TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Verkehrstechnik

Data-Driven Modeling of Lane Changing on Freeways

-

Application for Automated Vehicles in Mixed-Traffic

Nassim Motamedidehkordi, M.Sc.

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigten Dissertation.

Vorsitzender: Prof. Dr.-Ing. Constantinos Antoniou

Prüfer der Dissertation:

1. Prof. Dr.-Ing. Fritz Busch
2. Prof. Dr. rer. nat. Francisco Camara Pereira, Technical University of Denmark

Die Dissertation wurde am 16.04.2019 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 19.09.2019 angenommen.

Acknowledgement

I would like to express my deepest appreciation to Prof. Fritz Busch for providing me the opportunity and the encouragement to commence my doctoral studies. He continually conveyed a spirit of adventure regarding my research, and this dissertation would not have been possible without his guidance and persistent support.

I would also like to extend my appreciation for the supporting guidance provided by Prof. Francisco Camara Pereira. Prof. Pereira’s thorough peer review and constructive feedback contributed greatly to the quality enhancement of this dissertation. I would like to also thank Prof. Pereira’s team at the Technical University of Denmark, who made my stay at their department a memorable experience.

To my colleagues at the Chair of Traffic Engineering and Control at the Technical University of Munich – Thank you for your supervision, cooperation and long-lasting friendship. Your continuous support and enduring positive attitude were encouraging through the most challenging times, and for that, I am grateful.

Special thanks to my mentor Prof. Silja Hoffmann, for her enormous support. Prof. Hoffman’s unrelenting encouragement and positive energy have motivated me to continue following my career aspirations with an open mind.

Furthermore, I would like to acknowledge – with much appreciation – the pivotal role that the staff at UnternehmerTUM had in this dissertation. The collection of important traffic data would not have been possible without their support in video data collection.

To my loving family – Thank you for actively supporting me in my determination to find and realize my potentials. My mother, Nasrin, has been a source of encouragement and inspiration throughout my entire life; my father, Shahram, has always been and always will be my role-model for determination and hard work; my sister, Noushin, and my brother-in-law, Kaveh, have supported me spiritually throughout my life and of course, the evolution of this thesis.

Last but not least, I am grateful to my husband, Dr. Sina Habibollahi, who endured the struggles and ensuing discomfort of an additional three years of long-distance relationship during this dissertation. I can never thank him enough for his endless emotional support and patience, for understanding the importance of my career dreams, and for respecting my decision.

Abstract

Lane changing is known to have a substantial influence on traffic flow characteristics due to its interfering effect on the surrounding traffic, most importantly the increase in drivers’ workload and stress level. In recent years, technological innovation in the automotive industry accelerated the emergence of automated vehicles allowing technology to alleviate some demanding driving tasks, including lane changing maneuvers. One of the significant challenges continues to be the penetration of automated vehicles into the current transportation system and the associated mixed-traffic. Operating automated vehicles in conjunction with human drivers introduces mixed-traffic complexities that present unique challenges to effectively predict the driving and mimic the overall social behaviors of drivers.

A commonly used tool for developing and evaluating the impact of automated vehicles in various scenarios, including mixed-traffic situations is microscopic traffic simulation. Unfortunately, most of the available microscopic simulation tools are based on rule-based lane changing models. These models do not account for all factors including driving behavior norms and heterogeneous driving behavior, which – as the research suggests – ultimately results in unrealistic and biased findings because that the simulation does not account for realizable trade-offs among the factors such as speed, acceleration, jerk, and safety when performing the task of driving.

The goal of this dissertation is to provide a generic data-driven approach for modeling lane changing behavior via a series of data gathering techniques based entirely on real-world observations. The proposed approach allows the lane changing model to dynamically assess and learn socially acceptable driving behavior based on observations of the surrounding driving environment, including other human drivers facing the same real-world conditions. This method provides a realistic, non-rule-based model that accounts for both the tactical and operational behavior of lane changing. The proposed approach also accounts for stochasticity among human drives, which enables more accurate investigations and outcome projections using traditional microscopic traffic simulation.

This modeling technique requires real data from the driving behavior of human drivers, and once collected, uses it to develop data-driven models for the tactical and operational behavior of drivers for the application in microscopic traffic simulations. The data was collected from traffic video recordings at an on-ramp to an urban freeway in Munich. After post-processing 13 hours of recorded videos, individual vehicle trajectories were extracted and processed using clustering methods to categorize human driving behaviors into three clusters: “timid”, ”moderate” and “aggressive”. Driving behaviors categorized as “aggressive” were then purposely excluded from the training data, which is used subsequently for modeling the lane changing behavior for automated vehicles to eliminate the undesirable characteristics and unsafe maneuvers from the dataset.

The decision to change lanes is recognized as a tactical decision and is executed in varying units of time ranging from seconds to minutes based entirely on immediate driving, traffic and environmental circumstances. Modeling the tactical lane changing behavior was done by deploying a supervised Machine Learning (ML) algorithm that simultaneously takes into account the status of the “ego vehicle”, surrounding dynamic objects such as other road users, and environmental components such as road infrastructure. Defining these parameters in the feature vector enabled the algorithm to establish the classifier as a prediction – with a high degree of certainty – on whether the driver will ultimately decide to perform a lane change to the right, to the left or remain in the current lane.

At a subconscious level, a driver executes action patterns in milliseconds to respond to the immediate traffic situation. Operational behavior modeling during lane changing was developed using the ‘Inverse Reinforcement Learning’ (IRL) method. In this method, the agent’s policy or behavioral history is an established variable, and the goal of ML is to determine the reward function that explains the given behavior, resulting in the automatic extraction of the respective rewards from the respective observation. This model uses the ‘Markov Decision Process’ (MDP), which is a mathematical framework for modeling decision-making situations where outcomes are partly random and partly controlled by the decision-maker.

Both models were integrated into the SUMO (Simulation of Urban MObility) software to evaluate the overall ability of the approach to simulate traffic realistically and compared its performance and transferability when faced with unseen events in the training dataset. The integrated model was evaluated by comparing the network throughput, the distribution of Time-To-Collisions (TTC) and the number of lane changes with the default lane changing model in SUMO. The simulation results indicate that the proposed integrated modeling approach is capable of realistically simulating the lane changing behavior of both normal drivers and automated vehicles and it outperforms the underlying models in SUMO. Moreover, the model successfully executes plausible lane changing maneuvers in unseen traffic situations.

Kurzfassung

Fahrstreifenwechsel haben einen erheblichen Einfluss auf die Eigenschaften des Verkehrsflusses. Dies resultiert aus der störenden Wirkung auf den umgebenden Verkehr. Zusätzlich erhöhen Fahrstreifenwechsel die Arbeitsbelastung und das Stresslevel der Fahrzeuglenker. In den letzten Jahren beschleunigten technologische Innovationen in der Automobilindustrie die Entwicklung automatisierter Fahrzeuge, die die Erleichterung einige der anspruchsvollen Fahraufgaben, einschließlich Fahrstreifenwechselmanöver, ermöglichen. Eine der großen Herausforderungen ist nach wie vor die Durchdringung automatisierter Fahrzeuge und der damit einhergehende Mischverkehr. Der Betrieb automatisierter Fahrzeuge in Interaktion mit menschlichen Fahrern führt zu komplexen Verkehrssituationen. Dies stellt einzigartige Herausforderungen für die effektive Vorhersage der Fahrmanöver und Trajektorien und die Nachahmung des Verhaltens menschlicher Fahrer dar.

Ein gängiges Tool zur Entwicklung und Bewertung der Auswirkungen automatisierter Fahrzeuge in verschiedenen Szenarien, einschließlich Mischverkehrssituationen, ist die mikroskopische Verkehrssimulation. Allerdings basieren die meisten der verfügbaren mikroskopischen Simulationswerkzeuge auf regelbasierten Fahrstreifenwechselmodellen. Diese Modelle berücksichtigen bisher allerdings noch nicht alle Faktoren im notwendigen Detaillierungsgrad, einschließlich Fahrverhaltensnormen, welches - wie diese Forschung zeigt - letztlich zu unrealistischen und verzerrten Ergebnissen führt. Dies geschieht, da die Simulation nicht den realisierbaren Kompromiss zwischen den Faktoren Geschwindigkeit, Beschleunigung, Ruck und Sicherheit bei der Erfüllung der Fahraufgabe berücksichtigt.

Das Ziel dieser Dissertation ist es, einen generischen datengetriebenen Ansatz für die Modellierung des Fahrstreifenwechselverhaltens zu entwickeln, der vollständig auf realen Beobachtungen basiert. Der vorgeschlagene Ansatz ermöglicht es ein gesellschaftlich akzeptables Fahrverhalten dynamisch zu bewerten und zu erlernen, welches auf realen Beobachtungen der Umgebung, einschließlich anderer menschlicher Fahrer basiert. Diese Methode bietet ein realistisches, nicht regelbasiertes Modell, das sowohl das taktische als auch das operationale Verhalten beim Fahrstreifenwechsel berücksichtigt. Der vorgeschlagene Ansatz berücksichtigt auch das stochastisches Verhalten menschlicher Fahrer. Dies ermöglicht genauere Untersuchungen und Ergebnisprognosen auf Basis von traditionellen mikroskopischen Verkehrssimulationen.

Diese Modellierungstechnik benötigt reale Fahrverhaltensdaten menschlicher Fahrer und entwickelt daraus datengesteuerte Modelle für das taktische und operationale Verhalten von Fahrern für die Anwendung in mikroskopischen Verkehrssimulationen. Die Daten wurden aus Verkehrsvideoaufnahmen an einer Einfahrt zu einer Stadtautobahn in München gesammelt. Nach der Nachbearbeitung von aufgezeichneten Videos mit einer Dauer von 13 Stunden wurden einzelne Fahrzeugtrajektorien extrahiert und mit Hilfe von Clustering-Methoden verarbeitet, um das menschliche Fahrverhalten in drei Cluster einzuteilen: "konservativ", "moderat" und "aggressiv". Als "aggressiv" eingestuftes Fahrverhalten wurde dann bewusst

aus den Daten ausgeschlossen, welche als Basis für die weitere Modellierung fungieren. Genauer gesagt werden unerwünschte Eigenschaften und unsichere Manöver aus dem Datensatz, der anschließend zur Modellierung des Fahrstreifenwechselverhaltens für automatisierte Fahrzeuge verwendet wird, eliminiert.

Die Entscheidung, die Fahrstreifen zu wechseln, wird als taktische Entscheidung anerkannt und wird im Zeitbereich zwischen einer Sekunde bis zu einer Minute durchgeführt, die ausschließlich auf unmittelbaren Fahr-, Verkehrs- und Umweltbedingungen basieren. Die Modellierung des taktischen Fahrstreifenwechselverhaltens erfolgte durch den Einsatz eines überwachten maschinellen Lernverfahrens, welches gleichzeitig den Status des "ego" Fahrzeugs berücksichtigt, das dynamische Objekte wie andere Verkehrsteilnehmer und Umweltkomponenten, sowie die Straßeninfrastruktur umgibt. Die Definition dieser Parameter im Merkmalsvektor ermöglichte es dem Algorithmus, den Klassifikator als Vorhersage zu etablieren, ob der Fahrer sich letztendlich für einen Fahrstreifenwechsel nach rechts oder links bzw. auf das Verbleiben auf dem aktuellen Fahrstreifen entscheidet.

Auf einer unterbewussten Ebene, führt ein Fahrer Aktionsmuster im Millisekundenbereich aus, um auf die aktuelle Verkehrssituation zu reagieren. Die Modellierung des Betriebsverhaltens oder operativen Verhaltens beim Fahrstreifenwechsel wurde mit der Methode "Inverse Reinforcement Learning" (IRL) entwickelt. Bei dieser Methode wird davon ausgegangen, dass die Agenten sich optimal verhalten. Das Ziel des maschinellen Lernens ist es, die Belohnungsfunktion zu bestimmen, die das gegebene Verhalten erklärt. Dies führt zur automatischen Extraktion der jeweiligen Belohnungen aus der jeweiligen Beobachtung. Dieses Modell verwendet den "Markov Decision Process" (MDP), einen mathematischen Rahmen zur Modellierung von Entscheidungssituationen, bei denen die Ergebnisse teilweise zufällig und teilweise vom Entscheidungsträger kontrolliert werden.

Beide Modelle wurden in die SUMO (Simulation of Urban MObility) Software integriert, um die Gesamtfähigkeit des Ansatzes zur realistischen Verkehrssimulation zu bewerten und seine Leistung und Übertragbarkeit bei unsichtbaren Ereignissen im Trainingsdatensatz zu vergleichen. Das integrierte Modell wurde durch den Vergleich des Netzwerkdurchsatzes, der Verteilung von Time-To-Collisions (TTC) und der Anzahl der Fahrstreifenwechsel im Vergleich zum Standard-Fahrstreifenwechselmodell in SUMO bewertet. Die Simulationsergebnisse zeigen, dass der vorgeschlagene integrierte Modellierungsansatz in der Lage ist, das Fahrstreifenwechselverhalten von normalen Fahrern und automatisierten Fahrzeugen realistisch zu simulieren, und er übertrifft die zugrunde liegenden Modelle in SUMO. Darüber hinaus führt das Modell erfolgreich plausible Fahrstreifenwechselmanöver in zuvor unbeobachteten Verkehrssituationen durch.

To my parents,

for letting me pursue my dreams 3500 kilometers away from home.

To my sister,

for supporting my craziest dreams.

To my husband,

for giving me new dreams to pursue 6600 kilometers away from him.

Table of Contents

"If you torture the data long enough, it will confess to anything."

Darrell Huff, *How to Lie with Statistics,* 1954

1. Introduction

In this chapter, the motivation and framework of this research work are explained. Firstly, the importance of lane changes in traffic efficiency and safety as well as the motivation to hand over this task to automated vehicles are discussed. Moreover, the developments and current problems and challenges related to automated vehicles' driving behavior, especially lane changing are explained. The research questions pursued in this dissertation are then defined based on the foregoing review and the methodology used to investigate the research questions is outlined. Finally, the outline of the dissertation is discussed.

1.1 Motivation

Lane changing has a substantial influence on traffic flow characteristics due to the interfering effect that it has on surrounding vehicles [DAGANZO ET AL., 1999; MAUCH & CASSIDY, 2002; SASOH & OHARA, 2002; MUÑOZ & DAGANZO, 2002; CHEN ET AL., 2004; AL-KAISY ET AL., 2005]. Several recent experiments have concluded that the development of congestion at bottlenecks is mainly attributed to lane changing maneuvers [CASSIDY & RUDJANAKANOKNAD, 2005; BERTINI & LEAL, 2005; LAVAL & DAGANZO, 2006; HOU ET AL., 2014b]. Lane changing is also believed to increase the stress and workload of drivers and has a significant impact on traffic safety and be responsible for crashes [CHOVAN ET AL., 1994; TOLEDO ET AL., 2003; PANDE & ABDEL-ATY, 2006].

In recent years, technological innovation in the automotive industry accelerated the emergence of automated vehicles and handing over the demanding driving tasks such as lane changing to automated vehicles has increased attention. Automated vehicles are believed to increase traffic safety [EUROPEAN TRANSPORT SAFETY COUNCIL (ETSC), 2016], traffic efficiency [HOOGENDOORN ET AL., 2014], economy [FAGNANT & KOCKELMAN, 2015] and improve the mobility [ALESSANDRINI ET AL., 2015]. These vehicles are capable of sensing their environment and draw a virtual picture of external traffic, road and environmental conditions with the help of cameras, sensors, radars, high-definition maps and large quantities of algorithms and navigating without human input. However, different vehicles will operate at varying levels of autonomy, depending on the technologies they use. As shown in Fig 1.1, the Society of Automotive Engineers [SAE INTERNATIONAL, 2016] identifies six levels of driving automation from "no automation" to "full automation" based on the functional aspects of technology.

SAE Level	Name	Narrative definition	Execution of steering and acceleration/ deceleration	Monitoring of driving environment	Fallback performance of dynamic driving task	System capability (driving modes)	BASt Level	NHTSA level
Human driver monitors the drivng environment								
0	No Automation	the full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a	Driver only	0
1	Driver Assistance	the driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	Human driver and system	Human driver	Human driver	Some driving modes	Assisted	1
2	Partial Automation	the driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the human driver perform all remaining aspects of the dynamic driving task	System	Human driver	Human driver	Some driving modes	Partially automated	2
Automated driving system monitors the environment								
3	Conditional Automation	the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene	System	System	Human driver	Some driving modes	Highly automated	3
4	High Automation	the driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task, even if a human driver does not respond appropriately to a request to intervene	System	System	System	Some driving modes	Fully automated	3/4
5	Full Automation	the full-time performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver	System	System	System	All driving modes	-	

Fig 1.1 Levels of automation based on the definition of SAE international [SAE INTERNATIONAL, 2016] and compared with those developed by Federal Highway Research Institute of Germany [FEDERAL HIGHWAY RESEARCH INSTITUTE (BAST), 2016] and US National Highway Traffic Safety Administrations [NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA), 2017].

The penetration of automated vehicles into the current transportation system will lead to mixed-traffic as in the initial phase automated vehicles – regardless of their level of automation – will share the road with human drivers. The goal within the automated vehicle development is to design safer, more efficient and interactive vehicles while maintaining the persona of an ideal human driver, without having the mistakes that human drivers make. Thus, in mixed-traffic, these vehicles should not only safely navigate around human drivers but also know how to interact with them, be able to mimic the social behaviors and communicate their intentions to other drivers. Sadigh et al. [2016a] observed that an automated vehicle's action could also affect neighboring human drivers' behavior, and studied how humans will react when the automated vehicle performs specific actions [SADIGH ET AL., 2016b]. As a result, these vehicles should engage with pedestrians, bicyclists, as well as human drivers in a socially acceptable manner consisting of actions that are easily understandable to the surrounding road users [HUANG ET AL., 2018]. A socially acceptable automated driving behavior would be behavior in which automated vehicles, when interacting with other road users, operate smoothly and in an appropriate interactional manner; vehicles that behave neither too aggressively nor yield incessantly to other road users, neither impede the normal flow of traffic nor cause undue notice. In other words, socially acceptable driving means that automated vehicles can smoothly integrate into the flow of traffic and handle traffic interactions without disrupting other road users.

One of the main challenges in the development of automated vehicles is the difficulty of modeling a human driver and specifying precisely how a human driver would drive and trade off between different aspects of driving such as safety, speed, comfort, gas usage, legal risks, and the many other elements that are taken into account while driving. In order to capture how the human thinks, it would require the cognitive architecture of human behavior, knowing exactly how the human brain works and knowing how every single driver in the world drives, which is despite the efforts to model the cognitive architecture of humans [BOER & HOEDEMAEKER, 1998; SALVUCCI ET AL., 2001; KUGE ET AL., 2000; KUMAGAI ET AL., 2003; SALVUCCI, 2006], not a viable option. Therefore, developing driving behavior algorithms for automated vehicles are done mostly based on rule-based control algorithms. However, the conventional rule-based models driving are incapable of representing many factors that drivers take into account and it is not possible to reduce human driving decisions to a few IF-THEN rules. For example, these models usually do not respond according to different traffic situations with specific driving regulations or driving cultures in which the vehicle might be.

Prediction of the exact driving behavior of humans is a very burdensome task because the human intent is highly dependent on external factors such as emotions, distractions and weather conditions [KILPELÄINEN & SUMMALA, 2007; PÊCHER ET AL., 2011; STEINHAUSER ET AL., 2018]. Therefore, there is a distribution of possible actions that drivers may take in every traffic situation. Thus, human drivers' decisions may be thought of as a random process, and it is essential to model the associated distributions as accurately as possible. Many approaches have tried to learn drivers' intent and behaviors directly from the human source, such as using eye and facial trackers or from the traffic data captured from outside of the vehicle [D'ORAZIO ET AL., 2007; HE ET AL., 2018; MANGAROSKA ET AL., 2018]. With the help of this information and by planning around and anticipating possible behaviors the drivers may take, the developers can minimize conservatism or aggressiveness in the decision-making policies of automated vehicles, which can lead to more efficient traffic flows [FRIEDRICH, 2016], yet still maintaining an acceptable level of safety. The driving capability of the vehicles after this learning strategy, in which they are exposed to a vast number of traffic situations, can be analyzed through microscopic traffic simulations.

For the development and impact assessment of automated vehicles, microscopic traffic simulation tools are widely used [BAGLOEE ET AL., 2016]. These tools utilize car-following and lane-changing models to represent each driver's maneuvering behaviors in traffic and have been applied in intelligent transportation systems (ITS) and automated vehicles studies as a cost-effective alternative to field tests for many years [MOTAMEDIDEHKORDI ET AL., 2016; HARTMANN ET AL., 2017]. These simulation tools can be used to create virtual scenarios in which the lane-changing model is an essential component for replicating real-world traffic conditions.

The accuracy of the modeling and simulation of traffic behaviors is crucial to obtain credible results and ensuring that models are clearly understood, appropriately designed, and carefully calibrated is of high importance [RAHMAN ET AL., 2013]. Accurate driving behavior models are critical for realistic simulation of driving scenarios and can contribute to advance research in the development of automated vehicles [KUEFLER ET AL., 2017]. With the limitations of the state-of-the-art models, their implementation in a simulation environment can result in unrealistic traffic characteristics and lead to errors in the corresponding analysis and bias in planning and policy decisions [CHOUDHURY, 2007], especially in studying the impact of new technologies such as automated vehicles. Therefore, it is necessary to rethink the currently existing microscopic traffic models. Moreover, these models do not fully account for the heterogeneous behavior of drivers. Unfortunately, most of the widely used models in microscopic traffic simulation tools require a lot of time and effort for calibration and validation of the models and tweaking the parameters, which should be conducted for each new situation specifically. The adaptation of a microscopic model to a local situation is time-consuming and often the necessary measurement data is difficult to obtain [FELLENDORF & VORTISCH, 2001].

Nowadays, advances in computing technologies, the growing volumes and varieties of available data, as well as affordability of data storage have motivated the use of data-driven methods to automatically acquire models of driving behavior [GINDELE ET AL., 2015]. Data-driven models focus on ML methods that can be used to build models for complementing or replacing physical-based models. As illustrated in Fig 1.2, the physical models understand a driving behavior, formulate it with a mathematical or physical model, and calibrate the model to enable the model to make predictions, whereas the data-driven approaches deploy ML tools and model to understand the underlying behavior from the data [The Royal Society, 2017]. In other words, the data-driven models, in contrast to the physical models, which are based on well-established mathematical or physical models, build relationships between input and output data using statistical or ML techniques, without worrying too much about the underlying physical model. Hence, adapting the model in data-driven models to the new situations is easier than the conventional rule-based models in terms of tweaking the parameters and calibration. In order to adapt the model to new situations, the small details should be adjusted, and the adaptation is faster and more convenient than the time-consuming calibration and validation of a physical model.

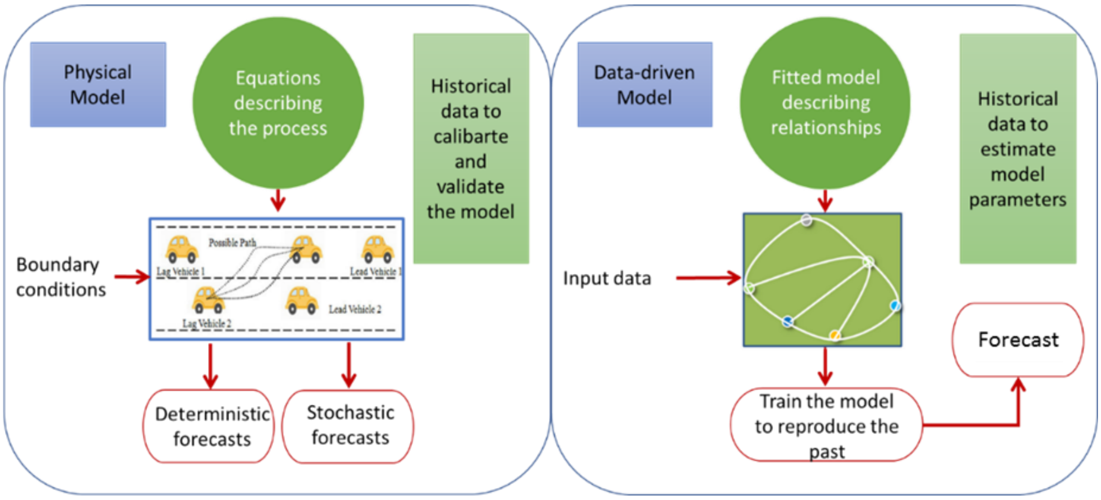


Fig 1.2 Difference between physical models and data-driven models

Based on the abovementioned factors, this dissertation aims to provide a general approach for modeling the tactical (decision-making) and operational (execution) lane change behavior from real-world observations. The advantage of this model is two-fold: firstly, this method provides realistic models for the tactical and operational level of lane changing of conventional human-driven vehicles and accounts for stochasticity among drives, which enables more accurate investigations and projection of the impacts of automated vehicles using microscopic traffic simulation. Secondly, this model facilitates the development of socially acceptable lane changing behavior of automated vehicles from observations of humans, acquired and used under the same real-world conditions.

The model proposed in this dissertation employs a common approach to gather real driving data and use ML to imitate and model the lane changing behavior from the observed data. This model uses ML methods to extract and reflect the detailed information buried in the data and deploys the mathematical tools that have also been implemented for the motion planning of automated vehicles. This model, however, adapts its level of detail to have a reasonable yet realistic complexity when combined with the microscopic modeling of lane changing.

1.2 Research Questions

The following questions guide the research conducted within this dissertation:

- How can the socially acceptable driving behavior be sampled from the dataset of individual vehicle trajectories?
- Can the tactical and operational lane changing behavior for automated vehicles be learned from the observations of lane changing behavior of human drivers?

- How can this methodology be generalized for the behavior of automated vehicles in different driving situations?

1.3 General Approach

Fig 1.3 illustrates the general approach to learn the lane change from human drivers. First of all, literature about the driving behavior task, the conventional lane changing models, the models used for motion planning in automated vehicles and the data-driven models are reviewed extensively. Moreover, widely used microscopic traffic simulation tools and their capabilities in modeling human factors and external elements are assessed. Based on the reviewed literature, the limitations of the currently existing lane changing models and the research gaps are identified.

In the following chapter, various types of traffic data collection techniques are explained and the method that can fulfill the requirements to answer the research questions is identified. Moreover, various options for extracting the trajectories from the video data collection are discussed in detail. After conducting the data collection, the videos are post-processed and individual vehicles trajectories are extracted from them.

The extracted individual trajectories from video observations need to be preprocessed before being used for model development. Therefore, the drivers were clustered into “timid”, “moderate” and “aggressive” based on their driving behavior. The clustering process looks at driver norms and identifies the driving behaviors that deviate from the norm and are more aggressive than other drivers. In order to use the database for developing a model for automated vehicles, the aggressive driving behavior is purposefully filtered out from the data.

Later, the models that represent the behavior of human drivers and automated vehicles are learned in two steps. First, the decision process of lane changing is modeled with classification. In this model, based on the state of the ego vehicle and the environment model, which are translated into features, the model predicts lane changing decisions of the vehicles. Secondly, the execution of lane changing is modeled with Inverse Reinforcement Learning (IRL) method.

After reaching the reconstructed trajectories with the desired accuracy, the model is implemented in a simulation software SUMO [KRAJZEWICZ ET AL., 2012] to check the plausibility and evaluate the performance of the model in the traffic scenarios which were not existent in the training dataset. In the end, conclusions are drawn about the transferability of the developed models.

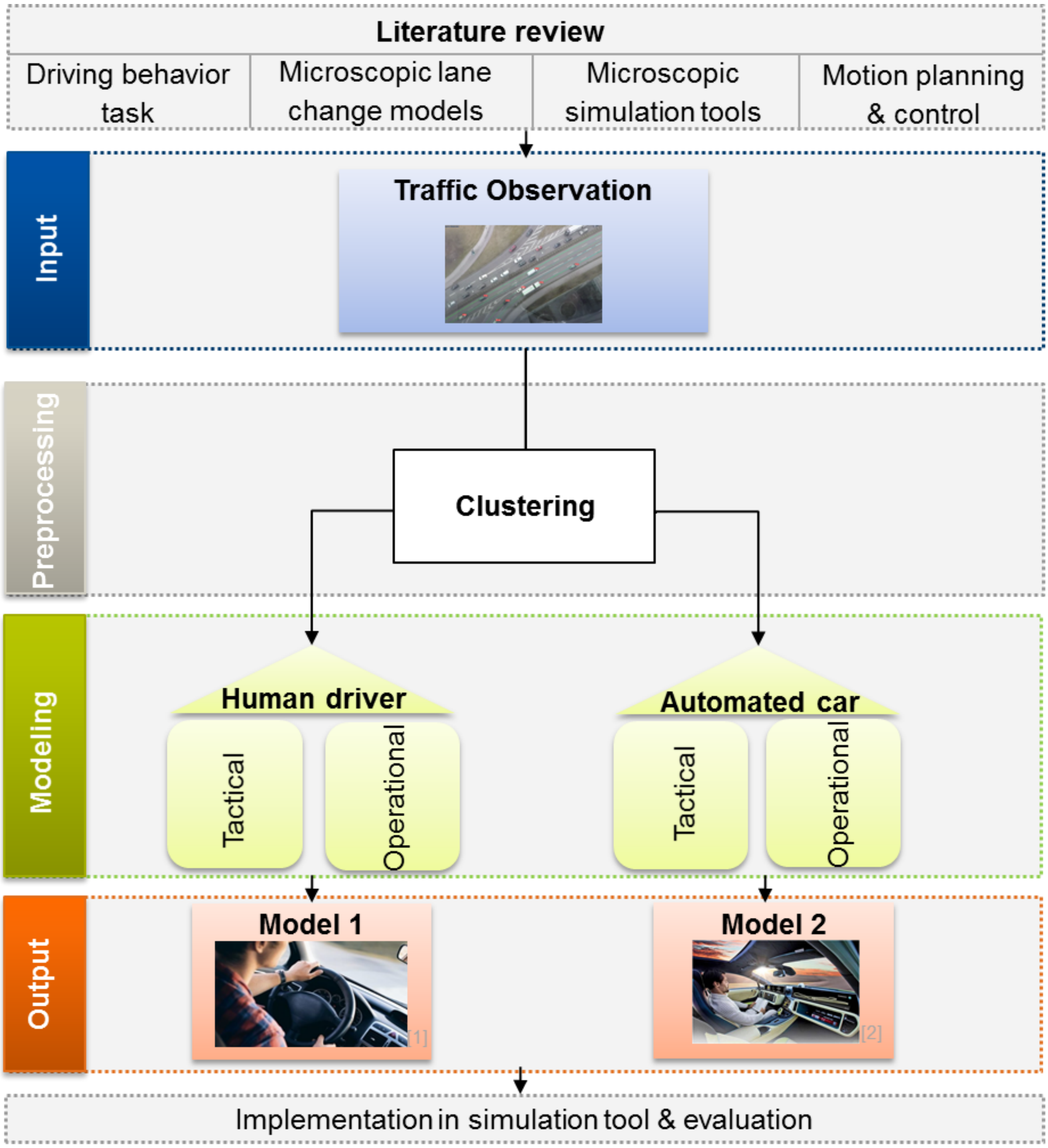


Fig 1.3 Research workflow

1.4 Thesis Outline

In chapter 2, a literature review of the existing lane changing models and motion planning algorithms for automated vehicles are presented. The data collection and trajectory extraction are discussed in chapter 3. Chapter 4 summarizes the clustering of driving behavior. In chapter 15, the modeling process of this research is presented. First, a methodology to estimate the decision-making for changing the lane from discrete trajectory data is presented. Then, the model for learning the execution of lane changing maneuvers is explained. The simulation experiment is presented in chapter 6.

2. Literature Review

The modeling of lane changes is usually considered a multi-step process. Therefore chapter 2.1 explains the most well-known models for driver behavior modeling, which is followed by influencing factors on driving behavior. In chapter 2.2, the extensive body of research about microscopic lane change modeling is explained. Chapter 2.3 concentrates on widely-used microscopic traffic simulation tools and assesses their capabilities. Chapter 2.4 summarizes the methods used for motion planning and control of automated vehicles. Finally, chapter 2.5 summarizes all the literature and methods and analyses the research gap in the currently existing methods.

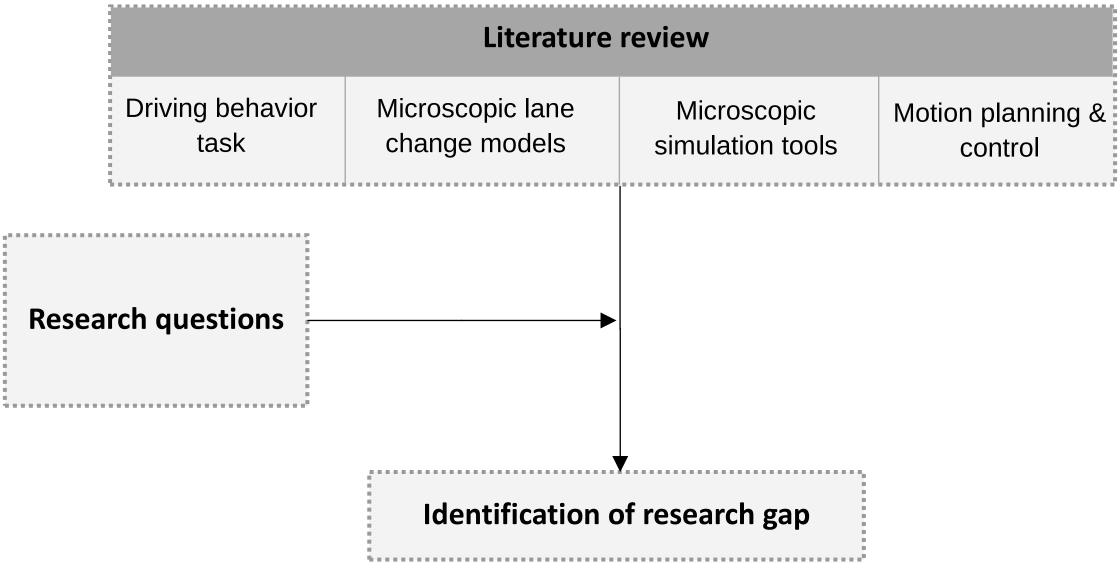


Fig 2.1 Identification of research gap workflow

2.1 Driving Behavior Task

Driving behavior is determined by the characteristics of the driving task and by a variety of external and internal factors. This section, therefore, reviews the most well-known models used for describing the task of driving behavior and discusses the factors which can influence the driving behavior.

2.1.1 Description of Driving Behavior

One of the general ways of dividing the driving's task is to use the three levels of behavior and control: These three levels are called strategic, tactical and operational based on Michon [1985] and later called navigation, guidance and control by Lunenfeld and Alexander [1990].

The highest level in the hierarchical model of the driving task shown in Fig 2.2 is the strategic level which is the planning stage of a trip, incorporating the determination of trip goal, route and vehicle choice, and evaluation of the risks and costs associated with it [MICHON, 1985]. This level sets criteria for factors at the lower level, such as speed control and risk levels. The driver makes a series of tactical decisions while executing the strategic level decision [MORIDPOUR ET AL., 2010b]. The tactical level includes choices and actions related to the interaction with the road and other road users, such as changing the lane, keeping the distance, and taking a turn [MICHON, 1985]. The time resolution of this behavior can be from seconds to several minutes. At the tactical level, maneuvers are selected to fulfill short term goals such as a decision to pass a slow-moving vehicle or maintaining the desired speed. At the tactical level, the time required for making and executing the decisions is between five and 30 seconds [ALEXIADIS ET AL., 2014]. The actual behavior, which is displayed under the strategy and the tactics of the trip, is control of the lowest level. At this level, the maneuvers are converted to control operations and the basics of steering and braking come directly into the play [MICHON, 1985]. In this level, the control actions take place on a time scale of less than five seconds [ALEXIADIS ET AL., 2014].

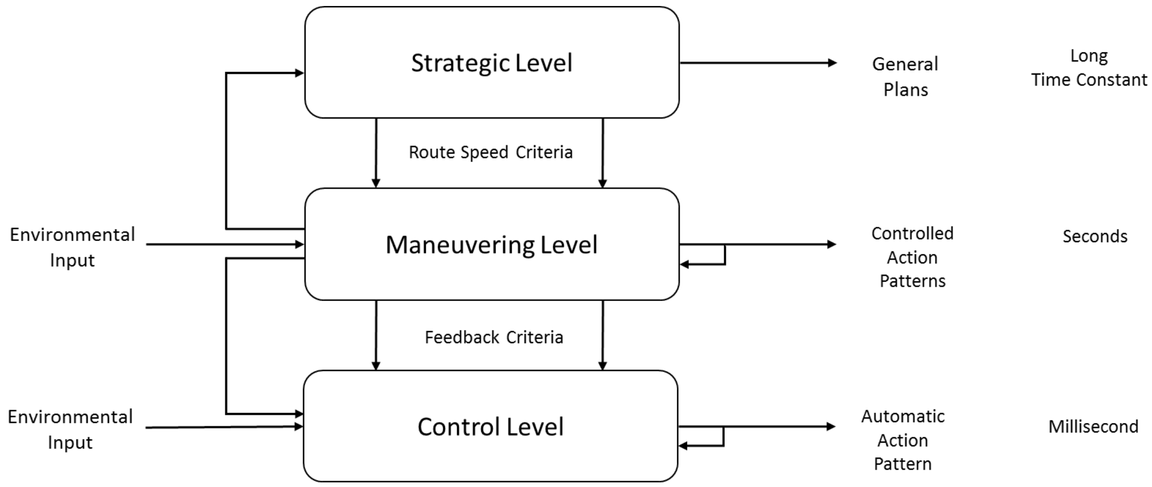


Fig 2.2 A hierarchical model of the task of driving [MICHON, 1985]

A very similar hierarchical order of driving behavior in strategic, tactical and operational task levels has been proposed by van der Molen and Bötticher [1988]. In their model, the driver's behavior is connected to the physical environment and the perception made of it. Moreover, their approach provides a concept for behavior alternatives and subjective probabilities and also considers perceptual, judgmental and decision processes of traffic participants at all levels of behavior [LÜTZENBERGER & ALBAYRAK, 2014]. As can be seen in Fig 2.3, this model offers many elements that are not easy to control in an empirical study [KESKINEN ET AL., 2004].

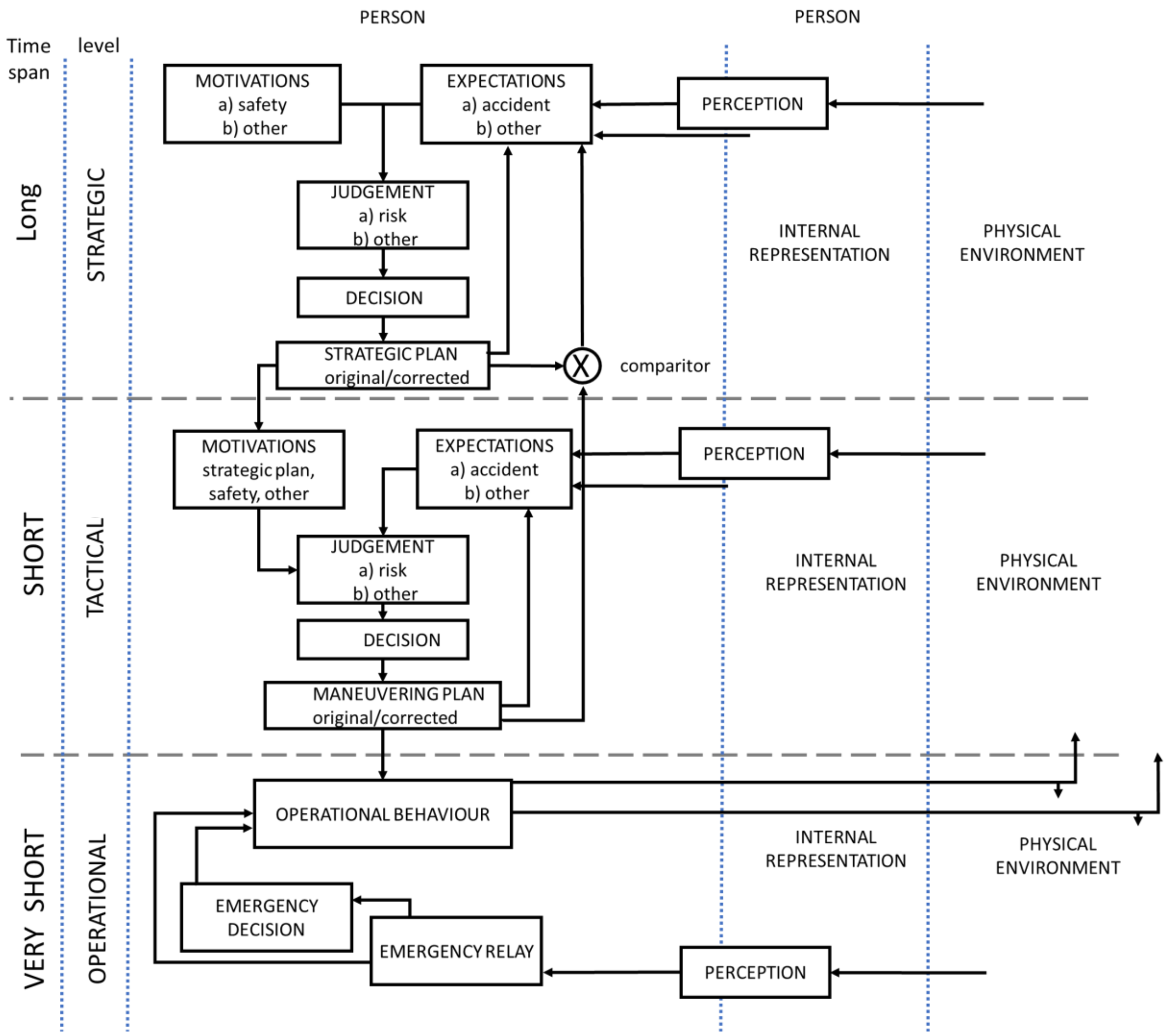


Fig 2.3 Hierarchical risk model for traffic participants [VAN DER MOLEN & BÖTTICHER, 1988]

The Matrix of Tasks [HALE ET AL., 1990] extends the hierarchical control model of Michon not just by adding extra levels or more detailed specifications, but by an additional horizontal dimension which includes factors such as different levels of expertise, familiarity with the surrounding situation and incorporates motivational aspects. These conditional factors will be of two sorts [HALE ET AL., 1990]:

- Constant conditions that have a time dynamic of at least minutes if not hours such as traffic density, visibility conditions and state of the road surface.

