests and gradient boosting machine models function more or less like "black boxes", as the process that produces the results cannot be really observed. Therefore, the user can influence only a few parameters and the size of the models. In those cases, it is important to ensure that the models have not been overfitted on the highD dataset or, in other words, to make sure that the model does not conform too closely to the training set, as is the case in many car-following models (Ciuffo et al., 2012). To fulfill this requirement, unseen datasets have to be plugged into the trained models and the new precision metrics have to be evaluated and compared to the metrics from the mathematical model.

7.5. Validation of the data-driven and mathematical car-following models

7.5.1. Validation in congested phase

The re-extracted NGSIM dataset of Coifman and Li (2017) was chosen for validating the above data-driven car-following models. This dataset features car, motorcycle and truck trajectories (196400 observations) from I-80 between 04:00 – 04:15 and its simple format makes it suitable for use in place of the highD dataset. Only the velocity trajectories of cars were used for this study, which resulted in a substantially lower number observations than the initial dataset – coupled with the necessary adjustments due to the reaction time, reaching 12255 ($\tau = 0.4\ s$), 4090 ($\tau = 1.2\ s$), 2430 ($\tau = 2\ s$) and 1755 ($\tau = 2.8\ s$) observations. Another important reason for the reduction of that magnitude is that double car-following models require one preceding and one leading vehicle, which seems to be a somewhat uncommon occurrence in NGSIM.

Even so, there exists an additional advantage that justifies the suitability of NGSIM as validation dataset. The clustered velocity-density and velocity-flow diagrams (Figure 6.3) have shown that the transition from the congested to the free phase happens at around 70 km/h in highD. The histogram of velocity from NGSIM (Figure 7.17) shows that the vast majority of the observations stands under 70 km/h, which hints that the entire dataset from I-80 between 04:00 – 04:15 would have been clustered under the congested phase. Although this observation serves the purpose of validating the data-driven models in the congested phase for the time being, it should be taken with caution. On the one hand flow, density and speed depend largely on infrastructure, speed limits, traffic flow composition etc., and on the other hand observations of higher speeds are not available in the dataset of (Coifman & Li, 2017), which means that higher speeds might not be possible at all on this road segment.

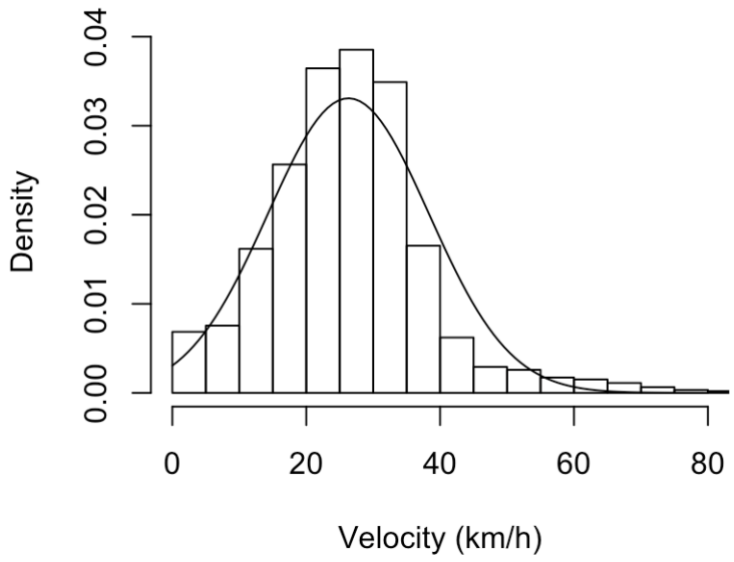


Figure 7.17 Distribution of longitudinal velocity in the re-extracted NGSIM dataset of Coifman and Li (2017)

Nevertheless, this dataset still constitutes a good alternative to highD. The only missing variable is the time-to-collision (TTC) safety surrogate measure, which can be calculated from the existing variables after applying the reaction time emulation procedure. After that, the linear, elastic, random forest and GBM models are applied on the dataset for reaction times of 0.4, 1.2, 2.0 and 2.8 s and the usual precision metrics are calculated. The selection of the reaction times is not random; 0.4 s is the best reaction time in Gipps' model, 1.2 s was used in the training procedure and is the best reaction time in Krauss' congested phase calibrated model, 2.0 s is the optimal reaction time in Krauss' free phase model and 2.8 s are the maximum reaction time to be included in this study. The results are shown in Figure 7.18, where RMSE and MAE are given in metres per second and RMSE and R^2 are dimensionless.

Many observations can be made based on Figure 7.18. Krauss' model showed a more stable performance than Gipps', with the former performing worse for reaction times 0.4 and 1.2 s and better at 2.0 s than the latter, confirming the results of the original sensitivity analysis of the highD dataset in congested phase. The data-driven models overperformed the mathematical models with the exception of Gipps' at 0.4 s, which is the optimal value coming from the sensitivity analysis. Similarly, the errors of data-driven models were lowest at 1.2 s, which justifiable given that they were trained on a dataset based on that reaction time. Among them, there seems to be no clear winner, as the elastic net, random forest and gradient boosting machines models return quite similar performance metrics, whereas the first linear regression model stands somewhat worse. Generally, double car-following models returned satisfactory results in NGSIM, almost regardless of the reaction time.

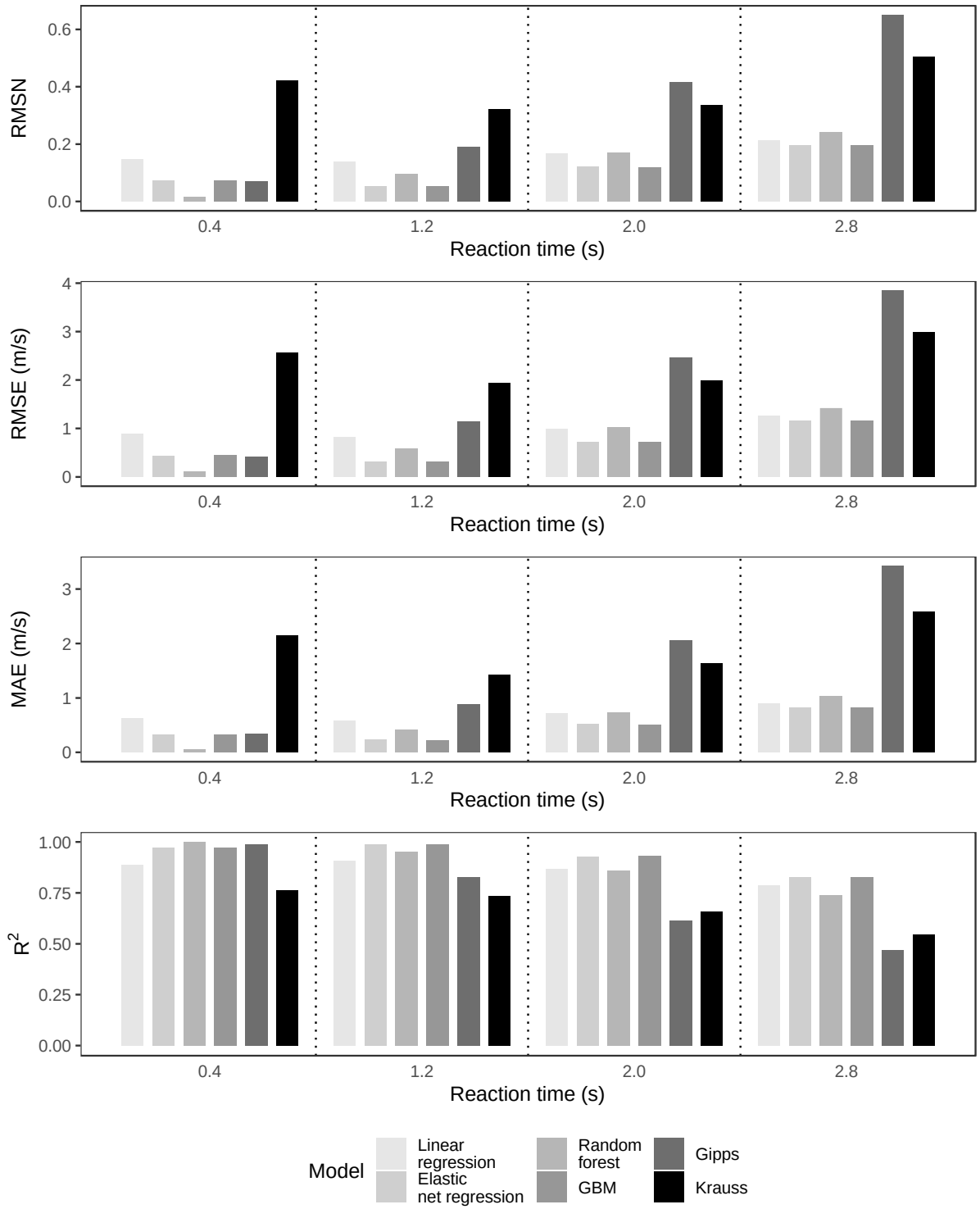


Figure 7.18 Validation results in congested phase

7.5.2. Validation in free phase

All two-lane tracks from high D were not included in GMM clustering. The reason is that flow and density on two-lane highways are not comparable with those of three-lane tracks and therefore, they had to be handled differently than the rest of the dataset. However, they present an opportunity for validating the data-driven and mathematical car-following models, as they can be considered as a dataset "not seen" by the calibrated models.

It is obvious that all two-lane tracks would have been clustered in the free phase. Figure 7.19 shows the distribution of $xVelocity$ from two-lane tracks, where an average velocity of over 100 km/h is apparent. Recalling the fact that the transition from congested to free phase happens at around 70 km/h (on three-lane highways), the overwhelming majority of two-lane observations are well above this speed level. Of course, different road geometries definitely result in different clustering results; however, the average velocity of two-lane trajectories is close to 130 km/h i.e. the drivers were most probably limited by the speed limit rather than by their interactions with the preceding vehicles.

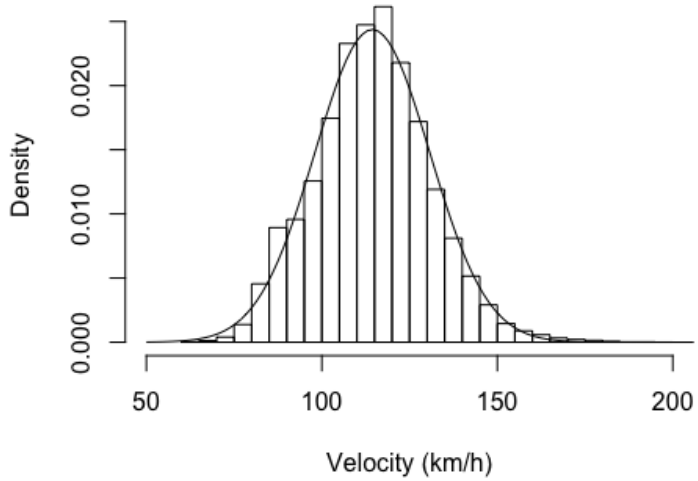


Figure 7.19 Distribution of longitudinal velocity in two-lane Tracks of highD

The results from the validation of the free-phase models show similarities with those from the congested phase. Both linear models (with and without $precedingXVelocity$) constantly outperform the mathematical models of Gipps and Krauss, despite the fact that the second linear model does not include the velocity of the preceding vehicle at all. Therefore, it can be said that the velocity of the ego vehicle at $t + \tau$ can be estimated from its velocity, acceleration and acceleration of the preceding vehicle at t. Also, neither linear model includes any variable related to the distance headway as it usually happens with car-following models.

Among the mathematical models, Gipps' car-following model performs well when confronted with time-lagged velocity trajectories at $\tau = 0.4\ s$, which is the optimal value of this parameter, but later it produced higher errors than the data-driven models. Krauss' model showed an

almost stable behaviour, with errors constantly higher than those of the data-driven models. However, the R^2 coefficient of determination was low, which probably reveals that Krauss' model was not able to capture any accelerations and decelerations and instead, produced the exactly opposite result (Figure 7.20).

The supremacy of the linear models can be attributed to the fact that almost all vehicles on the highway move at a stable velocity close to the speed limit, which is usually a linear function of the leader's velocity or the velocity of the ego vehicle in the previous time interval. Gipps' and Krauss' models assume nonlinear interactions and therefore, they cannot adapt to very high and steady velocities in highD.

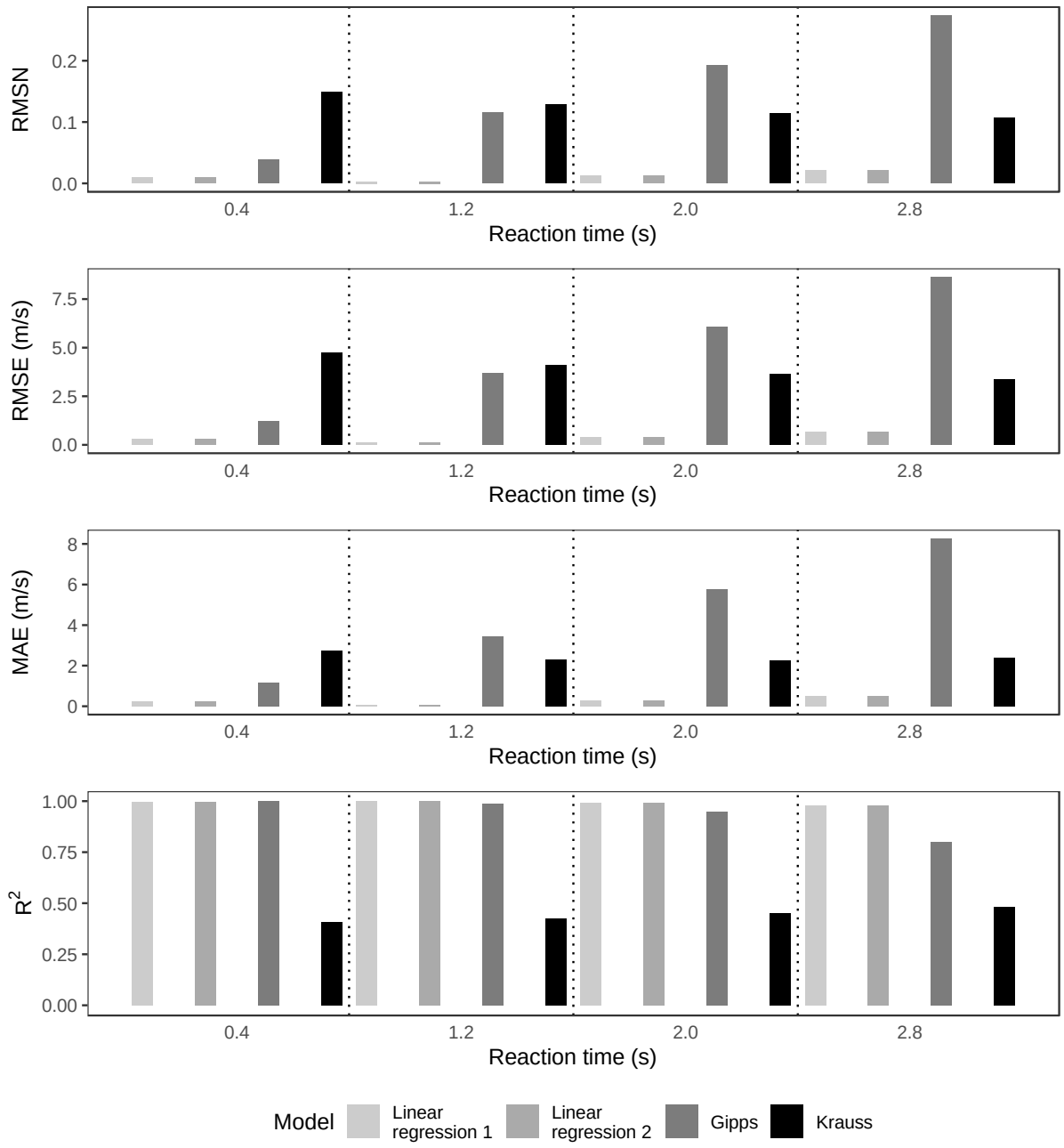


Figure 7.20 Validation results in free phase

Part IV: Further Proposals and Conclusions

8. Proposals for future work

The highD dataset offers many opportunities in microscopic and macroscopic traffic flow studies. From the analysis in this study, it has become apparent that many variables have been used only as aggregated measures or have not been used at all. The following paragraphs will focus on lane-based traffic density information and on vehicle types. The former has been extracted using virtual loop detectors as part of the exploratory data analysis and later, as part of traffic phase identification, where it produced inferior results in comparison to traffic density by direction. The latter has not been used at all in this study, as the developed data-driven and mathematical car-following models refer just to passenger vehicles (cars) and totally ignore trucks.

The correlation plot of all variables with velocity in the congested phase, including density information by lane, is shown on Figure 8.1. Obviously, traffic density is negatively correlated with velocity, because increased density presents an obstruction to free choice of speed and vice versa. However, it is important that the correlations of $densityRight$, $densityMiddle$, $densityLeft$ with $xVelocity$ are statistically significant, which means that these variables could potentially improve at least the linear regression models.

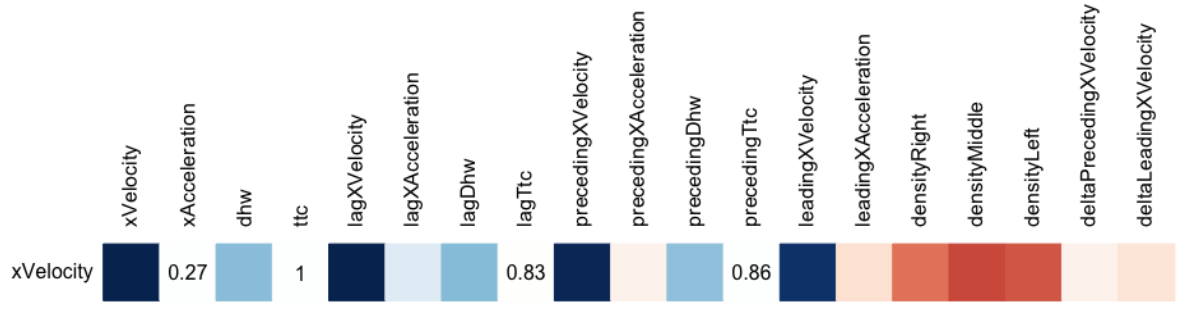


Figure 8.1 Correlation plot in congested phase including density per lane

On the other hand, there are a couple of limitations that should be considered when introducing macroscopic characteristics in microscopic car-following models. First, traffic density data depend largely on the geometry of a road; for example, data-driven models for three-lane highways might not be applied interchangeably on a highway with two lanes, as macroscopic data from one lane will be missing. In this case, a different model would have to be specially trained to account for this issue. Second, as car-following models look ahead in time, an important issue is the availability of traffic density data. Depending on the aggregation interval, which was 30 sec in this study, a projection of traffic density in the near future is necessary.

This limitation can be potentially overcome by integrating a sliding window for traffic density or through forecasting techniques from nearby inductive loop detectors.

Another information that has not been used in the present study is the type of each vehicle. HighD includes only two vehicle types (car and truck) in comparison to other datasets such as the NGSIM, which includes passenger vehicles, trucks and motorcycles, and pNEUMA (Barmpounakis & Geroliminis, 2020), which includes cars, taxis, busses, medium vehicles, heavy vehicles and motorcycles, but contains footage from busy urban settings. The boxplots of velocity, distance headway and time-to-collision by vehicle type from highD are shown in Figure 8.2, based on vehicle pairs (ego and preceding). As expected, cars have a higher average speed and keep lower distance headways and time-to-collision. Conversely, truck drivers generally develop lower speeds and keep a greater distance to preceding vehicles, as reflected on their distance headway and time-to-collision boxplots.