- Rapidly varying factors such as braking by a car in front, another vehicle signaling or moving out of the lane, the appearance of warning signs and noticing signs of drowsiness.

These factors represent the differing circumstances under which the task has to be carried out and the factors that trigger starting on one or another sub-task. Hale et al. [1990] simplified the reality into the matrix shown in Tab. 2.1. This matrix illustrates the three levels of task classification [MICHON, 1985] on the horizontal dimension against three levels of behavior

[RASMUSSEN, 1987] on the vertical dimension. The table contains typical tasks carried out in each box. Based on this model, knowledge-based behavior is applied when dealing with a difficult environment condition or unfamiliar traffic. Moreover, this behavior level is applied when the driver is still a novice. Rule-based behavior that accounts for the standard interaction with other road users is also applied when the driver should transfer his automatic routines to a new system. Rule-based behavior is applicable in familiar traffic situations [LÜTZENBERGER, 2014].

	Planning	**Maneuver**	**Control**
Knowledge	Navigating in strange town	Controlling a skid on icy roads	Learner on first lesson
Rule	Choice between familiar routes	Passing other cars	Driving an unfamiliar car
Skill	Home/work travel	Negotiating familiar junctions	Road holding round corners

Tab. 2.1 Matrix of Tasks [HALE ET AL., 1990]

Summala [1996] also conceptualized the driving task into a task cube, which combined three different concepts: As shown in Fig 2.4, a functional hierarchy that included vehicle choice, trip decisions, navigation and guidance and vehicle control. The functional taxonomy consists of passing maneuvers, crossing management, obstacle avoidance, headway control and lane keeping. The psychological processing dimension of the proposed cube has three layers: decision-making, attention control and perceptual-motor control. Later, Summala [1997] concluded that the motivational factors should be given higher importance, extended the driver task cube and developed the Filter Model of Risky Behavior and Road Accidents.

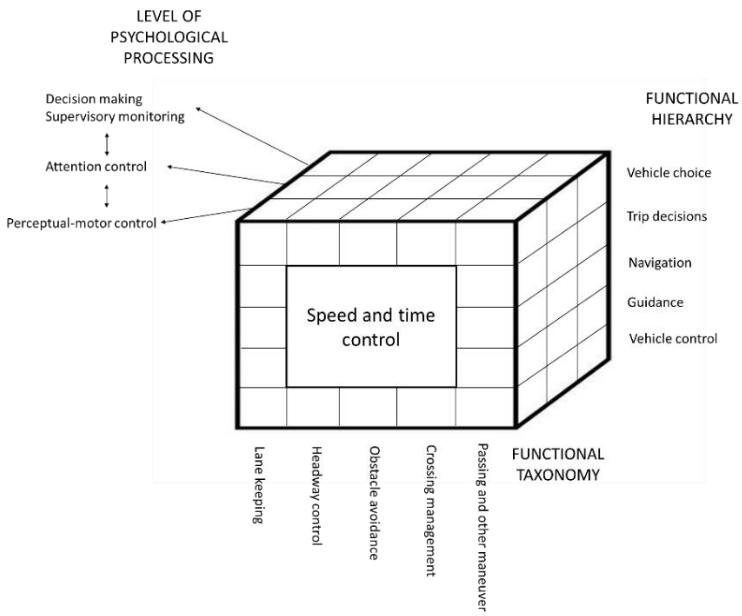


Fig 2.4 Driver task cube [SUMMALA, 1996]

Hierarchal levels of driving behavior [KESKINEN ET AL., 2004], shown in Fig 2.5, consist of four layers. On the Driving Maneuvering level, the driver controls the speed and position of the vehicle. On the Mastering Traffic Situations Level, the driver adapts to the current traffic situation. On the next level, Goals and Context of Driving level, drivers consider factors such as the purpose, their surrounding environment and their social context. Although the first three layers of this model are similar to the levels of Michon model [1985], a new component in this model is the connection between the strategic level and the driver's environment. Moreover, the fourth layer accounts for the personality of the drivers and allows the driver to be "less congruent with the norms of the society" [KESKINEN ET AL., 2004].

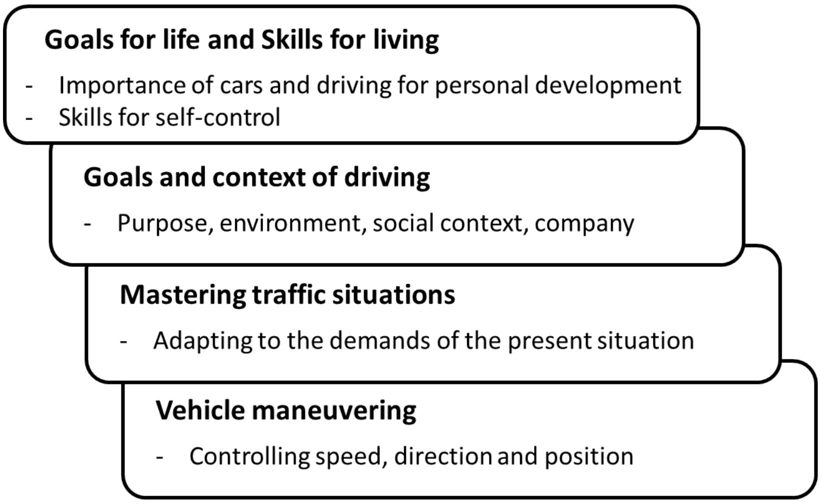


Fig 2.5 Hierarchical levels of driving behavior [KESKINEN ET AL., 2004]

Reviewing the abovementioned models has shown that most of the well-known models translated the driving behavior as a hierarchically ordered structure with various layers and considered comprehensive support for the drivers' cognitive ability. Some of these models added extra layers to Michon's model [1985] and argued that for having a comprehensive model for human driving behavior, additional layers or dimensions should be added to Michon's model [HALE ET AL., 1990; SUMMALA, 1997; VAN DER MOLEN & BÖTTICHER, 1988]. Although the model by Keskinen [2004] related the driving behavior of the driver to the environment and considered the adaptation of the driver to the social environment, other essential factors such as emotions, the capability of self-awareness were not included in any of these models.

The focus of this dissertation is limited to the maneuvering (tactical) driving behavior level and control (operational) behavior level of lane change modeling. However, inspired by the driving behavior model of Keskinen et al. [2004], the proposed model considers the current traffic situation into account and adjusts itself based on social context and norms.

2.1.2 Influential Factors on Driving Behavior

Many factors influence the driving behavior at both tactical and operational levels and lead to heterogeneous driving behavior, which are summarized in Tab. 2.2. This table categorizes the factors into the following groups: driving ability, development factors, personality characteristics, demographic factors, perceived environment and driving environment.

One of the leading factors in driving behavior is driver characteristics. Driving is the outcome of a psychological process that translates data, signals and messages into action, which continuously adapts to the exchange of varying stimuli between the driver, environment and the vehicle [PASETTO & BARBATI, 2011]. As can be seen from Tab. 2.2, in the last decades, various aspects of human driving behavior and their impacts on traffic flow and safety, driving speed have been extensively investigated. Some studies have shown that the motivation of drivers plays an important role in driving manners such as aggressiveness and risk-taking, time pressure [CŒUGNET ET AL., 2013; RENDON-VELEZ ET AL., 2016] and desire to reach the destination might motivate them to drive faster [FILDES ET AL., 1991; MOGHADDAM & JEIHANI, 2017].

During the lane changing process, the driver changes the lane to keep the comfortable driving environment, increase speed or gain access to the desired travel path [TOLEDO ET AL., 2003]. The underlying desire for lane changes and the execution process of lane changes are influenced by both external and internal factors [AHMED ET AL., 1996]. Factors such as geometric condition, surrounding traffic condition, acceleration/deceleration capability of the vehicle, and drivers' experience and preference that involve in the decision process are considered in the lane changing model. Effect of surrounding traffic characteristics has been studied by Moridpour et al. [2010a] and the results have shown that for the motivation to change the lane, the speed of surrounding vehicles in the current lane, the space gaps of surrounding vehicles in the current lane and the target lane are important. Moreover, the authors of similar studies [AL-KAISY ET AL., 2002; AL-KAISY ET AL., 2005] believed that heavy vehicles might impose some physical and psychological effects on surrounding traffic due to their physical (e.g., length and weight) and operational (e.g., acceleration and deceleration) characteristics.

Tab. 2.2 Factors influencing the driving behavior

Factor	Example	Reference
Driving ability	Skill	[Fuller & Santos, 2007; Lajunen & Summala, 1995]
	Experience	[Nabatilan et al., 2012; Patten et al., 2006; Shinar et al., 2005; McCartt et al., 2003; Groeger & Brown, 1989; Tao et al., 2017; Lajunen & Summala, 1995]
Development factors	Physical factors e.g., drowsiness	[Wang et al., 2016; Shinar, 2017]
	Psychological factors e.g., emotions	[Dumetz, 2016; Steinhauser et al., 2018; Pêcher et al., 2011]
	Behavioral factors e.g., alcohol or substance use	[Zhao et al., 2014; Shinar, 2017; Laude & Fillmore, 2015; Shyhalla, 2014; Brown et al., 2013; Ogden & Moskowitz, 2004]
Personality characteristics	Risk-taking propensity	[Iversen, 2004; Lajunen & Summala, 1995]
	Aggressiveness	[Tao et al., 2017; Lajunen & Summala, 1995]
	Susceptibility to peer pressure	[Bingham et al., 2016; Simons-Morton et al., 2011; Curry et al., 2012]
Demographic factors	Age	[Shinar et al., 2005; Cantin et al., 2009; Hakamies-Blomqvist et al., 1999; Claret et al., 2003; Groeger & Brown, 1989; Tao et al., 2017; McGwin et al., 1998; Fuller & Santos, 2007; Owens et al., 2007; Williams, 2003]
	Sex	[Bener & Crundall, 2008; Al-Balbissi, 2003; Claret et al., 2003; Groeger & Brown, 1989]
	Employment	[Moghaddam & Jeihani, 2017]
	Education	[Borrell et al., 2005]
Perceived environment	Social and peer norms	[Meesmann et al., 2015; Carter et al., 2014]
	Cultural norms	[Dumetz, 2016; Redshaw, 2011]
	Risk perception	[Carter et al., 2014; Ivers et al., 2009; Kuang et al., 2015; Rundmo & Iversen, 2004]
Driving environment	Darkness	[Salvendy, 2012; Tielert et al., 2010; Assum et al., 1999; Bella & Calvi, 2013; Bella et al., 2014; Owens et al., 2007]
	Road geometry	[Stutts et al., 2001; Hamdar et al., 2016; Abele & Møller, 2011]
	Weather condition	[Stutts et al., 2001; Salvendy, 2012; Hamdar et al., 2016; Elfaouzi et al., 2010; Amal & Hall, 1994; Tielert et al., 2010; Jägerbrand & Sjöbergh, 2016; Kilpeläinen & Summala, 2007]
	Purpose of trip	[Fildes et al., 1991; Moghaddam & Jeihani, 2017]
	Time pressure	[Cœugnet et al., 2013; Rendon-Velez et al., 2016]
	Traffic congestion	[Gugerty et al., 2004; Balk et al., 2006; Moghaddam & Jeihani, 2017]

2.2 Microscopic Lane Change Models

Developing behavior models for lane changing requires a thorough understanding of drivers' behavior during this driving task [RAHMAN ET AL., 2013] and traffic situation both in the original lane and in the host lane [SHINAR ET AL., 2005]. Moreover, modeling the lane changing is an essential and challenging part of microscopic traffic simulation tools as well as the development of automated vehicles. Therefore, there is a need to model the lane changing as precisely and realistically as possible. There is a substantial body of research on lane changing models, which will be summarized in this chapter.

2.2.1 Different Types of Lane Changing

Most lane change models categorize the lane changes, depending on the driver's motivation, in mandatory lane change (MLC) or discretionary lane change (DLC) [GIPPS, 1986; YANG & KOUTSOPOULOS, 1996; AHMED ET AL., 1996; HALATI ET AL., 1997; ZHANG ET AL., 1998; AHMED, 1999; HIDAS & BEHBAHANIZADEH, 1999; HIDAS, 2002]. These two lane changing types and the motivation behind each are described in this section.

The drivers execute MLC in order to follow their desired path or due to a lane drop or yielding to traffic near the ramp. Some variables that can affect such MLC decision include the number of lanes to cross to reach a lane connected to the next link, the remaining distance to the point at which lane change must be completed [TOLEDO & ZOHAR, 2007], and whether the subject vehicle is a heavy vehicle [MORIDPOUR ET AL., 2010a]. In case MLC involves crossing several lanes, drivers are likely to respond earlier. A longer delay makes a driver more anxious and leads to an increase in the likelihood of responding to MLC situations. Due to the lower maneuverability and larger gap length requirement of heavy vehicles, they have a higher likelihood of responding to the MLC conditions [MORIDPOUR ET AL., 2010a].

DLC occurs when drivers seek better traffic conditions, for instance, higher speed or safety. In this type of lane changing, the drivers intend to improve the perceived driving condition, and no urgency is linked to the maneuver. In some studies [LAVAL & DAGANZO, 2006; LAVAL & LECLERCQ, 2008], DLCs are concluded to be triggered by speed differences between adjacent lanes. These researchers assumed that the probability of lane changing proportionally increases as speed difference increases. However, the linear relation between the increasing probability of lane changing and the speed difference is not verified. Another study [KAN ET AL., 2009] presented a systematic investigation of drivers' motivation during discretionary lane changing movements on a multi-lane freeway section. Knoop et al. [2012a] studied the traffic conditions, speed difference and density values in the origin lane and the target lane, which led to a number of discretionary lane changes in free-flow conditions.

Lane changing can also be categorized into symmetric or asymmetric with respect to the lanes and the vehicles as well as the regulations [HABEL & SCHRECKENBERG, 2014]. In countries with symmetric lane changing, on roads with four or more lanes, vehicles may pass to the left or right of slower vehicles as long as the maneuver can be executed safely. In countries with asymmetric lane changing rules, 'pushy' drivers are believed to induce a lane change of a slower driver in front of them [KESTING ET AL., 2007]. Although the symmetric model is interesting for theoretical considerations, asymmetric models are known to be more realistic [RICKERT ET AL., 1996].

2.2.2 Lane Change Modeling Review

Algorithms and models for lane changing behavior encompass a broad range of physical and mathematical, statistical and ML techniques. Following the categorization of Rahman et al. [2013], the lane change models can be divided into four categories, as illustrated in Fig 2.6: 1) rule-based models, 2) incentive-based models, 3) discrete-choice probabilistic models and 4) Artificial Intelligence (AI) models.

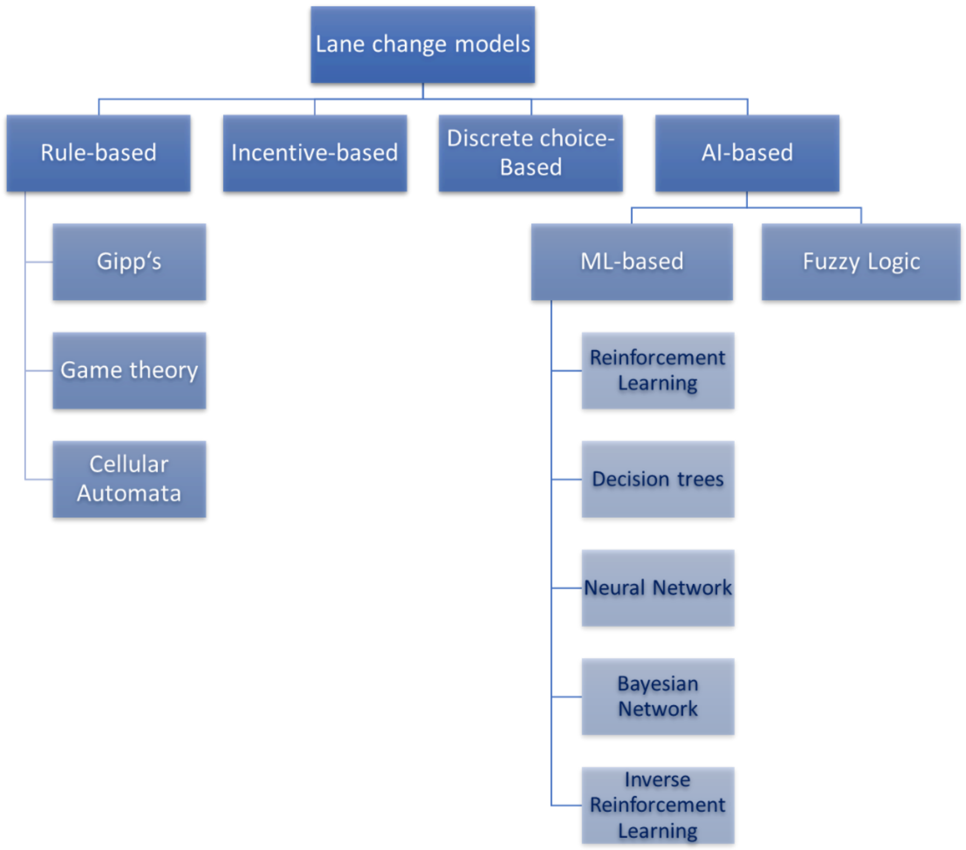


Fig 2.6 Classification of lane changing models

Rule-based models

Most of the conventional rule-based models consist of two steps: lane selection and gap acceptance. The first researcher who differentiated the desire to change the lane and the execution of the lane changing was Sparmann [1978]. His model implemented a psycho-physical threshold on the relative speed and the spacing to define the points that the drivers will respond to, based on the traffic observations of over a one-kilometer section of German freeway equipped with detectors at 100-meter intervals. This model established the relationship between frequencies of passing and lane changing as a function of traffic volume.

Gipps [1986] introduced a model that considered the necessity and desirability of the lane change. Gipps' model was developed for the urban driving situation in which external factors such as traffic signals play a role in lane changing decisions. In lane changing models, drivers evaluate the gaps between the lead and the lag vehicles in the target lane before executing the lane changing. The drivers compare the available gap with the critical gap and make a binary decision whether to accept it or not. In Gipps' model [1986], the drivers consider the lead and lag gap separately. In this model, the necessary decelerations for the ego vehicle to follow a new leader and for the new lag to follow the ego vehicle are calculated and in case these values fall below a certain threshold, the gaps are considered to be acceptable for execution of lane change.

INTEGRATION model [VAN AERDE ET AL., 1992] was the first attempt to provide a single model that considers both freeways and arterials, as well as both traffic simulation and traffic assignment. In this model, DLCs are executed by computing the potential speeds in adjacent lanes and comparing the speeds with the predefined values. The model then allows the vehicle to drive on the lane, which permits the driver to travel at the highest speed. However, the lane changing maneuver is subject to the availability of the minimum gap in the host lane. MLCs, on the other hand, are based on lane continuity and require that each vehicle should be in one of the lanes that is directly connected to the future turning movement of the vehicle.

MITSIM [YANG & KOUTSOPOULOS, 1996] defined a rule-based lane changing model in which the probability of initiating the MLC depends on the distance to the point that the lane change should be completed, the number of necessary lane changes and the density of traffic. Conflicting goals in this model are resolved probabilistically based on the utility theory models. In this model, DLCs are triggered when the speed is below the desired speed and the lane selection is based on a random utility model.

In CORSIM [HALATI ET AL., 1997], there are three different levels in the decision-making: motivation, advantage and urgency. The motivation of lane changing initiates from the dropping of the speed below a certain threshold. The advantage in lane changing means moving to a lane with a higher travel speed or lower queue length. The urgency is represented by the number of necessary lane changes to the point where the lane change must be completed. In this model, the urgency factor has an impact on the gap acceptance decision of drivers. However, in this model, the variability in the gap acceptance behavior of drivers is ignored.

SITRAS model developed by Hidas and Behnbahanizadeh [1999] defined different triggers for lane changing. Examples of these triggers are turning movements, lane blockages, lane drops and speed advantages. These different lane changing triggers lead to either MLC or DLC. Moreover, cooperative lane changing module of this model might be deployed by a vehicle in an MLC situation under heavily congested traffic conditions.

Kosonen [1999] proposed a model for urban traffic simulation that also included a DLC model based on the computation of the traffic pressure. This traffic pressure is an approximation of the potential deceleration rate caused by the leading vehicles and it depends on the desired speed of the driver. The pressure function is used to model drivers' lane changing decisions according to the model logic, which is combined with a minimum time before a new lane change is allowed in order to avoid a frequent lane changing behavior.

Cellular Automata:

The next type of rule-based models is Cellular Automata model developed by Nagel and Schreckenberg [1992], which is based on the assumption that a vehicle changes to another lane if a certain set of conditions are satisfied. Despite being very efficient and fast, one of the main disadvantages of cellular automata is the imposed artificial constraints of the grid (lattice discretization) which makes the simulation output hard to interpret [KNOSPE ET AL., 2004].

Nagatani [1993; 1994a] was one of the first researchers who used the Cellular Automata for two-lane traffic flow. This model examined a two-lane system with completely deterministic rules and the maximum speed of one, where cars either move forward or change lanes. An unrealistic feature of this model is states where blocks of several cars oscillate between lanes without moving forward, a feature that was fixed later in another study by introducing randomness into the lane changing [NAGATANI, 1994b].

Rickert et al. [1996] used a rule set for two-lane traffic, which consists of two parts: look ahead in own lane for obstruction and look into the host lane to check whether the adequate gap is available. Moreover, the authors analyzed the model macroscopically and pointed out the essential parameters that defined the shape of the fundamental diagram and investigated the importance of stochastic component with respect to real traffic. They concluded that discrete models are useful tools for understanding the fundamental relationship between microscopic rules and macroscopic measures.

In the model by Wanger et al. [1997], the authors successfully extended the Nagel-Schreckenberg model [1992] of single-lane traffic flow to multi-lane traffic with a set of lane changing rules. Moreover, they introduced the asymmetric lane change rules into the two-lane Cellular Automata. This model was used to model the lane changing behavior in Germany in which overtaking from the right is not allowed; therefore, the model was adapted in a way that overtaking on the right is possible, but in small velocities and with small probabilities only.

Knospe et al. [2002] developed a Cellular Automata-based model by analyzing the reproduction of the lane usage inversion and the density dependence of the number of lane changes. Their asymmetric two-lane model with a straight-forward and local lane changing rule set could successfully show the empirically observed lane usage inversion and reproduce the density dependence of the number of lane changes. Moreover, by analyzing the local measurements, they concluded that the car dynamics remained unchanged by the introduction of the lane changing rules. Therefore, their lane change extension of the single-lane model is appropriate to be used for the propagation of traffic state information to all lanes.

Li et al. [2006] took a factor for aggressive lane changing behavior into account and proposed asymmetric two-lane Cellular Automata model. In their study, they showed that aggressive lane changing behavior of fast vehicles could improve traffic flow in mixed-traffic in the intermediate density range by depressing the plug formed by slow vehicles. A model by Guzman et al. [2014] incorporated drivers' individual characteristics and acceleration constraints of vehicles in reality. For this aim, they modeled the human response by reconstructing the local behavior on a microscopic level and simulated their model on a two-lane system with periodic conditions and two types of vehicles with different lengths and different limited velocities. The simulation results showed that their extended model had better power to reproduce the empirical data and outperformed the previous lane change models.

Xu and Xu [2016] proposed an extended Cellular Automata-based lane change model, which accounts for risk description of vehicle lane changing in the driving and a better distance constraint rule compared to the previous models. Their simulation results illustrated that the distance constraint rule and traffic flow model could greatly increase the frequency of vehicle acceleration and lane changing, which is beneficial to the improvement of traffic speed, reduce the density of traffic flow and reduce road congestion, under the premise of ensuring the safety of symmetric same direction two-lane safe lane changing. Therefore, they concluded that their model could replicate the actual traffic condition more realistically than the previously-defined ones.

Li et al. [2018] analyzed the interaction between the following and the lane-changing vehicles to improve the lane changing behavior of vehicles in a microscopic traffic simulation by using the cellular automaton model. In their model, the following vehicle on the target lane was considered as the research object during the lane changing process, and the lane changing rules of three lane changing models, the free, forced, and cooperative lane changing models, were adopted.

Game Theory:

The next set of rule-based models were developed based on game theory, which is a mathematical model of strategic interaction between rational decision-makers. A game consists of a number of players, a set of strategies for each player and a payoff function to describe the outcome of each player throughout the game [NEUMANN & MORGENSTERN, 2007]. Kita [1999; 1993] and Yoo [2014] developed a driver model to better understand human drivers' various behaviors in the upcoming mixed situation of human drivers and autonomous vehicles with the game theoretic approach. He developed two different models: a driver decision model and a driving model. Wang et al. [2015] developed a game theoretic approach for lane changing and car-following control that had shown the numerical results for four vehicles at most. Talebpour et al. [2015] modeled the lane changing behavior in a connected environment with a game theory approach. However, their calibration results showed major limitations.

Game theory-based models assume agents with perfect rationality capable of determining which action will lead to an optimal payoff for him/her and the other drivers with whom he/she is interacting. Therefore, Cortés-Berrueco et al. [2016] integrated driver to driver interactions into their model; however, this model was not calibrated and validated with real data. Kang and Rakha [2017] modeled a merging traffic situation with the game theory approach. An extension of this model that included a decision-making model for merging maneuvers using a game-theoretical approach was developed later [KANG & RAKHA, 2018]. This model considered two drivers: the driver of the subject vehicle in an acceleration lane and the driver of the following lag vehicle in the host lane. In contrast to the previous model, which assumed that the driver of the subject vehicle and the driver of the following lag vehicle decide on an action at the first point only, their new model introduced the concept of a repeated game, assuming that a lane change decision is made repeatedly to adjust to changes based on surrounding conditions [KANG & RAKHA, 2018].

Incentive-based models:

The idea behind the incentive-based models, as the name represents, decides whether to change the lane or not to maximize the benefits. The first incentive-based model was Minimizing Overall Braking Decelerations Induced by Lane changes (MOBIL) model [KESTING ET AL., 2007; TREIBER & KESTING, 2009]. This model was developed based on two criteria: incentive and safety. The incentive criterion measures the attractiveness of a given lane based on its utility, and the safety indicator that measures the risk associated with lane changing. While the safety criterion prevents aggressive lane changes and collisions, the incentive criterion takes into account the advantages or disadvantages of other drivers associated with a lane change with the factor called 'politeness factor'. This model was applied to traffic simulations of cars and trucks with the Intelligent Driver Model (IDM) [TREIBER, 2002] as the underlying car-following model.

The next incentive-based model was Lane changing Model with Relaxation and Synchronization (LMRS) [SCHAKEL ET AL., 2012]. The driver's desire to change the lane is the combination of the route, speed, and keep-right incentives. In this model, a trade-off is considered while the route incentive is being dominant. Moreover, a relaxation phenomenon is defined in the model to replicate the fact that the drivers are willing to accept minimal gaps for large desires. The authors modified the IDM model and evaluated their lane change model in a simulation. They calibrated and validated their model in free-flow and congested traffic conditions with the data collected from loop detectors, which were closely spaced. Even though their model showed plausible results for free-flow conditions, the generalization of this model is unclear.

Discrete-choice probabilistic models:

Discrete choice models are used to explain or predict a choice from a set of discrete alternatives [BEN-AKIVA & LERMAN, 1985]. Most discrete-choice-based lane changing models are based on probit or logit models. For discrete-choice-based models, lane changing is usually modeled as either MLC or DLC by 1) checking lane change necessity, 2) choice of target lane and 3) gap acceptance. Each step can be formulated as a logit or probit model. Depending on the step and the number of lanes, the subject driver may face a binary or multichoice decision [RAHMAN ET AL., 2013]. These models were used for lane change modeling by several researchers.

Ahmed et al. [1996] developed a general utility-based function for the MLC and DLCs. For DLCs, not only the difference between the current speed and the desired speed is considered, but also the behavior of Heavy Goods Vehicles (HGV) and the existence of tailgating vehicles. If driving on the current lane does not fulfill the satisfaction of the driver, the drivers compare the situation on the adjacent lanes to choose the best lane. Ahmed et al. [1996] assumed that for a lane changing maneuver, both the lead and the lag gaps must be accepted. With this model, gap acceptance parameters could be estimated for both DLC and MLC situations and in MLC situations, the drivers accepted lower gaps than in DLC situations. Ahmed [1999] as well developed a model for forced merging behavior in congested traffic situations. A similar concept was developed by Zhang et al. [1998], in which MLC critical gaps are randomly distributed across the population. The mean value of the MLC critical gap is the function of the remaining distance to the location where the lane change execution should be completed. The authors validated the model but did not propose a framework for the calibration.

In order to tackle the problem of having a distinction between MLC and DLC, Toledo et al. [2003] introduced an approach that integrated both lane changes in one utility function for each lane. This model allows the drivers to consider the MLC and DLC jointly. A discrete choice framework was employed to model both the tactical and the operational behavior of drivers. For this model, the gap acceptance parameters were calibrated based on the maximum likelihood to observations.

Artificial-intelligence models:

During the last years, the progress in AI has made revolutionary progress in many areas such as robotics, autonomous vehicles and driving behavior modeling. Researchers attempted to use various AI and ML methods for modeling driving behavior. Because the driver’s physical and mental behavior is nondeterministic and highly nonlinear, it is difficult for traditional approaches to embody this kind of uncertain relationship [DING ET AL., 2013]. The methods used purely for the driving behavior for autonomous driving are explained in chapter 2.4 and this chapter focuses solely on the driving behavior modeling of human drivers.

Fuzzy logic:

fuzzy-logic-based lane changing models incorporate the uncertainty of lane changing and consider the natural and subjective perceptions by formulating the logical rules [KIKUCHI & CHAKROBORTY, 1992]. The unique nature of fuzzy logic models is that they can convert nonlinear systems into IF-THEN rules [MENDEL, 1995]. In FLOWSIM [MCDONALD ET AL., 1997], lane changing maneuvers are based on two theories: changing to a slower lane and changing to a faster lane. Das and Bowles [1999] proposed a new microscopic simulation methodology based on fuzzy rules for implementation in the AASIM software in which MLC fuzzy rules consider the distance to the next exit or merge point and the required number of lanes to change and DLC is based on the driver's speed. Another study by Moridpour et al. [2012] also developed a lane changing model using fuzzy logic to predict the lane changing maneuver of heavy vehicles on freeways.

HMM:

A Hidden Markov Model (HMM) is a statistical tool for representing the probability distributions over a sequence of observations [RABINER & JUANG, 1986]. HMMs are usually used for classification. Nishiwaki et al. [2008] modeled lane change patterns of each driver with an HMM, which is trained using longitudinal vehicle velocity, lateral vehicle position, and their dynamic features. Vehicle trajectories are generated from the HMM in a maximum likelihood criterion at random lane changing time and state duration. Their results conveyed that vehicle trajectories generated from the HMM included a similar trajectory to that of a target driver. Another study [LIU ET AL., 2016] proposed a probabilistic lane change behavior on HMM to detect the dangerous cut-in maneuvers for lane changing on highways.

Neural Network:

An Artificial Neural Network (ANN) is an information processing paradigm inspired by the information processing of the biological nervous systems. The key element of ANN is the structure of the information processing system, which is composed of a large number of highly interconnected processing elements (neurons) working together to solve specific problems [MCCULLOCH & PITTS, 1943]. Macadam and Johnson [1996] demonstrated the use of neural networks in representing the driver's steering behavior in double lane changing maneuvers and S-curve maneuvers. Due to the limited data variability for neural networks, their results were not promising and they concluded that the adaptive nature of neural networks should be used for modeling driver steering behavior under a variety of scenarios.

A study by Ding et al. [2013] predicted lane changing trajectory based on a time delay Back-Propagation Neural Network (BPNN) in real-time. The inputs of the model were the position, velocity, acceleration, and time headway of the vehicle and the future lane changing trajectory was considered as the desired output of the trained network. They validated their model with Next Generation SIMulation (NGSIM) trajectory data [Federal Highway Administration (FHWA), 2006] and a smoothing method.

Huikun et al. [2016] proposed a data-driven model to learn the lane changing process of vehicles from individual vehicle trajectories (NGSIM dataset). In this study, the lane changing was classified into two steps: the decision-making for lane changing, and the execution of lane changing. An ensemble of decision trees that decides whether or not to change lane and a back-propagation neural network is used for each step respectively. The result of their study showed more naturalistic lane changing behavior compared to the existing lane change models in simulation.

Dynamic Bayesian modeling:

Dynamic Bayesian networks (DBN) add the time component to the standard Bayesian network which allows the modeling of time series or sequences and represents the temporal evolution of variables over time [GHAHRAMANI, 1998]. In the directed acyclic graph of a DBN nodes represent random variables and edges represent probabilistic dependencies between variables across time. The key assumption in DBN is that the probability distributions describing the temporal dependencies are time-invariant. This means that the temporal evolution of the analyzed process can be reconstructed by knowing the temporal dependencies represented in the DBN graph [MURPHY, 2002].

Wheeler [2014] investigated a methodology for dynamic model construction based on a Bayesian statistical framework that was successfully employed in collision avoidance systems. The study used NGSIM Dataset and extracted of 162 features for the ego vehicle. These features included inherent vehicle properties, relative features between vehicles, roadway relative features, and aggregated features over a vehicle's history. In this study, features were inspired by those used in the literature [KASPER ET AL., 2012; SCHLECHTRIEMEN ET AL., 2014].

Gindele et al. [2015] modeled the decision-making process of drivers from traffic observations by building a hierarchical Dynamic Bayesian Model that describes physical relationships as well as the driver's behaviors and plans. In this model, they showed a novel approach for tactical decision-making for autonomous driving, which is based on a continuous Partially Observable Markov Decision Process (POMDP) that uses the presented model for prediction.

Reinforcement Learning approaches:

Reinforcement Learning (RL) has the capability of dealing with time-sequential problems and seeking optimal policies for long-term objectives by learning from trials and errors [WANG ET AL., 2018] and therefore has been used for modeling the driving behaviors.

Hoel et al. [2018] introduced a method for speed and lane change decision-making based on Deep Reinforcement Learning (DRL). In this study, the agent was trained in a simulated environment to handle the speed and lane change decision. Shalev-Shwartz [2016] applied DRL to the problem of forming long term driving strategies for autonomous vehicles. Kobert et al. [2013] developed a general framework for RL in robotics and Sallab et al. [2017] came up with a framework for using DRL for autonomous driving and used simulation results to demonstrate learning of autonomous maneuvering in a scenario of complex road curvatures and simple interaction of other vehicles.

Chen [2018] developed a model to generate human-like lane following and changing behaviors. In this study, the author used Long Short-Term Memory (LSTM) to make the vehicle's trajectory prediction by demonstrating of human driving trajectories. They divided the lane changing task into predicting the driver's discrete intention and forecasting the subsequent continuous trajectory and solved the sequential prediction task with LSTM. Finally, the author compared and evaluated the prediction results with real human driving trajectories in the NGSIM dataset.

Imitation learning approaches:

Imitation learning (IL) approaches rely on data provided through human demonstration with the aim to learn a policy that behaves similarly to an expert agent. One of the methods of IL is Behavioral Cloning (BC), which learns a policy as a supervised learning problem over state-action pairs from expert trajectories [Ho & Ermon, 2016]. The BC approach, developed by Michie et al. [1990], is conceptually sound plausible but fails in practice as small inaccuracies compound during the simulation. Inaccuracies result in the policy to states that are underrepresented in the training data, which leads to inaccurate predictions and ultimately to invalid situations. BC succeeds only with a large amount of data due to the error caused by the covariant shift [Ross & Bagnell, 2010]. These shortcomings motivated work on alternative IL methods, such as IRL.

IRL approaches:

IRL learning assumes that the expert follows an optimal policy with respect to an unknown reward function. If the reward function is recovered, a policy that behaves identically to the expert can be simply found using RL. IRL thus generalizes much more effectively and does not suffer from many of the problems of RL. Because of these benefits, there are more emphasis on modeling human driver behavior using IRL in recent efforts. IRL is explained extensively in chapter 5.2.1.

In the study from Gonzalez et al. [2016], the authors showed how the realistic driver model learned from demonstrations via IRL can be used to predict the long-term evolution of highway traffic scenes. In their model, each traffic participant is considered as a Markov Decision Process (MDP) in which the cost function is a linear combination of static and dynamic features. In particular, the static features capture the preferences of the driver while the dynamic features, which change over time depending on the other traffic participants' actions, capture the driver's risk-aversive behavior. However, the result of this study is for simulation and the ones which are tested on real data have an egoistic view from the car.

In a study by Sadigh et al. [2016a], the authors deployed the combination of IRL and game theory to model the driving behavior of human drivers. The main idea in their research is the fact that the drivers do not act in isolation and the actions of one driver affect the other ones and vice versa. Their model gave promising results when tested with humans in the loop; however, the model was derived using deterministic techniques.

2.2.3 Calibration of Lane Change Models

Verification, validation and calibration of models are critical to the reliability and credibility of the model results [HELLINGA, 1998]. Calibration is the process of systematically tweaking model parameters so that the model can reproduce the observed traffic conditions. Dowling et al.[2004] define calibration as below:

> *"Calibration is necessary because no single model can be expected to be equally accurate for all possible traffic conditions. Even the most detailed microsimulation model still contains only a portion of all the variables that affect real-world traffic conditions. Since no single model can include the whole universe of variables, every model must be adapted to local conditions."*

The calibration process is continued until the error between the performance measures taken from the field data and the performance measures calculated in the simulation is less than a predetermined margin of error [ALEXIADIS ET AL., 2014] and model validation is the process of checking to what extent the model replicates the reality [TOLEDO & KOUTSOPOULOS, 2004]. The adaptation of a microscopic model to a local situation is time-consuming and often, the necessary measurement data is difficult to obtain [FELLENDORF & VORTISCH, 2001].

For the calibration and validation of microscopic lane changing models, various empirical traffic data have been used. Some studies used individual trajectories [BALAL ET AL., 2016; YANG & KOUTSOPOULOS, 1996; LECLERCQ ET AL., 2007; THIEMANN ET AL., 2008; MORIDPOUR ET AL., 2010a; SCHAKEL ET AL., 2012; KNOOP & BUISSON, 2014; PARK ET AL., 2015; BEN-AKIVA ET AL., 2006; PATIRE & CASSIDY, 2011; ESSA & SAYED, 2016] but they also have some shortcomings. For example, some models used the vehicle trajectories in one lane but did not consider what happened between the vehicles in different lanes; some presented useful microscopic results; however, they did not have an integrated framework for lane changes [JIN ET AL., 2018].

Another dataset used for calibration and validation of lane changing is travel time [KIM, 2006; SCHAKEL ET AL., 2012]. Other researchers [SCHAKEL ET AL., 2012; KNOOP & BUISSON, 2014] used loop detector data, but this kind of observation limits the possibility of observing the performed lane changes. Therefore a method has been developed to determine the number of lane changes in loop detector counts [KNOOP ET AL., 2012b]. Moreover, some studies used macroscopic data, such as the capacity to tweak the parameters [GEISTEFELDT & GIULIANI, 2015].

2.2.4 Summary

Despite the great effort of rule-based models to model the lane changing behavior of humans with different gap-acceptance conditions and consequently different lane changing behavior for various situations, these models cannot represent and model the heterogeneous behavior of humans. Most of these models are deterministic, and even those who added the stochasticity to the model through estimated distributions fail to represent the behavior of humans realistically. Because of the multiplicity of possible driving conditions associated with discretionary and mandatory lane changes, these approaches tend to lead to complex models with many parameters [TREIBER & KESTING, 2009].

Incentive-based models account for the variability of drivers by the politeness factor in MOBIL and accepted headway, deceleration and level of desire in LMRS. Moreover, they have very few parameters to tweak and therefore, the realistic driving behavior such as heterogeneous driving behavior of the drivers cannot be modeled. However, it is not clear yet whether or not they can well represent the driving behavior in congestion [ABUALI & ABOU-ZEID, 2016].

In discrete choice models, the lane changing decisions are probabilistic instead of binary yes or no results. In Ahmed’s models [1996; 1999], the rigid separation between MLC and DLC is unrealistic in real-life driving and Toledo’s model [2003], the main weakness is the difficulty of determining the utility functions for various decision choices [RAHMAN ET AL., 2013]. In existing discrete-choice-based lane changing models, the heterogeneities in drivers such as driver aggressiveness, driving skill level have not been considered adequately.

AI-based models have the potential to be superior in dealing with complex situations without resorting to detailed hard-coded rules or pre-determined models. Some MLI-based models presented acceptable performance [HOEL ET AL., 2018; GONZALEZ ET AL., 2016; GINDELE ET AL., 2015; WANG, 2015]; however, these models are usually trained with the simulation or simulator data, and it is still not clear whether they can perform the same on the real data. Another shortcoming of these models is their computational time, which makes them impractical and unattractive for implementation in the microscopic traffic simulation tools.

This dissertation, therefore, tackles the abovementioned shortcomings in the existing models and proposes a model that learns the driving behavior from the individual vehicle trajectories. Moreover, a strong focus of this work is to develop a model that can make predictions and deliver results fast enough that it can be implemented in the microscopic simulation tool.