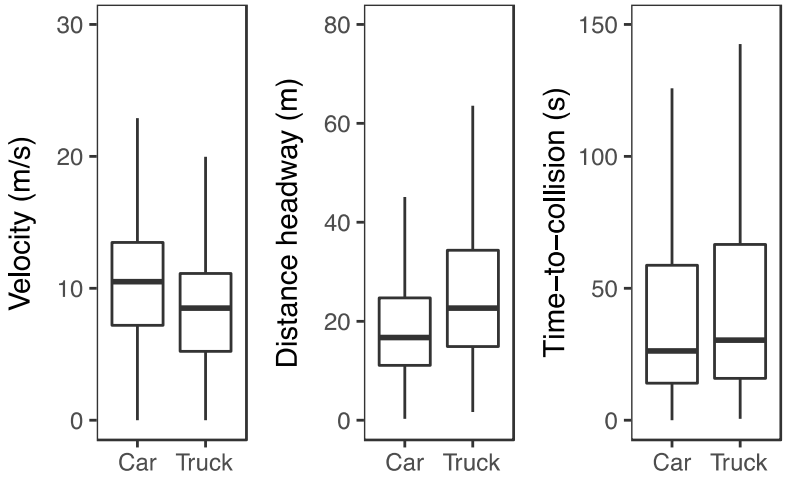


Figure 8.2 Boxplots of velocity, distance headway and time-to-collision for different vehicle types

The boxplots in Figure 8.3 deal with interactions of vehicle pairs. It can be seen that trucks constitute limiting factors when velocity is considered, as both car – truck and truck – car pairs have lower average velocity than car – car pairs. However, when considering the boxplots of distance headway, it is obvious that cars keep a shorter distance to any vehicle type, whereas trucks keep a longer one as in previous figure. Regarding the TTC, car – car and truck – car pairs are slightly lower than car – truck pairs, meaning that any vehicle behind a car tends to let smaller headway and velocity difference to the preceding car.

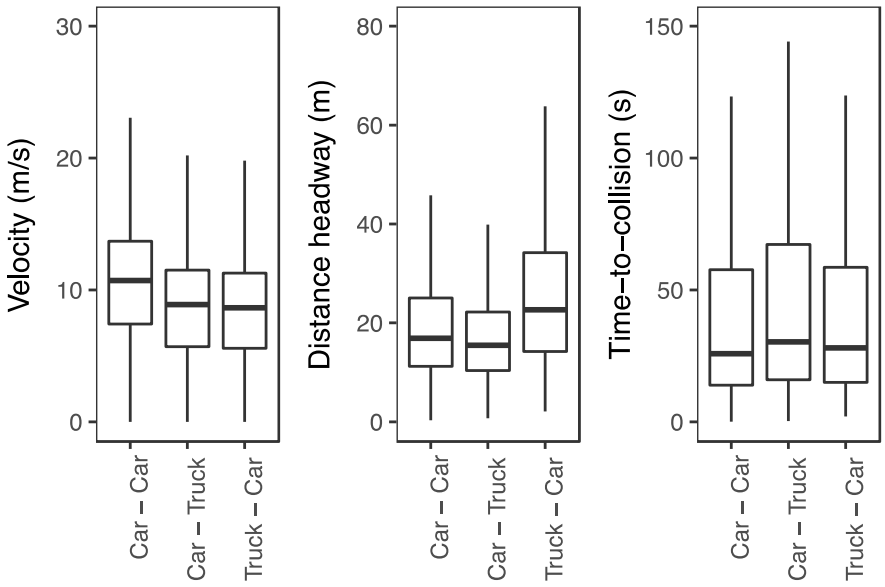


Figure 8.3 Boxplots of velocity, distance headway and time-to-collision for different vehicle pairs

To reinforce the observations from Figure 8.2 and Figure 8.3, some elementary statistical tests regarding the means of the samples car – truck and truck – car were performed. T-tests of dhw and $xVelocity$ at significance level $a = 0.05$ indicated that the means of those two populations are different, by rejecting the null hypothesis:

H_0 : The means of two populations are equal

Point biserial correlation measures the relationship between a binary and a continuous variable. Vehicle type is mapped as a binary variable $class$, where 'car' equals zero and 'truck' equals one. The point biserial correlation coefficients of $class$ with $xVelocity$ and dhw are -0.18 and 0.20 respectively; since the coefficient of $class$ and $xVelocity$ is negative, this indicates that when $class$ takes the value 1 (trucks), then $xVelocity$ tends to take lower values compared to when $class$ equals zero (cars). The opposite can be said for the correlation coefficient between $class$ and dhw, which is positive; $class$ 1 tends to take higher values than $class$ zero. The biserial correlation coefficients justify the boxplots in Figure 8.2 and Figure 8.3. Hence, it seems that variable $class$ is statistically significant i.e. differences in behaviour according to vehicle type pairs do exist.

All in all, involving more variables in model building can potentially increase the accuracy of the proposed models. Many subjects came up while developing the current thesis, especially concerning variations of the methodology, validation techniques and new applications. Possible subjects of investigation include:

- In depth investigation of the impacts of reaction time on the performance of mathematical and data-driven car-following models – real reaction times according to driver characteristics, driving conditions, road morphology and topology, vehicle type etc.

- Investigation of various aggregation intervals in macroscopic traffic analysis and use of different clustering algorithms based on the interpretability of the results. Introduction of more traffic phases.
- Exploration of different training, testing and validation strategies. In this study, all data-driven models were trained at reaction time 1.2 s and different reaction times were applied during validation despite the models being trained only for one reaction time. Another strategy would be to train models separately for each reaction time, validate each model only for its reaction time accordingly and compare the results with the current strategy.
- Extensive literature review of suitable car-following parameter values on highway and urban traffic. Choice of suitable parameter ranges for Krauss' car-following model, in order to bring its performance on par with Gipps' model.
- Application of data-driven models on arterials with dense traffic in order to investigate if they can reproduce traffic dynamics such as platoon formation, transition from free to congested phase, traffic stability etc.
- Embedding of traffic phase and vehicle type as discrete variables in data-driven models. Modelling of interactions with vehicles on adjacent lanes or with vehicles following the ego vehicle.
- Testing of the congested phase model on low-speed, urban traffic and comparison of the results to highway traffic.
- Application of the data-driven models in microscopic traffic simulation packages (e.g. SUMO) in place of mathematical car-following models and comparison of the simulation results. Special focus should be placed on the rationality of driver behaviour during acceleration, deceleration etc.

9. Main findings and conclusions

In the last chapter, firstly the main findings of this thesis are summarised and then, the main conclusions are drawn based on the previous chapters. In the end, a special chapter is dedicated to the references that were included in this report and an appendix with figures and tables that may help in understanding some results.

9.1. Main findings

This paragraph aims to summarise roughly the main results of this study, while avoiding all details that have already been mentioned previously.

- Model-based clustering of macroscopic properties can deliver satisfactory results regarding traffic phase identification – it is necessary to evaluate visually the outcome for its correspondence to current knowledge and past research on traffic phases.
- Data-driven models outperform the mathematical models both on seen and unseen datasets – with very few exceptions. Even simple data-driven models on free phase reproduce velocity trajectories more effectively than mathematical models.
- The calibrated Gipps' model shows good performance when the optimal parameters are applied. Its performance deteriorates quickly while increasing the reaction time. Krauss' model is more robust to changes in reaction times but delivers constantly worse results than data-driven models.
- Lane-based traffic density, vehicle types and many other unused parameters may be able to improve the data-driven models, as preliminary investigations have shown.

9.2. Conclusions

Car-following models are an integral part of any traffic simulation package. Many researchers have tried to construct mathematical models that regulate the velocity of the follower according to the spatiotemporal properties of traffic flow and of the immediate leader, while at the same time providing a physical explanation of those relationships. The recent availability of highly accurate data from drone videography has enabled the expansion of relevant research into models that offer almost no physical explanation but can mimic the car-following behaviour with high accuracy, characterised as data-driven models.

Data-driven car-following models can account for more complex interactions between a driver and the environment. It is possible to consider more than one leading vehicles, in comparison to mathematical formulas that almost always model a one leader – one follower relationship. The highD dataset offers numerous opportunities in this direction, as it includes trajectories from many different vehicles from highways around Cologne, Germany. Among the many variables of this dataset, the dynamic characteristics of each vehicle can be used to model its movement on a highway according to the preceding and/or the leading vehicle, as well as the phase of the traffic stream.

After an extensive exploratory data analysis focussing on the macroscopic properties of traffic, the first part of the methodology concerns the identification of traffic phases. The trajectories of highD are divided into two databases, one for the congested and one for the free phase, using model-based clustering algorithms. Subsequently, the two databases are used separately for the development of data-driven models. In congested phase, vehicles drive closer to one another and therefore, it is possible to include more than one leader. In free phase, drivers keep longer distances between them and it is not possible to model double following behaviours.

The goal of a car-following model is to estimate the velocity of the ego vehicle in the next time step. In this study, the time step refers to the reaction time of a driver, which is emulated using a specially designed data formatting procedure resembling a time-lagged trajectory rematching algorithm. Then, multiple linear regression including penalisation techniques, random forest and gradient boosting machine regression are applied and optimised, in order to create models that predict the velocity of the following vehicle in congested or free phase. An extensive sensitivity analysis of Gipps' and Krauss' mathematical models aims at discovering the optimal values of their car-following parameters.

Thereafter, a two-level evaluation procedure ensures the applicability of data-driven models on unseen data. First, parts of the highD dataset are plugged into the already trained models, where the usual evaluation metrics confirm almost the same performance as observed during training. Second, in the congested phase another dataset, this time from Interstate 80 freeway in California, USA, and in free phase the two-lane trajectories from highD, which have been excluded from the earlier analysis, are used as validation sets. The results are very promising, as explained in the next paragraph.

In general, data-driven models achieve lower errors between the simulated and the original observations of velocity. In congested phase, Gipps' model performs almost equally well for lower values of the reaction time τ but errors increase significantly at higher values of τ. Among the data-driven models, linear regression seems to be the weakest candidate, although it is less sensitive to the increases of τ. The models from elastic net regression, random forest and gradient boosting machines are almost always on par and perform much better than Gipps' and Krauss' models. In free phase, mathematical models are not able to capture the linear relationship between the velocity of the ego, the time-lagged velocity of the ego and the velocity of the preceding vehicle, as they assume a differential behaviour. Data-driven models outperform the mathematical in free phase.

Overall, it can be said that double car-following models are superior to single mathematical models such as Gipps' and Krauss' models. It is the first time that the latter was included in a study of this kind, which involves a sensitivity analysis of the model's parameters. However, the possibilities of highD have not been exhausted in this thesis; many opportunities to improve mathematical and data-driven models alike exist, mainly using information from surrounding vehicles and traffic states of the adjacent lanes. The results of this study are envisioned to improve the capabilities of car-following models through novel data sources and new modelling techniques.

Bibliography

Aghabayk, K., Sarvi, M., Forouzideh, N., & Young, W. (2013). New Car-Following Model Considering Impacts of Multiple Lead Vehicle Types. *Transportation Research Record, 2390*(1), 131-137.