2.3 Microscopic Traffic Simulation Tools

Microscopic simulation software is an indispensable powerful and essential tool in traffic analysis for modelers and transport planners. A large number of simulation software are developed and used worldwide, and they provide a cost-efficient and safe mean to test emerging technologies such as automated vehicles. In this chapter, the most widely used traffic

simulation software and the underlying model for their behavior are discussed. Human factors significantly determine traffic and developers of traffic simulation frameworks are accounting for these factors and including sophisticated models to mimic human decisions [LÜTZENBERGER & ALBAYRAK, 2014]. However, there is no model in which all decisive factors on human driving behavior listed in chapter 2.1.2 are included. This chapter describes the most-widely microscopic traffic simulation tools and their components.
It is noteworthy to mention that although some other simulation software like PELOPS (that is a combination of highly detailed sub-microscopic vehicle- and microscopic traffic technical model that allows investigations about the longitudinal vehicle dynamics as well as an analysis of the course of traffic [LUSMANN ET AL., 1997]) and some other sub-microscopic simulation software (e.g., CarMaker [IPG Automotive GmbH, 2015], DYNA4 [TESIS, 2018] and CarSim [Mechanical Simulation, 2015]) do exist, due to the incompatibility between the nature of these tools such as considering the vehicles dynamics and the limitations of the traffic observations of this study, they were not studied further for this dissertation.

2.3.1 SUMO

SUMO (Simulation of Urban MObility) is an open-source, continuous, microscopic and multi-modal traffic simulation tool developed by DLR (Deutschen Zentrums für Luft- und Raumfahrt German for German Aerospace Center). The driving behavior dynamics are determined by three models in SUMO [ERDMANN, 2014]:

1. Car-following model determines the speed of the ego vehicle in relation to the vehicle ahead of it. The Driver model is SUMO is a Gipps' model extension, which was invented and described by Krauss [1997]. In this model, vehicles have a collision-free behavior and the safe velocity of each vehicle is calculated based on the speed of the leading vehicle, gap to the leading vehicle, the driver's reaction time and the deceleration function [KRAJZEWICZ ET AL., 2002].

2. Intersection model determines the behavior of vehicles at different types of intersections with regard to right-of-way rules, gap acceptance and avoiding junction blockage.

3. Lane changing model determines lane choice on multi-lane roads and speed adjustments related to lane changing.

Based on Erdmann [2014], the lane changing model in SUMO has to fulfill two main steps. First, the software computes each vehicle's lane changing decision based on the vehicle's route and the current and historical traffic condition. In this software, the drivers do the lane changing strategically, meaning that they evaluate the lanes according to their desired route and determine the urgency of a lane change. Afterward, the model computes the necessary acceleration or deceleration for the ego vehicle and the surrounding vehicle to safely execute the lane change.

The speed of a vehicle in this model is mainly determined by the leader, which may be on the same lane or the preferred successor lane after the current lane. The car-following model calculates the speed for following the leader. A vehicle can only change its lane if there is enough physical space on the target lane and if it does not come close to its leader or follower on the target lane. In case neither of these conditions is met, the vehicle is believed to have a blocking leader or a blocking follower [LI ET AL., 2014]. A vehicle that moves to a lane on the next edge is said to advance the lane, and a vehicle that changes to a parallel lane on the same edge is said to change lane. Based on the description provided by Erdmann [2015], during each simulation step, the following sub-steps are executed in order for every vehicle:

1. Computation of preferred successor lanes.

2. Computation of safe velocities under the assumption of staying on the current lane and integration with lane changing related speed requests from the previous simulation step.

3. Lane changing model computes change request.

4. Either execute lane changing maneuver or compute speed request for the next simulation step. The speed changes depending on the urgency of the lane changing request.

During each simulation step, every vehicle executes the four sub-steps, and if any vehicle satisfies the lane changing condition, then the lane changing maneuver begins. Assuming the coordinates of the ego vehicle before lane changing are (X_0, Y_0) and the speed is V_0 and the coordinates of the ego vehicle after lane changing are (X_t, Y_t) and the speed is V_t.

$$X_t = X_0 + \frac{(V_t + V_0)T}{2}$$

$$Y_t = Y_0 + L$$

Where *T* is the simulation resolution and *L* is the lane width.

2.3.2 VISSIM

VISSIM [PTV AG, 2016] developed by PTV (Planung Transport Verkehr AG) group in Karlsruhe, Germany is one of the popular programs used for microscopic traffic simulation. This multimodal simulation software can be applied for simulation in both freeway and urban areas. Car-following mode in VISSIM is based on the well-known psycho-physical car-following model by Wiedemann [1974]. Based on the basic assumption in this car-following model, a driver can be in one of four driving modes:

1. Free-driving mode, where no influence is exerted from leading vehicles. In this mode, the driver attempts to reach and maintain the desired speed.

2. Approaching mode, when the driver of the follower vehicle consciously observes that he is approaching a slower vehicle in front.

3. Following mode, where the headway for a pair of vehicles is between the maximum following headway and the safe headway. In this mode, the follower vehicle is able to accelerate or decelerate in accordance with the vehicle in front.

4. Braking mode, when the headway between vehicles drops below the desired safety distance.

Lateral movement in VISSIM can be structured in lane selection, lane changing, and continuous lateral movement within one lane. VISSIM classifies lane changes into free lane change and necessary lane change. The underlying lane change model in VISSIM is based on an improved version of Sparmann's model [1978].

In VISSIM's lane changing model, as long as a driver is not aware of any necessary lane change because he is far away from the next relevant connector, he chooses the lane with the best interaction situation. However, lane selection is often governed by mandatory lane changes for desired turns [FELLENDORF & VORTISCH, 2010]. In the case of a free lane change, the lane changing model examines whether the available lag TTC between the following vehicle in the target and the subject vehicle lane satisfies the desired safety distance and minimum time headway parameters [PTV AG, 2016]. For a lane change in a queue, the model also checks the lead TTC between the preceding vehicle in the target lane and the subject vehicle. In VISSIM software, each connector has two distances attached: the lane change distance and the emergency stop distance. The lane change distance describes when a driver becomes aware of the upcoming connector. From that point on he will consider the connector in his lane selection. The emergency distance is the distance to the connector where a driver will stop when reaching the necessary lanes to change to the connector is not possible [FELLENDORF & VORTISCH, 2010].

VISSIM implements Microsoft Component Object Model (COM) as a programming interface. The functionality provided by a COM interface can be used by different programming languages [PTV Group, 2018]. This interface provides access to the modeled road network with all its attributes, signal control, evaluations, all vehicles in the simulation and their attributes, the simulation control and extends the applications of VISSIM. The COM interface can be used to easily include VISSIM in other applications.

2.3.3 AIMSUN

Advanced Interactive Microscopic Simulator for Urban and Non-urban Networks software known as AIMSUN [Aimsun, 2018] uses a Gipps' car-following model [1981] and a development of the Gipps' lane changing model [GIPPS, 1986]. Based on Barcelo et al. [2005], the implemented car-following model in AIMSUN takes into account the additional constraints defined by Mahut [1999] on the braking capabilities of the vehicles, imposed in the classical safe-to-stop-distance hypothesis. Based on this hypothesis, if the leader should begin to decelerate to a stop at some time *t*, the follower, starting at time $t+\tau$ (τ is the follower's response time), should be able to decelerate to a stop safely.

Lane changing is modeled as a decision process that analyses the necessity of the lane change as well as the desirability of the lane change, and the feasibility of the lane change [HIDAS, 2005; HIDAS, 2004]. Once a decision was made to change the lane, the maneuver is executed in AIMSUN as an instantaneous switch from one lane to the other. The decision process depends on the distance of the vehicle from the next required turning point. Based on Barcelo et al. [2005], AIMSUN divides this distance into three zones with distinct considerations in each zone:

1. Zone 1: this is the zone farthest from the next turning point. The vehicle is in this zone only considers lane changing desirability to reach its desired speed.
2. Zone 2: the intermediate zone in which lane changing decisions are made to reach the required turning lane without the behavior of vehicles in the turning lane being affected.
3. Zone 3: the final section before the turning point. In this zone, vehicles are forced to reach their required turning lane by reducing their speed or even stopping to wait for a gap, and vehicles in the target lane may modify their behavior in order to allow the other vehicles to move into the target lane.

Moreover, AIMSUN also has an optional 'Two-lanes Car-Following Model' that can be used to force vehicles in the faster lanes of multi-lane roads to slow down. The model is controlled by four user-defined parameters: the number of vehicles considered, maximum distance ahead, maximum speed difference, and maximum speed difference on the ramp [HIDAS, 2004; HIDAS, 2005]. If a vehicle is unable to reach its required turning lane, it will get stuck in the wrong lane, and after a given waiting time, which can be adjusted by the user, it will become a lost vehicle by continuing on the wrong route. AIMSUN has a plugin API, which enables modifications during a simulation run and allows the implementation of sophisticated external logic [CASAS ET AL., 2010].

2.3.4 PARAMICS

The microsimulation model PARAMICS (Parallel Microscopic Simulation) was developed by the company Quadstone and SIAS Ltd [Quadstone LTD., 2018]. At the microscopic level, driver and vehicle interactions are simulated based on individual driver behavior and vehicle kinematics. Vehicle-to-vehicle interactions are based on vehicle following, gap acceptance and vehicle kinematics. The driver behavior is based on research conducted at the Transport Research Laboratory, where it was concluded that driver aggressiveness and awareness are sufficient to describe the behavior of most drivers [KROGSCHEEPERS & KACIR, 2001]. This software depends on a psycho-physical car-following model based on the Fritzsche model [1994], which assumes that the driver can have one of five phases: Following-I, Following-II, Danger, Closing In, and Driving Freely. These modes are determined using six following thresholds [PANWAI & DIA, 2005]:

1. Perception-Threshold Negative (PTN) is defined as the negative relative speed of a pair of vehicles.
2. Perception-Threshold Positive (PTP) is defined as the positive relative speed of a pair of vehicles.
3. Desired Distance (AD) illustrated a comfortable distance headway of vehicles, which is related to the speed of the follower vehicle.
4. Risky Distance (AR) threshold shows conditions when the distance headway is too close to be categorized as comfortable driving.
5. Safe Distance (AS) represents the threshold to enter a driving condition when the driver is unable to decelerate fast enough to avoid a risky situation.
6. Braking Distance (AB) is a threshold, which is used to avoid collisions due to high speeds and late decelerations.

The lane changing behavior in PARAMICS is also based on Fritzsche model [1994]. Lane changing action follows the established gap acceptance theory and a historical record of suitable gap availability in a link. A driver's aggression is taken into account in the lane changing model by means of adjusting his signpost distance [ESSA & SAYED, 2016].

2.3.5 BABSIM

BABSIM is a non-commercial freeway traffic flow simulation tool developed by Institute for Transportation and Traffic Engineering and the Institute of Computational Engineering of Ruhr-University Bochum on behalf of the German Federal Highway Research Institute (BASt) [GEISTEFELDT ET AL., 2014]. This software has several modules that represent the intention of the following driving tasks: car-following, overtaking, keeping right, gap control, path following, and cooperation with surrounding drivers. For each driving situation, a vote will be returned for lateral and longitudinal movement of the vehicle from each module. The final decision for each maneuver is calculated by combining the votes [HARDING, 2008]. This simulation software has Wiedermann [1974] and Sparmann [1978] model for car-following and lane changing model respectively.

2.3.6 Summary

This section summarizes the traffic flow simulation tools reviewed in chapter 2.3, and gives an overview of their capabilities in simulating different levels of driving behavior task, and the influential factors in driving behavior discussed in section 2.1.2. Tab. 2.3 illustrates five simulation tools, and the color determines to which extend the component can be simulated in that software or its underlying behavior models. As can be seen from the table, most of the influential factors for the task of driving behavior are not included in the microscopic traffic simulation tools.

All the analyzed tools include models for the tactical level of driving behavior. In most cases, these models belong to the category of stimuli-response models, such that certain events cause drivers to change their behavior. All the examined simulation tools account for a simple perception of the driving environment in which they conceptualize the perception into the distance to the neighboring vehicles. However, none of these tools consider factors such as weather condition and congestion level in their perception. Moreover, factors such as social and peer norms, cultural norms, and risk perception are not included in their perceived environment. Although the personality characteristics of the driver are not included in the underlying driving model of these simulation tools, some modelers replicate different driving behavior (aggressive, conservative) characteristics by defining different vehicle types and applying corresponding driving behavior by tweaking the model parameters. Nevertheless, this approach is accompanied by a lot of effort for finding the parameter combination which can replicate a specific behavior.

Software	Behavior Level			Influencing Factors																				
				Driving Ability		Development Factors			Personality Characteristics			Demographic Factors			Perceived Environment			Driving Environment						
	Control	Tactical	Strategic	Skill	Experience	Physical	Psychological	Behavioral	Risk Taking	Aggressiveness	Susceptibility to Peer Pressure	Age	Sex	Employment	Social and Peer Norms	Cultural Norms	Risk Perception	Darkness	Road Geometry	Weather Condition	Purpose of Trip	Time Pressure	Traffic Congestion	Traffic Control
SUMO	Implemented	Implemented	Implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable	Indirectly Implementable	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable
VISSIM	Implemented	Implemented	Implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable	Indirectly Implementable	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable
AIMSUN	Implemented	Implemented	Implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable	Indirectly Implementable	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable
PARAMICS	Implemented	Implemented	Implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable	Indirectly Implementable	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Implemented	Not implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable
BABSIM	Implemented	Implemented	Implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable	Indirectly Implementable	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Not implemented	Implemented	Not implemented	Not implemented	Not implemented	Indirectly Implementable	Indirectly Implementable

Implemented   Indirectly Implementable   Not implemented

Tab. 2.3 Overview of traffic simulation tools and their capabilities

2.4 Decision-Making and Motion Planning for Automated Vehicles

The developed algorithms and models for automated vehicles are mostly divided into two driving behavior levels: tactical decisions and operational decisions. This section, therefore, reviews the modeling approaches used for automated vehicles both in simulation and in real life and focuses on the methodology used in each model.

Tactical decision-making for automated vehicles has been given increasing attention. A considerable number of models contained rule-based decision-making processes [NIEHAUS & STENGEL, 1991], whereas more advanced models tend to use utility functions [EHMANNS, 2003]. In a study by Schubert et al. [2010], the Bayesian network was used for situation assessment and decision-making for lane changes. Deceleration to Safety Time (DST) was used as a central criterion for lane change situation assessment. In another study [SCHUBERT, 2012], the authors picked some sample sequences of highway driving and illustrated the consequences of uncertainties on the ambiguity of the situation over time. Reichel et al. [2012] proposed a framework for the classification of lane change decision-making for merging situations, which is used to analyze if the ego vehicle is part of a convoy merging maneuver or if a convoy is about to merge to the ego lane. McAllister et al. [2017] suggested a model that estimated and propagated uncertainty from every component throughout the entire system pipeline using a principled Bayesian for automated vehicles to cope appropriately with high uncertainty.

Ulbrich & Maurer [2015] proposed a Dynamic Bayesian Network (DBN)-based model, which relates variables to each other over adjacent time steps, for the tactical behavior planning for lane changes. In their model, they included the perception uncertainties and integrated system's abilities and current skills. A decision-making concept for the lane changing decision of the highly automated BMWs in real traffic was presented in a study by Ardelt et al. [ARDELT ET AL., 2012]. This method deployed a decision tree to assess the feasibility of the maneuvers based on the defined goals and traffic situation. Another study [MOTAMEDIDEHKORDI ET AL., 2017] modeled the tactical lane change behavior with various supervised ML algorithms and classified the behavior into keeping the lane, lane change to the right and lane change to the left with SVM, decision tree, Random Forest (RF) classifiers. In a similar study, Vallon et al. [2017] trained a support vector machine for lane change decision-making with features composed of relative position and relative velocity. Mukadam et al. [2017] proposed a framework that leverages the strengths of traditional optimization or rule-based methods for low-level control and DRL for high-level tactical decision-making, by striking the right balance between both domains. In this study, the authors considered the problem of autonomous lane changing in a multi-lane-multi-agent setting and set up the problem environment in SUMO traffic simulation.

When the tactical decision is made, which could be cruise-in-lane, change-lane to the right or change lane to the left, the selected tactical decision has to be translated into a trajectory that can be tracked by the feedback controller. The resulting trajectory must be dynamically achievable for the vehicle, comfortable for the passenger, and avoid collisions with obstacles detected by the onboard sensors [PADEN ET AL., 2016]. The task of finding such a trajectory is the responsibility of the motion planning system.

The task of motion planning for an automated vehicle corresponds to solving the standard motion planning problem as discussed in the robotics literature, and for motion planning and control of automated vehicles (operational behavior), various methods have already been used by researchers. As shown in Fig 2.7, these methods can be divided into four main categories: 1) Optimal control, 2) RL methods, 3) Learning from demonstrations, and 4) Motion prediction. This chapter briefly explains the most important studies of each group which has already been implemented for autonomous vehicles.

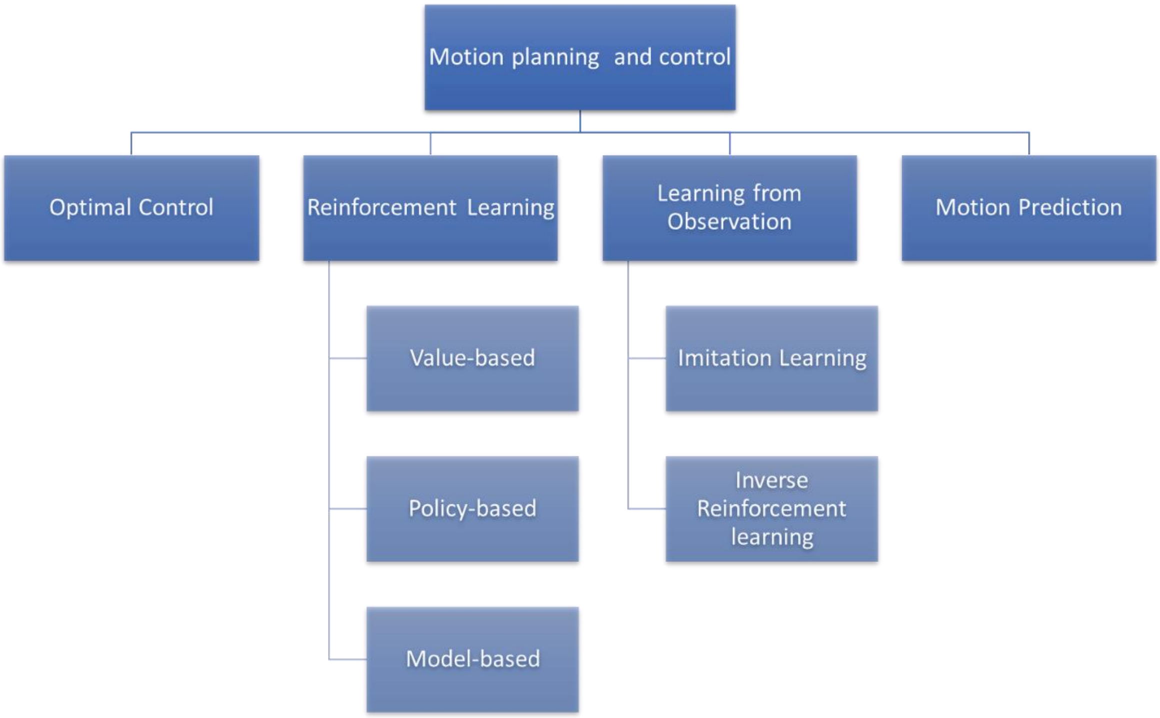


Fig 2.7 Summary of motion planning and control [https://github.com/eleurent/phd-bibliography]

2.4.1 Optimal Control

For operational behavior of automated vehicles, some studies focused on the Model Predictive Control (MPC) [CAMACHO & BORDONS, 2007] approaches. The main advantage of MPC is to anticipate the future consequences of control input. MPC allows dealing with the fulfillment of different objectives in complex missions using cost functions. MPC has been widely used to guide automated vehicles [ANDERSON ET AL., 2010; FALCONE ET AL., 2007; ALI ET AL., 2013; GRAY ET AL., 2012; GAO ET AL., 2012]. Turri et al. [2013] deployed linear MPC formulation to address the lane keeping and obstacle avoidance problems for a passenger car driving on low curvature roads. In another study, Rochefort et al. [2012] developed guidance of a group of autonomous cooperating vehicles using model predictive control. Their developed control strategy allowed finding a feasible near-optimal control sequence with a short and constant computation delay in all situations. A Non-linear model predictive controller (NMPC) [LINIGER ET AL., 2015] was developed for tracking the race cars based on mathematical optimization. Qian et al. [2016] proposed a new formulation of the trajectory planning problem as a Mixed-Integer Quadratic Program (MIQP). In their study, they illustrated that this formulation could be solved efficiently using widely available solvers, and the resulting trajectory is guaranteed to be globally optimal. However, the main drawback of MPC is the unpredictable computation delay of the optimization procedure. In order to create a more personalized autonomous lane change experience that satisfies safety and comfort constraints, Vallon et al. [2017] integrated lane change decision logic into MPC framework.

2.4.2 Reinforcement Learning

Another line of research learns driving behavior in simulation, which makes it suitable for RL. In the simulation, it is possible to observe failure cases during learning in a safe environment. RL was used for learning behavior models [KONIDARIS & HAYES, 2005; INFANTES ET AL., 2011; HE ET AL., 2016; SHTEINGART & LOEWENSTEIN, 2014]. In RL, the agent is provided with a reward function, and whenever the agent executes an action in some state, the reward function provides feedback about the agent's performance. The given reward function is used to obtain an optimal policy, one where the expected future reward is maximum. The problem with the RL is that in most tasks, there is no natural source for the reward function. Instead, it has to be carefully defined to represent the task accurately. The difficulty with designing a reward function that encourages the behaviors you want while still being learnable. Often, engineers manually tweak the rewards of the RL agent until the desired behavior is observed. A better way of finding a well-fitting reward function for some objective might be to observe an expert agent (human) performing the task in order to automatically extract the respective rewards from these observations with IRL [KAELBLING ET AL., 1996].

Wang & Chan [2017] developed a framework shown in Fig 2.8 for DRL to learn the merging behavior of autonomous vehicles on an on-ramp. For the interactive environment, they applied the Long Short-Term Memory (LSTM) architecture. From this architecture, an internal state containing historical driving information is conveyed to a Deep Q-Network (DQN). The DQN uses neural networks parameterized by θ to approximate the Q-function. Q-function takes the internal state as input and generates Q-values, denoted as $(s, a; \theta)$, as output for action selection. This DLR architecture allows capturing the historical impact of the interactive environment on the long-term reward and taking this reward into account for deciding the optimal control policy. This study has a certain limitation of considering the interaction among only three vehicles and not being verified and validated.

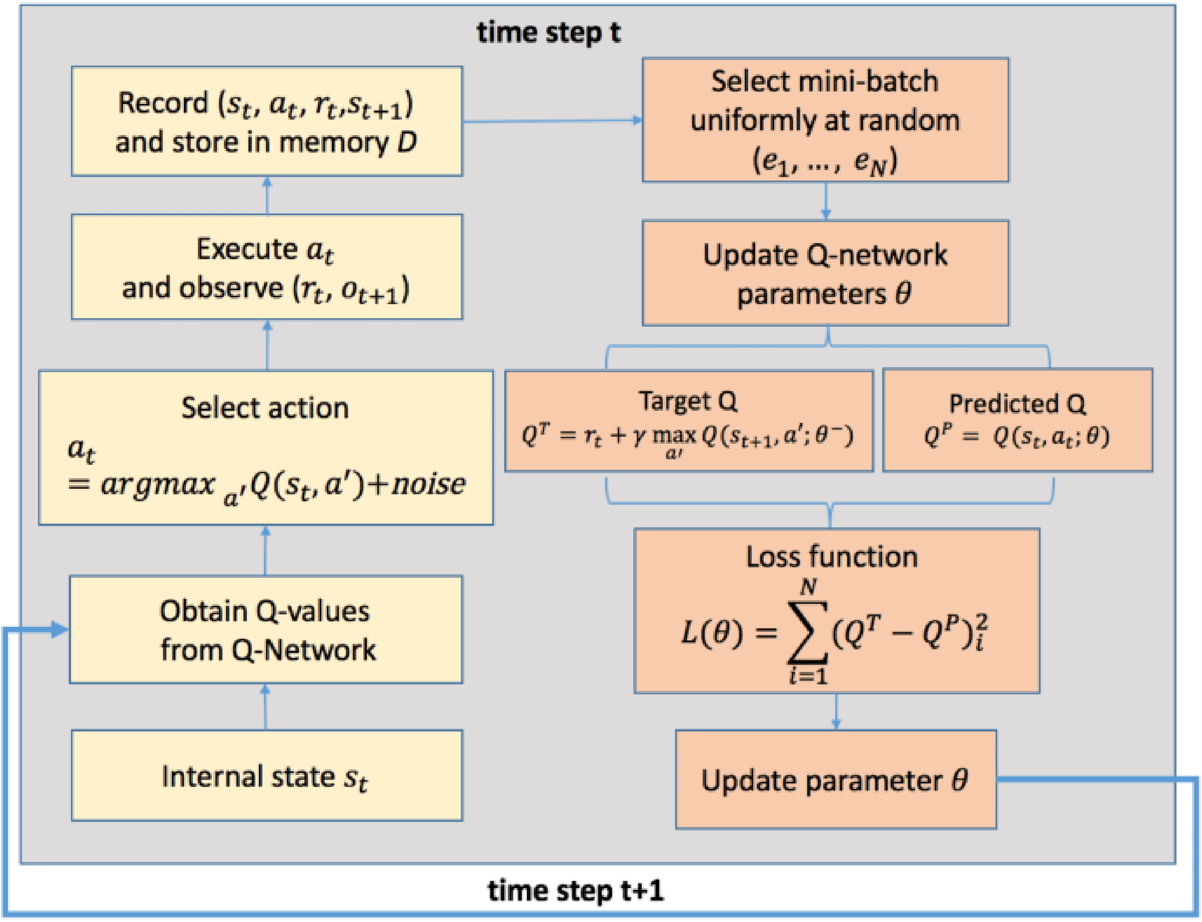


Fig 2.8 Deep Q-learning architecture in the methodology developed by Wang & Chan [2017].

In similar studies [2018; 2019], the authors proposed a RL-based approach to train the vehicle agent to learn an automated lane change behavior such that it can make a lane change under diverse and even unforeseen scenarios. They treated both state space and action space as continuous and designed a Q-function approximator that has a closed-form greedy policy, which contributes to the computation efficiency of the deep Q-learning algorithm. The reward function in this model was defined with yaw rate, yaw acceleration and lane changing duration for training a smooth and efficient lane change behavior and tested their model in a simulation platform with diverse simulation scenarios.

Berkenkamp et al. [2017] presented a learning algorithm that explicitly considers safety to overcome the fact that in RL, finding optimal policies is mostly associated with exploring all possible actions, which may be harmful to real-world systems. On the other hand, model-based RL has been used in Piergiovanni et al. [2018] to control robots. In their model, they followed the standard RL setting and formulated policy learning as learning in a Markov decision process (MDP).

2.4.3 Learning from Demonstrations

Learning from demonstrations is usually done by two methods: IL or IRL. IL is the problem of learning to perform a task from expert demonstrations, in which the learner is solely provided the sample of trajectories from the expert. IL is the process of directly learning the mapping between the states and the actions, considering this task as a supervised learning problem, whereas IRL assumes that the expert's policy is optimal with respect to an unknown reward function. Therefore, the first aim of the apprentice is to learn a reward function that describes the observed expert behavior. Then, using direct RL, it optimizes its policy based on this reward function and hopefully behaves as well as the expert [PIOT ET AL., 2017]. The first advantage of learning a reward over learning a policy is that the reward can be analyzed to better understand the expert's behavior [BLOCKEEL ET AL., 2013]. The second advantage of learning a reward is that it allows adapting to perturbations in the dynamics of the environment and its possibility to transfer to other environments [MUNZER ET AL., 2015].

The Generative Adversarial Imitation Learning (GAIL) model developed by Ho & Ermon [2016] developed a framework to extract a policy from data directly. They showed that their generative method from which they derive a model-free IL algorithm, performs better than the conventional IL models in large, high-dimensional environments. Bhattacharyya et al. [2018] developed an extended GAIL in order to address its shortcomings and considered multi-agent GAIL policy. They showed that their model could outperform the model with the single GAIL policy.

In the IRL framework, the task is to take a set of human-generated or simulation-generated driving behavior data and extract an approximation of the reward function for the task by using human driving data to automatically learn the feature weights for the reward [ABBEEL & NG, 2004]. Abeel & NG [2004] were the first researchers who motivated deploying the IRL for learning driving styles. Thereafter, researchers used different variations of IRL to model the driving behavior for automated vehicles. Babes et al. [2011] introduced the maximum likelihood IRL in which they maximized the likelihood of the data assuming an exponential family distribution. They introduced a method to cluster observed behavior styles and learn individual feature weights for each cluster. In another study [KUDERER ET AL., 2012], proposed learning from demonstration method and learned a cost function with a feature-based IRL that first the observed driving behavior style the best.

Some Research focused on developing driving behavior prediction based on Maximum Entropy IRL (MaxEnt IRL) [LEVINE & KOLTUN, 2012; LEVINE ET AL., 2011] in order to tackle the ambiguity of the IRL. Another study by Vasquez et al. [2014] proposed leveraging an existing driving simulator (Torcs) by developing a ROS communication bridge. They used it is as the basis for an experimental framework that can be used for the development of human-like autonomous driving based on IRL. Sharifzadeh et al. [2016] proposed IRL approach using Deep Q-Networks to extract the rewards in problems with large state space and tested their method in a simulation-based autonomous driving scenario. Being inspired by the study of Levine and Koltun [2012], which presented a method for IRL in continuous state and action spaces, Shimosaka et al. [2017] developed a prediction model using a graph-based state space, in which they propose a novel state-space discretization framework.

2.4.4 Motion Prediction Models

Reghelin et al. [2012] developed a centralized traffic controller for highway traffic, minimizing the sum of travel times of all vehicles. In their model, they assume aggressive actions and performing conservative behavior if expected collisions in the planning framework occur.

In a study by Lawitzky et al. [2013], a framework for prediction of the motion of vehicles on highways considering the interaction of the traffic users was presented, which fulfills the required time constraint for on-line computation. This framework assumes that drivers do not prefer the safe option over the maneuvers with high collision probability. Their model predicted the highway scenes, taking inter-vehicle dependencies and interactions explicitly into account while evaluating collision probabilities of different maneuvers.

Schwarting and Pascheka [2014] considered the decision-making problem of a host vehicle during lane changing with a number of object vehicles moving on neighboring lanes in the same direction in a highway traffic scenario. In this situation, they searched for cooperative maneuvers over a control horizon that considers preceding as well as following vehicles. They proposed a motion primitive-based recursive approach which combines egoistic behavior prediction, conflict detection, and conflict resolution with a cost function-based maneuver evaluation to mimic human-like cooperative decision-making.

Kim et al. [2017] proposed trajectory prediction method based on the recurrent neural network called long short-term memory (LSTM) to analyze the temporal behavior and predict the future coordinates of the surrounding vehicles. The proposed scheme feeds the sequence of vehicles' coordinates obtained from sensor measurements to the LSTM and produces the probabilistic information on the future location of the vehicles over an occupancy grid map. Their proposed method was similar to that used by Khosroshahi et al. [2016] for analyzing the driving behavior of the surrounding environment with LSTM.

2.4.5 Summary

The developed models for motion planning and control algorithms of automated vehicles are mostly based on two common approaches: Model-driven and Data-driven. Model-driven modeling represents the attempt to capture the understanding of how a system works through explicit representations and rules. Data-driven approaches focus on building a model that can identify the right prediction based on having seen or trained on a large number of examples. The strength of this approach is that it does not depend on humans accurately describing through a set of rules how the model should work. The higher the quality and variety of the training data, the better the model can make predictions. One of the significant concerns about data-driven models is the uncertainty caused by bias and under-representativeness of the training data. Another big challenge of these models is the generalization of the model to unseen events. Model-driven models, on the other hand, attempt to capture the knowledge and derive decisions through explicit representation and rules. Furthermore, modeling takes time and it is inherently a trial-and-error approach, rooted in the method of theory-based hypothesis formation and experiment-based testing. Therefore, finding a suitable model and refining it until it produces the desired results is often a lengthy process. Moreover, there are many different rules and exceptions to those rules and components that cannot be easily defined by humans.

2.5 Research Gap

This chapter reviewed the most well-known models for driver behaviors, the influencing factors for driving behavior, microscopic lane change modeling, widely-used microscopic traffic simulation tools, as well as the methods used for motion planning and control of automated vehicles. From the foregoing literature review, the major limitations of the existing microscopic lane change models to study the new forms of mobility, such as automated vehicles, become apparent. The literature review has highlighted an area where further research could overcome gaps and shortcomings.

Conventional Rule-based or physical lane changing models, model the lane changing behavior with varying gap-acceptance conditions and lane changing behavior for various situations, however, these models cannot represent the heterogeneous driving behavior of humans' drivers during lane changing realistically. Although the conventional models have some advantages, such as intuitive explanations and rational boundaries, many components cannot be adequately described by a mathematical equation of a simple agent [CASCETTA, 2001]. Most of these models are deterministic and even those who tried to add the stochasticity to the model through estimated distributions fail to represent the behavior of humans realistically. Because of the multiplicity of possible driving conditions associated with discretionary and mandatory lane changes, this approach tends to lead to complex models with many parameters [TREIBER & KESTING, 2009].

Incentive-based models consider the variability of drivers into account by the politeness factor in MOBIL and accepted headway, deceleration and level of desire in LMRS. Moreover, they have very few parameters to tweak and therefore, they fail to present a realistic model for human's lane changing behavior. In discrete choice models, the lane changing decisions are probabilistic instead of binary yes or no, however, the existing discrete-choice-based lane changing models, the heterogeneities in drivers such as driver aggressiveness and driving skill level are not considered adequately.

AI-based models, have the potential to be superior in dealing with complex situations without resorting to detailed hard-coded rules or predetermined models and some studies presented plausible performance [HOEL ET AL., 2018; GONZALEZ ET AL., 2016; GINDELE ET AL., 2015; WANG, 2015]; however, these models are usually trained with the traffic simulation or driving simulator data. Therefore, it is still not clear whether they can perform the same on the real data. Another shortcoming associated with these models is their computational time, which makes them impractical and not easily implementable in the microscopic traffic simulation tools. However, these factors and small variations in driving behavior can lead to aggregated effects on the traffic flow.

Some of the existing physical-based lane change models are implemented in microscopic traffic simulation tools, which are used for traffic flow studies. The driving behavior models in microscopic simulation tools should be as precise and realistic as possible to replicate the reality and draw meaningful conclusions about the impacts of automated vehicles on traffic flow characteristics. However, as discussed in Tab. 2.3, the most widely used microscopic traffic simulation tools do not account for the influential factors on driving behavior. Moreover, in the underlying rule-based models in microscopic simulation tools, there is a need to develop a new mathematical model for different driving scenarios, for example, a new model for freeway and urban areas. Besides, the calibration and validation effort and time of the conventional models is very high and adapting the model to different situations and different driving cultures mandates a new iterative calibration and validation process for tweaking the parameters. These shortcomings resulted in seeking the driving behavior modeling approaches that can better reflect the factors such as socially acceptable driving behavior, can be easily adapted to new situations, and replicate the real-world driving conditions more realistically.

As part of this effort, many models proposed for the decision-making algorithms and motion planning of automated vehicles were investigated. Optimal control models and motion-prediction models define models having various parameters. Despite the promising results of these algorithms, exact solutions to the motion planning problem are in most cases computationally intractable [PADEN ET AL., 2016]. A high-fidelity model which accurately reflects the response of the vehicle to various factors may complicate the planning and control problems due to the added details. This presents a trade-off between the accuracy of the selected model and the difficulty of the decision problems [PADEN ET AL., 2016]. Therefore, integrating the same models in microscopic traffic simulation models is not feasible.

Another category of developed models for control of automated cars is based on learning from demonstrations is usually done by two methods: IL or Inverse IRL. IL is the process of directly learning the mapping between the states and the actions (the policy), whereas IRL relies on the assumption that the expert's policy is optimal with respect to an unknown reward function. By means of IRL, the detailed information buried in the data, such as the correlation between different features as well as the significance (weight) of each feature can be captured. The result of the second modeling approach showed that it is capable of learning reward functions and reproducing different driving styles and socially acceptable driving behavior using data from real drivers [VINKHUYZEN & CEFFKIN, 2016].

A general modeling approach should be developed that, like the IRL-based model developed for control of automated vehicles, reflect the underlying nature of driving behavior in various traffic situations and in different driving cultures without the need to perform the iterative and time-consuming approach of calibration and validation. This approach should reflect the detailed information buried in the data such as the impact of the factors discussed in Tab. 2.2 on diving behavior, while being computationally reasonable and implementable in microscopic traffic simulation software. However, due to lack of data about the demographics of drivers, the focus of this dissertation is on the integration of the perceived environment (social and cultural norms) as well as the driving environment in the driving behavior model, specifically on lane changing behavior.

Among the reviewed modeling approaches, data-driven approach was found to be the most appropriate for achieving the goal of this study. This approach has the strength of not depending on human to accurately describe a set of rules on how the model should work. There are so many different rules and exceptions to those rules and components that cannot be defined by humans and capture the knowledge and derive decisions through explicit representation and rules from data. However, these models have specific data requirement with regard to the amount and variety and representativeness of training data. The proposed data-driven methodology for lane change will significantly increase the accuracy of the models used for impact assessment of automated vehicles with microscopic simulation tools.

3. Traffic Observations

Having access to detailed traffic data is a vital component in driving behavior modeling [LIMA AZEVEDO, 2014]. Calibration and validation of any lane changing model parameters require detailed trajectory data, including vehicle positions at discrete points in time, as well as other variables such as the number of lane changes, speeds, accelerations, headways, and intra-vehicle gaps in traffic [RAHMAN ET AL., 2013], many studies attempted to calibrate their models with empirical data [YANG & KOUTSOPOULOS, 1996; LECLERCQ ET AL., 2007; THIEMANN ET AL., 2008; MORIDPOUR ET AL., 2010a; SCHAKEL ET AL., 2012; KNOOP & BUISSON, 2014; PARK ET AL., 2015; JIN ET AL., 2018]. In this Chapter, the work carried out for the collection of vehicle trajectories for the specific case study is presented. First, state-of-the-art traffic data collection types are briefly reviewed. The selection of the observation area and video data collection system are presented in chapter 3.2 and 3.3. The method used for vehicle trajectory extraction is presented in chapter 3.4. Finally, the results of this collection are discussed in the last section.

3.1 Data Collection Types

In the last couple of decades, behavior modeling research has devoted efforts to collect and analyze detailed traffic data [LIMA AZEVEDO, 2014]. This data can be generally divided into two categories considering the collection procedure used:

a) Vehicle-based methods: The data provided the probe vehicles equipped with sensors and cameras which records the data while traveling in the traffic stream [National Highway Traffic Safety Administration (NHTSA), 2004; US Department of Transportation; National Highway Traffic Safety Administration (NHTSA), 2013; BRACKSTONE ET AL., 1999; SAYER ET AL., 2007; KONDYLI & ELEFTERIADOU, 2010; VAN SCHAGEN ET AL., 2011; SWOV Institute for Road Safety Research, 2013]. This type of data, also called naturalistic driving behavior data, provides an overview of the driving behavior of the driver and the adjacent vehicle, as long as they are in the field of view of the sensors. This data type provides data from the limited number of probe vehicles but in many locations and scenarios.

b) Site-Based methods: These methods provide the data from the sensors installed in a specific area. This type of data collection can provide a vast amount of data for a limited number of locations. This method is mainly done with the video cameras [BÉLISLE ET AL., 2017; KATHS, 2017; ST-AUBIN ET AL., 2015; KNOOP ET AL., 2012a; KOVVALI ET AL., 2007; LAURESHYN, 2010] although technologies such as RADAR [AOUDE ET AL., 2011; KOVVALI ET AL., 2007] and infrared [BHATTACHARYA ET AL., 2011] has also been used by the researchers. This type of data collection can be divided into two groups: Stationary and dynamic. In static observations, the sensor is installed on a static object such as buildings, poles or infrastructure whereas, in dynamic observations, airborne technologies such as drones, helicopters and airplanes are used [DAAMEN ET AL., 2010].

c) Mixed methods: These methods fuse data from various sources. Chan and Bougler [2005] deployed the data from different sensors to assess the conflicts at the intersection. Christoph et al. [2010] used the combination of both in-vehicle and site-based observations for naturalistic driving analysis.

In order to reach the aim of this dissertation, which is applying data-driver methods to learn the lane changing behavior, a vast amount of data is required. For the purpose of this dissertation, site-based stationary data observation method is selected for data collection. It is noteworthy to mention that despite delivering the required data, this kind of database has the weakness of considering only one single observation site with limited variability of the infrastructure environment.

3.2 Traffic Scenario Selection

In the traffic context, a scenario is a description that contains the actors, which are in this case road users, some background information about the history of movement of the actors and the assumptions about the environments and goal of each actor and this description can be defined as a list of what-is questions followed by the answered. Based on the requirements mentioned in section 3.2, it is necessary to collect enough amount of data from a traffic scenario in which a lot of lane changes happens. The traffic scenario was selected with the goal of having as many lane changes as possible with the stationary observation. The location of the data observation is the on-ramp from A9 freeway to the Mittlerer Ring, which is high-capacity ring road (urban highway) around the city center of Munich. The observations were made from the IBM Highlight tower 1, the location of which is shown in Fig 3.1.

Fig 3.1 Location of the research observation [google maps,2018]

3.3 Video Data Collection

Video data were collected using a GoPro Hero5 Black Edition in full HD resolution with multiple cameras with different views. For this study, a super-wide view of the camera was used to maximize the camera's Field Of View (FOV). The screen resolution was 3840x2160 and the videos have a resolution of 24 frames per second. The camera was mounted on the 23rd floor of the highlight tower 1, as shown in Fig 3.2. In order to record the traffic, cameras were installed in three different positions so that their views overlap.

Fig 3.2 Location of the IBM Highlight tower and the observations' view (in yellow). The figures at the bottom show the camera view from three positions.

Fig 3.3 illustrates the view of the camera from which the trajectories were extracted. As it can be seen in the figure, the camera was installed in the building, and the reflection of the scene in the glass as well as the sun can lead to problems when extracting the trajectories.

Fig 3.3 Camera view from the observation point

Videos were recorded in three days and at different times of the day to cover the various time of the year, traffic states, and different types of commuting traffic. Tab. 3.1 presents the summary of observations:

Tab. 3.1 Overview of traffic observations

Observation Number	Date	Time	Camera position
1	11.07.2017	08:50 - 16:00	1
2	09.08.2017	09:30 - 10:50	1
3	09.08.2017	10:58 - 12:18	1
4	09.08.2017	12:30 - 12:56	1
5	15.01.2018	11:30 - 16:50	1
6	15.01.2018	07:45 - 11:30	1
7	15.01.2018	11:30 - 16:50	2
8	15.01.2018	15:11 - 16:52	3

3.4 Trajectory Extraction

Individual vehicle trajectories were extracted from the videos by using the commercial software DataFromSky developed by RCE systems [RCE systems, 2018]. For the extraction of the trajectories from the video recordings, several inputs should be provided for the software. The first input is a reference image from the traffic scene captured by the same camera that recorded the videos. The videos are then registered to this image and georeferenced. The next parameter defined for the camera extraction is the intrinsic parameters of the camera used for the observations. These parameters in the intrinsic matrix are illustrated below:

$$k = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

These five parameters represent the focal length, image sensor format and principal point. The parameters $\alpha_x = f.m_x$ and $\alpha_y = f.m_y$ represent the focal length in terms of pixels, where m_x and m_y are the scale factors which relates pixels to distance and f is the focal length in term of distance [HARTLEY & ZISSERMAN, 2015]. γ represents the skew coefficient between x and y axis. The parameters u_0 and v_0 represent the principal point, which would be in the center of the image ideally. The video frames are rectified using the following parameters for the go pro Hero5 camera:

$$K = \begin{bmatrix} 862.5202 & 0 & 959.2693 \\ 0 & 886.36 & 500.8769 \\ 0 & 0 & 1 \end{bmatrix}$$

For the distortion of the camera, 8 parameters are determined. k_1, k_2, k_3, k_4, k_5 and k_6 are radial distortion coefficients. $P_1$and P_2 are tangential distortion coefficients and is shown as:

$$\begin{bmatrix} k_1 \\ k_2 \\ p_1 \\ p_2 \\ k_3 \\ k_4 \\ k_5 \\ k_6 \end{bmatrix} = \begin{bmatrix} 0.84511 \\ -0.14147 \\ 1.05431 * 10^{-4} \\ 6.902 * 10^{-4} \\ -6.89179 * 10^{-3} \\ 1.2127 \\ -3.8962 * 10^{-2} \\ -3.7148 * 10^{-2} \end{bmatrix}$$

The registration engine attempts to align the two sets of tie points to identical geographic locations using the adjustable parameters. In order to reference the map, at least four corresponding points, also called control points, with the correspondences between the provided reference image and real-life coordinates, should be provided to the software. Moreover, the traffic scene annotation file should be provided for the extraction of trajectories. Annotations are the areas within which vehicles are expected, can be designated as entry and exit.

The trajectory extraction provides the georeferenced trajectories of vehicles. Meaning that the trajectories have longitude and latitude in UTM coordinate system, the speed, longitudinal and lateral acceleration and the class of vehicles for each time step. In this stage, the vehicles are detected and localized by application of coupled weak and strong vehicle classifiers, across the whole area of the video frame to produce a set of detections [ADAMEC ET AL., 2017]. For the classification of the vehicles, the dimension settings shown in Tab. 3.1 are deployed:

Tab. 3.2 Dimensions of each vehicle category for trajectory extraction

Class	Vehicle Length [m]	Vehicle Width [m]	Vehicle Height [m]	Vehicle Pivot x (in Width) [m]	Vehicle Pivot y (in Length) [m]
Car	5.0	2.00	1.40	1.00	2.50
Medium vehicle	5.83	2.67	1.70	1.33	1.67
Bus	12.50	4.00	2.80	2.00	0.83
Heavy Vehicle	12.50	3.33	2.90	1.67	0.83

From the mentioned datasets in the table, the trajectories from observations number 1, 2 and 8 could not be extracted due to high sun reflection. Moreover, the focus lies only on a specific area of the view which is illustrated in Fig 3.4.

Fig 3.4 Research focus area

In total, the trajectories were extracted from twelve hours and 25 minutes of videos and saved in a MySQL database to facilitate the retrieving and accessing the data. The total length of trajectories is 9341.811 kilometers of trajectories and contains the trajectories from 46,918 road users. The extracted trajectories cannot capture all the driving behavior-decisive factors, which were mentioned in Tab. 2.2, however, it provides information about the heterogeneous diving behavior of human drivers under different traffic conditions and driving situations. In order to visualize the variations in the speed, longitudinal and lateral accelerations of vehicles, the histogram and density plots of each vehicle class were plotted and analyzed separately. As shown in Fig 3.5, a density plot is a smoothed, continuous version of a histogram estimated from the data. For these plots, Gaussian kernel density estimation is used. In this method, a continuous curve is drawn at every individual data point and then all of these curves are added to make a single smooth density estimation. The vertical axis in a density plot is the probability density function for the kernel density estimation, which is different than probability.

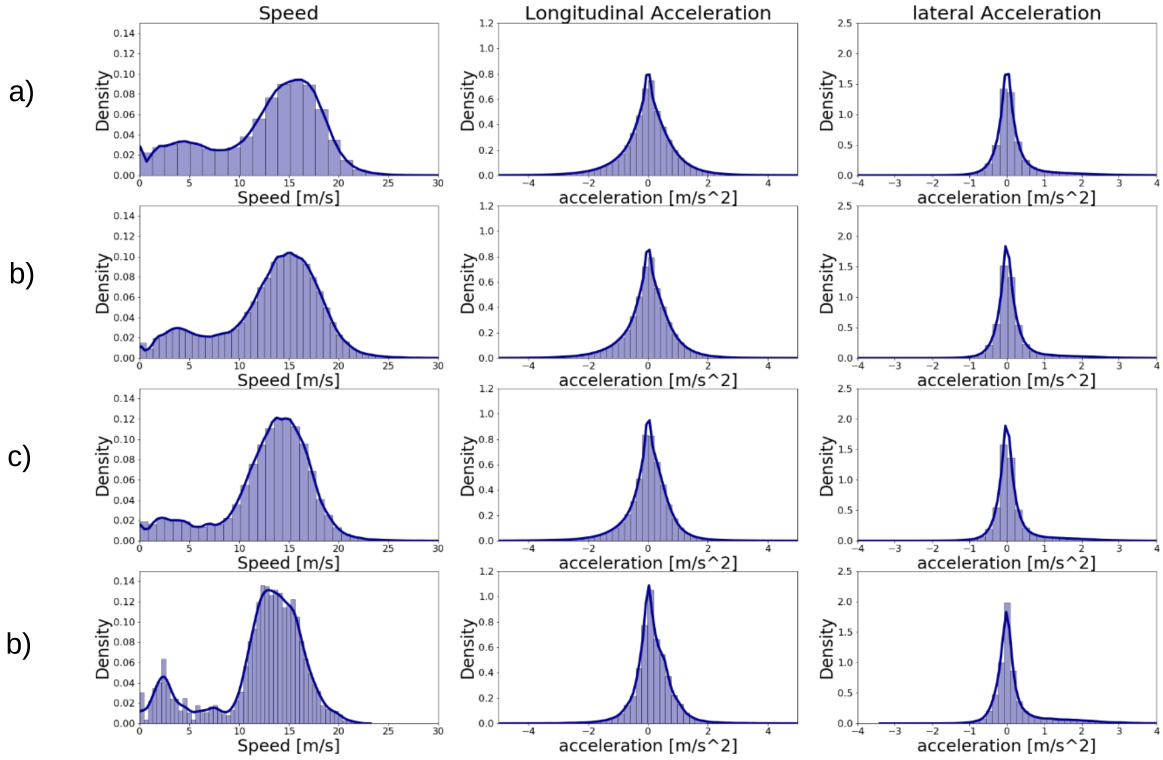


Fig 3.5 Histogram and density plots of speed, longitudinal acceleration and lateral acceleration for a) cars, b) medium vehicles, c) heavy vehicles and d) buses in the dataset.

As illustrated in Fig 3.6, the extracted trajectories show the precise location of the vehicles and the precise alignment of the vehicles within the lane. In this figure, the blue trajectories are the future paths that the vehicles will take, and the green lines illustrate the part of the trajectory that has already been driven by the vehicles.

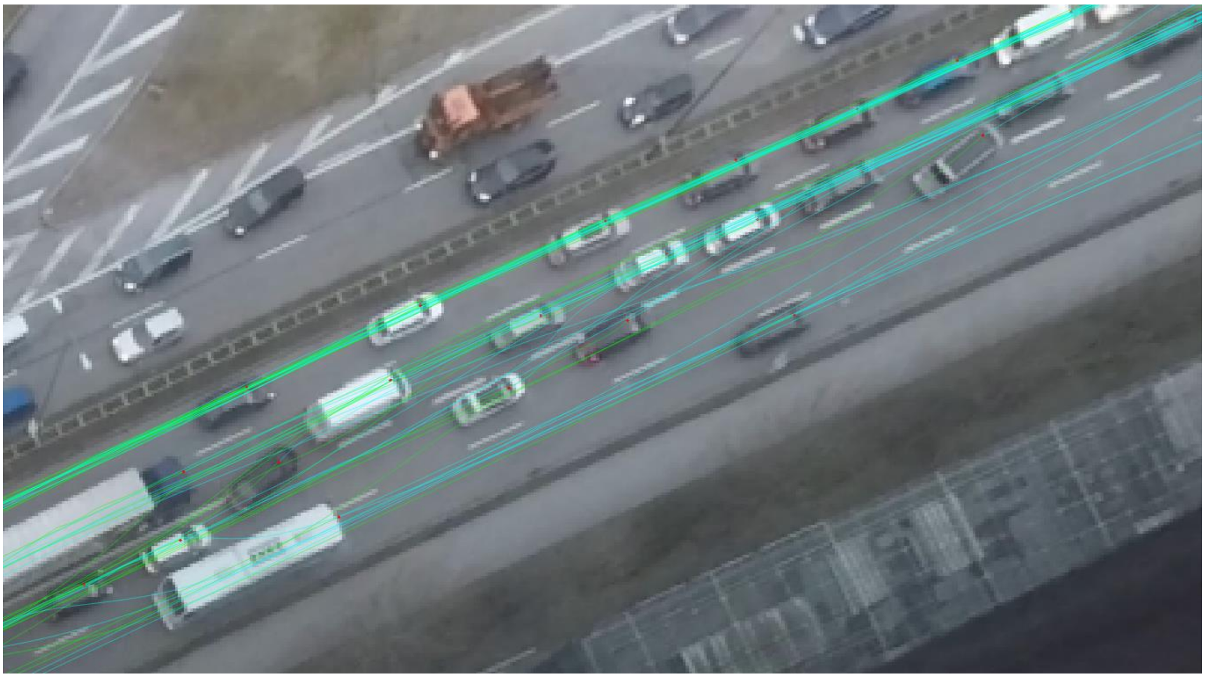


Fig 3.6 Resolution of vehicle trajectories

3.5 Summary

In this chapter, a description of the method used for the extraction of vehicle trajectories is presented. In order to collect detailed traffic variables required for microscopic lane change modeling, over twelve hours of video observations were recorded from real traffic on three days. The videos were post-processed with DataFromSky software and a large set of motion parameters, which allowed for the characterization of driving behavior were successfully extracted from the observations. As a result, the data collected consists of 46,918 vectors of 10 values of type vehicle ID, vehicle type, traveled distance, average speed, x and y coordinates, speed, tangential acceleration, longitudinal acceleration, and time step.

4. Clustering Driving Behavior

Identifying aggressive drivers is crucial in developing safe automated driving algorithms and advanced driving assistant systems [CHEUNG ET AL., 2018]. The term driving style usually refers to the typical behavioral patterns of a driver and defensiveness-aggressiveness is a metric that is commonly used for differentiating the driving styles [BASU ET AL., 2017]. Other studies categorized driving style to aggressive/assertive versus defensive [KARJANTO & YUSOF, 2015]; or mild versus moderate versus aggressive [XU ET AL., 2015]. Another study identified four broad driving styles: (1) reckless and careless driving; (2) anxious driving; (3) angry and hostile driving; and (4) patient and careful driving [TAUBMAN-BEN-ARI ET AL., 2004]. The driving style was categorized as angry driving, anxious driving, dissociative driving, distress-reduction driving and careful driving style in a similar study [VAN HUYSDUYNEN ET AL., 2015]. Another study [HONG ET AL., 2014] differentiated the driving behavior styles in terms of defensiveness, as well as the propensity for violation of rules.

The goal of the driving behavior categorization in this thesis is to extract a set of features from trajectories that can be mapped properly to driving behaviors and categorize the driving behaviors into separate driving behavior clusters. Although the captured data includes the vehicle classes, it fails to provide detailed information about the characteristics, such as the brand of each vehicle, engine type, the source of drivers' distraction, as well as drivers' mental and emotional condition, that might have an impact on driving behavior. The result of different number of clusters are discussed in the next section. Based on the result, this dissertation focuses mainly on three driving behavior of timid, moderate and aggressive, which are mutual categories in most of the driving behavior categorizations and can be captured with the observed data.

Due to the increase in data size, human manual labeling has become extremely difficult and expensive, and the use of automatic methods has gained interest. Clustering is a well-developed field of data analytics and one of the most widely-used tasks in the data mining process for discovering categories and automatically identifying interesting distributions and patterns in the underlying data [HALKIDI ET AL., 2001]. Therefore, In this dissertation, a clustering was deployed for mapping and extracting the driving behavior style from the observed trajectories. The clustering of driving behavior shows how certain behaviors repeat throughout the data between drivers. The basic steps to develop a clustering process are presented in Fig 4.1.

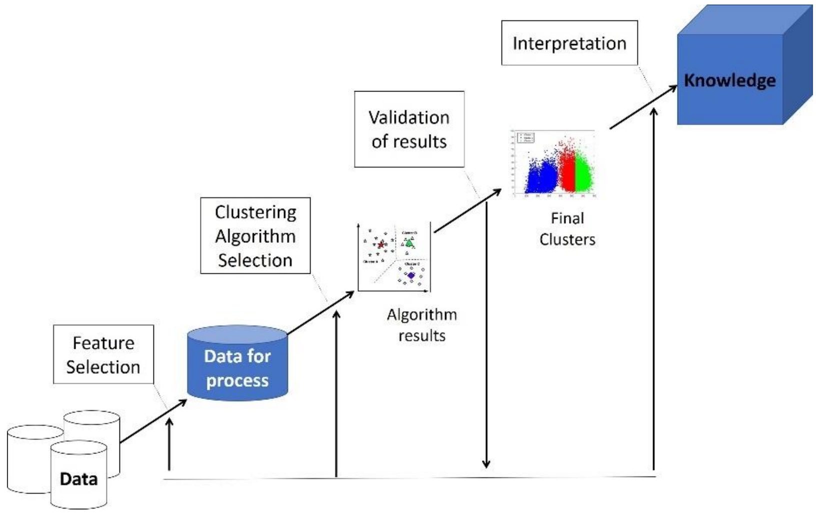


Fig 4.1 Steps of clustering [FAYYAD ET AL., 1966]

These basic steps can be summarized as follows [FAYYAD ET AL., 1966]:

- Feature selection: the goal is to properly define the features on which the clustering is to be performed to encode as much information as possible. The process in which from a set of features in the dataset, the ones that have the most significant influence on the outputs are automatically extracted is called feature selection [CHANDRASHEKAR & SAHIN, 2014].

- Clustering algorithm: This step involves the choice of an algorithm that results in a good clustering scheme considering the dataset. The clustering algorithm is mainly characterized by two criteria:

    1. *Proximity measure* quantifies the similarity between two data points. When selecting the proximity measure, it has to be ensured that all selected features have equal contributions to the computation of the proximity measure and none of the features dominate others.

    2. *Clustering criterion* can be expressed with a cost function with consideration of the type of the clusters which are expected to occur in the dataset.

- Validation of the results: The correctness of the results of clustering algorithm is examined using appropriate techniques and criteria [HALKIDI ET AL., 2001].

- Interpretation of the results: In the application area, the clustering results have to be integrated with other experimental evidence in order to draw the correct conclusion [HALKIDI ET AL., 2001].

4.1 Feature Selection

Feature selection also called pattern representation [JAIN & DUBES, 1988], is the process of selecting or extracting the features which describe the observation. Each feature defines a particular aspect of observation and can be continuous or categorical. Feature extraction is the augmentation of original features to provide information into the clustering algorithm that involves choosing directly measured variables. The selection and extraction of features that adequately describe the data and include important and relevant attributes that make it possible to separate the observations into clusters is a crucial step in the clustering process [KATHS, 2017]. Irrelevant features in the data set may degrade the quality of learning and consume more memory and computational time. From the clustering point of view, removing irrelevant features will not negatively affect clustering accuracy whilst reducing required storage and computational time [ALELYANI ET AL., 2013].
As mentioned in chapter 2.5, the focus of this dissertation lies on factors such as driver aggressiveness, social norms as well as driving environment such as traffic state. However, all these factors are not directly extractable from the available dataset. For this study, the features listed in Tab. 4.1 were extracted from the available features of the data set and used to create a profile for each individual driver.

Tab. 4.1 Features for clustering

Symbol	Notation	Description	Unit
f_0	$\overline{v}$	Average velocity	$\frac{m}{s}$
f_1	$\overline{\Delta v_{all}}$	Average Relative speed to the vehicles in the scene	$\frac{m}{s}$
f_2	$\overline{j}$	Average longitudinal jerk	$\frac{m}{s^2}$
f_3	$\overline{\Delta v_f}$	Average relative speed to the vehicle in front	$\frac{m}{s}$
f_4	$\overline{d_f}$	Average distance to the front vehicle	m
f_5	$\overline{\Delta v_b}$	Average relative speed to the vehicle in the back	$\frac{m}{s}$
f_6	$\overline{d_b}$	Average distance to the vehicle in the back	m

After the feature extraction, the frequency of values in the data set were plotted in histograms. The plots in Fig 4.2 represent the histogram of different features for all the datasets.

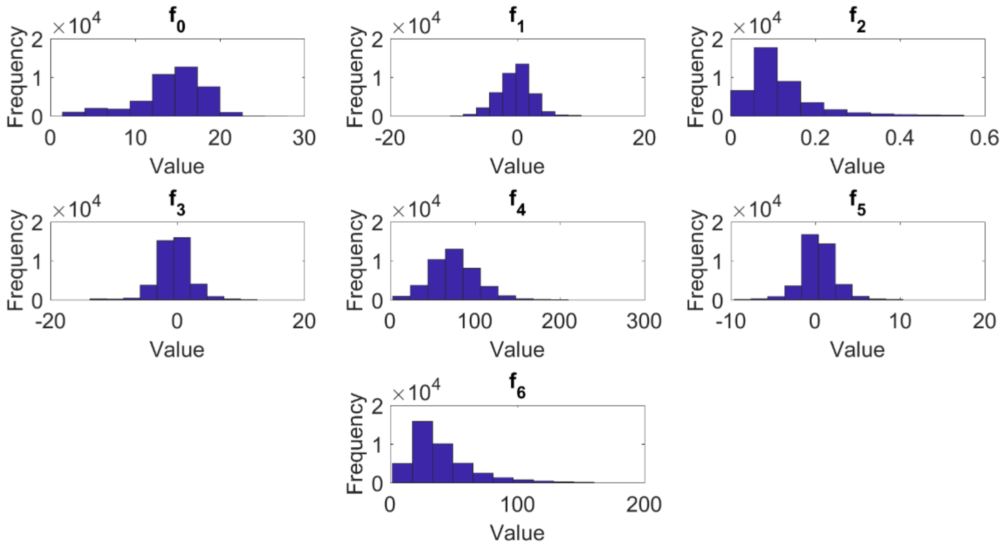


Fig 4.2 Histogram of features

As it can be seen from the plots shown in Fig 4.3, the features have different units, and this may lead to a very different clustering structure, giving more importance to the features with the largest scales during dissimilarity calculation and bias in the clustering result. In order to avoid the dependence clustering structure on the choice of feature units, a common practice is to standardize the features. Standardization is the process of scaling individual samples to have unit norm. This is a crucial step as the k-mean clustering method, which works with Euclidean distances. Therefore, the features were standardized with the z-transformation, which subtracts the mean of the data from all values and divides them by the standard deviation based on the following formula:

$$Z_{score} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

Where μ_j and σ_j represent respectively the mean and variance of feature f_j. After the z-transformation, the distribution of the data has a mean of zero and a variance of one. This transformation preserves the original distribution of the data and is less influenced by outliers. The plots in Fig 4.3 represent the values after the standardization.

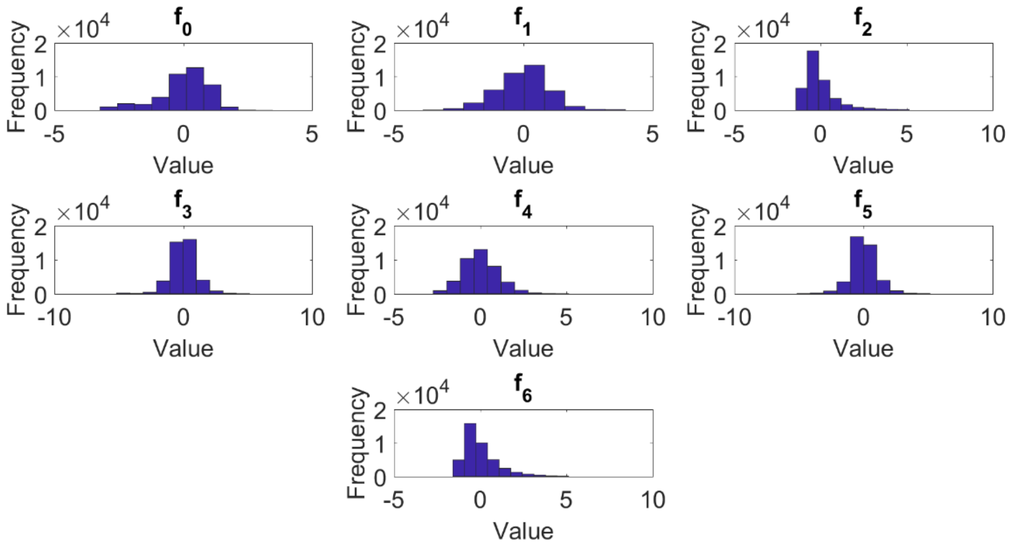


Fig 4.3 Histogram of standardized features

In clustering, it is essential to discover and quantify the degree to which variables (features) in the dataset are dependent upon each other due to the fact that multicollinearity (if two or more variables are tightly related) can deteriorate the performance of the algorithm. When the variables used in clustering are collinear, some variables are weighted higher than others. Basically, two variables that are perfectly correlated represent the same concept and the concept is represented in the data twice and hence gets twice the weight of all other variables. Therefore, the Pearson correlation coefficients [PEARSON, 1895] were calculated and significant correlation coefficients of the features were examined closely. The Pearson correlation coefficient is a nonparametric measure of linear correlation between two variables. For a sample of size n raw scores X_i ,Y_i , r_s is calculated from:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Where $\rho_{X,Y}$ denotes the Pearson correlation coefficient.

$cov(X,Y)$ is the covariance of variables.

σ_X and σ_Y are the standard deviations of the variables.

Fig 4.4 summarizes the correlations between the features in a correlogram. In this figure, histograms of the features appear along the matrix diagonal and the scatter plots of feature pairs appear in the off-diagonal. The slopes of the least-squares reference lines in the scatter plots are equal to the displayed correlation coefficients.

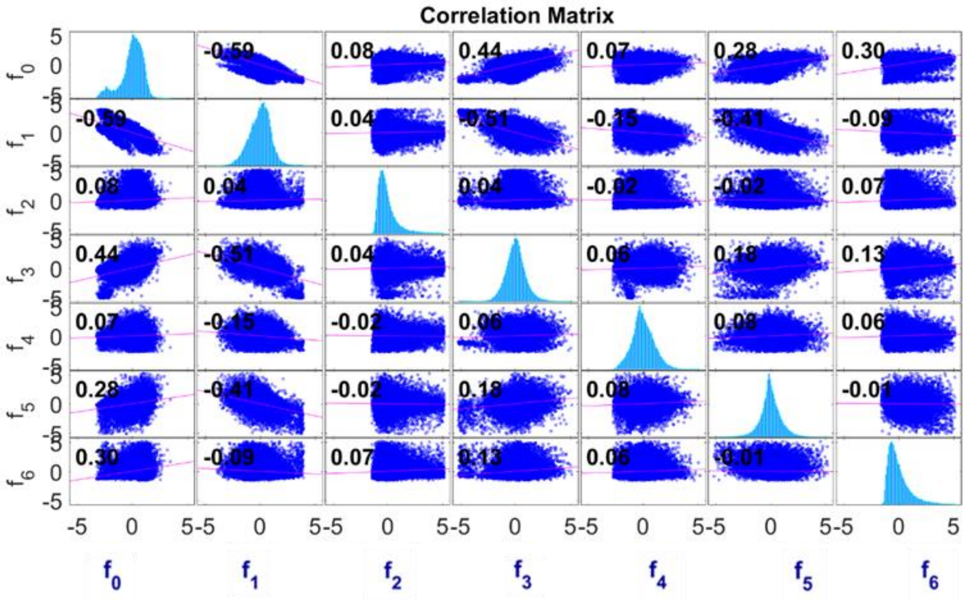


Fig 4.4 Correlogram between the features

The correlation matrix among pairs of features shown in Fig 4.5 confirms that some of the features are slightly correlated. To overcome this problem, the Principal Component Analysis (PCA), developed by Karl Pearson [1901] was used. PCA deploys the eigenvalue decomposition of data to find an orthogonal basis set that describes the variance in data points and convert the set of observation of possibly correlated variables into a set of values of linearly uncorrelated variables, also called principal components (PCs). This means that preserving as much variability as possible decodes into finding new variables that are linear functions of those variables in the original dataset, that maximize variance and are uncorrelated with each other [JOLLIFFE & CADIMA, 2016]. Finding such new variables (PCs), reduces to solving an eigenvalue or eigenvector problem. This analysis tends to simplify the complexity in the high-dimensional dataset while retaining trends and patterns. However, a disadvantage of this method is that the new PCs are usually linear functions of all the original features, and it is often the case that many features have non-trivial coefficients in the first few components, which makes the PCs difficult to interpret [JOLLIFFE & CADIMA, 2016].

After performing the PCA with the threshold of 0.85 for the variance, five remaining PCs were constructed. The correlation of these features is illustrated in Fig 4.5. As it can be seen, the correlation coefficients of the PCs are 0.

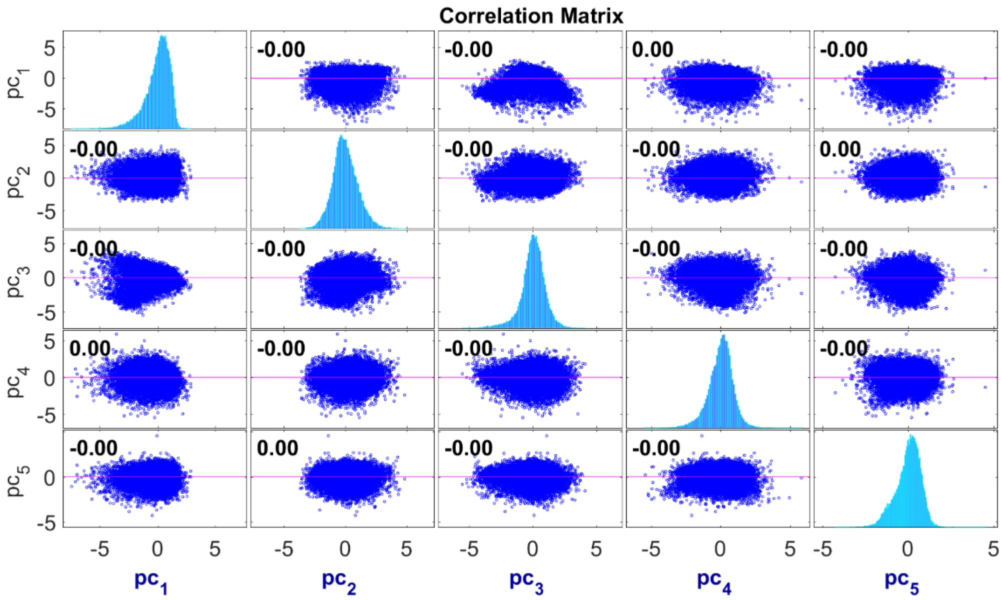


Fig 4.5 Correlogram between principal components

4.2 Clustering Algorithm

Clustering is a widely-used field of data analytics, and in recent years, a wide choice of clustering algorithms has been proposed in the literature. A complete overview of clustering algorithms can be found in *Introduction to Data Mining* [TAN ET AL., 2005] or *Data Mining and Predictive Analytics* [LAROSE & LAROSE, 2015]. Based on Halkidi et al. [2001], the representative algorithms can be categorized into the following groups:

Partitional algorithms:

Partitional clustering algorithms categorize the dataset into a set of disjoint clusters directly. In other words, these algorithms determine an integer number of partitions that optimize a certain criterion function. This criterion function emphasizes on the local or global structure of the dataset and is optimized in an iterative approach. Among the partitioning algorithms, the k-mean clustering algorithm developed by MacQueen [1967] is a commonly used algorithm. It is simple and can be used for a wide variety of data types. It is also efficient, even though multiple runs are often performed. k-mean is one of the most straightforward clustering algorithms that partitions n objects into k clusters in which each object belongs to the cluster with the nearest mean. The objective of k-mean clustering is to minimize the total intra-cluster variance, or squared error function. The objective function is formulated as follows:

$$J = \sum_{l=1}^{k} \sum_{i=1}^{n} \| x_i^l - c_l \|^2$$

Where k is the number of clusters, n is the number of cases, c_l is the centroid of each cluster and x_i^l is the case i. The k-mean clustering algorithm has the following steps shown in Algorithm 4.1:

Algorithm 4.1 K-means clustering algorithm

```
Randomly select k points as initial centroids where k is predefined.
While true
    Assign objects to their closest cluster center according to the Euclidean distance function
    Calculate the centroid or mean of all objects in each cluster.
    If centroids do not change:
        Break
```

The most widely used mean to measure the similarity between the observations in the dataset is the Euclidean distance. However, other measures such as Manhattan distance and categorical variables were also proposed for measuring the proximity [LAROSE & LAROSE, 2015]. The Manhattan distance is plausible for the grid-based situations, and categorical variables are used when measuring the physical distance is not possible [KATHS, 2017].

Partitional algorithms are computationally inexpensive and applicable mainly to numerical datasets. However, some variants of these algorithms can handle categorical data. These algorithms are unable to handle noise, and outliers and they are not applicable to the clusters with a non-convex shape. Another characteristic of these algorithms is that the number of clusters, except for Clustering Large Applications based on Randomized Search (CLARANS) algorithm [NG & HAN, 2002], should be specified in advance. [HALKIDI ET AL., 2001]

Hierarchical algorithms

Hierarchical algorithms proceed successively by merging smaller clusters into bigger ones or by splitting the large clusters into smaller ones. A tree of clusters, called a dendrogram, is obtained as a result of the algorithm, which shows the relation between the clusters. A clustering of the data items into disjoint groups is achieved by cutting the dendrogram at the desired level. Based on Theodoridis & Koutroumbas [2009] the hierarchical algorithms can be divided into according to the method of producing clusters:

- Agglomerative algorithms: These set of algorithms produce a sequence of clustering schemes of decreasing number of clusters at each step in which the clustering scheme results from the previous one by merging two closest clusters.

- Divisive algorithms: These divisive algorithms produce a sequence of clustering schemes of an increasing number of clusters at each step. In these algorithms, the clustering at each step results from the previous one by splitting a cluster into two separate clusters.

Hierarchical clustering algorithms are computationally expensive, and the number of clusters should be predefined. Although they are more efficient than partitional algorithms in handling noise and outliers, the finality of the merges can lead to an error when handling noisy or high-dimensional data.

Density-based algorithms

Density-based spatial clustering of applications with noise is a data clustering algorithm proposed by Ester et al. [1996]. The key idea of density-based clustering is to group neighboring objects of a data set into clusters based on density conditions. These algorithms typically regard clusters as dense regions of objects in the data space that are separated by regions of low density.

Density-based algorithms do not require one to specify the number of clusters in the data a priori. These algorithms can handle arbitrarily shaped collections of data (e.g., ellipsoidal, spiral, cylindrical) as well as clusters of different sizes. Moreover, they can efficiently separate noise [HALKIDI ET AL., 2001]. However, it is challenging to apply these algorithms to high-dimensional dataset and clusters with widely varying densities.

Grid-based algorithms are one type of density-based algorithms and are typically used for spatial data mining. These algorithms quantize the space into a finite number of cells and perform the operations on the quantized space. Firstly, each observation is labeled as either core, border or noise based on the number of observations within a predefined radius. After deleting the noise observations, the most suitable cluster core is assigned to the border observations. These algorithms are believed to be resistance noise and outliers and deliver good results when dealing with irregularly shaped data. Moreover, these approaches are very efficient for large databases, however, they might not be so efficient in high dimensional space [HAN ET AL., 2012].

After reviewing the clustering algorithms, based on the nature of the observations and the fact that the number of clusters is known a priori and can be fed as an input to the algorithm, the k-mean clustering algorithm is selected for this dissertation. However, for determining the optimum number of clusters, the elbow function [THORNDIKE, 1953] was used. As shown in Fig 4.6, the distortion goes down rapidly with K increasing from 1 to 2, and from 2 to 3, and then reaches an elbow at K=3, and then the distortion goes down very slowly after that. Therefore, the number of clusters is set to three in this study.

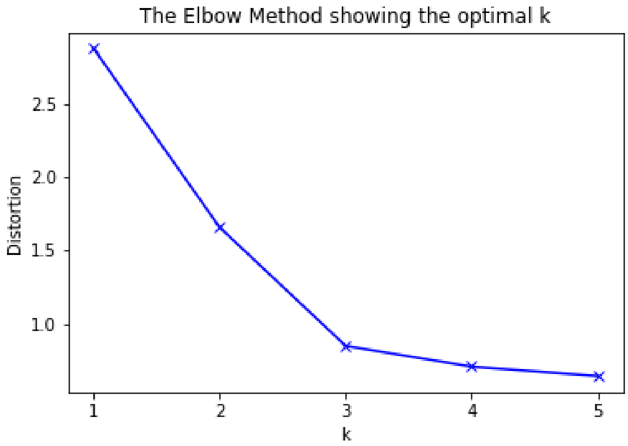


Fig 4.6 Result of elbow method for determining the optimal number of clusters

4.3 Validation of Results

One of the most critical issues in cluster analysis is the evaluation of clustering results to find the partitioning that can best fit the underlying data. The procedure for evaluating the result of the clustering algorithm is known as cluster validity [HALKIDI ET AL., 2001]. Based on Theodoridis & Koutroumbas [2009], the cluster validation can be divided into three categories:

- External criteria (indices): This approach is based on evaluating whether the points of the dataset are randomly structured or not and is used to measure the extent to which the cluster labels match externally supplied class labels. Entropy and purity are examples of external measures for cluster validity.

- Internal criteria (indices): In this approach, the goal is to evaluate the clustering result of the algorithm using only quantities and features inherent to the dataset. Cohesion, separation, Sum of Squared Error (SSE), and Silhouette coefficient are examples of the internal measures used to validate clustering.

- Relative criteria (indices): this criterion compares different clustering or clusters. Relative criteria are able to compare two clustering structures and point out which one is better in relative terms. These criteria are often based on the general idea of measuring somehow the balance between intra-cluster scattering and between-cluster separation, with differences arising mainly from different formulations of these two fundamental concepts [JASKOWIAK ET AL., 2016].

As a result of k-mean clustering procedure, the drivers were categorized into three different clusters. Once the clusters are developed, the quality of the clusters must be assessed to ensure that they reflect valid clusters in the data [LAROSE & LAROSE, 2015], and the similarities between the observations included in each cluster should be examined. For this aim, the silhouette value [ROUSSEEUW, 1987] was used in this study. The silhouette value is an internal criterion, which is used for interpretation and validation of consistency within a cluster of data and for each point measures how similar that point is to points in its cluster when compared to points in other clusters. The silhouette value for the ith point, s_i, is defined as:

$$S_i = \frac{b_i - a_i}{max(a_i, b_i)}$$

where a_i is the average distance from the ith point to the other points in the same cluster as i, and b_i is the minimum average distance from the ith point to points in a different cluster, minimized over all clusters. The silhouette value ranges from -1 to +1. A high silhouette value indicates that observation i is well-matched to its own cluster and poorly-matched to neighboring clusters. A high silhouette value for most points means that the clustering solution is appropriate. On the other hand, many points with a low or negative silhouette value represent that the clustering solution may have either too few or too many clusters. The silhouette clustering evaluation criterion can be used with any distance metric. Tab. 4.2 illustrates the average silhouette score and the number of each cluster.

Tab. 4.2 Number of samples and average Silhouette score in each cluster

Cluster	#	Mean Silhouette Score
1	18490	0.71
2	18326	0.69
3	3953	0.79

4.4 Interpretation of Results

In the application area, the clustering results have to be carefully investigated and integrated with other experimental evidence and analysis to ensure the correct interpretation of the results. In order to fulfill the aim of clustering in this study, which was the identification of aggressive driving behavior and elimination of the corresponding driving behavior from the dataset for the model development for automated vehicles, first, the histograms of the PC values for each cluster were analyzed. However, as mentioned in chapter 4.1, the interpretation of the meaning of each PC is not possible. Therefore, the histograms of the features for each cluster are illustrated in Fig 4.7. Some features like relative distance or relative speed are traffic-situation specific and cannot be used alone to judge the style of driving behavior. However, drivers in cluster 3 had considerably higher average longitudinal jerks. This kind of jerky driving behavior can be linked to a higher risk of accidents. Therefore, this cluster is considered to contain the profile of aggressive drivers. It is noteworthy to mention that the dataset with more information about the vehicle type can further improve the accuracy of the proposed clustering method for driving behavior analysis.

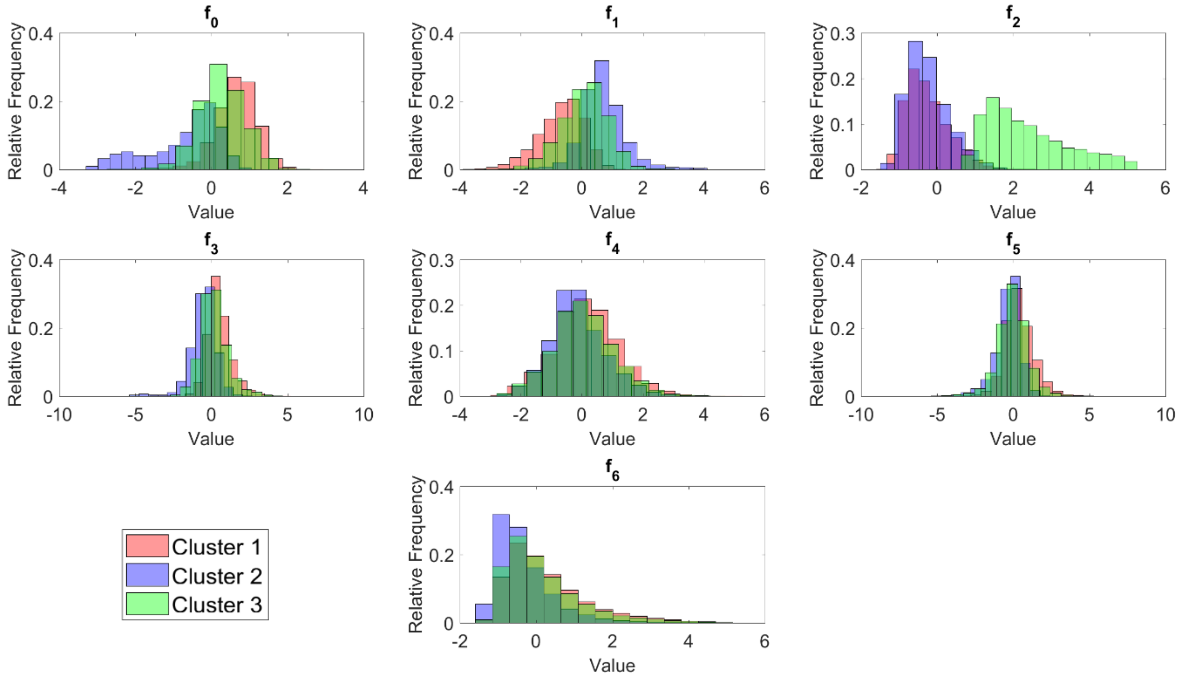


Fig 4.7 Histogram of features for each cluster

4.5 Summary

This chapter demonstrated how to use clustering, which is a common unsupervised learning approach, to identify distinct driving patterns from a large-scale observational data set collected from individual vehicle trajectories. Specifically, the average speed and relative speed to the neighboring vehicles, average jerk and distance to the preceding and following vehicles were used to create a driver profile. After profiling the drivers with the features, a k-mean cluster analysis was performed to identify various driver groups. It has been shown that driving behavior can be classified into three distinct groups such as aggressive drivers, moderate drivers and timid drivers. For developing the driving behavior of automated vehicles in the next chapters, the observations from the aggressive drivers were filtered out from the data.