Alessandrini, S., & Sperati, S. (2017). Characterization of forecast errors and benchmarking of renewable energy forecasts. In G. Kariniotakis, *Renewable Energy Forecasting* (pp. 235-256). Sawston, United Kingdom: Woodhead Publishing.

Alvarez Lopez, P., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Wießner, E. (2018). Microscopic Traffic Simulation using SUMO. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2575-2582). Maui, Hawaii, USA: IEEE.

Antoniou, C., Koutsopoulos, H. N., & Yannis, G. (2013). Dynamic data-driven local traffic state estimation and prediction. *Transportation Research Part C: Emerging Technologies, 34*, 89-107.

Auguie, B. (2015). gridExtra: Miscellaneous Functions for "Grid" Graphics. Retrieved from http://CRAN.R-project.org/package=gridExtra.

Bakshi, C. (2020, June 8). *Random Forest Regression*. Retrieved January 31 2021, from gitconnected - Medium: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

Bando, M., Hasebe, K., Nakayama, A., Shibata, A., & Sugiyama, Y. (1995). Dynamical model of traffic congestion and numerical simulation. *Physical Review E, 51*(2), 1035-1042.

Barmpounakis, E. N., Vlahogianni, E. I., & Golias, J. C. (2016). Unmanned Aerial Aircraft Systems for transportation engineering: Current practice and future challenges. *International Journal of Transportation Science and Technology, 5*, 111-122.

Barmpounakis, E. N., Vlahogianni, E. I., & Golias, J. C. (2018). Identifying Predictable Patterns in the Unconventional Overtaking Decisions of PTW for Cooperative ITS. *IEEE Transactions on Intelligent Vehicles, 3*(1), 102-111.

Barmpounakis, E. N., Vlahogianni, E. I., Golias, J. C., & Babinec, A. (2019). How accurate are small drones for measuring microscopic traffic parameters? *Transportation Letters*, *11*(6), 332-340.

Barmpounakis, E., & Geroliminis, N. (2020). On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. *Transportation Research Part C: Emerging Technologies, 111*, 50-71.

Basak, K., Hetu, S. N., Li, Z., Azevedo, C. L., Loganathan, H., Toledo, T., Ben-Akiva, M. (2013). Modeling reaction time within a traffic simulation model. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)* (pp. 302-309). The Hague, Netherlands: IEEE.

Bexelius, S. (1968). An extended model for car-following. *Transportation Research, 2*(1), 13-21.

Bock, J., Krajewski, R., Moers, T., Vater, L., Runde, S., & Eckstein, L. (2019). The inD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories at German Intersections. *arXiv preprint: 1911.07602*.

Boehmke, B. (n.d.). *Gradient Boosting Machines*. Retrieved December 17, 2020, from University of Cincinnati Business Analytics R Programming Guide: http://uc-r.github.io/gbm_regression

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123-140.

Bruce, P., & Bruce, A. (2017). *Practical Statistics for Data Scientists*. Sebastopol, CA, USA: O'Reilly Media, Inc.

Buckley, D. J. (1962). Road traffic headway distributions. *Australian Road Research Board (ARRB) Conference, 1st,* (pp. 153-87). Canberra, Australia.

Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition, 28*(5), 781-793.

Chandler, R. E., Herman, R., & Montrol, E. W. (1958). Traffic Dynamics: Studies in Car Following. *Operations Research, 6*(2), 165-184.

Chen, J., Liu, R., Ngoduy, D., & Shi, Z. (2016). A new multi-anticipative car-following model with consideration of the desired following distance. *Nonlinear Dynamics, 85*, 2705–2717.

Ciuffo, B., Punzo, V., & Montanino, M. (2012). Thirty Years of Gipps' Car-Following Model: Applications, Developments, and New Features. *Transportation Research Record, 2315*(1), 89-99.

Coifman, B., & Li, L. (2017, October 31). *I-80 NGSIM validation*. Retrieved December 23 2020, from Benjamin Coifman's publications: http://www2.ece.ohio-state.edu/~coifman/documents/I80-NGSIM/

Coifman, B., & Li, L. (2017). A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset. *Transportation Research Part B: Methodological, 105*, 362-377.

Cumming, R. W. (1963). The analysis of skills in driving. *Journal of the Australian Road Research Board, 1*(9), 4-14.

Dabiri, S., & Abbas, M. (2018). Evaluation of the Gradient Boosting of Regression Trees Method on Estimating Car-Following Behavior. *Transportation Research Record, 2672*(45), 136-146.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1-38.

Diehl, F., Brunner, T., Truong Le, M., & Knoll, A. (2019). Graph Neural Networks for Modelling Traffic Participant Interaction. *arXiv e-prints*.

Dülgar, Y., Menth, M., Rehborn, H., & Koller, M. (2019). Analysis of Microstructures in Traffic Jams on Highways Based on Drone Observations. *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. Cairo, Egypt: IEEE.

Dülgar, Y., Menth, M., Rehborn, H., & Koller, M. (2019). Heterogeneity of Microscopic Congested Traffic Data Based on Drone Measurements. *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. Graz, Austria: IEEE.

Edie, L. C. (1961). Car-Following and Steady-State Theory for Noncongested Traffic. *Operations Research*, *9*(1), 66-76.

Elsinghorst, S. (2018, November 29). *Machine Learning Basics - Gradient Boosting & XGBoost*. Retrieved December 17, 2020, from Shirin's playgRound: https://www.shirin-glander.de/2018/11/ml_basics_gbm/

Eurostat. (2020, September 20). *Modal split of passenger transport 1990-2018*. Retrieved January 21 2021, from https://ec.europa.eu/eurostat/databrowser/view/tran_hv_psmod/default/bar?lang=en

Evans, L., & Rothery, R. (1977). Perceptual Thresholds in Car-Following—A Comparison of Recent Measurements with Earlier Results. *Transportation Science, 11*(1), 60-72.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, *29*(5), 1189-1232.

Fu, R., Li, Z., Sun, Q., & Wang, C. (2019). Human-like car-following model for autonomous vehicles considering the T cut-in behavior of other vehicles in mixed traffic. *Accident Analysis and Prevention*, *132*, 105260.

Gattuso, D., Cassone, G. C., & Malara, M. (2021). Traffic Flows Surveying and Monitoring by Drone-Video. In C. Bevilacqua, F. Calabrò, & L. Della Spina (Ed.), *New Metropolitan Perspectives. NMP 2020. Smart Innovation, Systems and Technologies, 178*, pp. 1541-1551. Cham, Switzerland.

Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, *15*(2), 105-111.

Greenwell, B. M. (2017). pdp: An R Package for Constructing Partial Dependence Plots. *The R Journal, 9*(1), 421-436.

Greenwell, B., & Boehmke, B. (2020). Variable Importance Plots—An Introduction to the vip Package. *The R Journal*, *12*(1), 343-366.

Greenwell, B., Boehmke, B., & Cunningham, J. (2019). gbm: Generalized Boosted Regression Models. Retrieved from https://CRAN.R-project.org/package=gbm.

Guang-Han, P., & Di-Hua, S. (2009). Multiple car-following model of traffic flow and numerical simulation. *Chinese Physics B, 18*(12), 5420-5430.

Guo, L., Zhao, X., Yu, S., Li, X., & Zhongke, S. (2017). An improved car-following model with multiple preceding cars' velocity fluctuation feedback. *Physica A, 471*, 436-444.

Hahsler, M., Piekenbrock, M., & Doran, D. (2019). dbscan: Fast Density-Based Clustering with R. *Journal of Statistical Software, 91*(1), 1-30.

Hamner, B., & Frasco, M. (2018). Metrics: Evaluation Metrics for Machine Learning. Retrieved from https://CRAN.R-project.org/package=Metrics.

Hao, S., Yang, L., & Shi, Y. (2018). Data-driven car-following model based on rough set theory. *IET Intelligent Transport Systems, 12*(1), 49-57.

Hastie, T., & Tibshirani, R. (2020). *Statistical Learning*. Retrieved December 5 2020, from https://www.edx.org/course/statistical-learning. [edX course]

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.).* New York, USA: Springer. pp. 337–384.

Ho, T. K. (1995). Random Decision Forests. *Proceedings of 3rd International Conference on Document Analysis and Recognition* (pp. 178-282). Montreal, Quebec, Canada: IEEE.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics, 12*, 55-67.

Hoogendoorn, S. P. (2016, February 15). *Traffic Flow Theory and Simulation*. Retrieved December 23 2020, from TU Delft - OpenCourseWare: https://ocw.tudelft.nl/wp-content/uploads/Chapter-4.-Fundamental-diagrams.pdf

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software, 26*(3), 1-22.

Jiang, R., Wu, Q., & Zhu, Z. (2001). Full velocity difference model for car-following theory. *Physical Review E, 64*.

Kanagaraj, V., Asaithambi, G., Kumar, C. H., Srinivasan, K. K., & Sivanandan, R. (2013). Evaluation of Different Vehicle Following Models under Mixed Traffic Conditions. *Procedia - Social and Behavioral Sciences, 104*, 390-401.

Kanagaraj, V., Asaithambi, G., Toledo, T., & Lee, T.-C. (2015). Trajectory Data and Flow Characteristics of Mixed Traffic. *Transportation Research Record, 2491*(1), 1-11.

Kassambara, A. (2020). ggpubr: 'ggplot2' Based Publication Ready Plots. Retrieved from https://CRAN.R-project.org/package=ggpubr.

Kaufmann, S., Kerner, B. S., Rehborn, H., Koller, M., & Klenov, S. L. (2018). Aerial observations of moving synchronized flow patterns in over-saturated city traffic. *Transportation Research Part C: Emerging Technologies, 86*, 393-406.

Kerner, B. S. (2004). Three-phase traffic theory and highway capacity. *Physica A, 333*, 379-440.

Kerner, B. S., & Rehborn, H. (1996). Experimental properties of complexity in traffic flow. *Phys. Rev. E, 53*(5), 4275-4278.

Kesting, A., & Treiber, M. (2008). Calibrating Car-Following Models by Using Trajectory Data: Methodological Study. *Transportation Research Record, 2088*(1), 148-156.

Khakzar, M., Rakotonirainy, A., Bond, A., & Dehkordi, S. G. (2020). A Dual Learning Model for Vehicle Trajectory Prediction. *IEEE Access, 8*, 2169-3536. doi: 10.1109/ACCESS.2020.2968618.

Khan, M. A., Ectors, W., Bellemans, T., Janssens, D., & Wets, G. (2017). UAV-Based Traffic Analysis: A Universal Guiding Framework UAV-Based Traffic Analysis: A Universal Guiding Framework Based on Literature Survey. *Transportation Research Procedia*, *22*, 541-550.

Kockelman, K. M. (2001). Modeling traffic's flow-density relation: Accommodation of multiple flow regimes and traveler types. *Transportation, 28*, 363-374.

Koutsopoulos, H. N., & Farah, H. (2012). Latent class model for car following behavior. *Transportation Research Part B: Methodological*, *46*(6), 563-578.

Krajewski, R., Bock, J., Kloecker, L., & Eckstein, L. (2018). The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. *21st IEEE International Conference on Intelligent Transportation Systems,* (pp. 2118-2125). Maui, Hawaii, USA.

Krajewski, R., Moers, T., Bock, J., Vater, L., & Eckstein, L. (2020). The rounD Dataset: A Drone Dataset of Road User Trajectories at Roundabouts in Germany. *23rd IEEE International Conference on Intelligent Transportation Systems 2020*. Rhodes, Greece: Manuscript submitted for publication.

Krajewski, R., Moers, T., Neger, D., & Eckstein, L. (2018). Data-Driven Maneuver Modeling using Generative Adversarial Networks and Variational Autoencoders for Safety Validation of Highly Automated Vehicles. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2383-2390). Maui, Hawaii, USA: IEEE.

Krajzewicz, D., Hertkorn, G., Rössel, C., & Wagner, P. (2004). SUMO (Simulation of Urban MObility) An open-source traffic simulation. *Proceedings of the 4th Middle East Symposium on Simulation and Modelling*, (pp. 183-197). Sharjah, United Arab Emirates.

Krauss, S. (1998). *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. (Doctoral Thesis, Deutsches Zentrum für Luft– und Raumfahrt). Retrieved from https://sumo.dlr.de/pdf/KraussDiss.pdf.

Kuang, H., Wang, M.-T., Lu, F.-H., Bai, K.-Z., & Li, X.-L. (2019). An extended car-following model considering multi-anticipative average velocity effect under V2V environment. *Physica A, 527*, 121268.

Kuhn, M. (2020). caret: Classification and Regression Training. Retrieved from https://CRAN.R-project.org/package=caret.

Kurtc, V. (2020). Studying Car-Following Dynamics on the Basis of the HighD Dataset. *Transportation Research Record, 2674*(8), 813-822.

Leduc, G. (2008). *Road Traffic Data: Collection Methods and Applications*. Seville, Spain: JRC Technical Notes.

Lenz, H., Wagner, C. K., & Sollacher, R. (1999). Multi-anticipative car-following model. *The European Physical Journal B, 7*, 331-335.

Li, C., Jiang, X., Wang, W., Cheng, Q., & Shen, Y. (2016). A Simplified Car-following Model Based on the Artificial Potential Field. *Procedia Engineering, 137*, 13-20.

Li, L., Jiang, R., He, Z., Chen, X. (., & Zhou, X. (2020). Trajectory data-based traffic flow studies: A revisit. *Transportation Research Part C: Emerging Technologies, 114*, 225-240.

Li, L., Jiang, R., He, Z., Chen, X. M., & Zhou, X. (2020). Trajectory data-based traffic flow studies: A revisit. *Transportation Research Part C, 114*, 225-240.

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18-22.

Liu, X., Liang, J., & Xu, B. (2019). A Deep Learning Method for Lane Changing Situation Assessment and Decision Making. *IEEE Access*, 133749 - 133759.

Lüdecke, D., Makowski, D., Waggoner, P., & Patil, I. (2020). performance: Assessment of Regression Models Performance. Retrieved from https://easystats.github.io/performance/.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., & Hornik, K. (2019). cluster: Cluster Analysis Basics and Extensions. R package version 2.1.0. Retrieved from https://cran.r-project.org/web/packages/cluster/citation.html.

Mahajan, V. (2019). *Real-time Driving Intention Prediction and Crash Risk Estimation from Naturalistic Driving Data using Machine learning*. (Master's Thesis, Technical University of Munich, Germany). Retrieved from https://mediatum.ub.tum.de/doc/1523133/458415899571.pdf.

Mahajan, V., Katrakazas, C., & Antoniou, C. (2020). Prediction of Lane-Changing Maneuvers with Automatic Labeling and Deep Learning. *Transportation Research Record, 2674*(7), 336-347.

Maklin, C. (2019, July 16). *Gaussian Mixture Models Clustering Algorithm Explained*. Retrieved December 4 2020, from towards data science: https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e

Makrigiorgis, R., Kolios, P., Timotheou, S., Theocharides, T., & Panagiotou, C. G. (2020). Extracting the fundamental diagram from aerial footage. *2020 IEEE 91st Vehicular Technology Conference*. Antwerp, Belgium: IEEE Xplore.

McGonagle, J., Pilling, G., Dobre, A., Tembo, V., Kurmukov, A., Chumbley, A., Khim, J. (n.d.). *Gaussian Mixture Model*. Retrieved December 4 2020, from Brilliant: https://brilliant.org/wiki/gaussian-mixture-model/

Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., & Nashashibi, F. (2019). Non-local Social Pooling for Vehicle Trajectory Prediction. *Intelligent Vehicles Symposium (IV)*, 2019. Paris, France.

Montanino, M., & Punzo, V. (2013). Making NGSIM Data Usable for Studies on Traffic Flow Theory: Multistep Method for Vehicle Trajectory Reconstruction. *Transportation Research Record, 2390*(1), 99-111.

Montanino, M., & Punzo, V. (2015). Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transportation Research Part B: Methodological*, 80, 82-106.

NGSIM. (2006, December). Retrieved November 14, 2020, from U.S. Department of Transportation | Federal Highway Administration: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm

Oleszak, M. (2019, November 12). *Regularization: Ridge, Lasso and Elastic Net*. Retrieved December 12 2020, from datacamp: https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net [datacamp tutorial]

Papageorgiou, M., Mountakis, K.-S., Karafyllis, I., & Papamichail, I. (2019). Lane-free Artificial-Fluid Concept for Vehicular Traffic. arXiv:1905.11642.

Papathanasopoulou, V., & Antoniou, C. (2015). Towards data-driven car-following models. *Transportation Research Part C: Emerging Technologies, 55*, 496-509.

Papathanasopoulou, V., & Antoniou, C. (2016). Flexible car–following models incorporating information from adjacent lanes. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 701-706). Rio de Janeiro, Brazil: IEEE.

Papathanasopoulou, V., & Antoniou, C. (2018). Flexible car–following models for mixed traffic and weak lane–discipline conditions. *European Transport Research Review, 10*(62).

Plotly Technologies Inc. (2015). *Collaborative data science*. Retrieved from https://plot.ly

Punzo, V., Borzacchiello, M. T., & Ciuffo, B. (2011). On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. *Transportation Research Part C: Emerging Technologies, 19*(6), 1243-1262.

Quandt, R. E. (1958). The Estimation of the Parameters of a Linear Regression System Obeying Two Separate Regimes. *Journal of the American Statistical Association, 53*(284), 873-880.

R Core Team. (2019). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Reynolds, D. (2009). Gaussian Mixture Models. *Encyclopedia of Biometrics*, 659-663. (S. Z. Li, & A. Jain, Eds.). Boston, MA, USA: Springer.

Ripley, B. (2019). tree: Classification and Regression Trees. Retrieved from https://CRAN.R-project.org/package=tree.

Robicquet, A., Sadeghian, A., Alahi, A., & Savarese, S. (2016). Learning Social Etiquette: Human Trajectory Understanding in Crowded Scenes. *14th European Conference on Computer Vision*, (pp. 549-565). Amsterdam, the Netherlands.

RStudio Team. (2018). RStudio: Integrated Development Environment for R. Boston, Massachusetts, USA: RStudio, Inc. http://www.rstudio.com/.

Salvo, G., Caruso, L., & Scordo, A. (2014). Urban traffic analysis through an UAV. *Procedia - Social and Behavioral Sciences, 111*, 1083-1091.

Scrucca, L., Fop, M., Murphy, B., & Raftery, A. E. (2016). mclust: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal, 8*(1), 289-317.

Simon, L., Young, D., & Pardoe, I. (2018). *Applied Rregression Analysis*. Retrieved December 6 2020, from PennState Eberly College of Science: https://online.stat.psu.edu/stat462/node/180/

Song, J., Wu, Y., Xu, Z., & Lin, X. (2014). Research on car-following model based on SUMO. *The 7th IEEE International Conference on Advanced Infocomm Technology* (pp. 47-55). Fuzhou, China: IEEE.

Strobl, C., & Zeileis, A. (2008, August 12-14). *Why and how to use random forest variable importance measures (and how you shouldn't).* Retrieved December 13, 2020, from The R User Conference 2008: http://www.statistik.tu-dortmund.de/useR-2008/slides/Strobl+Zeileis.pdf

Sun, L., & Zhou, J. (2005). Development of Multiregime Speed–Density Relationships by Cluster Analysis. *Transportation Research Record, 1934*(1), 64-71.

Theil, H. (1961). *Economic forecasts and policy*. Amsterdam, the Netherlands: North-Holland Pub. Co.

Therneau, T., & Atkinson, B. (2019). rpart: Recursive Partitioning and Regression Trees. Retrieved from https://CRAN.R-project.org/package=rpart.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological), 58*(1), 267-288.

Umweltbundesamt. (2020, February 14). *Fahrleistungen, Verkehrsaufwand und Modal Split [Mileage, Traffic Volume and Modal Split]*. Retrieved January 21, 2021, from https://www.umweltbundesamt.de/daten/verkehr/fahrleistungen-verkehrsaufwand-modal-split#fahrleistung-im-personen-und-guterverkehr

Van de Geer, S. A. (2005). Least Squares Estimation. In B. S. Everitt, & D. C. Howell, *Encyclopedia of Statistics in Behavioral Science* (pp. 1041-1045). Chichester, United Kingdom: John Wiley & Sons.

van Wageningen-Kessels, F., van Lint, H., Vuik, K., & Hoogendoorn, S. (2015). Genealogy of traffic flow models. *EURO Journal on Transportation and Logistics, 4*, 445-473.

Venables, W., & Ripley, B. (2002). *Modern Applied Statistics with S*. New York, New York, USA: Springer.