5. Modeling Lane Change Behavior

Lane changing has a substantial impact on traffic operation and safety. Research studies have revealed the crucial role of lane changing in traffic breakdown and capacity drop [CASSIDY & RUDJANAKANOKNAD, 2005; JIN, 2013], traffic oscillation [ZHENG ET AL., 2011], relaxation [LECLERCQ ET AL., 2007; LAVAL & LECLERCQ, 2008], and moving bottleneck [LAVAL & DAGANZO, 2006]. Therefore, accurate modeling of the lane changing behavior for traffic studies and especially for studying the impact of emerging forms of mobility such as automated vehicles is of high importance. As Toledo [2007] mentioned, lane change models usually assume two-step decision process, lane selection, and lane change execution. This chapter also describes the modeling of lane change decisions (section 5.1) and lane change execution (section 5.2). For each part, the core design of the model, objective and assumptions, as well as the results of each model are described.

5.1 Modeling Lane Change Decision

In this chapter, the methodology used for modeling the tactical behavior of lane changing is explained. Part of this chapter was published in *Modeling tactical lane-changing behavior for automated vehicles: A supervised machine learning approach* [MOTAMEDIDEHKORDI ET AL., 2017]. The goal of this modeling approach is to learn the tactical behavior of drivers from the available dataset. As the advantages of ML methods over the traditional physical models were highlighted in chapter 2, a ML approach was used for this task. ML has different types: unsupervised, semi-supervised, supervised, RL [MARSLAND, 2014].

In this study, the tactical decisions made by drivers are embedded in the database, and the ML task infers the function from the labeled dataset. Therefore, for modeling the tactical behavior of lane changing, supervised ML was used in which the input variables (x) and an output variable (y) are given by the data an algorithm is used to learn the mapping function from the input to the output.

$$y = f(x)$$

The goal is to approximate the mapping function so well that when new input data (x) is provided to the model, it can predict the output variables (Y) for that data. The general process for a ML process is shown in Fig 5.1.

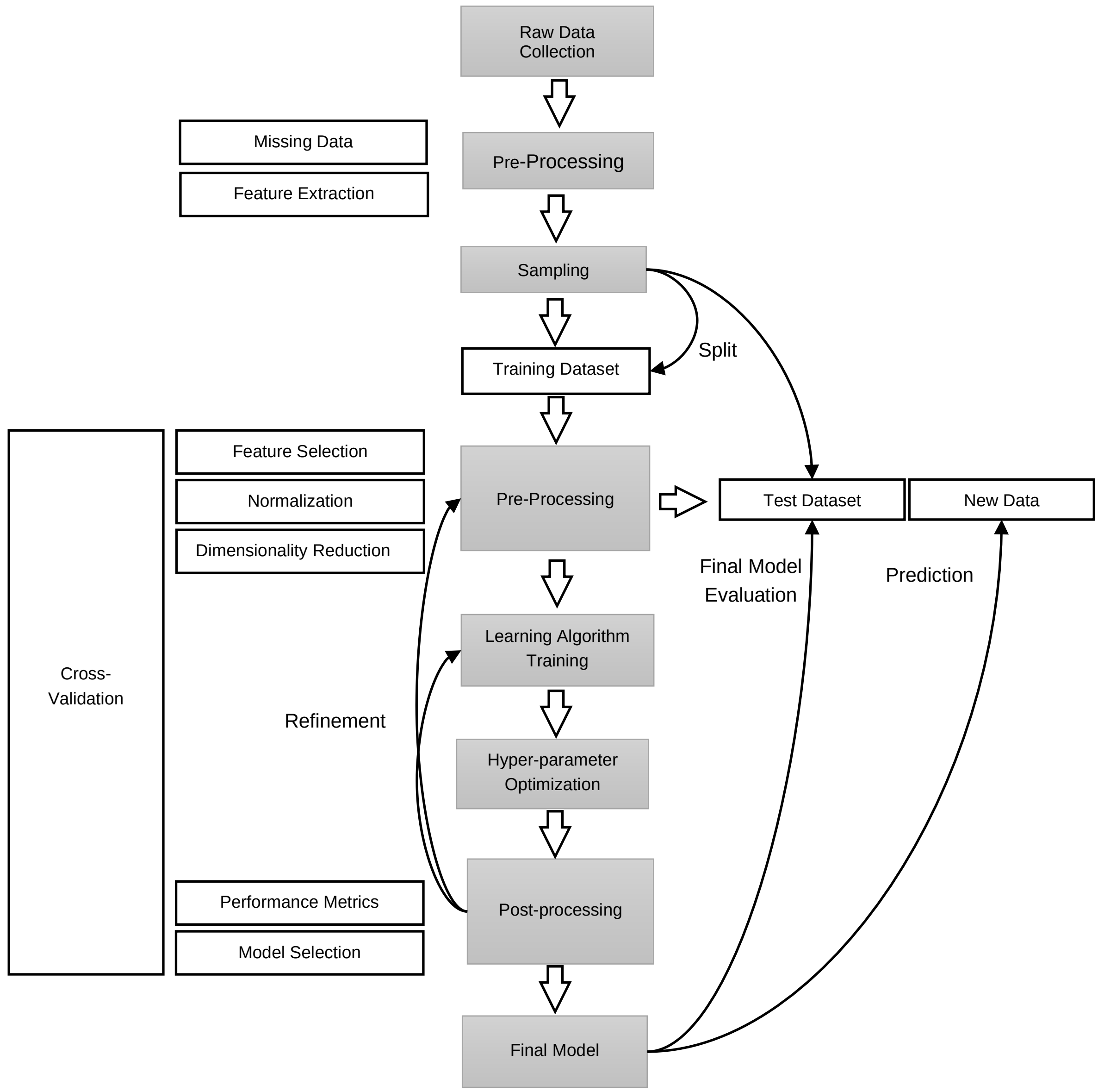


Fig 5.1 Supervised machine learning framework

Supervised ML algorithms have been vastly used in different disciplines to understand and model human behaviors, including driving behavior task. A Feature-based method was used in one study [KUDERER ET AL., 2012] for predicting trajectories of pedestrians by a robot with the help of maximum entropy. In another study [KRETZSCHMAR ET AL., 2014], the authors have presented an approach to learn the composite behavior of multiple interacting agents from demonstrations. In their research, the Hamiltonian Markov chain model was used to learn the behavior of pedestrians. In [PAPATHANASOPOULOU & ANTONIOU, 2015] and [PAPATHANASOPOULOU & ANTONIOU, 2016], the authors deployed data-driven methods (local regression method) to model car-following models. Some studies used ML algorithms to understand, analyze and model decision-making processes [Huikun Bi et al., 2016] with different classifiers (naive Bayes and decision trees [HOU ET AL., 2014b], regression tree [MENG & WENG, 2012]) to predict the drivers' behavior.

5.1.1 Core Design, Objectives and Assumptions

The prediction of tactical lane changing behavior is defined as a classification problem. Classification predictive modeling aims to approximate a mapping function (f) from input variables (x) to discrete output variables (y), and the mapping function predicts the class for a given observation [KOTSIANTIS, 2007]. In this model, the feature vector is the input, and the classifier predicts whether the driver would decide to change the lane to the right, to the left or stay within one lane in each specific driving situation. Therefore, the classification is a multi-class classification problem. The flowchart of the classification problem is illustrated in Fig 5.2.

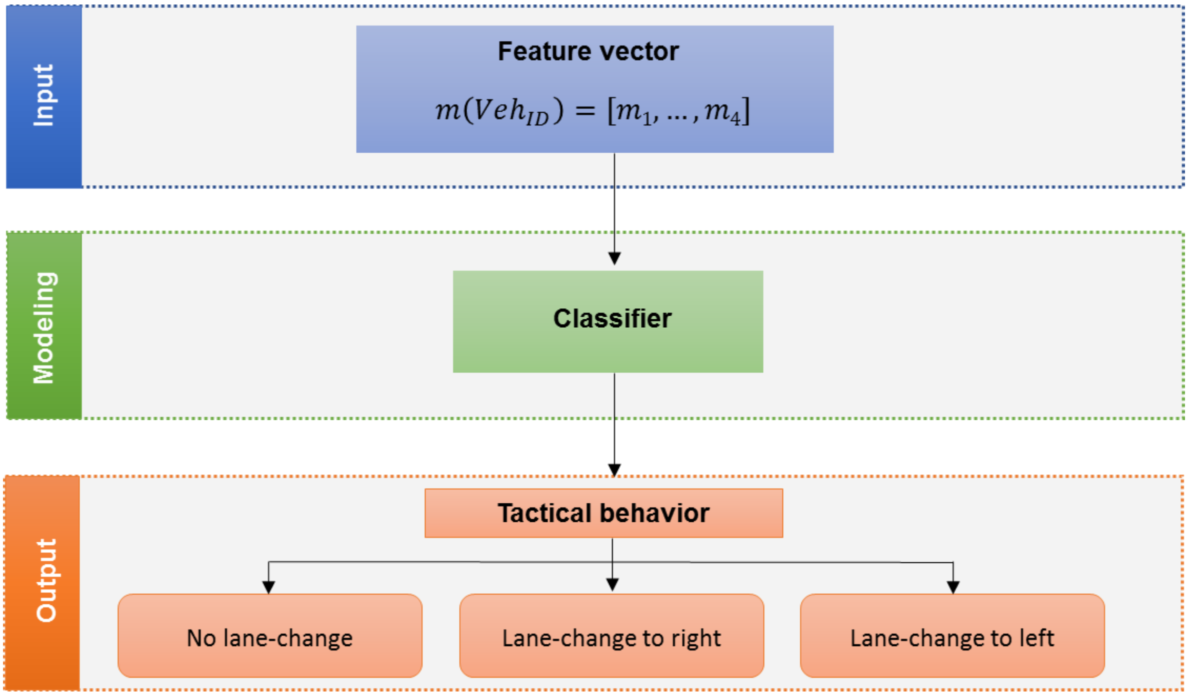


Fig 5.2 Classification flowchart

As it can be seen in Fig 5.2, the main core of the modeling is on the classifier. The are many algorithms for solving classification problems: Examples of these are: Naïve Bayes, Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel and linear kernel, Logistic Regression (LR), k-Nearest Neighbors (kNN), Extra Trees classifier, Decision Trees, Artificial Neural Network (ANN) and ensemble methods such as RF. These classification algorithms are here briefly reviewed and compared:

Naïve Bayes classifiers are a set of supervised learning algorithms based on applying Bayes' theorem [Bayes & Price, 1763] with the naive assumption of independence between pairs of features. In other words, Naïve Bayes classifier technique assumes that the predictors are independent. This classifier is easy to build, appropriate when the dataset is high-dimensional and represents how generative assumptions and parameter estimation can simplify the learning process [Shalev-Shwartz & Ben-David, 2014].

SVM, also known as support vector networks [CORTES & VAPNIK, 1995], is appropriate for learning linear predictors in high dimensional feature spaces [SHALEV-SHWARTZ & BEN-DAVID, 2014] and are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. SVM classifier tries to find a hypothesis h, for which the true error can be guaranteed to be the lowest. RBF kernel nonlinearly can map samples into a higher dimensional space to handle the nonlinear relationship between attributes and class labels. SVM algorithm is suitable for both regression and classification tasks and can handle multiple continuous and categorical variables. This classifier is known to have high accuracy for small and clean datasets; however, it is not suitable for large datasets. Moreover, the classifier does not perform well when dealing with noisy or overlapping datasets.

LR is named after the core of the method, the logistic function [VERHULST, 1838]. LR describes the relationship between explanatory variables and a discrete response variable [YAN ET AL., 2005]. In LR, the goal is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest and a set of independent variables. Maximum-likelihood estimation should be used to estimate the coefficients of the LR algorithm from the training data [SHALEV-SHWARTZ & BEN-DAVID, 2014]. The output of the linear function is squashed within the range of [0,1] using the sigmoid function (logistic function) [CYBENKO, 1989]. As shown in Fig 5.3, the sigmoid function is a bounded differentiable real function defined for all real input values and that has a positive derivative everywhere. This S-shaped curve can take any real-valued number and map it into a value between the range of 0 and 1. Moreover, it shows a sufficient degree of smoothness and is also a suitable extension of the soft limiting nonlinearities used previously in neural networks [HAN & MORAGA, 1995].

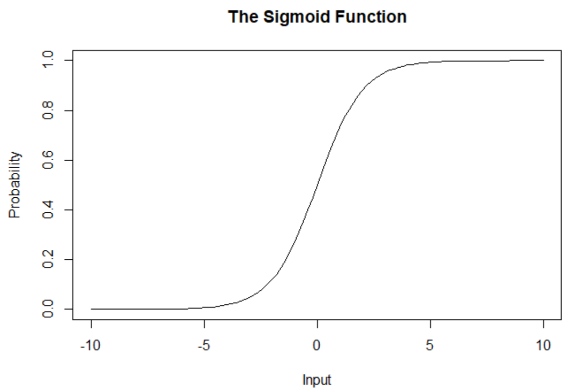


Fig 5.3 Sigmoid function

Typically, if the squashed value is higher than a threshold value, it is assigned to a label 1, else it is assigned to a label 0. Therefore, LR classifier is typically used for the binary classification; however, for extending LR to multiple classes two methods are used:

1) One-vs-All/ One-vs-Rest: This strategy consists of fitting one classifier per class against all the other classes. This method is computationally efficient and has high interpretability of results. Since each class is represented by one classifier only, it provides the possibility to gain knowledge about the class by inspecting its corresponding classifier. [BUITINCK ET AL., 2013].

2) One-vs-One: This strategy consists of fitting one classifier per class pair and selecting the class which received the most votes at the prediction time. Since it requires to fit $\frac{number\ of\ classes * (number\ of\ classes - 1)}{2}$ classifiers, this method is usually slower than One-vs-Rest, due to its complexity [BUITINCK ET AL., 2013].

3) Softmax function: This function is a generalization of the sigmoid function that is used to handle the multi-class classification [BOLTZMANN, 1868]. This function takes values as an input and returns the proper probabilities. In this function, the sum of returned values is equal to 1. The mathematical formulation of this function is as follows:

$$S(y_i) = \frac{e^{y_i}}{\sum_1^i e^{y_i}}$$

kNN classification, which is based on k-nearest neighbor rule [FIX & HODGES, 1989], is one of the most fundamental and simple classification methods mostly used when there is a lack of prior knowledge about the data distribution. The nearest neighbor method is based on memorizing the training dataset and finding a predefined number of training samples closest in distance to the new instance, and predicting the label from them [Shalev-Shwartz & Ben-David, 2014]. K represents the number of clusters or neighbors that the algorithm will check. For calculating the class membership, each object is classified by a majority vote of its neighbors, and the object is assigned to the class most common among its k nearest neighbors. Although kNN classification algorithms are simple to understand and interpret and usually show high accuracy, they are computationally expensive and require a lot of memory.

Decision Tree [BREIMAN, 1984] is a non-parametric supervised learning method used for classification and regression and tries to find the best split of data at each point. The goal is to develop a model that can predict the value of a target variable by learning decision rules derived from the data features. Decision Tree builds regression or classification models following a tree structure. In this process, the dataset is broken down into smaller and smaller subsets while at the same time an associated decision tree, with decision nodes and leaf nodes, is gradually developed. A decision node has two or more branches, and a leaf node represents a classification or decision. Decision Trees have several advantages such as simplicity for understanding, being applicable to both categorical and numerical data, and mirroring human decisions more closely; however, they have the risk of creating over-complex trees without generalizing the data well [QUINLAN, 1986; MITCHELL, 1997].

RF classifier consists of tree predictors, and each tree depends on the values of a random vector sampled with the same distribution and independently for all trees in the forest [BREIMAN, 2001]. Based on [BUITINCK ET AL., 2013], this classifier fits several tree classifiers on various subsamples of the dataset and improves the predictive accuracy and controls overfitting by averaging them. RF classifier is among the very easy to use algorithms due to the limited number of hyper-parameters that are straightforward to understand and usually delivers good accuracy. RF also tackles one of the most significant problems in ML, namely overfitting. Overfitting the classifier to fir the model can be controlled by defining enough number of trees. The main limitation of RF is that a large number of trees can make the algorithm slow down in real-time predictions. In general, these algorithms are efficient in training, but they might be slow to create predictions on test dataset when the number of trees is high.

Extra Tree Classifier, also known as an “Extremely randomized trees” classifier, is a variant of a RF in which the splits are selected randomly instead of using some criteria [GEURTS ET AL., 2006]. The difference between the Extra Tree classifier and the classical Decision Trees is their splitting structure. As part of looking for the best split to separate the samples of a node into two groups, random splits are drawn, and the selection of features is made completely randomly, and the best split is chosen. Therefore, their structure is independent of the output values of the learning data set [GEURTS ET AL., 2006].

ANN classifier uses the neural network for the classification tasks. Neural network is composed of a number of simple, interconnected neurons working in parallel within the network. ANN has the capability of developing an internal representation of a signal pattern that is presented as input to the network [MCCULLOCH & PITTS, 1943]. A key feature of ANN is an iterative learning process. In this process, records are presented to the network one at a time, and the weights associated with the input values are tweaked each time. After presenting all cases, the process is often repeated, and the network trains by adjusting the weights to predict the appropriate class label of input samples. In the training phase of classification, the network’s calculated values for the output nodes are compared to the correct values of each class an error term for each node is calculated. These errors are later used to adjust the weights in the hidden layers in a way that in the next iteration, the output values will be closer to the correct values [WAN, 1990].

Tab. 5.1 summarizes the advantages and disadvantages of the classification algorithms. After reviewing the complexities of the algorithms and comparing their advantages and disadvantages, the RF and LR classifiers were chosen and implemented to model the lane changing decision of drivers.

Classification method	Advantage	Disadvantage
Naïve Bayes	• based on simple probability principles • fast • works well with high dimensions	• interdependencies between the attributes
SVM	• high accuracy • works well even with non-linearly separable base features	• High complexity and extensive memory requirement • Susceptible to overfitting/training issues depending on kernel
LR	• convenient probability scores for observations • efficient implementations available across tools • Multi-collinearity is not an issue and can be countered with L2 regularization to an extent • pretty efficient in terms of time and memory requirement	• Prone to bias
kNN	• classes should be linearly separable • usually robust with regard to noisy data • well-suited for multi-dimensional classes	• finding the nearest neighbors in large training dataset can be time-consuming • sensitive to noise and irrelevant attributes
Decision Trees	• simplicity for understanding • applicable to both categorical and the numerical data	• prone to overfitting • possible issues with diagonal decision boundaries
ANN	• easy to implement and use, with few parameters to adjust • applicable to wide range of real-life problems	• requires high processing time if neural network is large • difficult to determine the number of necessary neurons and layers • slow learning
RF	• high predictive accuracy compared to decision trees • control overfitting by averaging • easy to use algorithm • low number of hyper-parameters	• can be slow down in real-time predictions • not as easy to visually interpret

Tab. 5.1 Classification methods comparison

RF was chosen due to its simplicity to interpret the results. The only disadvantage linked to RF is the slow speed with the existence of many trees, which can be easily addressed and controlled by avoiding to choose a high number of trees as a hyper-parameter. This hyper-parameter is the number of trees the algorithm builds before taking the maximum voting or taking averages of predictions [BUITINCK ET AL., 2013]. Generally, a higher number of trees increases the performance and improves the stability of the prediction, but it also slows down the computation. LR does not have a discrete output. Instead of delivering discrete outputs, it delivers class membership probabilities associated with each observation as an output. For LR models, it is possible to test the statistical significance of the coefficients in the model [HOSMER JR ET AL., 2013].

In this study, the information about the ego vehicle's environment is extracted as a form of features that describe the basic properties of the surrounding vehicles. Since the decision for lane changing depends on the traffic situation for consecutive steps, a finite length of features was considered, which was called a scenario-slice in the study by Reichel et al. [2010]. A scenario-slice $S[t]\epsilon\, R^{L*Q}$ consists of L time-series each of length Q. The task is to find a mapping from $S[t]$ to a decision $y[t]\ \epsilon\,\{-1,0,1\}$, where $y[t]=1$ indicates that at the time instance n the ego-car decided to change lane to the right, $y[t]=0$ there was no lane change and $y[t]=-1$ a lane change to the left happened. However, in the classical classification problem, it is assumed that the observations are the realizations of random variables. In this context, the random variables from the same scenario-slice e.g., $s[k]$ and $s[k-1]$, are statistically dependent. Therefore, independent entire scenarios were defined as $S\epsilon R^{L*(t_{end}+Q+1)}$, which consists of t_{end} scenario-slices $s[t]$, where $t=1,\ldots,t_{end}$. Therefore, when looking for a classifier:

$$f\colon R^{L\times Q}\rightarrow\{-1,0,1\}, s[t]\mapsto \hat{y}\,[t]$$

It maps a scenario-slice $S[t]$ to a decision,

$$\hat{y}\,[t]=f(s[t])\epsilon\,\{-1,0,1\}$$

In order to simplify a complex classification task, namely mapping f from $S[t]$ to $\hat{y}\,[t]$, is divided into two steps: 1) a feature generation step where $S[t]$ is transformed into a feature vector $x[t]$and a classification step where $x[t]$ is mapped to $\hat{y}\,[t]$.

Feature extraction:

The first step in classification is the translation of the vehicles and its surrounding into the features. In order to percept the individual driving situation and environment of the driver, the general skeleton defined for situation aspect models introduced by [REICHEL ET AL., 2010] was deployed. According to his theory, the situation can be modeled by a protagonist (ego vehicles), further actors, the stage, and constellation of all of them. In this case, these elements are the data about the ego vehicle, dynamic objects such as other road users, environment components such as road infrastructure, and static objects. Fig 5.4 illustrates the model used for defining the situation.

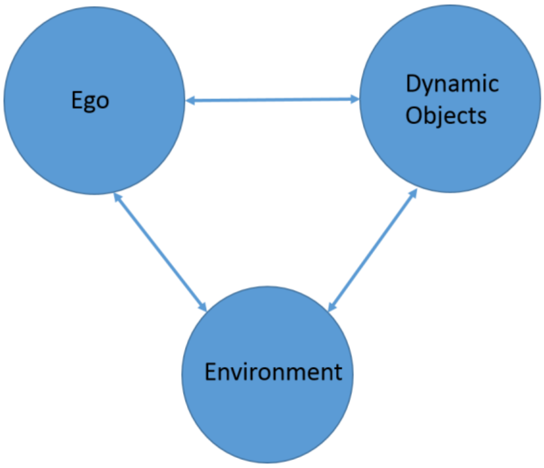


Fig 5.4 General situation aspect model [REICHEL ET AL., 2010]

In order to translate the individual driving situation and environment of the vehicle as an input vector consisting of features, the description of the situation aspect using the abovementioned model for an ego vehicle, which is shown in red in Fig 5.5, was used. The inputs summarize the state of the ego car, which consists of internal and external information as a feature.

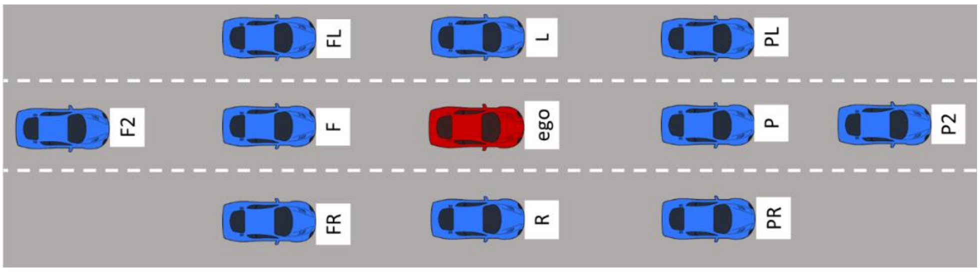


Fig 5.5 Ego vehicle and its environment. Vehicle j can be one of the following vehicles: P: preceding vehicle, P2: preceding vehicle of the preceding vehicle, PL: preceding vehicle on the left lane, PR: preceding vehicle on the right lane, L: vehicle on the left lane, R: vehicle on the right lane, F: following vehicle, F2: following vehicle of the following vehicle, FL: following vehicle on the left lane, FR: following vehicle on the right lane.

A set of features for the target vehicle (for which the future action is to be computed) and for its surrounding vehicles as described above were generated from the dataset. These features can be divided into three categories: 1) core feature for ego vehicle, 2) roadway features for the environment and 3) vehicle relative features for dynamic objects. The features constructed for this study are inspired by those used in the literature [SCHLECHTRIEMEN ET AL., 2014; KASPER ET AL., 2012; WHEELER, 2014; SCHLECHTRIEMEN ET AL., 2015].

The core features category addresses the ego vehicle and its dynamic characteristics. For the target vehicle, the features such as absolute speed, longitudinal and lateral speed, and acceleration were defined. As mentioned in chapter 2.1.2, the driving environment, such as roadway geometry and traffic situation, are decisive factors on driving behavior. Driver's reactions not only depend on the adjacent vehicles but also vary with the traffic conditions in a broader sense. Therefore, the second set of features translates these attributes into the feature vector. An influential factor is general traffic characteristics, such as the average velocity on a lane, or the traffic state. As a result, the mean relative velocity of the ego vehicle from the other vehicles in the scene was defined as a feature in the input vector. The choice of feature of other vehicles was made to replicate the information a human driver is likely to base his decisions upon the features from surrounding vehicles are all relative to the target vehicle, as the drivers are expected to usually make decisions based on perceived distances and relative speeds rather than their values in an absolute frame [ALTCHÉ & LA FORTELLE, 2017]. For each vehicle $j \in \{PL, P, P2, PR, L, R, FL, F, F2, FR\}$, the following features were defined:

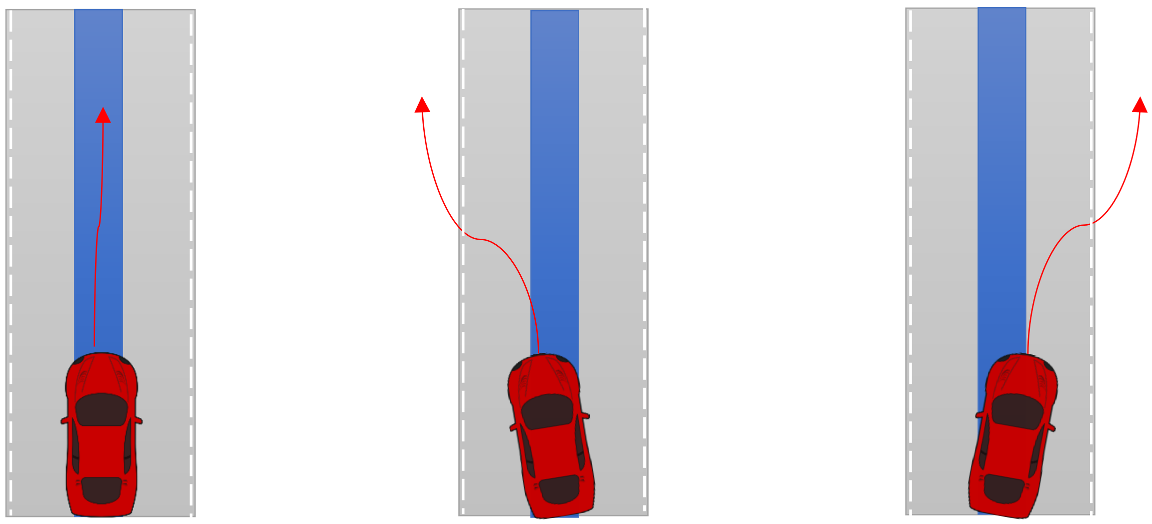
- Distance
- Longitudinal relative speed
- Longitudinal acceleration
- Lateral acceleration
- Type of vehicle j (car, medium sized vehicle, truck) considering the unique physical characteristics of heavy vehicles (e.g., length and size) and their operational characteristics (e.g., acceleration, deceleration, and maneuverability) [RAHMAN ET AL., 2013].

As a history of movement of each vehicle plays an important role and decides whether other vehicles are behaving cooperatively or not, the value of the feature for three consecutive time steps was defined in the input vector. Tab. 5.2 summarizes the features and specifies whether they are continuous or discrete features.

			Core features of ego features
$\|\|v\|\|$	$\frac{m}{s}$	continuous	Speed
v^{ego}	$\frac{m}{s}$	continuous	Speed in lane
a_x^{ego}	$\frac{m}{s^2}$	continuous	Longitudinal acceleration in lane
a_y^{ego}	$\frac{m}{s^2}$	continuous	Lateral acceleration in lane
			Driving environment features
$lane$	-	discrete	Index of the closest lane
Δv_{yx}^{scene}	$\frac{m}{s}$	continuous	The mean velocity of vehicles in the scene
			Vehicle relative features
d_j	m	continuous	Distance between the ego vehicle and vehicle j
Δv_j	$\frac{m}{s^2}$	continuous	Relative speed between ego vehicle and vehicle j
$a_{x,j}$	$\frac{m}{s^2}$	continuous	Longitudinal acceleration of vehicle j
$a_{y,j}$	$\frac{m}{s^2}$	continuous	Lateral acceleration of vehicle j
$Class_j$	-	discrete	Class of vehicle j

Tab. 5.2 Features for classification. Vehicle j can be any of the surrounding vehicles $\{BL, B, B2, BR, L, R, FL, F, F2, FR\}$

One step toward explanatory models is annotating datasets with theoretically motivated labels or semantic features. Therefore, the next step was to identify the lane changing events, label them based on the features and assign them a target class value. The target class values were assigned in response to the question "Does the ego vehicle perform a lane change to the maneuver to the right or to the left?". In order to label the events, as shown in Fig 5.6, a blue band which has the width of the quarter of the lane width was defined in the middle of each lane. An event was labeled as lane changing in case the lateral position of the vehicle shifts from the defined band of the lane without any oscillations. The classifier here calculated the membership of each instance to the three classes.



$\hat{y}\ \epsilon\{No\ lane\ change\}$ $\hat{y}\ \epsilon\{Lane\ change\ to\ left\}$ $\hat{y}\ \epsilon\{Lane\ change\ to\ right\}$

Fig 5.6 Labeling lane change events

Feature selection:

For selecting the features that contribute the most to the prediction and reduce the dimensionality on the sample set, a feature importance analysis was performed on the dataset. As the dataset was high-dimensional, the dimensionality reduction might help to boost the estimator's performance. Different methods such as recursive feature elimination, feature importance ranking, and univariate selection are available to perform feature selection. The feature selection task is generally divided into three types [Chandrashekar & Sahin, 2014; Miao & Niu, 2016; Tang et al., 2014]:

1. Filter methods select the most discriminative features through the data character. The feature selection in this method is done before the classification and is divided into two

steps. First of all, the variables are ranked, and then K highest ranked features are selected [SÁNCHEZ-MAROÑO ET AL., 2007].

2. Wrapper methods use the intended learning algorithm itself to evaluate the features. One of the most known wrapper methods is Recursive Feature Elimination (RFE) [GUYON ET AL., 2002]. The goal of RFE is to select features by repeatedly considering smaller and smaller sets of features. After estimating the importance of features, the least important features are pruned from the current set of features. That procedure is recursively repeated until the desired number of features to select is eventually reached [BUITINCK ET AL., 2013].

3. Embedded models in which the feature selection is embedded in the process of model construction. These methods are independent of the classifier and avoid the cross-validation step. Therefore, they are computationally very efficient. However, they do not take into account the biases of the classifier [TANG ET AL., 2014].

For feature importance in the classification task of this dissertation, the extra tree classifier was used. Extremely-randomized Tree is a method for supervised classification problems which strongly randomizes the attribute and cut-point while splitting a tree node. In this method, when looking for the best split to separate the samples of a node into two groups, unlike RF, the idea of using bootstrap copies of the learning sample drops. Instead of trying to find an optimal cut-point for each one of the randomly chosen features at each node, random cut-points are drawn for the K randomly selected features and the best split among those is chosen. This method is believed to have good performance in high-dimensional datasets [JODOGNE ET AL., 2006] and a better bias-variance due to the explicit randomization of the cut-point and attribute combined with ensemble averaging [GEURTS ET AL., 2006]. The tabular version of the Extra-Trees splitting procedure for numerical attributes developed by Guerts et al. [2006] is given in Algorithm 5.1. The splitting procedure has two parameters: K which is the number of attributes randomly selected at each node and n_{min}, which is the minimum sample size of the splitting a node. This parameter is used several times with the original learning sample to generate an ensemble model. The final prediction of the model is the aggregation of the prediction of trees.

Algorithm 5.1 Extra trees splitting algorithms [GEURTS ET AL., 2006]

Split_a_node(S)
$Input$: the local learning subset S corresponding to the node we want to split
$Output$: a split $[a < a_c]$ or nothing
– If **Stop_split** (S) is TRUE then return nothing.
– Otherwise select K attributes $\{a_1, \ldots, a_K\}$ among all non-constant (in S) candidate attributes;
– Draw K splits $\{S_1, \ldots, S_K\}$, where S_i = **Pick_a_random_split** (S, a_i),$\forall\ i = 1, \ldots, K$;
– Return the split s_* such that Score $(s_*, S) = max_{i=1,\ldots,K}$ Score(s_i, S).

Pick_a_random_split(S, a)
$Inputs$: a subset S and an attribute a
$Output$: a split
– Let a_{max}^S and a_{min}^S denote the maximal and minimal value of a in S;
– Draw a random cut-point a_c uniformly in $\left[a_{min}^S, a_{max}^S\right]$;
Return the split $[a < a_c]$.

Stop_split(S)
$Input$: a subset S
$Output$: a Boolean
– If $|S| < n_{min}$ then return TRUE;
– If all attributes are constant in S, then return TRUE;
– If the output is constant in S, then return TRUE;
– Otherwise, return FALSE.

It is noteworthy to mention that before performing the classification algorithms, as the features had different scales, the features were standardized with the same method mentioned in chapter 4.2.

The results of feature importance with extra tree classifier for human drivers' and automated vehicle dataset are illustrated in Fig 5.7. In this plot, the horizontal axis shows the features ranked based on their importance, and the value of the contribution of each feature to the result is shown on the vertical axis. This value is between 0 and 1 and the higher values correspond to the higher importance of the feature during the lane changing process. The list of the features is illustrated in Tab. 5.3. Five most important features in this model are: Speed of ego vehicle, longitudinal and lateral acceleration of ego vehicle, average relative speed to the other vehicles in the scene, lane number, and distance to the preceding vehicle. On the other hand, the class of surrounding vehicles is not influencing factors in the decision-making process of lane changing behavior. The results for the automated vehicles showed the same trends and very similar importance scores for the features.

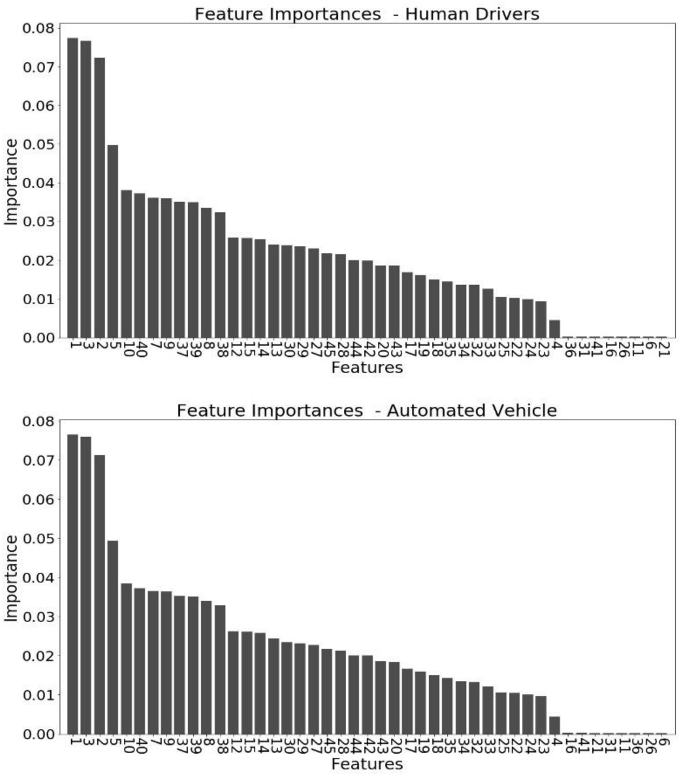


Fig 5.7 Feature importance result for human drivers and automated vehicles

Number	Feature	Number	Feature	Number	Feature
1	v^{ego}	16	$Class_B$	31	$Class_{BL}$
2	a_x^{ego}	17	Δv_B	32	Δv_{BL}
3	a_y^{ego}	18	$a_{x,B}$	33	$a_{x,BL}$
4	$lane$	19	$a_{y,B}$	34	$a_{y,BL}$
5	Δv_{yx}^{scene}	20	d_B	35	d_{BL}
6	$Class_F$	21	$Class_{B2}$	36	$Class_{FR}$
7	Δv_F	22	Δv_{B2}	37	Δv_{FR}
8	$a_{x,F}$	23	$a_{x,B2}$	38	$a_{x,FR}$
9	$a_{y,F}$	24	$a_{y,B2}$	39	$a_{y,FR}$
10	d_F	25	d_{B2}	40	d_{FR}
11	$Class_{F2}$	26	$Class_{FL}$	41	$Class_{BR}$
12	Δv_{F2}	27	Δv_{FL}	42	Δv_{BR}
13	$a_{x,F2}$	28	$a_{x,FL}$	43	$a_{x,BR}$
14	$a_{y,F2}$	29	$a_{y,FL}$	44	$a_{y,BR}$
15	d_{F2}	30	d_{FL}	45	d_{BR}

Tab. 5.3 List of features

In the classification of high-dimensional data containing groups of correlated features, the requirements of model sparsity and retrieving of all predictive features are in direct competition [DO ET AL., 2010]. In applications in which assessment of feature importance is the main objective, models that give priority to the latter requirement should be preferred. However, as mentioned at the beginning of this chapter, having too many features might lead to overfitting, which is referred to as the "curse of dimensionality" [BELLMAN, 1957]. Before eliminating the unimportant features, it should be examined whether the features are in any sense correlated and can be transformed into a set of values of linearly uncorrelated variables called PCs. In this case, the PCs can capture most of the information without losing any of the features.

Dimensionality reduction:

As mentioned in chapter 4.14.1, If the features are perfectly correlated, then one does not add any additional information, so if the number of features is too high, reducing the number of features through dimensionality reduction techniques such as PCA would be beneficial. In classification problems, a high correlation between the explanatory features might decrease the classification accuracy. Dimensionality reduction is the introduction of a new feature space where the original features are represented. The new space is of a lower dimension than the original space. Therefore, the correlation between the features should be examined. The correlations between the variables are determined using the Spearman method [SPEARMAN, 1904], in which the correlation between the variables is determined based on their monotonic relationship and not their linear relationship.

In order to visualize the correlations between the features, the correlation matrixes of features were plotted. As can be seen from Fig 5.8, the matrix is symmetrical, and the bottom left of the matrix is the same as the top righthand the color shows the correlation between the variables. It can also be seen that each variable is perfectly positively correlated with each other in the diagonal line from top left to bottom right. As can be seen from the matrix, some features are correlated, for instance, a high correlation can be observed between the speed of the ego vehicle and the average speed difference from the surrounding vehicles.

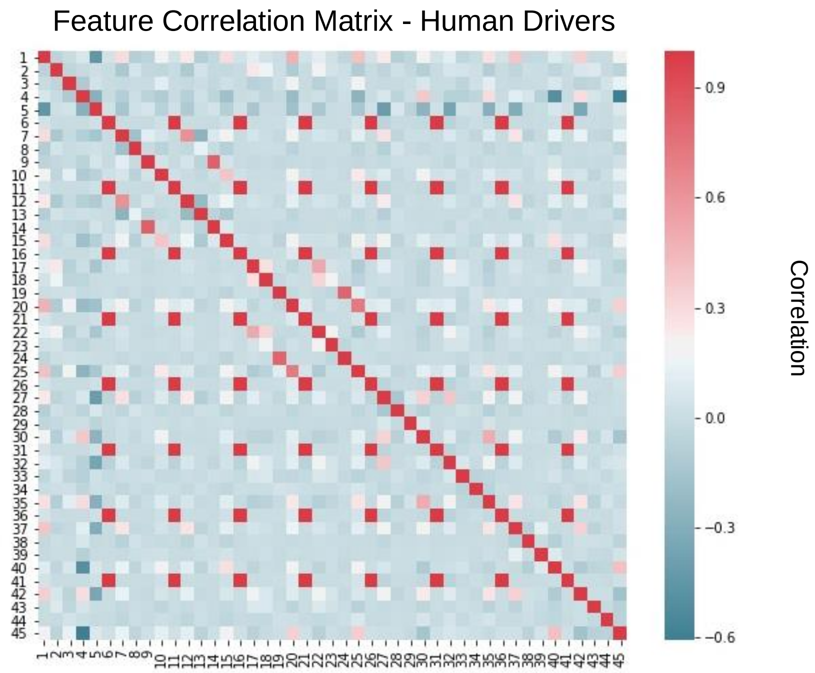


Fig 5.8 Correlation with Spearman method between the features for human drivers

The correlation matrix for the automated vehicles shown in Fig 5.9 has a very similar trend as one for normal drivers, and apart from very slight differences in the correlation values, the matrix is very close to one of human drivers.