Wang, X., Zhu, M., Chen, M., & Tremont, P. (2016). Drivers' rear end collision avoidance behaviors under different levels of situational urgency. *Transportation Research Part C: Emerging Technologies, 71*, 419-433.

Washington, S. P., Karlaftis, M. G., & Mannering, F. L. (2011). *Statistical and Econometric Methods for Transportation Data Analysis*. Boca Raton, FL, USA: CRC Press.

Wei, J., Dolan, J. M., & Litkouhi, B. (2010). A learning-based autonomous driver: emulate human driver's intelligence in low-speed car following. *SPIE Defense, Security, and Sensing, 2010*. Orlando, Florida, USA.

Wei, T., & Simko, V. (2017). R package "corrplot": Visualization of a Correlation Matrix. Retrieved from https://github.com/taiyun/corrplot.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. New York, New York, USA: Springer.

Wickham, H., Francois, R., Henry, L., & Müller, K. (2020). dplyr: A Grammar of Data Manipulation. Retrieved from https://CRAN.R-project.org/package=dplyr.

Wickham, H., Hester, J., & Francois, R. (2018). readr: Read Rectangular Text Data. Retrieved from https://CRAN.R-project.org/package=readr.

Wilde, G. J. (1982). The Theory of Risk Homeostasis: Implications for Safety and Health. *Risk Analysis, 2*, 209-225.

Woo, H., Ji, Y., Tamura, Y., Kuroda, Y., Sugano, T., Yamamoto, Y., Asama, H. (2018). Dynamic State Estimation of Driving Style Based on Driving Risk. *International Journal of Automotive Engineering, 9,* 31-38.

Woo, H., Madokoro, H., Sato, K., Tamura, Y., Yamashita, A., & Asama, H. (2020). Prediction of Following Vehicle Trajectory Considering Operation Characteristics of a Human Driver. *2020 IEEE/SICE International Symposium on System Integration (SII)* (pp. 712-717). Honolulu, Hawaii, USA: IEEE.

Wu, P., Gao, F., & Li, K. (2019). A Vehicle Type Dependent Car-following Model Based on Naturalistic Driving Study. *Electronics, 8*(4), 453.

Xiang, T.-Z., Xia, G.-S., & Zhang, L. (2019). Mini-Unmanned Aerial Vehicle-Based Remote Sensing: Techniques, applications, and prospects. *IEEE Geoscience and Remote Sensing Magazine, 7*(3), 29-63.

Yan, Q. (2019, November 27). R Visualization Tips: Using the scales package. Retrieved from https://bookdown.org/Maxine/ggplot2-maps/posts/2019-11-27-using-scales-package-to-modify-ggplot2-scale/.

Yan, Y. (2016, May 13). Machine Learning Evaluation Metrics. Retrieved from https://github.com/yanyachen/MLmetrics.

Yang, D., Zhu, L., Liu, Y., Wu, D., & Ran, B. (2019). A Novel Car-Following Control Model Combining Machine Learning and Kinematics Models for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems, 20*(6), 1991-2000.

Yao, W., Zhao, H., Davoine, F., & Zha, H. (2012). data, Learning lane change trajectories from on-road driving. *2012 IEEE Intelligent Vehicles Symposium* (pp. 885-890). Alcala de Henares, Spain: IEEE.

Zheng, L., & He, Z. (2015). A new car following model from the perspective of visual imaging. *International Journal of Modern Physics C, 26*(08), 1550090. doi:10.1142/S0129183115500904.

Zhou, M., Qu, X., & Li, X. (2017). A recurrent neural network based microscopic car following model to predict traffic oscillation. *Transportation Research Part C: Emerging Technologies, 84*, 245-264.

Zhou, Y., Fu, R., Wang, C., & Zhang, R. (2020). Modeling Car-Following Behaviors and Driving Styles with Generative Adversarial Imitation Learning. *Sensors, 5034*, 1-19.

Zhu, M., Wang, X., & Wang, Y. (2018). Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies, 97*, 348-368.

Zhu, M., Wang, X., Tarko, A., & Fang, S. (2018). Modeling car-following behavior on urban expressways in Shanghai: A naturalistic driving study. *Transportation Research Part C: Emerging Technologies, 93*, 425-445.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301-320.

Appendix

A. Model-based clustering with two clusters; traffic phases by direction

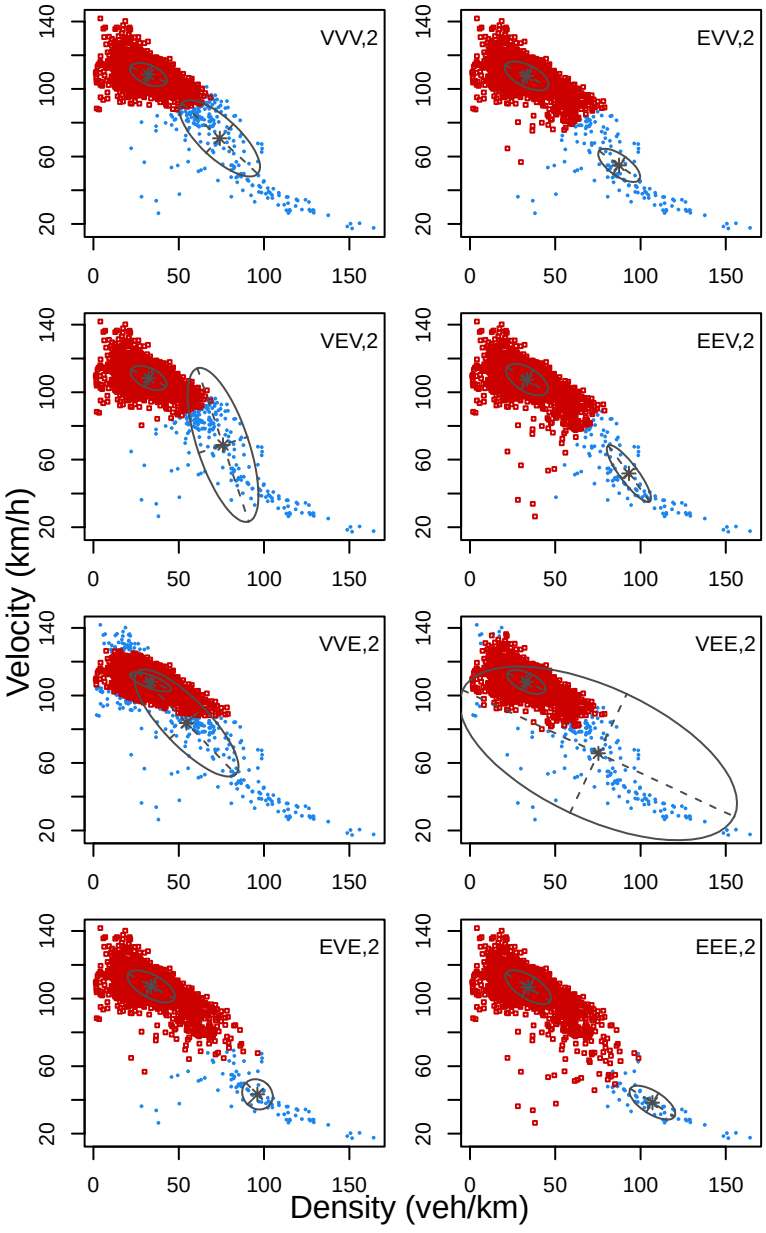


Figure 9.1 Density - velocity diagram with two clusters (by direction)

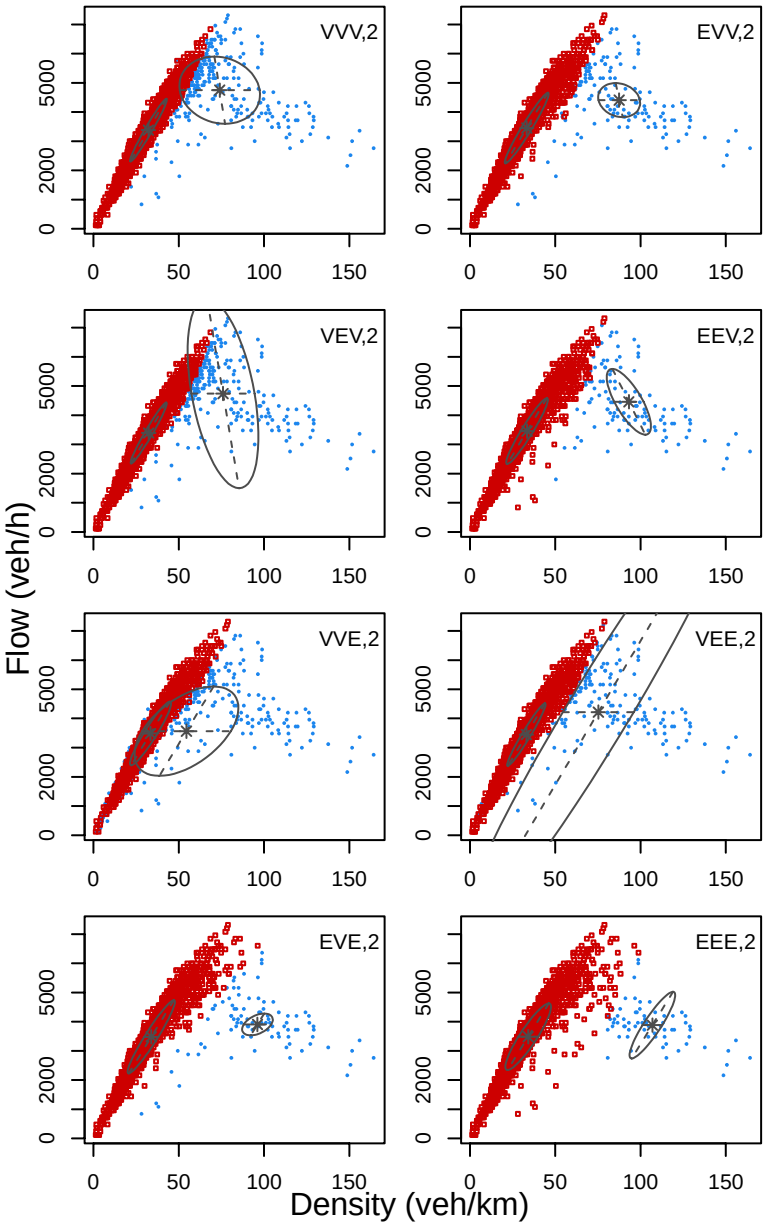


Figure 9.2 Density - flow diagram with two clusters (by direction)

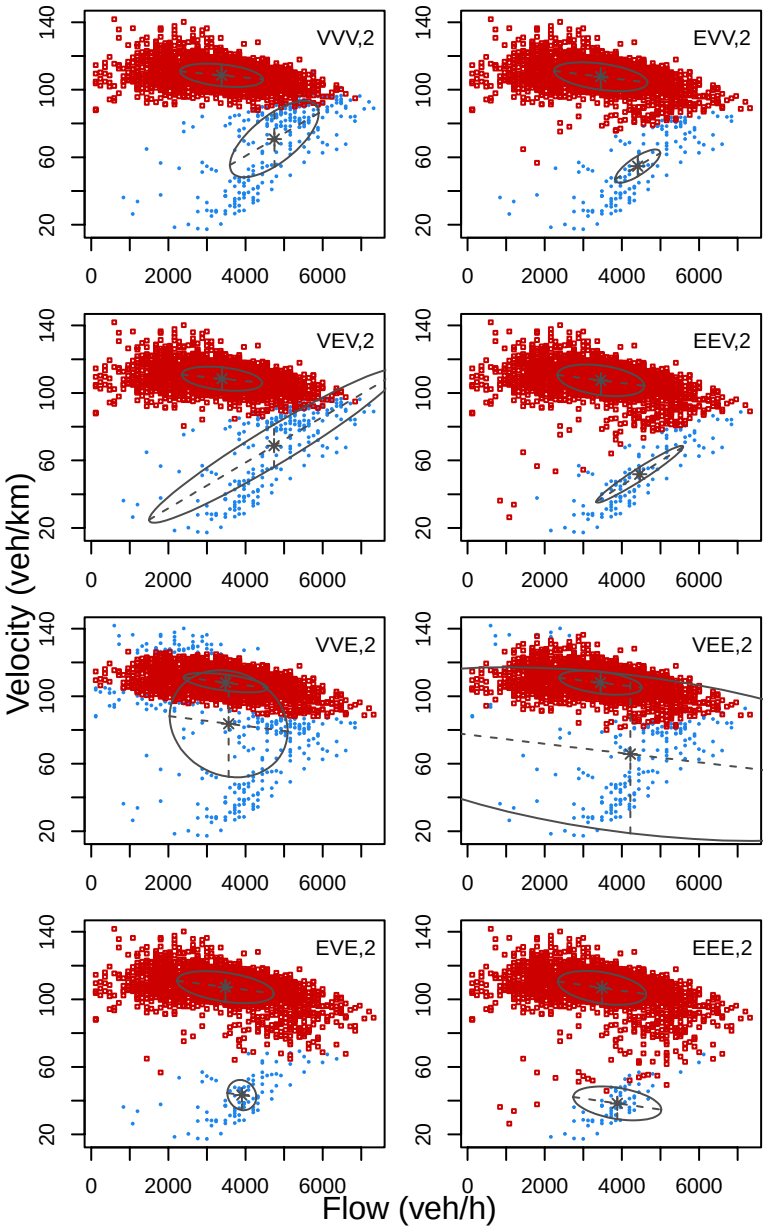


Figure 9.3 Flow - velocity diagram with two clusters (by direction)

B. Model-based clustering with two clusters; traffic phases by direction

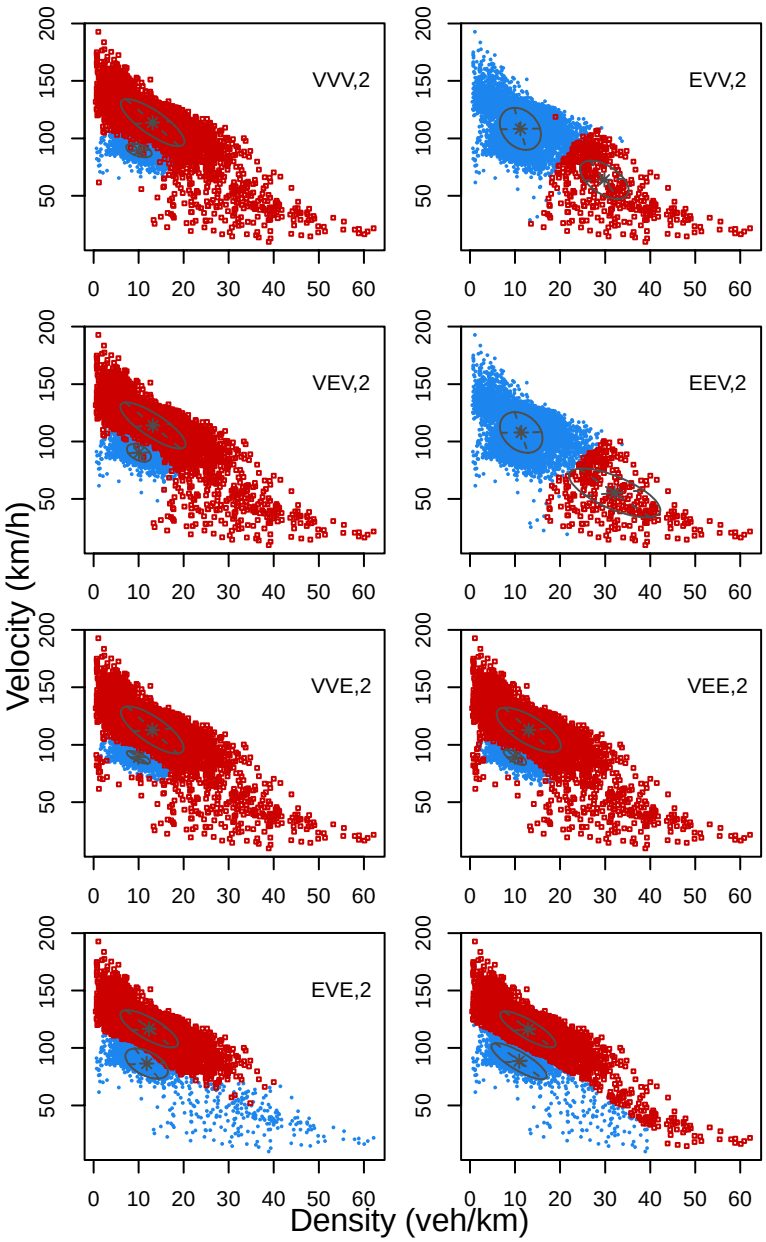


Figure 9.4 Density - velocity diagram with two clusters (by lane)

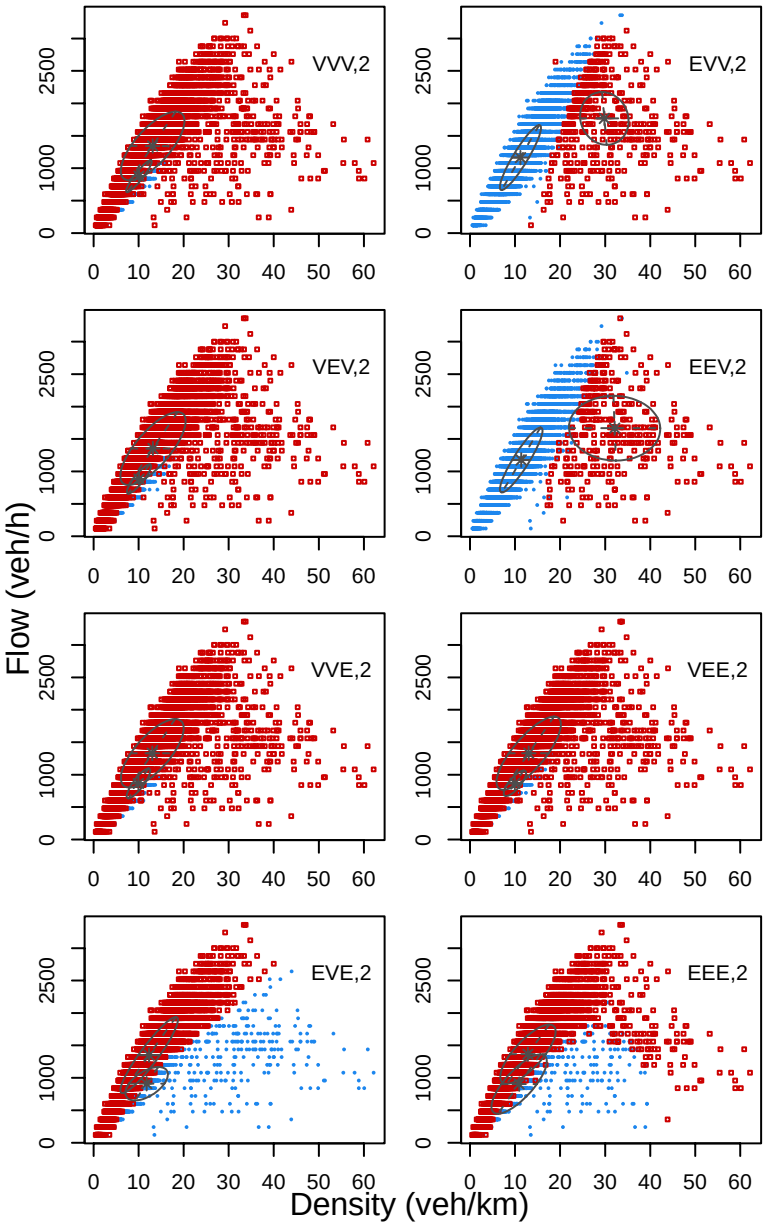


Figure 9.5 Density - flow diagram with two clusters (by lane)