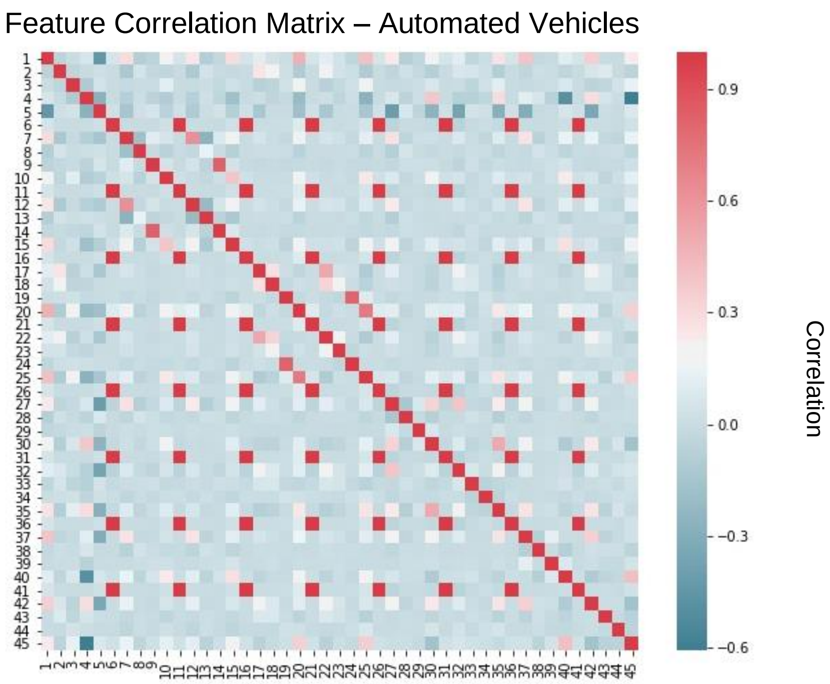


Fig 5.9 Correlation with Spearman method between the features for automated vehicles

A good strategy for reducing the correlation bias is to group the correlated features prior to model fitting and derive corresponding feature representatives as a summary of each group. Therefore, a PCA was conducted in which the features were grouped in 11 PCs. The importance of each PC is for human drivers and automated vehicles are shown in Fig 5.10.

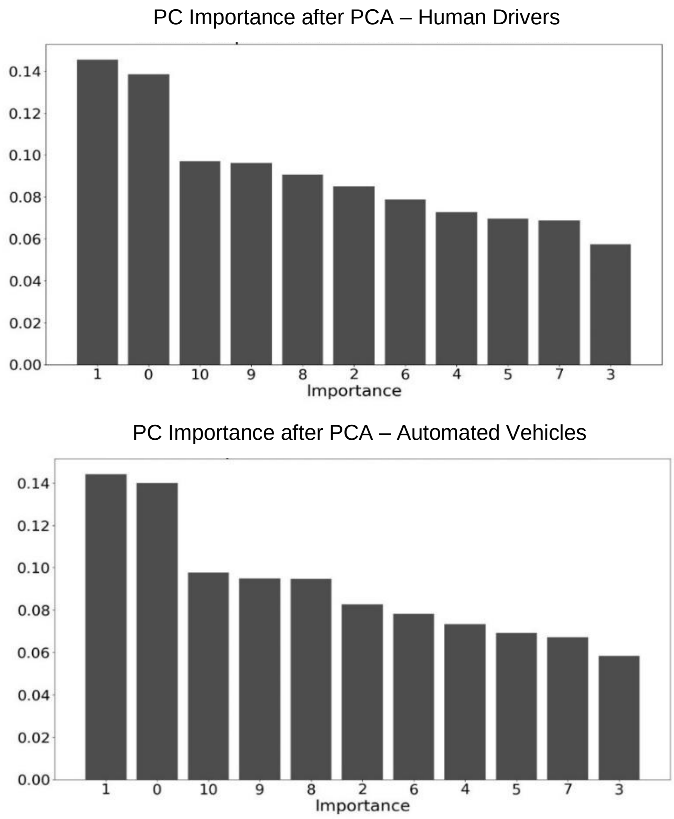


Fig 5.10 Principle components importance for human drivers and automated vehicles

5.1.2 Model Training and Testing

Based on the reviewing the classifiers in chapter 5.1.1, extra tree classifier and LR were chosen as a classifier for this dissertation and the models were trained in python with help of Scikit-learn [BUITINCK ET AL., 2013; PEDREGOSA ET AL., 2011] the details of which are explained below:

RF training:

A number of estimators is the number of trees in the RF classification. In the RF used in this dissertation, ten trees were chosen. In the RF, criterion is the function to measure the quality of a split. Supported criteria in scikit-learn are “gini” for the Gini impurity and “entropy” for the information gain. Gini impurity is a measure that determines how often a randomly chosen element from the set would be labeled incorrectly if it was randomly labeled based on the distribution of labels in the subset [BREIMAN, 1984]. Gini impurity for a set of items with J

classes, suppose $i \in \{1, 2, \ldots, J\}$, and let p_i be the fraction of items labeled with class i in the set.

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2$$

Information gain, on the other hand, is based on the notation of entropy which characterizes the impurity of an arbitrary set of examples [QUINLAN, 1993]. The expected information provided by a message with respect to the class membership can be expressed as:

$$H(T) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

Where $p_1, p_2, \ldots$ are fraction that add up to 1 and represent the percentage of each class present in the child node which results from a split in the tree. For this study, the entropy option has been picked.

LR training:

As the classification problem in this study is multi-class case, the training algorithm which uses One-vs-Rest scheme was deployed. In LR classification problem, one can specify the norm to be used in the penalization. *L1*-norm loss function is minimizing the sum of the absolute differences (S) between the target value (y_i) and the estimated values ($f(x_i)$):

$$S = \sum_{i=1}^{n} |y_i - f(x_i)|$$

L2-norm loss function is minimizing the sum of the square of the differences (S) between the target value (y_i) and the estimated values ($f(x_i)$):

$$S = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2$$

In order to solve the optimization problem for classification, saga solver [DEFAZIO ET AL., 2014] is a variant of sag solver [SCHMIDT ET AL., 2017] was used. Despite being fast in large datasets like sag solver, saga has the advantage of supporting the non-smooth penalty *L1* option. This makes this solver appropriate for sparse multinomial LR. Saga solver can only handle *L1* penalty. Therefore, in the *L1* norm used in the penalization. Moreover, parameter *C* in this model specifies the inverse of regularization strength and the smaller values specify stronger regularization. For this study, the regulation value was set to 0.9.

In the dataset, the number of lane changing and non-lane changing events is not equal, which means that the training dataset is imbalanced. In this case, the classifiers are typically more sensitive to detecting the majority class, and less sensitive to the minority class and the classification method might be biased toward always predicting the majority class [CHAWLA, 2005], which in this study is remaining on the lane. Therefore, there is a need to address and solve this problem. Two different methods were tested for this dissertation: Oversampling the minority class and undersampling the majority class.

A simple method for oversampling the minority class is to duplicate the entries. However, if this step is done before the cross-validation, it might result in overfitting and misleading the results. Oversampling the dataset with the SMOTE [CHAWLA ET AL., 2002] method, in which the new samples of the minority class are artificially generated using the nearest neighbors of these cases, and training the model resulted in a very slow learning process.

In undersampling, the majority class, n samples are chosen randomly from the majority class, where n represents the number of samples for the minority class and are used in the training phase. While different techniques have been proposed for undersampling, testing more advanced methods like undersampling specific samples that are further away from the decision boundary [JAPKOWICZ, 2000] and informative undersampling [LIU ET AL., 2009] on a small subset of data did not improve the accuracy significantly compared to selecting samples at random. Therefore, the random undersampling was used further, and only the results of random undersampling are presented here. After conducting the undersampling, 1807 events were in each class.

In ML methods, there is a need to test the stability of the model and make sure that the model does not overfit. Overfitting is the situation that the model would have a perfect prediction score but would fail to predict properly on the yet-unseen data [KOHAVI, 1995]. k-fold cross validation is performed as per the following steps:

1. Partition the original training data set into k mutually exclusive equal subsets of equal size. Each subset is called a fold. Let the folds be named as $f_1, f_2, \dots, f_k$.
2. For $i = 1$ to $i = k$
3. Keep the fold f_i as validation set and keep all the remaining $k-1$ folds in the cross-validation training set.
4. Train the ML model using the cross-validation training set and calculate the model accuracy by validating the predicted results against the validation set.
5. Estimate the model accuracy by averaging the accuracies derived in all the k cases of cross validation.

In k-fold cross validation method, all the entries in the original training data set are used for both training as well as validation. This method is believed to reduce the bias as most of the data is also being used for fitting and also reduced variance significantly as most of the data is used in the validation set. The authors in [HASTIE ET AL., 2009] suggest setting k=5 or k=10 and the performance of cross-validation is the average of the values computed in the loop.

5.1.3 Results

Once the classification is done, its performance should be measured. This section presents the results of the classifier for the prediction of lane change decisions of the drivers. A systematic analysis of 24 performance measures used in the complete spectrum of ML classification tasks was introduced by Solokova & Lapalme [2009]. Many of these measures are based on the confusion matrix. The confusion matrix is a performance measurement for ML classification problems. Each row of the matrix illustrates the instances in the actual class while each column represents the instances in the predicted class. Four important terminologies used for the confusion matrix are:

- True positive: the prediction was positive and it is true.
- True negative: the prediction was negative and it is true.
- False positive (type 1 error): the prediction was positive and it is false.
- False negative (type 2 error): the prediction was negative and it is false.

Some other performance measures derived from the confusion matrix are recall, precision, *F1*-score and Reciever Operating Characteristics (ROC) curve. In multi-class classification problems, the measures are calculated for each class and averaged over all the classes.

Recall estimates how many of the actual positives of the model capture through labeling it as positive (True Positive). Recall shall be the model metric for selecting the best model when there is a high cost associated with False Negative. Precision illustrates the percentage of the predicted positive cases that were true positives. In summary, recall expresses the ability to find all relevant instances in a dataset, and precision expresses the proportion of the data points our model says was relevant actually were relevant. Precision and recall are defined as:

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

The *F1* score is defined as the harmonic mean of precision and recall taking both metrics into account, and is the best metric when seeking the balance between precision and recall. *F1* score is calculated based on the following equation:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

The ROC curve illustrates how the recall versus precision relationship changes as the threshold for identifying a positive in the model varies. The threshold determines the value above which a data point is considered in the positive class. ROC curve plots the true positive rate on the y-axis and the false positive rate on the x-axis. The True Positive Rate (TPR) is the recall, and the False Positive Rate (FPR) is a false alarm probability. Both of these can be calculated from the confusion matrix:

$$true\ positive\ rate = \frac{true\ positives}{true\ positives + false\ negatives}$$

$$false\ positive\ rate = \frac{false\ positives}{false\ positives + true\ negatives}$$

Any classifier that has discrete outputs such as decision trees is designed to produce only a class decision, i.e., a decision for each testing sample, and hence it generates only one confusion matrix, which in turn corresponds to one point into the ROC space [THARWAT, 2018]. However, many methods were introduced for generating full ROC curve from a classifier instead of only a single point such as using class proportions [ZOU, 2002] or using some combinations of scoring and voting [FAWCETT, 2006]. In order to quantify a model's ROC curve, the total Area Under the Curve (AUC) can be calculated. This metric falls between 0 and 1 with a higher number indicating better classification performance.

RF results:

The total confusion matrices for both human drivers and automated vehicles are summarized in Tab. 5.4. This matrix considers the confusion matrices for each fold and sums their values to get the total confusion matrix.

	Human Drivers			Automated Vehicles		
Actual vs. Predicted	Lane Change to Left	No Lane Change	Lane Change to Right	Lane Change to Left	No Lane Change	Lane Change to Right
Lane Change to Left	1687	61	66	1761	21	25
No lane Change	51	1722	49	37	1752	23
Lane Change to Right	63	62	1714	42	38	1749

Tab. 5.4 Total confusion matrix for human drivers and automated vehicles with RF classifier

Tab. 5.5 illustrates the recall, precision, and *F1* for each class and the overall indicators, which are the average of each indicator for a single class for both human drivers and automated vehicles. Overall, the prediction accuracy of both models was good; however, the overall performance of the model for automated vehicles, due to the elimination of aggressive drivers whose behavior and actions are unpredictable, was better than the model for human drivers. Moreover, none of the models performed better in the prediction of one class compared to the other two, and the prediction accuracy of all the classes are similar.

Model	class	Recall	Precision	F1
Human drivers	Lane change to left	0.94	0.93	0.93
	No lane change	0.93	0.95	0.94
	Lane change to right	0.94	0.93	0.93
	Overall performance	0.94	0.94	0.94
Automated vehicles	Lane change to left	0.97	0.97	0.96
	No lane change	0.97	0.97	0.97
	Lane change to right	0.96	0.96	0.97
	Overall performance	0.97	0.97	0.97

Tab. 5.5 Classification performance for each class with RF classifier

LR results:

Total confusion matrix results for the classification with the LR method are illustrated in the matrixes shown in Tab. 5.6. As can be seen from the results, the classification capability of automated vehicles model is, as the RF classifier, higher than the model for human drivers. However, the general prediction capability of RF outstands the LR for this specific task.

	Human Drivers			Automated Vehicles		
Actual vs. Predicted	Lane Change to Left	No Lane Change	Actual vs. Predicted	Lane Change to Left	No Lane Change	Actual vs. Predicted
Lane Change to Left	1587	115	112	1520	93	96
No lane Change	126	1512	109	172	1482	168
Lane Change to Right	142	184	1514	130	124	1575

Tab. 5.6 Total confusion matrix for human drivers and automated vehicles with LR classifier

The indicators resulting from the confusion matrices are shown in Tab. 5.4 for both human drivers and automated vehicles are depicted in Tab. 5.5.

Model	class	Recall	Precision	F1
Human drivers	Lane change to left	0.87	0.86	0.85
	No lane change	0.86	0.85	0.83
	Lane change to right	0.82	0.85	0.87
	Overall performance	0.85	0.85	0.85
Automated vehicles	Lane change to left	0.89	0.87	0.86
	No lane change	0.88	0.88	0.88
	Lane change to right	0.86	0.87	0.88
	Overall performance	0.87	0.88	0.88

Tab. 5.7 Classification performance for each class with LR classifier

As mentioned earlier, one common way of assessing the usefulness of a binary classifier is the ROC curve as well as the area under the ROC curve (AUC), which shows the trade-off between the TPR and the FPR of a classifier for various choices of the probability threshold. A model with high differentiation ability will have high sensitivity and specificity, resulting in a

ROC curve, which goes close to the top left corner of the plot. Therefore, the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test. Moreover, AUC represents the degree or measure of separability to discriminate between the events, and the closer this value is to 1, the better the separation capability of the model. When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class. When AUC is approximately 0, the model is reciprocating the classes. In other words, the model is predicting negative class as a positive class and vice versa. In a multi-class model, ROC Curves for three classes are plotted using One-vs-All methodology. Fig 5.11 illustrates the ROC curve and AUC of the classifier for three classes as well as the micro- and macro-average curve for each. In micro-average method, individual true positives, false positives, and false negatives of the system for different sets are summed up, whereas in the macro-average method, the average of the precision and recall of the system on various sets are calculated. In other words, for extending the ROC curve to multi-class classification, the output is binarized, and a ROC curve is once drawn by considering each element of the label indicator matrix as a binary prediction (micro-averaging) and once drawn by giving equal weight to the classification of each label (macro-averaging).

As can be seen from the graph, the prediction capability of the lane change to the right is the lowest among the three classes and the AUC score for this curve is 0.71. This means that there is 71% chance that the model will be able to distinguish between lane change to right class and other classes. The prediction capability of the model for no lane change class and lane change to the left is almost the same.

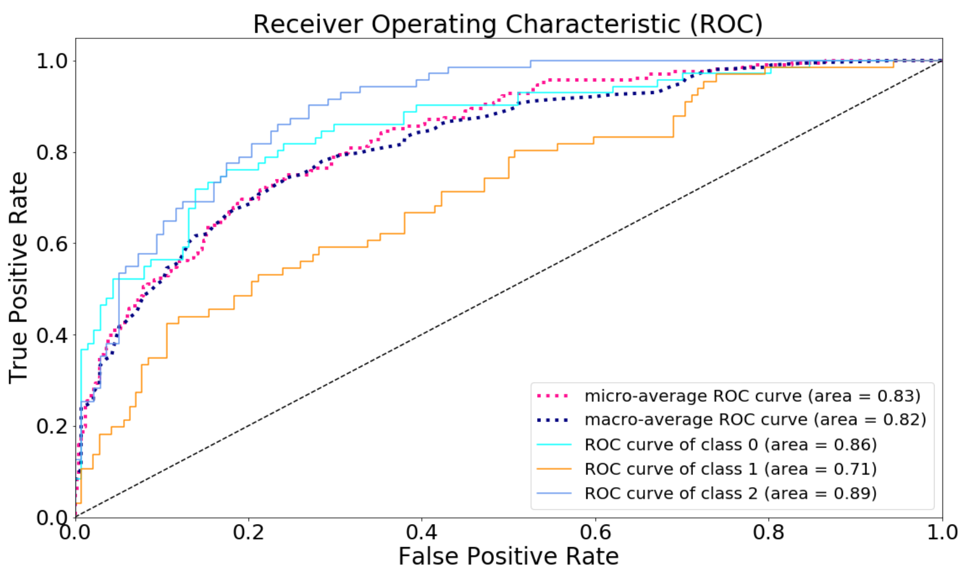


Fig 5.11 ROC curve

In summary, the prediction capability of RF for this specific task was better than LR. It is noteworthy to mention that although the model could predict the tactical lane changing behavior of the vehicles based on their surrounding environment (feature vector), the parameters such as motivation behind the lane changing and the desired speed of the driver were not existent in the database and therefore are not reflected in the model. However, by having a more comprehensive and detailed database, one can capture those variables in the driving behavior model developed by this modeling approach.

5.2 Modeling Lane Change Execution

This chapter focuses on the operational lane changing behavior and explains the maximum entropy IRL-based model which was deployed for learning the operational lane change behavior from demonstrations (individual vehicle trajectories).

5.2.1 Core Design, Objectives and Assumptions

As mentioned in chapter 2.4.2, many studies use RL defined by [SUTTON & BARTO, 1998] for learning behavior models [KONIDARIS & HAYES, 2005; INFANTES ET AL., 2011; HE ET AL., 2016; SHTEINGART & LOEWENSTEIN, 2014]. In RL, the agent is provided with a reward function, which provides feedback about the agent's performance whenever it executes an action in some state. The given reward function is used to obtain an optimal policy, one where the expected future reward is maximum. The problem with the RL is that in most tasks, there is no natural source for the reward function. Instead, it has to be carefully defined to accurately represent the task [SUTTON & BARTO, 1998]. The difficulty comes with trying to design a reward function that encourages the desired behaviors while being learnable. The common approach is to manually tweak the rewards of the RL agent until the desired behavior is observed. A better way of finding a well-fitting reward function for some objective might be to observe an expert agent (human) performing the task to automatically extract the respective rewards from these observations with IRL [KONIDARIS & HAYES, 2005].

In IRL, the setting is exactly the opposite of RL. In this method, the agent's policy or a history of behavior (which is in this study the individual vehicle trajectories) is given, and the goal is to learn reward function that explains the given behavior. The motivation for IRL is that it is often very difficult to manually specify a reward function for a task. In the real world, it is often not clear what the reward should be and there are rarely intrinsic reward functions. Manually designing and tweaking a reward function for a task is very challenging and comes with pitfalls and potential for errors [RATNER ET AL., 2018]. In many cases, it is easier to observe expert's behavior instead and let them demonstrate how to achieve the desired goal. Instead of simply imitating the expert's behavior, IRL tries to learn the underlying unknown reward function which is articulable as a linear combination of known features and the expert is trying to maximize [NG & RUSSEL, 2000]. In this case, IRL will provide a theoretical model that describes the objectives of the observed agents and can be deployed in many different scenarios. In this method, the strong assumption is that the experts acted optimally and always chose the best possible action. As described in a paper by Ng & Russel [2000], The reward function is a transferable description of the task. Even when the observed agent is very different from the target agent, the reward function contains all relevant information in a way that is usable for the other agent, which is very important when trying to transfer human tasks to robots or automated vehicles. Fig 5.12 explains the general methodology used for IRL:

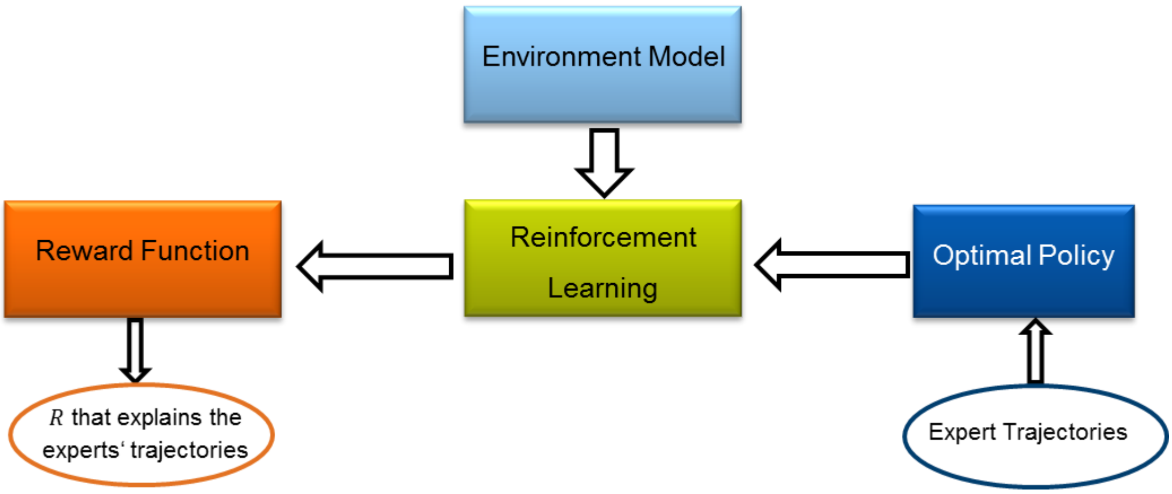


Fig 5.12 Inverse reinforcement learning framework

As defined by Ng & Russel [NG & RUSSEL, 2000], the model is informally characterized as follows:

Given:

1. measurements of an agent's behavior over time, in a variety of conditions
2. if needed, measurements of the agent's sensory inputs (environment states)
3. if available, a model of the environment (transition probabilities)

Determine: the reward function being optimized.

Notation and formulation:

The formulation of the model in this dissertation is inspired by the model formulation by [NG & RUSSEL, 2000]. This model uses the Markov decision process (MDP) which provides a mathematical framework for modeling the decision-making situations where outcomes are partly random and partly controlled by the decision maker. There are mainly three types of IRL problem formulation:

1. Finite-state MDP with known optimal policy.
2. Infinite-state MDP with known optimal policy.
3. Infinite-state MDP with unknown policy, but with given demonstrations.

Since in practical problems, mostly the experts' demonstrations are available rather than the explicit policy, the third method is the most widely-used one [LEE & POPOVIĆ, 2010].

MDP is defined via a state space, an action space, a function of transition probabilities between the states, a reward function, a discount factor for future reward and initial state distribution. The process is shown as a tuple $\langle S, A, R, T, \gamma, D \rangle$ with:

S: is the finite set of n environment states.

A: is the set of k actions.

R: is the reward function which maps each state to a real valued reward $S \rightarrow \mathbb{R}$.

$T: S \times A \times S \rightarrow [0,1]$ contains the transition probabilities, e.g. $T(s'|s, a)$ is the probability of landing in state s' after performing an action a in state s.

γ: is the discount factor for future rewards, which has the value between 0 and 1.

D: initial-state distribution, from which the start state s_0 is drawn.

In this process, the decision maker (agent) takes action and get a reward from its environment while the environment changes its state. The state transitions are probabilistic and depend only on the current state and the action taken by the agent. The reward obtained by the agent depends on the action taken, and on both the original and the new state of the environment. Assuming that the trajectory is an ordered list of states and actions which were sampled from some policy the trajectories can be defined as:

$\zeta = \langle (s_0, a_0), (s_1, a_1), \ldots \rangle$.

R is the real reward function (only known to the demonstrator/expert). A reward is gained when taking an action a in the state s and the environment changes to the successor state s'. The reward function of a trajectory is expressed as a linear combination of the feature counts.

It was assumed that there is some vector of features $\emptyset: S \times A \rightarrow [0,1]^k$ over states where the true reward function $R(s) = \omega . \emptyset(s)$. In driving behavior modeling, $\emptyset$ is a vector of features indicating the various desiderata in the driving that should be trade-off [ABBEEL & NG, 2004]. Incorporating feature construction into IRL has been a challenging task and most of the studies in this area focus on the parametrization of the reward function based on pre-determined features and expressed as a weighted linear combination of hand-selected [ZIEBART ET AL., 2008; ABBEEL & NG, 2004; LOPES ET AL., 2009; RATLIFF ET AL., 2006]. The difficulty is that many of the features may have important logical relationships that make it impossible to represent the reward function as their linear combination [LEVINE ET AL., 2010]. For this dissertation, the feature vector is the linear combination of the features determined in feature importance defined in chapter 5.1. However, all the features which are relevant for the decision-making during lane changing are not necessarily relevant for the execution of lane changing. When trying to imitate the behavior of the expert, utilizing all the available features in the environment might not be necessary. Using all features may lead to overfitting, especially in the lack of a lot of training data [Kim, Beomjoon and Pineau, Joelle, 2016]. To tackle this problem, a regularization technique was implemented within the IRL. This model selection technique prevents overfitting by eliminating unnecessary features using a penalty term. The popular choice for the penalty term for model selection is the L_1 norm [Robert Tibshirani, 1994]. The optimization target then becomes:

$$\omega^* = argmax \left\{L(\omega) + \beta\left||\omega|\right|_{\mathbf{1}}\right\}$$

The vector ω specifies the relative weighing between the desiderata relevant for the execution of lane changing and β is the regularization parameter; higher β means fewer features are considered, and vice versa. Following the method used by Kim et al. [2016], modification of Newton-Raphson method [FAN & LI, 2001] was used in this dissertation. On the other hand, a policy π maps the states to the actions. If the policy is stationary deterministic, it outputs exactly one deterministic action for each state: $\pi: S \to A$. If the policy is stochastic, it will contain probabilities of choosing each action for each state: $\pi: S \times A \to [0,1]$.

$V^{\pi}(s)$ is the value function for the policy π evaluated at any state s under policy π. By Bellman equation [1952], this value function would turn into:

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} T\left(s'\middle|s, \pi(s)\right) V^{\pi}(s')$$

$Q^{\pi}(s, a)$ is the value of state-action pairs under the policy π and can be derived from the following formula:

$$Q^{\pi}(s, a) = R(s) + \gamma \sum_{s' \in S} T(s'|s, a)V^{\pi}(s')$$

It is assumed that the agent attempts to optimize a function that linearly maps the features of each state, to a state reward value representing the agent's utility for visiting that state. The optimal policy is retrieved if and only if, for all $s \in S$:

$$\pi(s) \in arg\max_{a \in A} Q^{\pi}(s, a)$$

An alternate method had been proposed by Abbeel & Ng [2004], which was is based on the matching feature expectations between an observed policy and a learner's behavior. However, both normal IRL and the alternate method have ambiguity in the matching of feature counts and can be optimal for many reward functions and many policies lead to same feature counts. Therefore, Ziebart et. al [2008] proposed the MaxEnt IRL to overcome the ambiguity, in which they employ the principle of maximum entropy developed by Jaynes [1957]. As shown in Ziebart's work, principal benefits of the Maximum Entropy paradigm include the ability to handle expert suboptimality as well as stochasticity by operating on the distribution over possible trajectories [WULFMEIER ET AL., 2015].

For the stochastic policies, the probability of taking an action is weighted by the expected exponentiated rewards of all the trajectories that begin with that action. Maximizing the entropy of the distribution over the trajectory subject to the feature constraints imposed from observed data suggests that the model maximizes the likelihood of the observed data under the maximum entropy distribution over the demonstrations. In this model, optima can be obtained using gradient-based optimization methods where the gradient is the difference between expected empirical feature counts and the learner's expected feature counts [ZIEBART, 2010; ZIEBART ET AL., 2008]. In other words, gradient-based optimization minimizes the norm of the distance between the resulting feature values and the feature values of the behavior that was to be imitated. At the maxima, the feature expectations match and it guarantees that the learner performs equivalently to the agent's demonstrated behavior regardless of the actual reward weights the agent is attempting to optimize [ABBEEL & NG, 2004]. Value iteration is one from Sutton and Barto [1998] and can be formulated as shown in Algorithm 5.2:

Algorithm 5.2 Approximation of value iteration algorithm

1: $V(s) = -\infty$
2: $\boldsymbol{repeat}$
3: $V_t = V;\ V\left(s_{goal}\right) = 0$
4: $Q(s,a) = r(s,a) + E_{T\left(s,a,s'\right)}\left[V\left(s'\right)\right]$
5: $V = softmax_a Q_i(s,a)$
6: $\boldsymbol{until}\ \max_s\,(V(s) - V_t(s)) < \varepsilon$
7: $\pi(a|s) = e^{Q(s,a)-V(s)}$

The framework used in this dissertation for the implementation of IRL for learning the operational behavior of lane changing is illustrated in Fig 5.13. As can be seen in the diagram, first the features are extracted from the observation and imported into the reward function. With the assumption of the optimality of experts' trajectories, IRL learns the underlying reward function from the trajectories. This process is done by learning the weight of each feature from the data. Thereafter, the reward function is used to make predictions in the future.

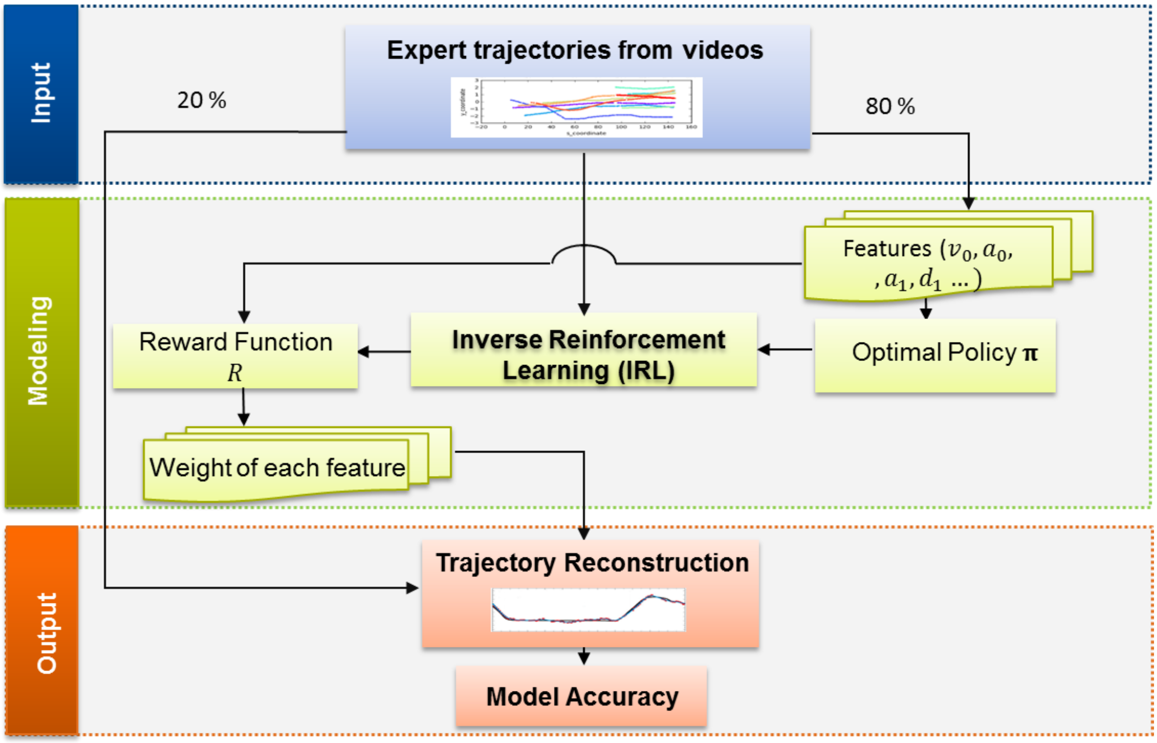


Fig 5.13 Maximum Entropy IRL framework

5.2.2 Model Training and Testing

In order to train the model, first, the grid world should be created. A grid world is an idealization of a robot in an environment. At each timestep, the agent is at some location and can move to neighboring locations, collecting rewards. The majority of existing approaches for robot mapping assume a static world, an assumption that does not hold in most practical application domains [Meyer-Delius et al., 2011]. In a study by Tanzmeister et al. [2014], an approach was presented that estimates a uniform, low-level, grid-based world model including dynamic and static objects, their uncertainties, as well as their velocities. Previtali et al. [2015] proposed a representation of the monitored environment based on a set of non-uniform grids, thus allowing for an effective and efficient representation of the monitored environment. The grid is periodically updated based on the information provided and high-density locations are described richly by increasing the granularity of the grid mapping around that location, whereas parts of the environment having fewer interactions are described sparsely. Their approach is linearly scaled with respect to the size of the environment as well as the computational resources required for solving the optimization problem. Although their model showed effective and computationally efficient results, their approach was tested only for the autonomously planning robots and pedestrians, and its plausibility for driving behavior is not clear.

For this dissertation, a uniform grid is implemented and since the geometry of the road, specifically the number of lanes, varies within the study area, two offline grids were predefined in the model. The grid size is a trade-off between the computation power and accuracy of the model. The two grids defined in this study area are matrices of 28x40 and 2x40 as the border region. Each cell in the grid captures the movement of up to 0.5 m in the lateral direction and 1 m in the longitudinal direction. The 4-Lane Markov Grid are matrices of 56x40 and 2x40 as the border region. Each cell in the grid captures the movement to 0.5 m in the lateral direction and 1 m in the longitudinal direction. For the environment dynamics, a stochastic grid world was defined. In order to incorporate the dynamic nature of the surrounding environment, the grid of each vehicle is rolling every 120 ms as the ego vehicle moves within the study area and the probability of moving to the existing cells in the Markov grid is calculated. However, if a dynamic obstacle lies outside of the abovementioned grids, the features such as relative distance and relative speed are still considered in the reward function. The reward dedicated to the static obstacles such as study area borders and dynamic obstacles such as moving cars are -1 and -0.5, respectively, to reduce the probability of an agent moving to a preoccupied cell.

The discount factor of MDP, which is denoted as γ, is for penalizing future awards and regulating how far ahead in time the algorithm looks for penalizing future rewards. The values close to one prioritize rewards in the distant future, whereas a discount factor closer to zero indicates that only rewards in the immediate future are being considered, implying a shallow lookahead. As discussed in a study by François-Lavet et al. [2015], the discount factor may have an impact on the stability and convergence of deep reinforcement learning algorithms. Having a discount factor smaller than one assures that the total expected value converges to a finite solution. If no discounting factor is defined, the total expected value explodes and would not converge [CARMON, 2007]. In other words, if it were not a discounted problem (γ =1), all policies that have to acquire on average a positive reward at each time step would sum up to infinity. In this study, after testing the values between 0.5 and 1, the discount factor is set to 0.8.

Another parameter that was set in the training is the learning rate. Learning rate is a hyper-parameter that can control how much the weights can be adjusted with respect to the loss gradients. Like any other hyper-parameter, learning rate needs optimization which is the problem of optimizing a loss function over a graph-structured configuration space [BERGSTRA ET AL., 2011]. Choosing a low value for learning rate assures that the model will not miss any local minima, however, the model will take a long time to converge while too large learning rate will make the training algorithm diverge [BENGIO, 2012]. Moreover, the learning rate can affect the training time by order of magnitude. In order to find the right learning rate, the model was trained with a subset of data with 48 different learning rates, from 0.000001 to 100 at logarithmic intervals. Based on the result of the test, the gradient descent learning rate for this dissertation was set to 0.01.

The number of trajectories that illustrated the number of sampled point from trajectories was set to 200 points. As the problem is solved by the gradient descent method, the algorithm is iterative meaning that there is a need to get the results multiple times the avoid underfitting and overfitting and get the optimum solution. The gradient descent iteration was set to 20 meaning that the weights updated with 20 epochs.

The actions are the set of possible alternative choices the agent can choose [RABINER & JUANG, 1986]. With the assumption that the actions are stochastic, there is a probability distribution over the resulting states given the respective action and the state. The main problem of solving an MDP is to find the optimal action to take in each particular state of the world [LITTMAN ET AL., 1995]. As this model is dealing with the vehicles driving on a freeway-like road segment, any abrupt movements of the vehicles and driving backward is not allowed. Therefore, the agent can choose the following four actions shown in Fig 5.14 to move around in the grid. However, the determination of the action alone cannot replicate the movement of vehicle realistically because the speed by which the vehicles perform an action is not predicted. The actions can illustrate the direction of the movement; however, they are not capable of predicting the speed by which the vehicle moves properly. To overcome this issue and to incorporate the speed prediction into the model, regression analysis with RF was employed.

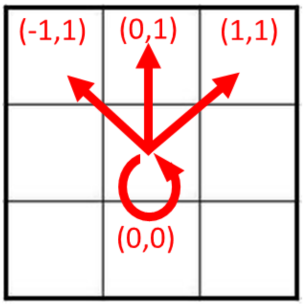


Fig 5.14 Actions in the Markov grid

Regression analysis estimates the functional relationships between the independent (predictor or explanatory) variables and a dependent (response or outcome) variable [DRAPER & SMITH, 2014]. The dependent variable is the main factor that the model tries to understand or predict and independent variables are the factors that the model hypothesizes have an impact on the dependent variable [ABRAHAM & LEDOLTER, 2006]. Linear regression is a classical parametric method that requires explicit modeling of nonlinearities and interactions, when necessary, and is believed to be robust if the number of observations is distinctly larger than the number of variables [BERK, 2004]. RFs, on the other hand, are non-parametric and allow learning the nonlinearities and interactions from the data without the explicit need model them [GRÖMPING, 2009].

In this dissertation, a RF regressor was embedded in the IRL model. A RF is an ensemble method that fits several decision trees on various subsamples of the dataset and uses averaging to increase the accuracy of prediction and control overfitting [BREIMAN, 2001].

Regression trees are used as base learners in the RF regression algorithm. A regression tree is built by recursively portioning the sample into more homogenous groups called nodes, down to the terminal nodes. Each split is based on the values of one variable and is selected according to the splitting criterion [GRÖMPING, 2009]. After selecting the number of trees in the forest, each regression tree is grown on a separate bootstrap sample derived from the initial training data. Each node in a tree represents a binary test against the selected predictor variable. The sub-sample size is always the same as the original input sample size, however, the samples are drawn with replacement by default [BUITINCK ET AL., 2013]. Moreover, when splitting node during the construction of the tree, the chosen split is no longer the best split among all the features. Alternatively, the picked split is the best among the random subset of the features. Although during this process, as the result of randomness, the bias might slightly increase, due to averaging the variance decreases usually more than the compensating for the slight increase in bias, hence resulting in an overall better model [BUITINCK ET AL., 2013].

The RF regressor in this study takes the speed of the vehicle in the previous time step, lane id, relative speed and distance to the preceding vehicle as input to predict the speed of the vehicle for each time step. Although it has been discussed that the RF does not require an external cross-validation procedure to estimate the model accuracy as the bagging during training preserves the data from overfitting [SIROKY, 2009], there is no guarantee that all the instances will be used at least once. Therefore, in this study, a 10-fold cross validation discussed in chapter 5.1.2 was performed. The last step of solving a ML problem is to evaluate the performance. For regression, problems the metrics used to evaluate an algorithm are often mean absolute error, mean squared error, and root mean squared error. Mean absolute error was calculated as the function to measure the quality of a split, which can be calculated with the following formula for n observation points:

$$MAE = \frac{\sum_{i=1}^{n} |predicted\ speed_i - actual\ speed_i|}{n}$$

The model was trained on the NVIDIA Quadro P4000 Graphic Processing Unit (GPU). The model was trained using two datasets once training with the whole dataset to replicate the result for the human drivers and once with the dataset excluding the aggressive drivers to account for the automated vehicles.

5.2.3 Results

The results discussed here include two types of results: first, the result of the regression analysis embedded in the IRL algorithm and second: the result of the trajectory prediction of the whole IRL Model.

Speed Prediction result:

The estimated speed comes from the predictions made on the data withheld from training in each round of cross-validation. MAE, Mean Squared Error (MSE) and Root Mean Squared

Error (RMSE) were deployed as the indicator for evaluating the performance of the regressor. For this model, the number of estimators, which is the number of trees in the forest, was tested with 5, 10, 20, 50, 100 and 200 trees. First of all, the feature importance was calculated to estimate to what extent the input samples' features contribute to speed. As can be seen from Fig 5.15, increasing the number of trees did not have a substantial influence on the importance of each feature. Moreover, for both normal drivers and automated vehicles, the relative distance to the preceding vehicle contributes the most to the speed prediction. This is followed by the speed of the vehicles in the previous timestep. The third influencing factor on the speed of the ego vehicle is the relative speed and the feature with the least important was the ID of the lane.

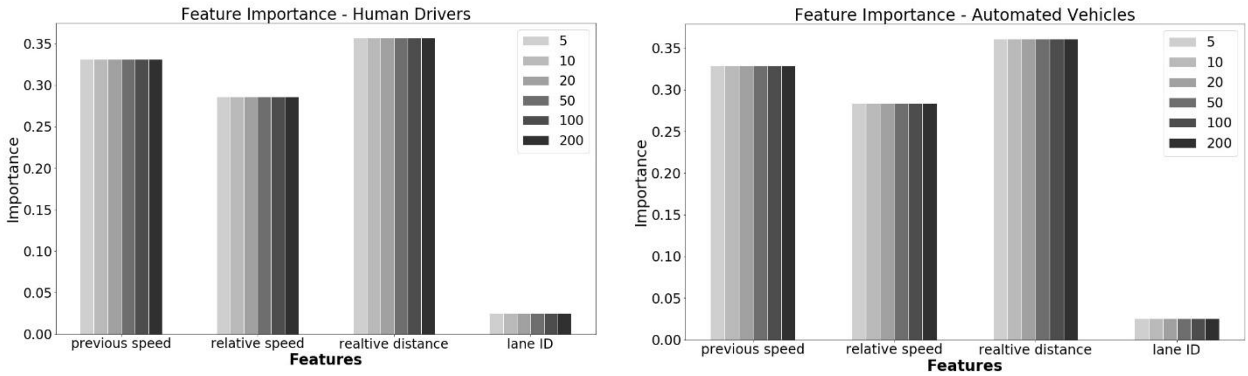


Fig 5.15 Feature importance for human drivers and automated vehicles with various number of trees

Fig 5.16 illustrates the average of regression results for all ten folds. In this plot, the horizontal axis represents the number of estimators, and the vertical axis illustrates the average MAE, MSE and RMSE of the regressor in degrees. The bag error rate has shown that after 20 trees, the error levels out and that an increase in the number of trees does not boost the prediction ability of the regressor.

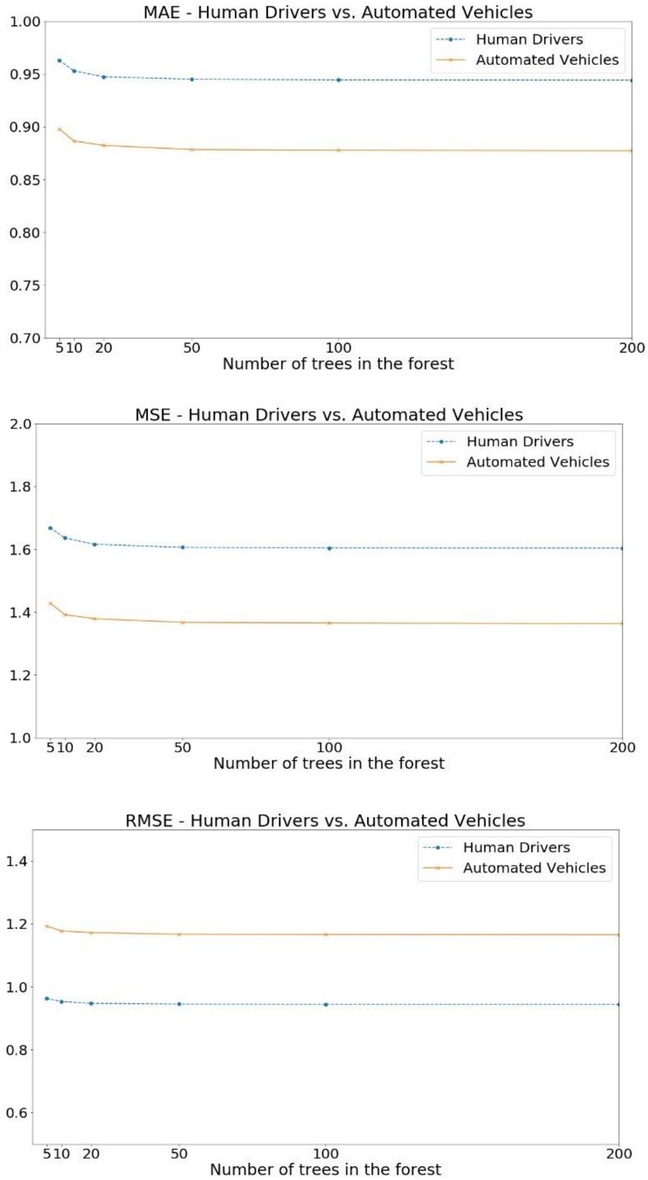


Fig 5.16 MAE, MSE and RMSE result of the RF regressor for speed prediction of human drivers and automated vehicles.

In order to make the prediction for the speed of automated vehicles, the aggressive drivers, which were identified with the means of clustering the driving behavior (Chapter 4) were filtered out from the dataset, and the RF regressor was trained with the data. As can be seen from Fig 5.16, the prediction accuracy of the regressor for human drivers is not considerably different than the one for human drivers. Moreover, increasing the number of trees more than 20 did not lead to improvement in the prediction capability of the system. Based on the result of the speed prediction, the RF regressor with 20 trees was deployed in the IRL model in order to predict the speed of the vehicle in each time step.

Trajectory prediction result

In order to analyze the difference between the observed behavior and the modeled most probable trajectory, as suggested by Kuderer et al. [2012; 2015], the norm of the discrepancy between the behavior predicted by the model and the recorded trajectories averaged over all ten folds. The following metrics were deployed for evaluating the prediction capability of the model and computing the error:

1. Total displacement error is the sum of the Euclidean distance over all the estimated points at each time step in the predicted trajectory and true trajectory is calculated. Assuming that the two curves P and Q represent the observations and the predictions respectively and N is the number of points, this indicator can be calculated based on:
$$\sum_{i=1}^{N} \sqrt{\left(x_i^P - x_i^Q\right)^2 + \left(y_i^P - y_i^Q\right)^2}$$
2. Average displacement error is similar to the metric used in a study by Pellegrini [2009]. This indicator computes the mean Euclidean distance over all estimated points at each time-step in the predicted trajectory and true trajectory.
$$\frac{\sum_{i=1}^{N} \sqrt{\left(x_i^P - x_i^Q\right)^2 + \left(y_i^P - y_i^Q\right)^2}}{N}$$
3. Fréchet distance is the measure of similarity between the curves. This method developed by M. Maurice Fréchet [1906] is to measure how much two given curves resemble each other. As illustrated in Fig 5.17, supposing that a man is walking his dog and he is constrained to walk a curve (blue curve) and his dog on another curve (red curve) and both the man and dog are allowed to control their speed independently, but they are not allowed to go backward, then the Fréchet distance of the curves is the minimal length of a leash that is necessary. Mathematically the Fréchet distance between the two curves is defined as:

   $$\delta_F(P, Q) = \min\{\max d(\, P(\alpha(t)), Q\,(\beta(t)))\}$$

   $$\alpha[0,1] \to [0, N]\; t \in [0,1]$$

   $$\beta[0,1] \to [0, M]$$

   where $\alpha(t)$ and $\beta(t)$ range over continuous and increasing functions with $\alpha(0) = 0$, $\alpha(1) = N$, $\beta(0) = 0$ and $\beta(1) = M$ only. This equation reads like: for every possible function $\alpha(t)$ and $\beta(t)$, find the largest distance between the man and its dog as they walk along their respective path; finally, keep the smallest distance found among these maximum distances. For this study the discrete Fréchet function developed by Eiter and Mannila [1994] was deployed.

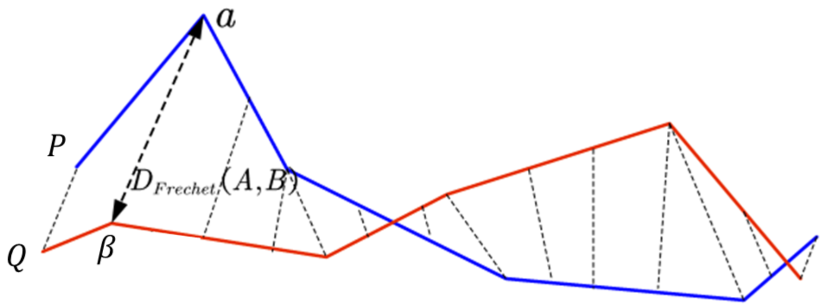


Fig 5.17 Fréchet distance

In order to make the predictions, first, the subset of the trajectory is analyzed and plotted for better illustrations. For each scenario, the corresponding reward function at the specific situation is plotted. As can be seen from the graphs in Fig 5.18, the ego vehicle takes the path in which it receives the highest award. In these graphs, the cells with very low reward are those occupied by other vehicles.

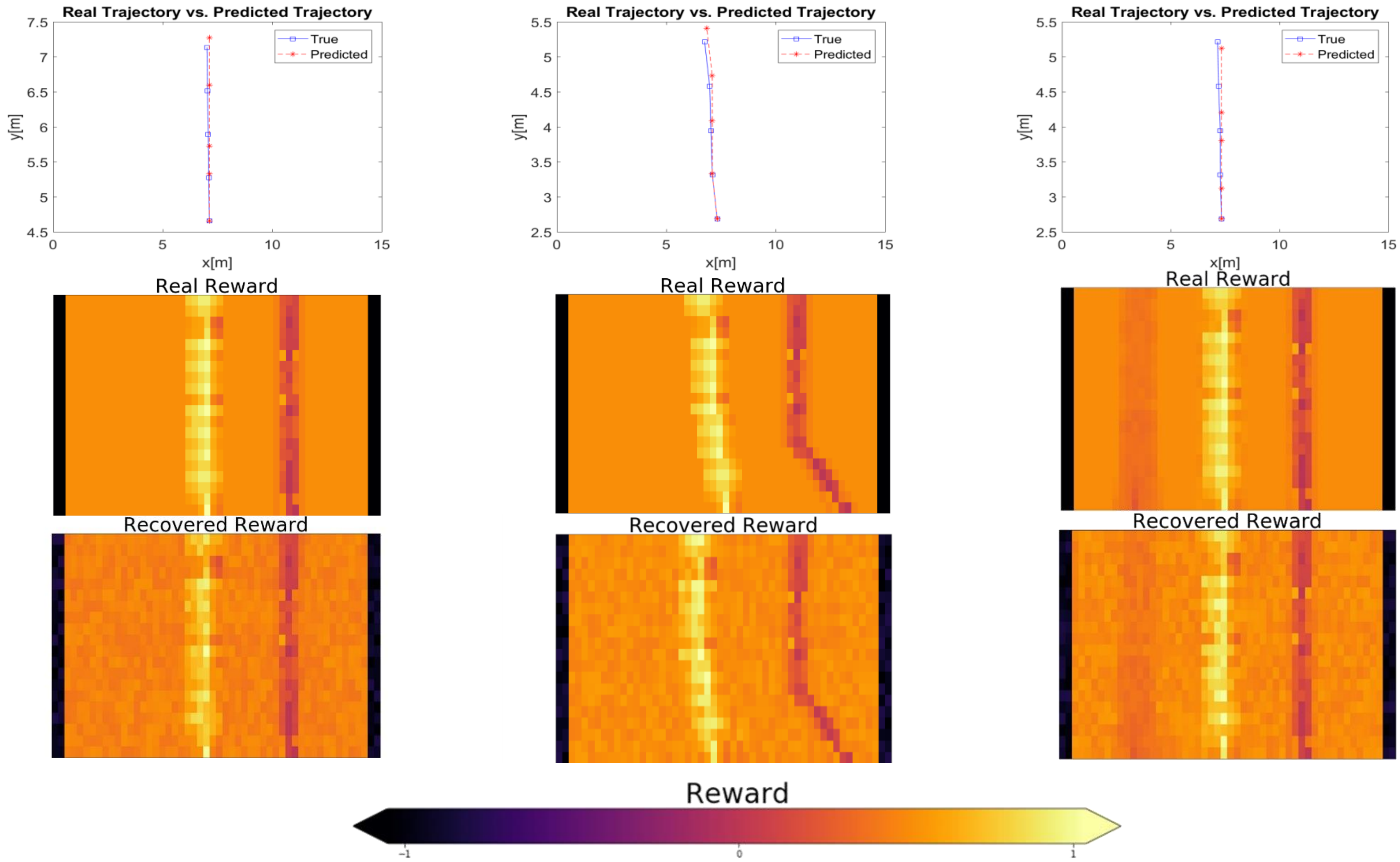


Fig 5.18 Examples of real and recovered reward function and predicted trajectories

During the testing process, in some specific traffic scenarios, the model failed to predict the vehicle movement correctly. Fig 5.19 illustrates two examples of anomalous scenarios. As can be seen from plot a), the driver decided to move to the left lane as the scene was empty, while the model predicted the highest probability for the action of going straight in an empty scene. The Fréchet Distance for this particular prediction was 3.4 m. The predicted trajectory follows a straight line, whereas the drives' movements were very stochastic. Due to the fact that the observed data does not represent the motivation of the drivers as well as the rareness of this situation in the dataset, the model could not learn the stochasticity of the driver's behavior in this situation.

The situation depicted in plot b) shows a vehicle entering the scene from the ramp where no other vehicles were driving on the ramp. This situation was infrequent in the dataset, and therefore in the prediction, the ego vehicle entered the scene with diagonal movements using both lanes, and when it reached the borders of the road, it was found to be stuck at a particular position. Analysis of the data shows that not many training samples encountered this particular scenario; hence, the action predicted was to stay at the current position. The Fréchet Distance for this particular prediction is 4.68 m.

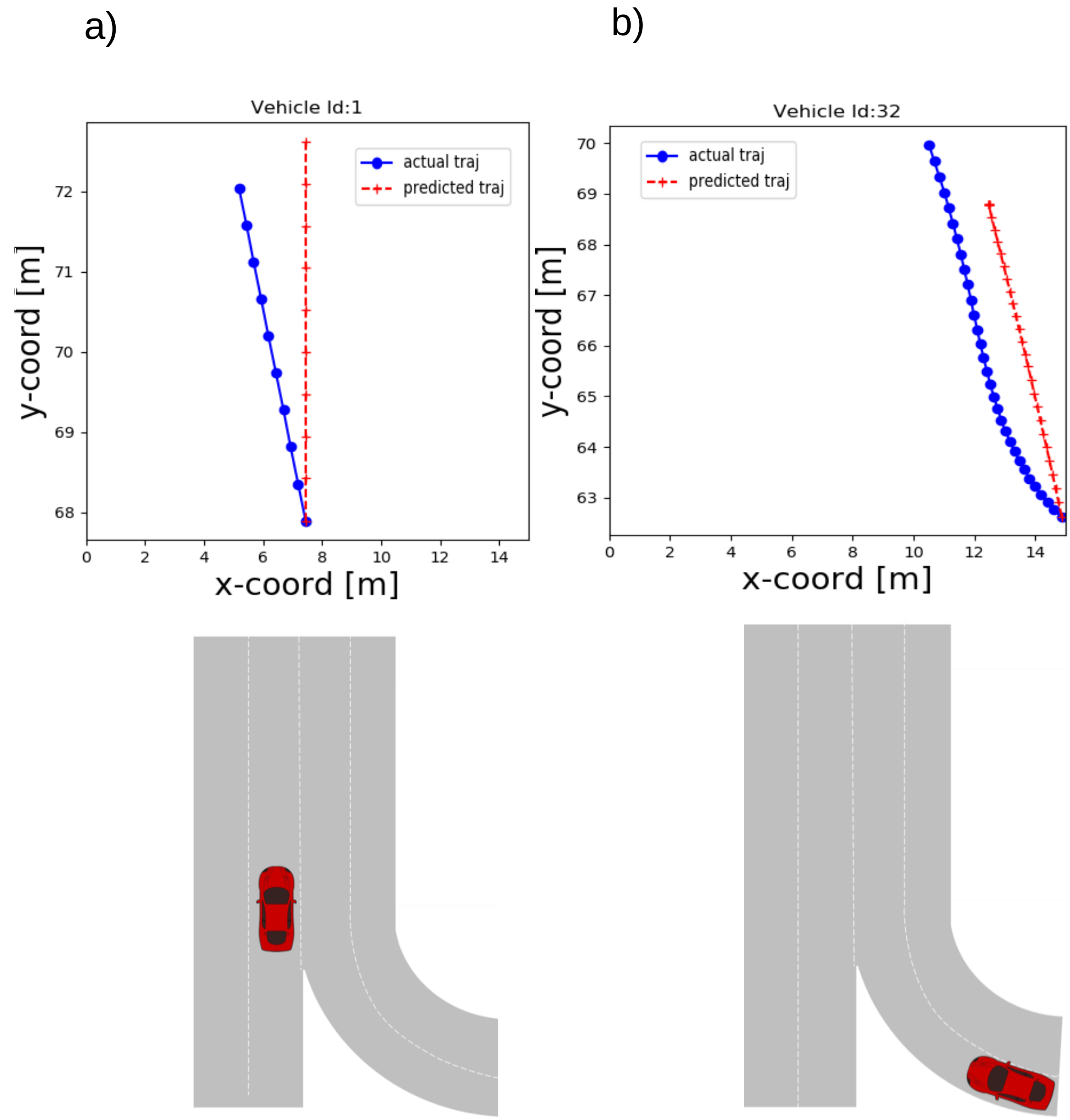
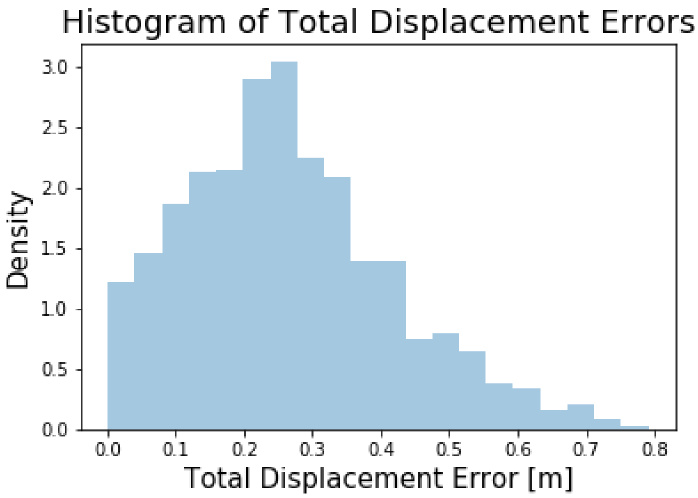
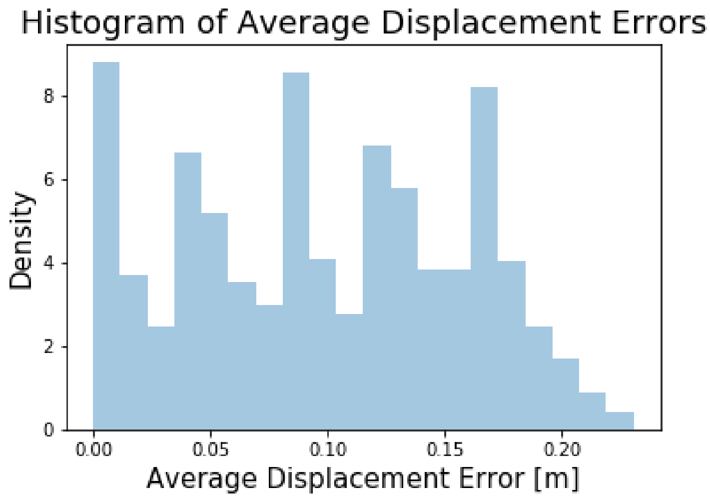
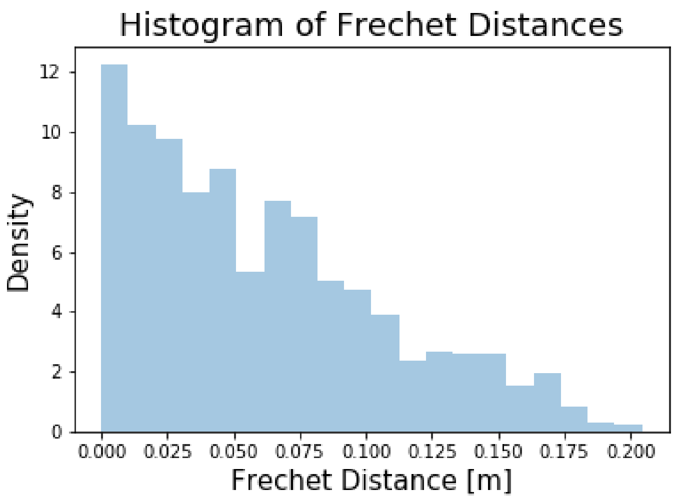


Fig 5.19 Examples of anomalous scenarios

Tab. 5.8 summarizes the results of the model prediction over all the trajectories, excluding the anomalous scenarios. As the table illustrates, the average total displacement error and average displacement error of all trajectories was 0.26 m of 0.098 m, respectively. It is noteworthy to mention that by increasing the number of cells in the Markov grid and reducing the grid size, which results in higher computation time, better results can be achieved.

Tab. 5.8 Overview of the trajectory prediction results

Indicator	Average [m]	Histogram
Total Displacement Error	0.265	
Average Displacement Error	0.098	
Fréchet Distance	0.061	

5.3 Summary

This chapter discussed the approach used for modeling the operational behavior of lane changing of drivers from the observed trajectories in the study area. For modeling the highly stochastic behavior of drivers, an IRL framework was deployed to solve sequential planning and decision-making problems in the stochastic environment under study. The key notion in IRL is that the driver act to optimize a reward function defined as a linear function of features. The IRL problem is then reduced to recovering the weights of the features in the reward function that induces the demonstrated behavior. However, IRL might be ambiguous, meaning that each policy can be optimal for many reward functions and many policies lead to the same feature counts. To resolve the ambiguity, the principle of maximum entropy was used. Moreover, a RF regression model was integrated into the model for making predictions about the speed of the ego vehicle.

The results showed that the model could recover the real reward of vehicles from the demonstrated trajectories and as a result, the model has the capability of making predictions about the operational behavior of drivers in most situations. The resulting variation the actual trajectory and predicted one in specific situations stems from the discrete nature of Markov grid and the rareness of those situations, which can be tackled by gathering more observations and reducing the resolution of Markov grid.

6. Simulation Experiment

Microscopic simulation software is an indispensable powerful, and important tool in traffic analysis for modelers and transport planners. A well-calibrated microscopic simulation model is able to replicate the statistical distribution of detailed traffic variables and emulate the flow of individual vehicles through a road network [LIMA AZEVEDO, 2014; BARCELO ET AL., 2005]. Moreover, microscopic simulation tools provide a cost-efficient and safe mean to test emerging technologies such as automated vehicles; however, for achieving this goal, the microscopic traffic simulation should realistically replicate the microscopic level driving behavior and consider the detailed interactions between the vehicles.

In this chapter, the process and results of using the existing microscopic traffic simulation software SUMO [KRAJZEWICZ ET AL., 2012] to analyze the feasibility of the developed model are explained. The main reason for choosing SUMO as the simulation software was its API that treats the SUMO simulation as a server and allows users to gain information from a traffic simulation or modify the simulation [WEGENER ET AL., 2008]. This feature makes SUMO an attractive tool for reinforcement learning and Inverse reinforcement learning studies. A computational framework for deploying reinforcement learning for autonomous vehicles and its integration in microscopic traffic simulation SUMO was proposed in *Flow* project [KHETERPAL ET AL., 2018] and Genders and Razavi [2016] applied deep reinforcement learning to build a truly adaptive traffic signal control agent SUMO. As the focus of this dissertation is on lane changing, this model was studied more precisely, and the details are described in the following chapter.

6.1 Simulation Framework

SUMO provides the possibility to import the network from OpenStreetMaps as an OSM-data, which contains an extract of the OpenStreetMap database, is saved in an XML structure and always has WGS84 geo coordinates [KRAJZEWICZ ET AL., 2002]. This coordinate system converts automatically to the UTM coordinate system, which is compatible with the observation data, when using the netconvertor and creating the SUMO network file. A SUMO network file describes the traffic-related part of a map, the roads and intersections the simulated vehicles run along or across and is encoded as XML files. However, the data from OpenStreetMap is not simulation-ready and should be further enhanced. Not only the geometry of the network was improved, but also the speed limit and type of the street. Fig 6.1 provides an overview of the steps to create the simulation network for SUMO.

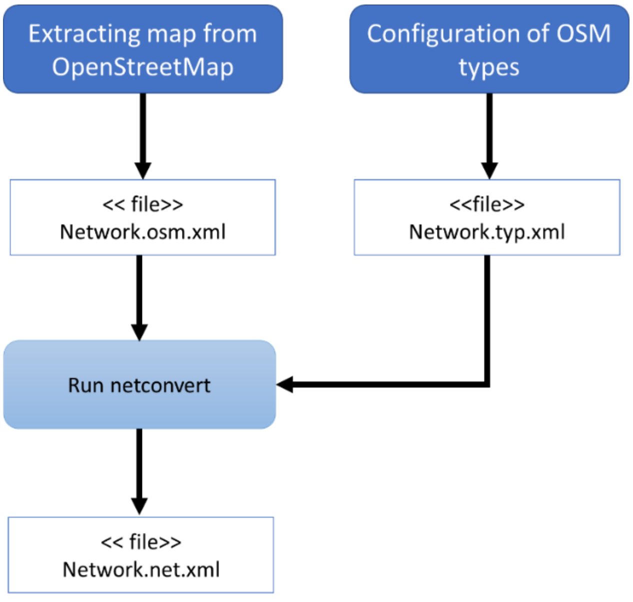


Fig 6.1 Overview of the steps to create the SUMO network file from OpenStreetMaps

After converting the network with the SUMO netconvertor, improving the geometry and attributes the network is simulation-ready. Fig 6.2 illustrates the simulation ready network plotted on the background map.

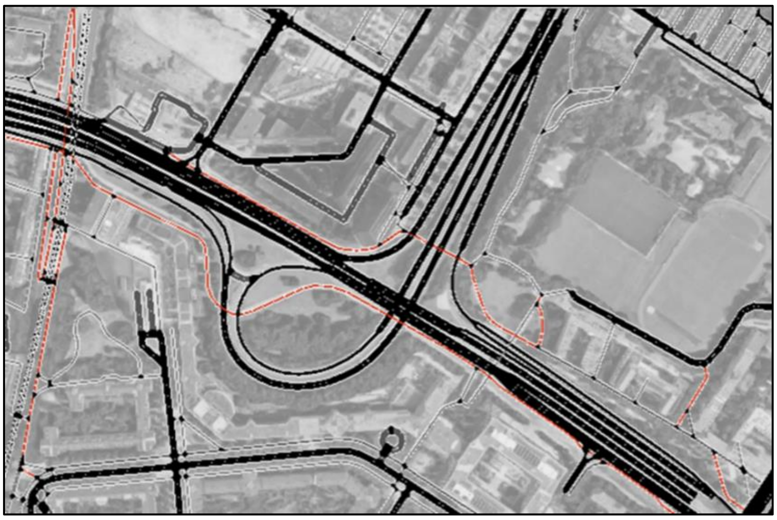


Fig 6.2 Simulation network in SUMO (Background image is from Google maps, 2018)

In order to check the plausibility of the model for unseen data with the simulation tool, observation number three, which was left aside and not used during the training process, was used. The duration of this video is 30 minutes; therefore, the simulation was conducted for 30 minutes with 15 minutes warmup time. Fig 6.3 summarizes the speed, longitudinal and lateral acceleration of each vehicle type in the dataset used for calibration and validation of the model. As can be seen from the speed graphs, the data was chosen in a way that the building up of congestion (low speeds) and dissolving of the congestion are included in the dataset.

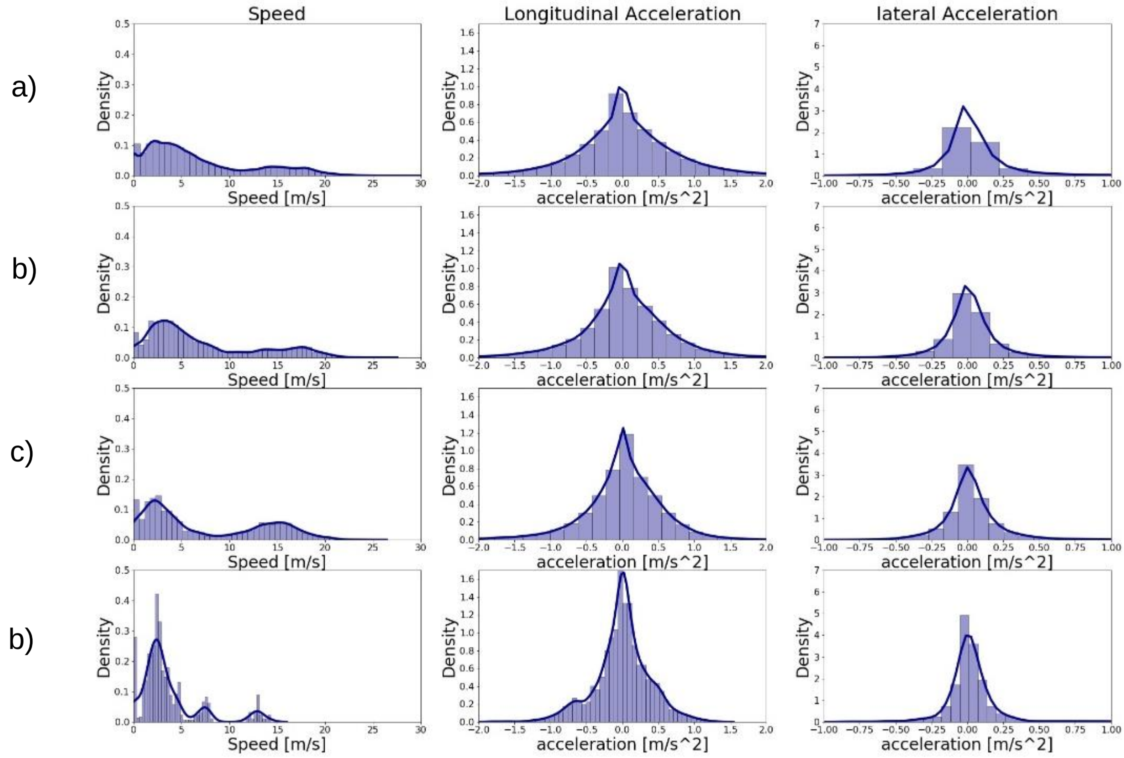


Fig 6.3 Calibration dataset histogram and density plots of speed, longitudinal acceleration and lateral acceleration for a) cars, b) medium vehicles, c) heavy vehicles and d) buses in the dataset

In order to check the safety of the situation, TTCs of the vehicles were calculated, and the normalized histogram of all TTCs was analyzed to display relative frequencies. It then shows the proportion of cases that fall into each of several categories, with the sum of the heights equaling 1. Fig 6.4 illustrates the histogram of TTCs below 10 seconds. As can be seen from the graph, the diagram is skewed to the left, and the traffic situations with very low TTCs were not frequent events in the dataset.

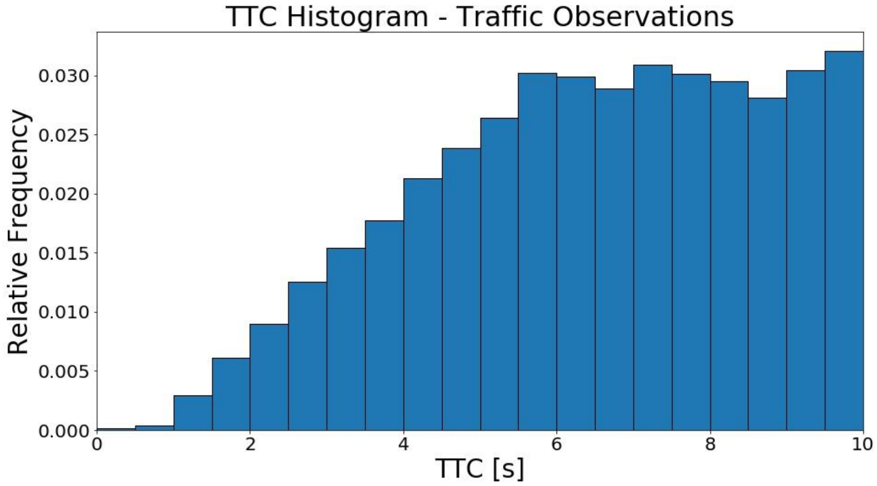


Fig 6.4 Histograms of TTC values of normal drivers in traffic observation

Analyzing the distribution of the lanes is also of high importance. As it can be seen in Fig 6.5, in the major stream, where two streams come together, the third lane was the most frequently used lane, whereas the rightmost lane was not popular among the drivers.

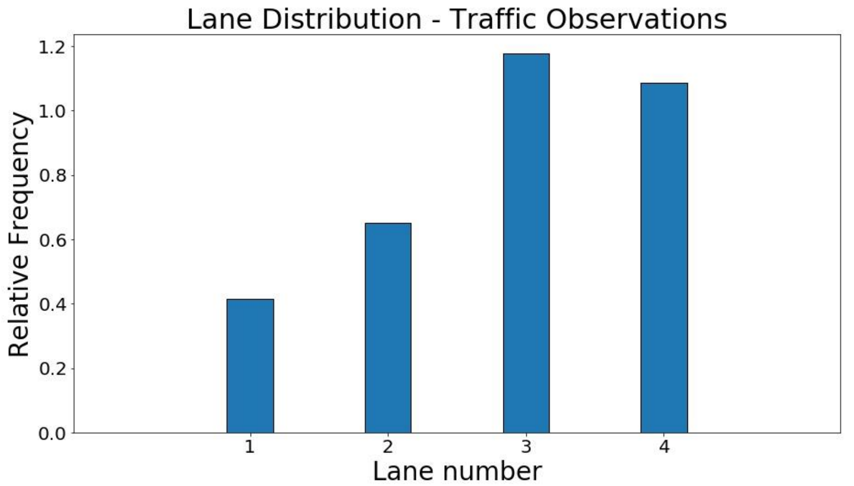


Fig 6.5 Distribution of lanes in the observed dataset

As mentioned in chapter 2.2.3, in order to have credible results, the base scenario should be first calibrated. Calibration is necessary to improve the ability of the model to accurately reproduce the traffic condition observed in this dataset. Many studies have already used individual vehicle trajectories for the calibration and validation of SUMO [RIVOIRARD ET AL., 2016; POURABDOLLAH ET AL., 2017; KATHS, 2017].

For the calibration, the approach proposed by U.S. federal highway administration [DOWLING ET AL., 2004] was used. This guideline suggested keeping the number of adjustable parameters as small as possible to reduce the amount of effort associated with the calibration. The modeler should trade-off between the number of chosen parameters to adjust, and the degrees of freedom to better fit the calibrated model to a specific traffic situation. This approach subdivides further the adjustable parameters based on their magnitude of effect into global parameters and localized parameters. The global parameters should be calibrated first, and then the model should be fine-tuned by tweaking the local parameters. The Three-step strategy that has been suggested by this methodology is as follows:

1. Calibrate capacity parameters.
2. Calibrate route choice parameters.
3. Calibrate overall model performance (travel times and queues).

As the study area of this dissertation does not have any parallel routes, the second step is not relevant in this study.

The goal of the capacity calibration step is to come as close as possible to matching the field measurements for traffic. For calibration of capacity in the study area, the part of the dataset in which at least 15 minutes queues persist was identified, and the hourly flow rates were estimated. Then the model parameters were changed to replicate the reality. The list of parameters that can be changed in sumo for car-following and lane changing is shown in Appendix I. For car-following of normal vehicles the default model in SUMO, which is a modification of the model defined by Stefan Krauss [1997]. For lane changing model, LC2013 default model developed by Jakob Erdmann [2014] was chosen. In the first step, the parameters with a significant impact on the capacity should be identified.

In order to calibrate the model, the values which lie within the acceptable range for each attribute were selected. Geistefeldt et al. [2015] attempted to calibrate SUMO for different freeway segments, including on ramps. Although the study area of this research is an urban freeway, it has free flow characteristics similar to E 5-2 segment (based on German highway capacity manual [Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV) - The committees of the Road and Transportation Research Association, 2015b]) of freeways. Therefore, the values selected for this study were defined around those values. The simulation ran with different combinations of the abovementioned set of parameters. In some cases, not all vehicles could make the lane change in the desired time and remain stationary. In order to avoid unrealistic disturbances, the time after which a stationary vehicle was taken out of the simulation was set to -1 in order to avoid vehicles disappearance. It is important to mention that the calibration and validation process was performed only for passenger cars and for the behavior of trucks and buses the default SUMO parameters were deployed.

SUMO, like many other microscopic simulation tools, a series of random numbers is used to generate randomness when loading vehicles (type distribution, speed deviation, etc.), probabilistic flows, vehicle driving dynamics and vehicle devices. SUMO implements the Mersenne Twister algorithm for generating random numbers. Mersenne Twister algorithm was developed by Matsumoto and Nishimura [1998] and is by far the most widely used general-purpose pseudorandom number generator [MARSLAND, 2014]. This variation has a large impact on simulation outputs as different simulation runs could produce very different results [BURGHOUT, 2004]. Therefore, it is essential to calculate the minimum number of runs to achieve reliable estimates of measures of performance. In order to account for stochastic behavior of traffic simulation, the required number of simulations were calculated based on the following formula and choosing the 95% confidence interval:

$$n \geq \frac{t(\alpha, n-1)^2 * s^2}{C^2}$$

Where:

n = Number of necessary simulation runs

$t(\alpha, n–1)$ = Value from the student distribution for a one-sided error

s = Standard deviation of the examined parameter

C = Desired confidence interval

The objective function of calibration is set to minimize the mean squared error (RMSE) between the estimated model and the field measurements. As the requirements to calculate the capacity in the observation dataset was not fulfilled, average travel times measurements were chosen as an indicator to compare the model and each set of repetitions has a single set of model parameters with eight different random seeds for each repetition within the set.

After finding the optimal global parameter set, the fine tuning was done. In the last step of calibration, the overall traffic performance of the model was calibrated. For this aim, the traffic volumes are considered and the parameters were further refined to reach the observed values. In order to compare the simulation result with the calibration acceptance target, defined by the guidelines [Wisconsin DOT, 2002], the GEH statistics is calculated as follows:

$$GEH = \sqrt{\frac{(E-V)^2}{\frac{E+V}{2}}}$$

where E is the model estimated volume and V is the measured volumes on the road. For the simulation model, GEH value was calculated 1.58 for the simulation based on the traffic volumes from data and from observations are illustrated in Tab. 6.1. Based on the calibration criteria table, the GEH value is below 5 for individual link flow and therefore, the value lies within the calibration acceptance range.

		Observation		Simulation	
Stream	Vehicle Type	Inflow [veh/h]	Outflow [veh/h]	Inflow [veh/h]	Outflow [veh/h]
Major Stream	Car	1681	3094	1679	3012
	Medium Vehicle	123	201	119	198
	Heavy Vehicle	58	122	55	115
	Bus	0	7	0	7
Ramp	Car	1515	-	1498	-
	Medium Vehicle	88	-	82	-
	Heavy Vehicle	70	-	67	-
	Bus	8	-	8	-

Tab. 6.1 Inflows and outflows in observation and simulation

In order to examine the goodness of the chosen parameter set and validate the model, a part of the dataset that was not used for calibration was chosen, and it was simulated with the exact entry of each vehicle in the network. Then, the travel time of each individual vehicle in the network was compared with those of traffic data observations. Fig 6.6 illustrates the plot of the real and simulated travel time of each vehicle with bivariate and univariate graphs. The graph on the left shows the hexagonal bins, whereas the graph on the right shows the scatter for better illustration of the deviations. Comparing the densities of real data and simulation data proves that the calibrated model can reproduce a credible result.

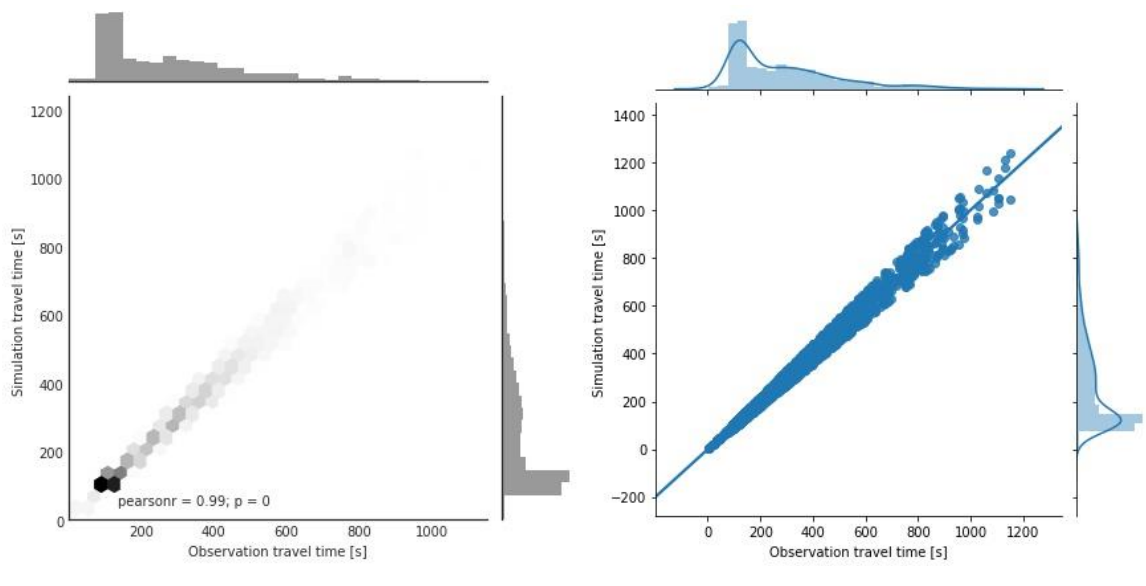


Fig 6.6 Bivariate and univariate graphs of real and simulated travel time

6.2 Model Implementation

In order to implement the model, the TraCI interface in SUMO was used. TraCI is the short term for Traffic Control Interface. Giving access to a running road traffic simulation, it allows to retrieve values of simulated objects and to dynamically manipulate their behavior [WEGENER ET AL., 2008]. As illustrated in Fig 6.7, TraCI provides the opportunity that for each simulation step, the surrounding environment of the ego vehicle be transferred to the driving behavior manager and after deciding about the actions, the actions be implemented in SUMO. The experts' trajectories, which are saved in the database, are the inputs of the IRL Lab. IRL Lab is the name given to the process performed for learning the lane changing behavior which was described in chapter 5. The resulting reward function is transferred to the behavior manager module for each following traffic situation which is translated into features. The reward is computed, and the corresponding action is decided upon. These actions are then converted into the new position and new speed of the ego vehicle and implemented in SUMO. In SUMO, the simulation time step length can be specified. As the observations were recorded with the resolution of 40 ms, the simulation timestep was set accordingly.