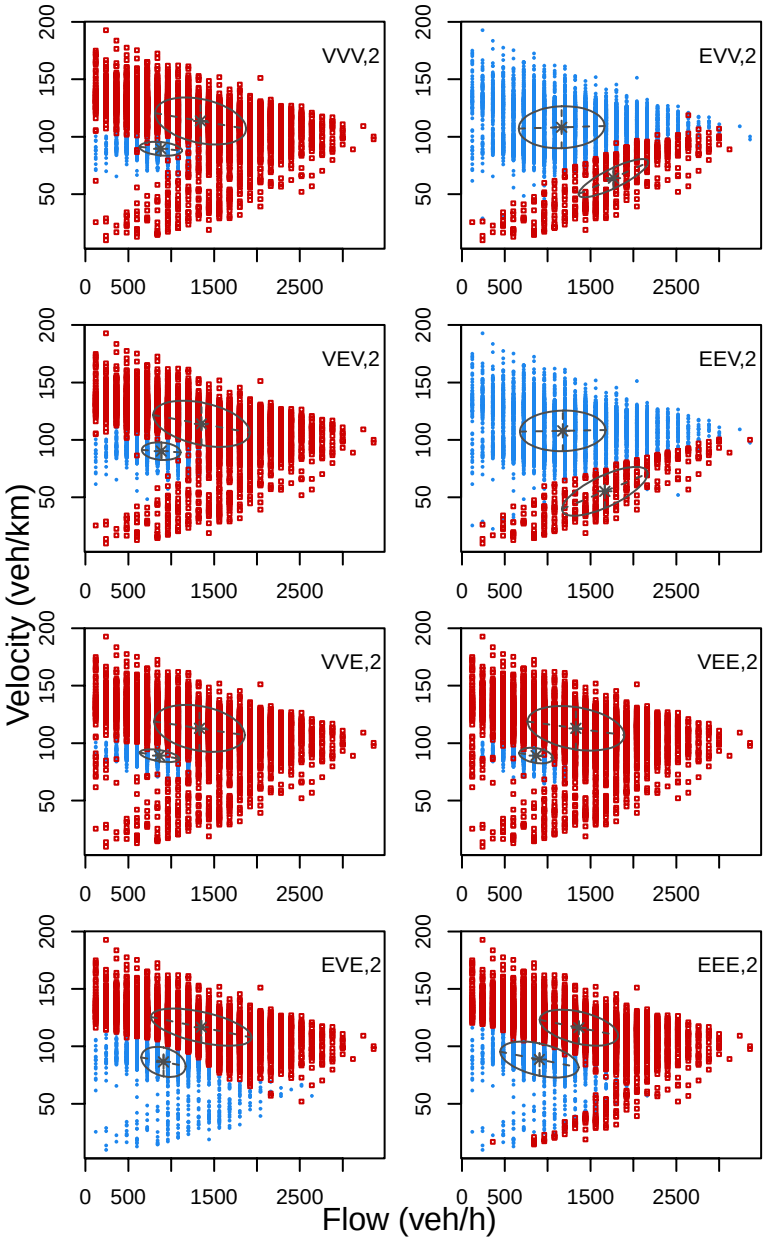


Figure 9.6 Flow - velocity diagram with two clusters (by lane)

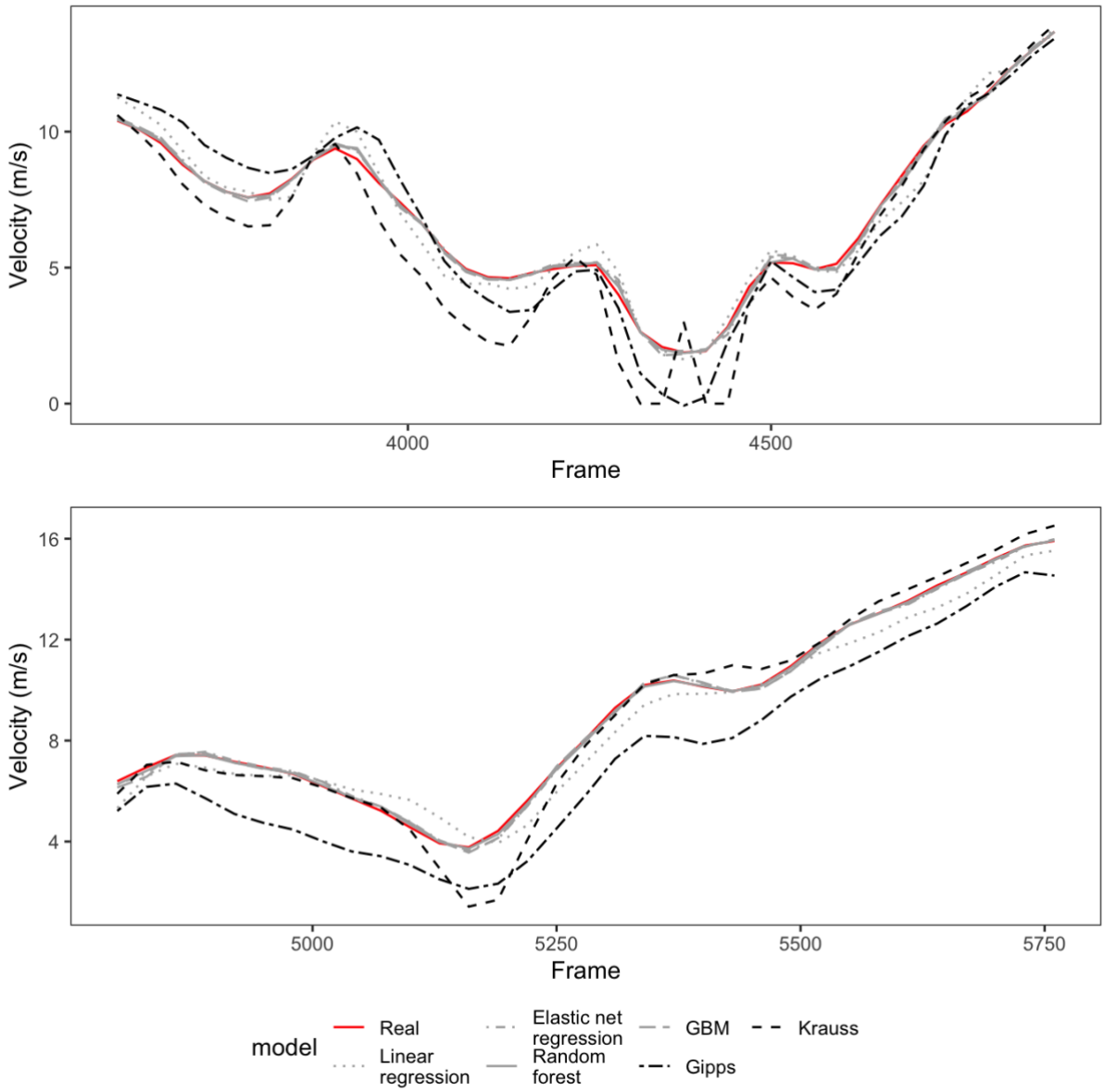


Figure 9.7 Comparison of real and estimated velocity trajectories (data-driven and mathematical models) from Track 24, Vehicle IDs: 402 & 500

Table 9.1 Gradient boosting machines parameter grid search

No. of trees	Reduction of the learning rate	No. of splits at node	No. of observations at terminal node	RMSE
3000	0.01	1	5	0.25776544
3000	0.01	1	10	0.258739
3000	0.01	1	15	0.26429399
3000	0.01	3	5	0.16543514
3000	0.01	3	10	0.16887093
3000	0.01	3	15	0.18022107
3000	0.01	5	5	0.14459703
3000	0.01	5	10	0.14782629
3000	0.01	5	15	0.15831423
3000	0.05	1	5	0.17781948
3000	0.05	1	10	0.18154093
3000	0.05	1	15	0.19407968
3000	**0.05**	**3**	**5**	**0.13851053**
3000	0.05	3	10	0.141901
3000	0.05	3	15	0.15108146
3000	**0.05**	**5**	**5**	**0.13725359**
3000	**0.05**	**5**	**10**	**0.13988221**
3000	0.05	5	15	0.14843414
3000	0.1	1	5	0.17909556
3000	0.1	1	10	0.18366915
3000	0.1	1	15	0.19557662
3000	**0.1**	**3**	**5**	**0.13793873**
3000	0.1	3	10	0.14197675
3000	0.1	3	15	0.15226382
3000	**0.1**	**5**	**5**	**0.13929867**
3000	0.1	5	10	0.14201641
3000	0.1	5	15	0.15070705
5000	0.01	1	5	0.23260121
5000	0.01	1	10	0.23383552
5000	0.01	1	15	0.24132587
5000	0.01	3	5	0.1439951
5000	0.01	3	10	0.14742733
5000	0.01	3	15	0.15864414
5000	**0.01**	**5**	**5**	**0.1366966**
5000	**0.01**	**5**	**10**	**0.13972956**
5000	0.01	5	15	0.14936432
5000	0.05	1	5	0.15708306
5000	0.05	1	10	0.16194543

5000	0.05	1	15	0.17460132
5000	**0.05**	**3**	**5**	**0.13663294**
5000	0.05	3	10	0.14040847
5000	0.05	3	15	0.14968854
5000	**0.05**	**5**	**5**	**0.13684592**
5000	**0.05**	**5**	**10**	**0.1395608**
5000	0.05	5	15	0.14812339
5000	0.1	1	5	0.16751931
5000	0.1	1	10	0.17347759
5000	0.1	1	15	0.1863165
5000	**0.1**	**3**	**5**	**0.1370476**
5000	0.1	3	10	0.1407789
5000	0.1	3	15	0.15014597
5000	**0.1**	**5**	**5**	**0.13929423**
5000	0.1	5	10	0.14201641
5000	0.1	5	15	0.15062995
7000	0.01	1	5	0.21421108
7000	0.01	1	10	0.21583314
7000	0.01	1	15	0.22508469
7000	**0.01**	**3**	**5**	**0.13841206**
7000	0.01	3	10	0.14151104
7000	0.01	3	15	0.15172662
7000	**0.01**	**5**	**5**	**0.13496402**
7000	**0.01**	**5**	**10**	**0.13794625**
7000	0.01	5	15	0.147034
7000	0.05	1	5	0.15103855
7000	0.05	1	10	0.15540157
7000	0.05	1	15	0.16669587
7000	**0.05**	**3**	**5**	**0.13605965**
7000	**0.05**	**3**	**10**	**0.13975116**
7000	0.05	3	15	0.14925011
7000	**0.05**	**5**	**5**	**0.13684592**
7000	**0.05**	**5**	**10**	**0.1395608**
7000	0.05	5	15	0.14812339
7000	0.1	1	5	0.16143688
7000	0.1	1	10	0.16782734
7000	0.1	1	15	0.18081643
7000	**0.1**	**3**	**5**	**0.1370476**
7000	0.1	3	10	0.14068644
7000	0.1	3	15	0.14985148
7000	**0.1**	**5**	**5**	**0.13929423**
7000	0.1	5	10	0.14201641
7000	0.1	5	15	0.15062995