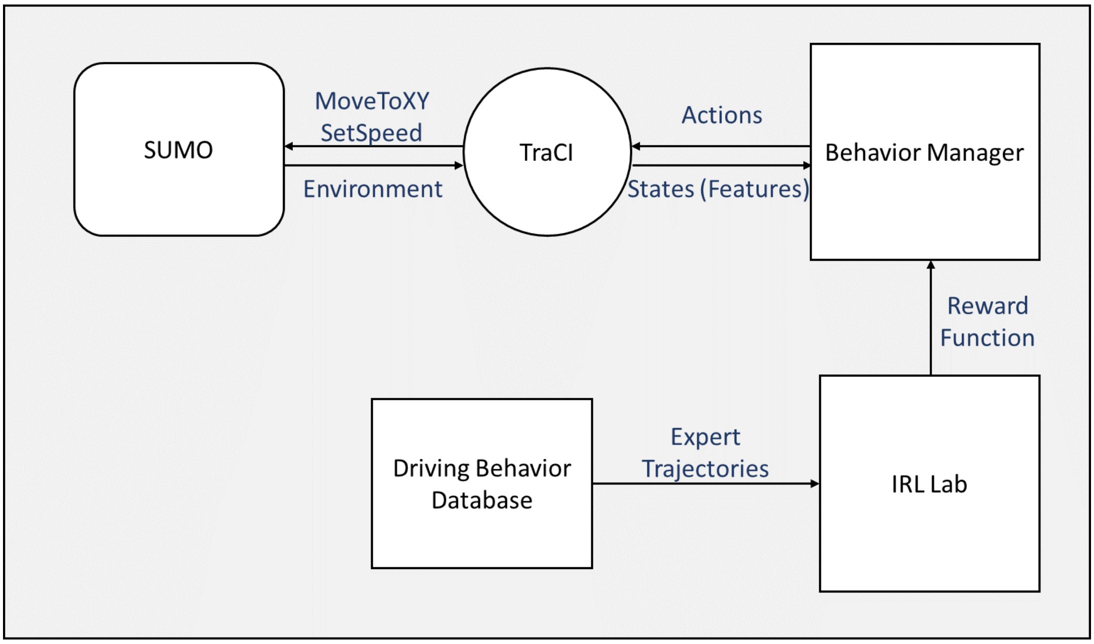


Fig 6.7 System architecture and an operation example of the command-response exchange between the SUMO and driving behavior manager module

When implementing an external model with TraCI, for the internal lane change motivations, the following options can be independently configured [ERDMANN, 2015] :

a) Ignore the internal request.

b) Ignore the internal request when in conflict by an external request.

c) Always follow the internal request regardless of the external request.

Furthermore, the following options for configuring the urgency of external requests:

a) Following request regardless of surrounding vehicles, perform urgent speed adaptions.

b) Follow request unless it would cause an immediate collision but ignore safety gaps to surrounding vehicles, perform urgent speed adaptions.

c) Only change if all safety constraints are met, perform urgent speed adaptions.

d) Only change if all safety constraints are met, perform no speed adaptions.

As the IRL-based model calculated the new positions and new speeds, the implemented model ignores the internal request by SUMO model completely and follows the request.

6.3 Simulation Scenario

In this section, the simulation scenarios are discussed. With the objective of testing the capabilities of the developed models, various real-world traffic situations were modeled with 50% penetration rate of automated vehicles. In total, four traffic scenarios are considered, and for every test, the simulation time is set to 1.5 hours to ensure the run-time is long enough. The simulation scenarios are as follows:

Scenario 1: the calibrated and validated model

The first simulated scenario aims to examine the capability of the developed models in the base scenario, compare their performances with the internal SUMO model and validated them with the data. In this scenario, all the traffic conditions are identical except the fact that the driving behavior model of passenger cars is overwritten by either of the two developed models.

Scenario 2: Accident on the second lane

In order to examine the transferability of the model to other traffic scenarios, as shown in Fig 6.8, an artificial traffic incident was created on the second lane of the major stream downstream of the ramp. The objective in developing this scenario to compare the underlying lane change model in SUMO with the developed models for human drivers and automated vehicles and examine the transferability of the data to the scenarios which were nonexistence in the training data. The artificial accident happened in minute 5 of the simulation and lasted for 10 minutes. Thereafter, the collided vehicles were removed, and the capacity of the road returned back to normal. In SUMO, the vehicles which cannot find the necessary gaps and execute the lane changing within a certain time are removed from the network. This behavior is one of the unrealistic behaviors in the microscopic simulation, which is in contrast with the conservation law of traffic flow [LIGHTHILL & WHITHAM, 1955]. In order to avoid the disappearance of the vehicles on the network, the teleoperating time in SUMO was set to -1.

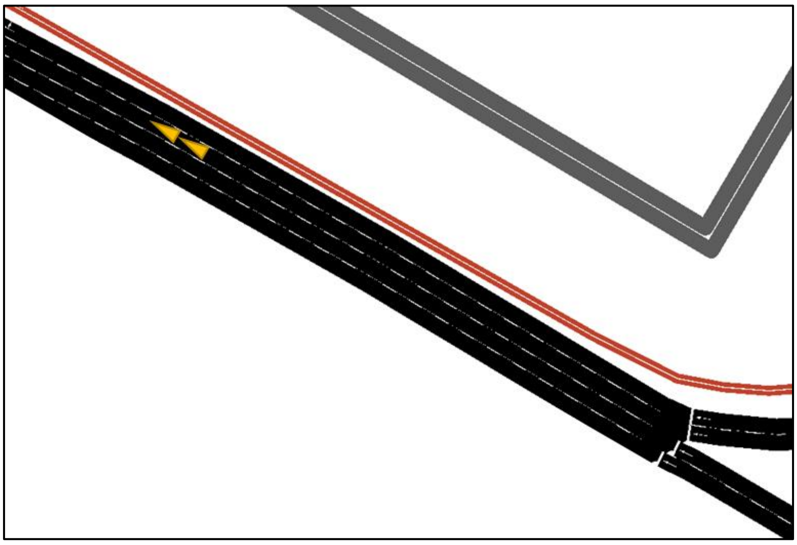


Fig 6.8 Location of accident in the network

Scenario 3: high pressure from on-ramp traffic

Traffic is generally more turbulent around freeway ramps due to route choice-related lane changes and anticipatory or cooperative maneuvers [VAN BEINUM ET AL., 2018] and the ramp inflow also varies the number of courtesy gaps that have to be created. In order to evaluate the merge action of the model in the situation with the higher traffic flow from the on-ramps, in the base scenario, the inflow from the ramp was increased by 15%, and the main inflow volume was reduced by 15% to ensure the merging ability of vehicles and the simulations run were completed. Tab. 6.2 illustrates the original and new inflows of vehicles.

Stream	Vehicle Type	Original Inflow [veh/h]	New Inflow [veh/h]
Major Stream	Car	1681	1428
	Medium vehicle	123	104
	Heavy vehicle	58	49
	Bus	0	0
Ramp	Car	1515	1742
	Medium vehicle	88	101
	Heavy vehicle	70	80
	Bus	8	9

Tab. 6.2 Traffic Flows in scenario 3

Scenario 4: higher speed limit

One way to examine the transferability of the model to other situations is the variation of traffic regulations and constraints that the vehicles might face. The speed limit is one of those limitations, the effects of which on various traffic aspects have been studied extensively in the literature [YANG ET AL., 2012; SORIGUERA ET AL., 2017; ROCK, 1995; OSSIANDER & CUMMINGS, 2002]. A speed limit enforcement can reduce the difference in speed between vehicles and reduce the number of lane changes [KNOOP ET AL., 2012a]. For this aim, in this scenario, an evaluation of the speed limit increase from 60 km/h to 80km/h on the major road was performed.

6.4 Evaluation

In this section, simulation results of the developed model are reported and compared against SUMO's default lane change model. In order to evaluate the performance of the models, the following three indicators are looked at mainly:

- Distribution of TTCs: TTC is often a critical element of a driver's trajectory management decision-making process [HOU ET AL., 2014a], and is the parameter frequently utilized in the literature to assess the interaction intensity among vehicles lane changing maneuvers. For instance, Farah et al. [2008] developed a model which predicts the risk in terms of the TTC associated with the passing behavior. In this study, the distribution shows the TTCs below 10 seconds are analyzed.

- Throughput of the network: This is an indicator of the productivity of the system, the number of vehicles that managed to go through the traffic network during the analysis period [Federal Highway Administration (FHWA), 2017]. The throughput can be compared for different alternatives to determine the relative productivity of each alternative.

- Number of lane changes: A driving simulator experiment by Yang et al. [2018] has shown that as traffic density increases, drivers' lane changing and overtaking intentions are enhanced. Both initial overtaking distance and headway decrease with traffic density, which might influence road safety.

Scenario 1: the calibrated and validated model

Fig 6.9, compares the TTC distributions of the different models and the number shown in red and dark blue are the number of vehicles that managed to go through the traffic network within the simulation period and the number of lane changes, respectively. As can be seen from the graphs, In SUMO model, in total more vehicles could perform a lane change maneuver when compared to IRL-based models; however, those which were involved performed the maneuvers with lower TTCs. Moreover, when comparing the two IRL-based models, a lower number of lane changes and higher TTC values prove that the models for automated vehicles reacted more conservatively during lane changing.

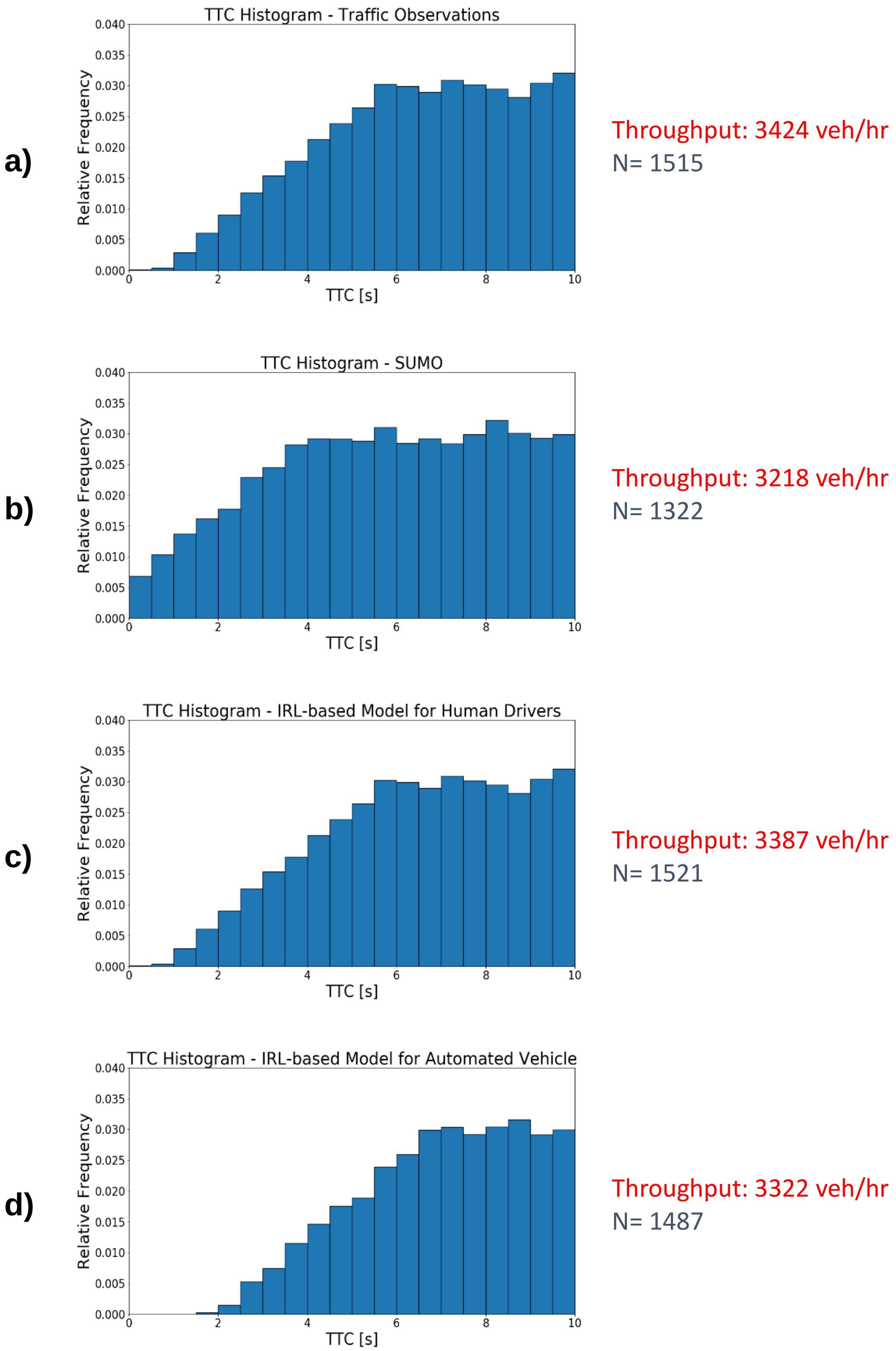


Fig 6.9 TTC distributions of vehicles for scenario 1 in a) traffic observations, b) SUMO model, c) IRL-based model for human drivers and d) IRL-based model for automated vehicles.

.

Scenario 2: accident on the second lane

As mentioned in Chapter 6.3, this scenario includes an artificial accident within the network. Fig 6.10 compares the results of scenario 2 with different models. Comparing the results of IRL-based models with the default SUMO model proves that the model could successfully perform the lane changes, although this specific situation was not existent in the dataset. Moreover, the graphs illustrated that the IRL-based model for automated vehicles was more conservative when compared to the developed model for human drivers.

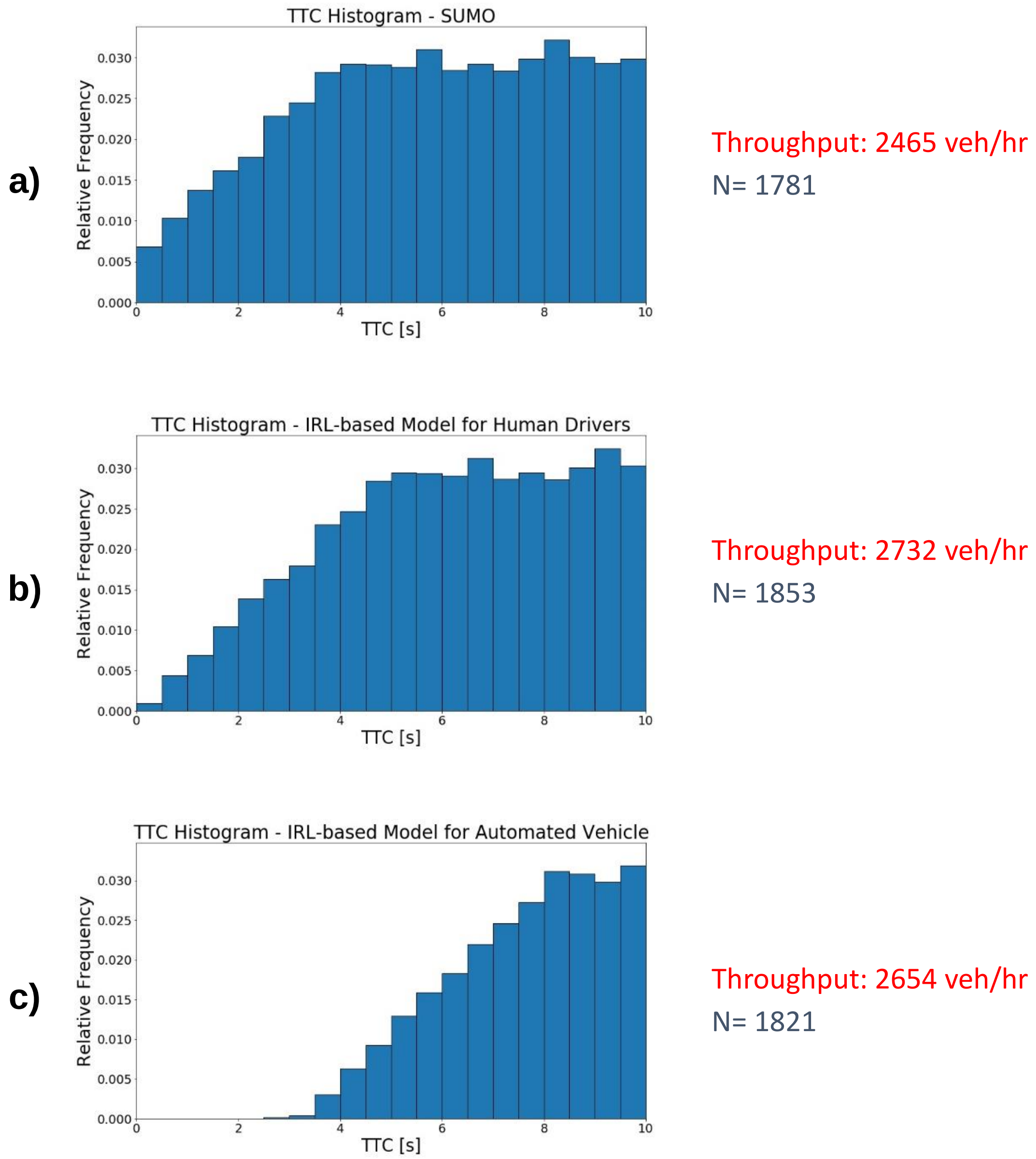


Fig 6.10 TTC distributions of vehicles for scenario 2 in a) SUMO model, b) IRL-based model for human drivers and c) IRL-based model for automated vehicles.

The next step is to compare the macroscopic indicators of three models. The comparison of three scenarios conveyed that after removing the traffic incident, in the scenario with SUMO lane change model the queue was dissolved after 14 min and 10 seconds, whereas these values were 10 minute and 26 seconds for the IRL model for human drivers and 9 minute and 15 seconds with the IRL model for automated vehicles.

Scenario 3: Higher pressure from on-ramp traffic

Fig 6.11 illustrates the result of scenario 3 in which the traffic flow from the on-ramp is higher. In this scenario, the difference between the TTC distributions are not as considerable as the previous scenarios, but the result of Scenario 3, especially the throughput of the network confirms that in case of higher traffic flow from the on-ramp, the vehicles controlled by underlying model in SUMO had difficulty merging into traffic stream, whereas the vehicles controlled by IRL models show more cooperative behavior. This has led to higher throughput for both human drivers model and automated vehicles model compared to SUMO model.

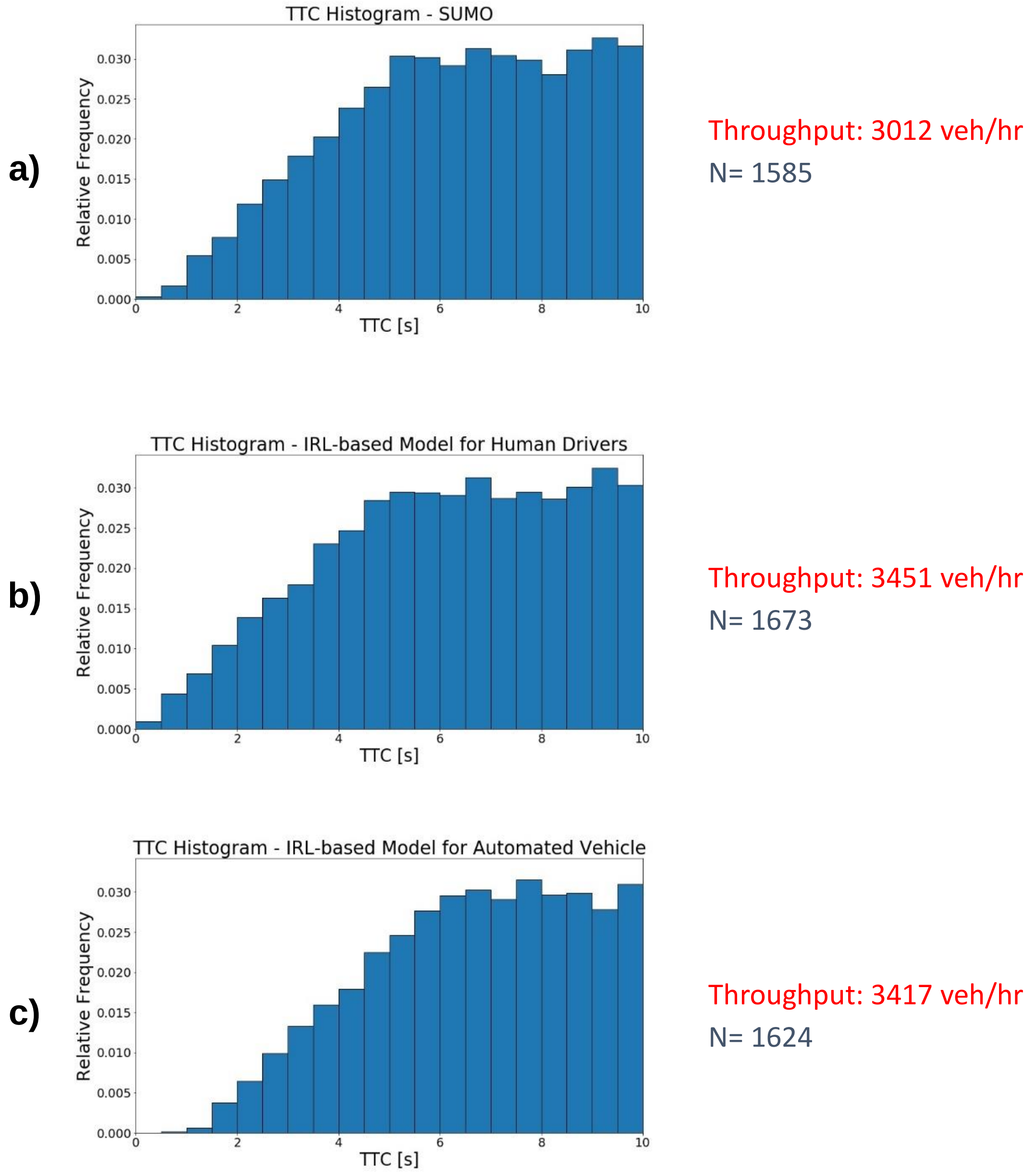


Fig 6.11 TTC distributions of vehicles for scenario 3 in a) SUMO model, b) IRL-based model for human drivers and c) IRL-based model for automated vehicles.

Scenario 4: Higher speed limit

Scenario 4 intended to examine the effect of traffic regulation, namely speed limit, on the performance of the developed models. As the results in Fig 6.12 illustrate, the values are not significantly different from scenario 1. The results prove that the increased speed limit did not influence the performance of the model, neither in the SUMO model nor in IRL-based models. Higher speed limits increased the throughput slightly in all the models. However, in the SUMO model, the increase in speed limit resulted in fewer successful lane change executions.

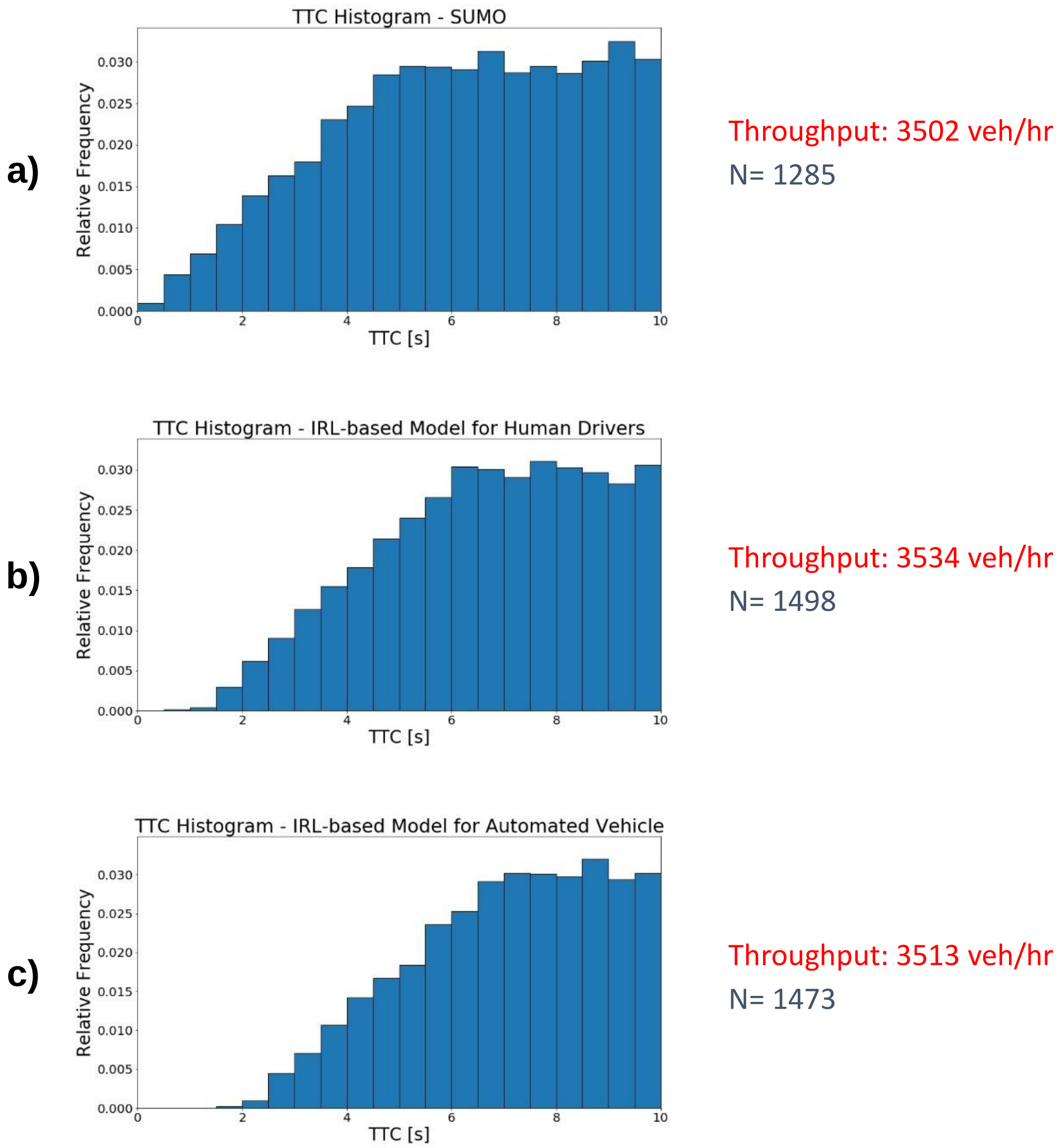


Fig 6.12 TTC distributions of vehicles for scenario 4 in a) SUMO model, b) IRL-based model for human drivers and c) IRL-based model for automated vehicles.

6.5 Summary

In this chapter, the simulation framework for examining the transferability of the IRL based model was discussed. The process for calibration and validation of the base model with the subset of the dataset that was not used during training was explained in detail. Furthermore, the framework used for the implementation of developed external IRL-based models (discussed in chapter 5) SUMO with TraCI interface was described.

In order to test the capability of the developed models, various traffic situations, which were not existent in the training dataset, were simulated. Simulation results illustrated that both models captured trade-offs between that a human driver would take in such a traffic situation and could execute the lane changing safely and efficiently. However, the safety gains were more significant in the model for automated vehicles, where aggressive driving behavior was not considered during the model training. It is noteworthy to mention that the integration of lane change model into simulation tools was only for passenger cars and developing a socially-acceptable model for heavy vehicles can further make the traffic flow smooth and improve the lane changing behavior model.

7. Conclusions

This chapter summarizes the research presented in this dissertation, highlights the major contributions achieved, and points out the limitations. Some directions for future research are suggested in the closing section.

7.1 Summary

The initially stated overarching aim of this research was to deploy a data-driven approach to learn the lane change behavior of vehicles from traffic observations and help improve the microscopic simulation models in terms of replicating the behavior of drivers in mixed-traffic. In order to conclude the work, the three research questions were answered:

- How can the socially acceptable driving behavior be sampled from the dataset of individual vehicle trajectories?

In terms of the initial objectives, the dissertation successfully recruited the clustering approach. The k-mean clustering algorithm analyzed the driving profile of each driver resulted from collected traffic observations and categorized the driving behaviors into three groups: “timid”, “moderate” and ”aggressive”. As intended, the clustering algorithm extracted the driving style of the drivers in the scene and could identify which driving behavior should be used as a training sample in the model development step.

- Can the tactical and operational lane changing behavior for automated vehicles be learned from the observations of lane changing behavior of human drivers?

The tactical lane changing behavior of the models was modeled with classification methods. For this aim, the state of the ego vehicle and the surrounding environment were translated into the feature vector. The significance of the features in the feature vector identified how drivers trade-off between different factors when making their tactical decision of changing the lane. Moreover, they were very useful in determining which factors can be addressed through driving behavior development for automated vehicles. The results showed that the tactical decision of the drivers could be successfully modeled in this particular traffic situation of a multi-lane freeway with keep-right traffic regulation.

The operational behavior of drivers during lane changing was learned from the observed trajectories in the study area with an IRL method. Due to the stochastic nature of the operational behavior of drivers, Markovian processes, that have long been previously used to model stochastic environments, were deployed. The IRL framework has been implemented to solve sequential planning and decision-making problems in the stochastic environment under study. Besides, a speed predictor with random forests was embedded in the IRL framework for making predictions about the speed of the ego vehicle. The results showed that in the defined study area and its specific nature, the model has the capability of making predictions about the operational behavior of vehicles in situations with reasonable accuracy.

- How can this methodology be generalized for the behavior of automated vehicles in different driving situations?

The driving behavior models in this study were built on observed data in a specific traffic situation. However, the challenging question is to know if the model will predict the future and unseen data. To answer this question, traffic flow simulations were used due to the lack of data from various traffic scenarios. The developed models were integrated into the microscopic simulation model SUMO to evaluate the plausibility of the models and examine their performance in new events, unseen traffic situations and traffic regulations which were non-existence in the training dataset: creating an artificial accident on the freeway, increasing the vehicle flow from the ramp and increasing the speed limit. The simulation results proved that the model for human drivers and automated vehicles could both capture the trade-offs that a conventional driver would take in such traffic situations and could execute the lane changing safely and efficiently. However, the model for automated vehicles resulted in higher TTC values than human drivers.

7.2 Transferability

Motorized road traffic comprises a system with very diverse players whose goal is to convey road users to their destinations safely, quickly, and without incident, and the heterogeneous driving behavior of drivers adds more complexity when modeling their driving behavior. The modeling approach in this study, instead of supplying the vehicle with a fixed structure (rule-based model) from which the right action for each scenario can be inferred, feeds the model with many traffic situations and specify the correct action for each situation by learning the reward function. The model then searches by itself for the best configuration of internal parameters (feature weights in reward function) and internal decision logic (actions), which allow it to act correctly in all of those situations. The proposed modeling approach in this dissertation could successfully model the tactical and operational lane changing behavior in the traffic situation under study. However, the question of the degree of transferability of the modeling approach to other locations and traffic situations has important implications on the practical application of this model. In order to transfer the modeling approach to other traffic situations, in addition to a substantial amount of traffic observations, some modeling details should be adapted to each situation. In this chapter, examples of traffic situations and the necessary changes to adjust the proposed modeling approach are discussed briefly.

In order to drive automated in urban areas, the complexity of the driving situations is very high, and the vehicles face more road users such as pedestrians and bicyclist. These traffic road users impose major challenges for the researchers and developers of the necessary technologies. These road users have more degree of freedom in their movement and predicting their actions and movements becomes more complex than predicting the movement of the vehicles on the freeway. Moreover, more complex road layouts in the urban areas lead to more possible scenarios that can happen in comparison to freeway driving, in which the range of behavioral states for the vehicle is limited. Applying the proposed methodology in this research to urban areas requires a more comprehensive feature vector and the corresponding movement in all directions for different road users such as pedestrians and bicyclists.

Another complex traffic situation for both normal drivers and automated vehicles is overtaking in the opposite lane. This driving situation is accompanied by a lot of uncertainties and risks [FARAH, 2016] and blind spots can arise from perception limitations [ANDERSEN ET AL., 2017]. For instance, the automated vehicle may have to move slightly into the opposite lane, as human drivers do, to clearly see in front of a car ahead. However, as this event is a rare event and the exact location of the overtaking is not clear in comparison with lane changing, which was the focus of this dissertation, naturalistic driving data are more convenient for this purpose. In this traffic situation, crucial features such as the speed and distance to the approaching vehicle in the opposite lane should be added to the feature vectors and the size of Markov grid should be dynamically adapted to include all the players in the scene.

In some situations, human drivers are allowed to break the traffic regulations in order to avoid a deadlock situation. An example of this would be crossing a solid lane marking to overtake a broken-down truck on a two-lane street. The human drivers would break the rules and drive around the truck when the road is clear. The proposed methodology in this dissertation can be used and adapted to learn the behavior of human drivers in this particular situation or similar situations and overcome the deadlock situation problem that automated vehicles are currently facing.

Construction zones are the next challenging situation to model for automated vehicles [FERGUSON & BURNETTE, 2013]. The signs and markings in the construction zone change based on the location and the undergoing task. Due to the dynamic nature of work and different lane markings, a rule-based model and a single way to standardize how the vehicle interprets and reacts to the construction zone is not sufficient. Therefore, the application of the proposed methodology in this dissertation can provide the opportunity to study the behavior of human drivers in the construction zones, model their behavior and safely navigate the automated vehicles in construction zones based on the learned knowledge. For this aim, the distance to the construction zone, type of construction, type of barrier and the magnitude of deviation from the original lane are the candidate components for adding to the feature vector.

7.3 Limitations

Although this dissertation has reached its aims, there were some unavoidable limitations. First of all, the focus dissertation was on site-based traffic observation. However, the other influencing factor of drivers, such as emotion and aggressiveness, as well as the motivation behind the lane changing and strategic behavior are not observable with this dataset. The combination of the method provided here with some naturalistic data and driving simulator study, in which the drives can also be questioned during and after the experiment, can empower the understanding of the motivation behind human's action during driving and enhance the quality of the provided methodology.

The foundational methods of IRL are able to achieve their results by leveraging the knowledge obtained from a policy that has been executed by a human expert. However, the goal for ML systems is to learn from a wide range of human behavior data and perform tasks that are beyond the abilities of human experts without error. A strong assumption in the IRL method is that the expert demonstrations are locally optimal, which in general is not true for drivers, and it depends on hand-crafted features. Furthermore, the reward function is not unique and depends on the features. Despite allowing for the tractability of the problem, this possesses several limitations, as it relies on human experience and domain knowledge in order to select appropriate features. Even with well-selected features, its discriminative power is often unknown before the learning process. However, this limitation was tackled in this thesis by creating a generic set of features that can translate various driving patterns significantly in different driving cultures and IRL was used as a simple and tractable way of modeling human drivers.

Markov decision processes, which are a standard method for decision-theoretic planning, have the drawback that they require an explicit state representation. The IRL in this study also deployed a discrete Markov decision and Maximum Entropy used a grid discretization for both states and actions. Although a large amount of work on approximate inference in continuous state spaces does exist [AGHASADEGHI & BRETL, 2011; LEVINE & KOLTUN, 2012], these algorithms are computationally more expensive and whether or not these models can be solved in a reasonable time or the possibility of their implementation in a traffic simulation tool is not clear yet.

7.4 Future Research

There are a number of gaps in this dissertation that follow from the findings and would benefit from further research:

1. The primary difference between the data-driven and rule-based modeling approaches is the level of analysis. Due to the fact that the data-driven modeling approach is built on learned information, it enables the analysis of the past and to predict the future under the same condition when training. On the other hand, since the rule-based modeling approach is based on underlying physical models, it enables a higher level of analysis and interpretation in comparison with the data-driven approach. In summary, the data-driven modeling approach has an advantage in that analysis is possible regardless of knowledge of the system, but it has a weakness regarding the depth of analysis. Conversely, while the rule-based modeling approach requires

knowledge, it enhances the depth of analysis and facilitates the conduction of profound knowledge. Therefore, a new integrated modeling approach that is able to overcome the deficiencies and combine the merits of both modeling approaches should be examined in the future line of research.

Most of ML algorithms are only capable of interpolating data and making predictions about situations that are among known situations and events. For instance, Lohninger [1999] showed how neuronal networks behaviors are consistent with training data but cannot do anything outside of this subspace. These algorithms usually fail to extrapolate data and make predictions about situations outside of the known events because they learn to locally fit the known data as closely as possible, regardless of how it performs outside of these situations. Moreover, collecting sufficient data for plausible interpolation and the required data from extreme or dangerous situations is both time- and resource-intensive. A recent study [SAHOO ET AL., 2018] has already proposed a new method, based on their previous study [MARTIUS & LAMPERT, 2016], which strives to reveal the true dynamics of the situation for robots. This method takes in data and returns the equations that describe the underlying physics using a shallow neural network approach. The authors of this study claim that their approach allows understanding functional relations and generalizing them from observed data to unseen parts of the parameter space. In other words, the algorithm can make predictions about situations outside of the known. However, this method has not been tested for the task of driving, and future research should cover the methods capable of extrapolation of the model for driving task to the unseen situations.

2. Connected vehicles are vehicles that use different communication technologies to communicate with other vehicles on the road (vehicle-to-vehicle [V2V]), roadside infrastructure (vehicle-to-infrastructure [V2I]), and the cloud [V2C]. This technology can be used to not only improve traffic safety but also to improve traffic efficiency [JURGEN, 2012]. Adding connectivity to automated vehicles can provide the opportunity to have more cooperative driving behavior and should be studied more extensively [FARAH & KOUTSOPOULOS, 2014]. For instance, a communication component enables the vehicle to know when another vehicle far ahead intends to change lanes or starts to decelerate even if it is beyond the line of sight. Next research should target the cooperative driving behavior with the existence of the communication.

3. The proposed model in this dissertation is applicable to the timeline when the roads are shared among human drivers and automated vehicles. In a scenario with 100% penetration rate of automated vehicles, the infrastructure can be adapted accordingly, and the driving behavior can be optimized differently. In this case, other methods such as swarm intelligence [DARIO ET AL., 1993] can be applied for controlling the vehicles in which vehicles communicate and relay their planned trajectory to other vehicles in the local vicinity [PRAKASH ET AL., 2014]. Therefore, the cumulative intelligence of all

these vehicles enables them to think and work together and share data in the effort of finding a solution and have more cooperative driving behavior.

4. Automated vehicles are deemed to eliminate human errors, the cause of most traffic accidents [National Highway Traffic Safety Administration (NHTSA), 2015]. These vehicles will not get tired, distracted, impaired by alcohol and drugs, will not get upset, will not take risks and will always be on high alert. However, there are still certain limitations regarding detecting their surroundings with their sensors. Although with the help of their sensors, the distance and velocities can be calculated more precisely than humans, there are still certain limitations linked to the sensor technology [COMBS ET AL., 2018]. The accompanying uncertainty with the sensor technology and how it might affect the driving behavior of automated vehicles should be studied more into detail.

5. The proposed data-driven approach conveyed successful results for the present case study. Yet, a couple of future enhancements were detected during the development and application tasks:

   - With regard to the modeling approach with IRL, when discussing potential avenues for further research, the problem of sub-optimality in the agent's actions has already been pointed out as an important problem [NG & RUSSEL, 2000]. While human experts can provide a baseline level of performance that RL algorithms could learn from, it is rarely the case that the expert policy given is the global optimum. The next problem to solve in IRL would be to extract a reward function without the assumption that the human policy is optimal.

   - Another possible line of research would consider the reward functions, which are not necessarily the linear combination of features. The next line of research should combine IRL methods to work with deep learning systems. For some specific tasks, understanding the underlying policy is beyond human's capability. Therefore, combining the deep learning models and the IRL can facilitate the application of this method to more complex tasks.

   - In order to deliver good results, IRL requires a lot of observations, and the learning is not at a rapid pace. Other lines of research should apply the methods such as one-shot [SANTORO ET AL., 2016a] and few-shot learning, which have already been studied for image recognition [VINYALS ET AL., 2016; KOCH, 2015; SANTORO ET AL., 2016b], generative modeling [EDWARDS & STORKEY, 2016; REZENDE ET AL., 2016], learning fast reinforcement learning agents with recurrent policies [SHIMOSAKA ET AL., 2017; DUAN ET AL., 2016] and imitation learning [DUAN ET AL., 2017].

References

ABBEEL, P.; A.Y. NG [2004]: Apprenticeship learning via inverse reinforcement learning,*ICML '04 Twenty-first international conference on Machine Learning.* Banff, Alberta, Canada.

ABELE, L.; M. MØLLER [2011]: The Relationship between Road Design and Driving Behavior,*3rd International Conference on Road Safety and Simulation.* Indianapol, USA.

ABRAHAM, B.; J. LEDOLTER [2006]: Introduction to regression modeling, Thomson Brooks/Cole, Belmont, Calif.

ABUALI, N.; H. ABOU-ZEID [2016]: Driver Behavior Modeling: Developments and Future Directions. In: *International Journal of Vehicular Technology*, 2016 (2), S. 1–12

ADAMEC, V.; B. SCHULLEROVA; A. BABINEC; D. HERMAN; J. POSPISIL [2017]: Using the DataFromSky system to monitor emissions from traffic,*Proceedingss of the Aiit International Congress on Transport Infrastructure and Systems.* Rome, Italy.

AGHASADEGHI, N.; T. BRETL [2011]: Maximum entropy inverse reinforcement learning in continuous state spaces with path integrals,*2011 IEEE/RSJ International Conference.* San Francisco, CA, USA.

AHMED, K.I. [1999]: Modeling drivers' acceleration and lane changing behavior, PhD thesis

AHMED, K.I.; M. BEN-AKIVA; H.N. KOUTSOPOULOS; R.G. MISHALANI [1996]: Models of Freeway Lane Changing and Gap Acceptance Behavior,*Transportation and Traffic theory. Proceedings of the 13th International Symposium Transportation and Traffic Theory.* Lyon, France.

AIMSUN [2018]: Aimsun Next.

AL-BALBISSI, A.H. [2003]: Role of gender in road accidents. In: *Traffic injury prevention*, 4 (1), S. 64–73

ALELYANI, S.; J. TANG; H. LIU [2013]: Feature Selection for Clustering: A Review,*Data Clustering: Algorithms and Applications*

ALESSANDRINI, A.; A. CAMPAGNA; P.D. SITE; F. FILIPPI; L. PERSIA [2015]: Automated Vehicles and the Rethinking of Mobility and Cities. In: *Transportation Research Procedia*, 5, S. 145–160

ALEXIADIS, V.; J. SLOBODEN; G. CORDAHI; R. VAN GORDER [2014]: Guidance on The Level of Effort Required to Conduct Traffic Analysis Using Microsimulation. Federal Highway Administration (FHWA). https://www.fhwa.dot.gov/publications/research/operations/13026/index.cfm.

ALI, M.; P. FALCONE; C. OLSSON; J. SJOBERG [2013]: Predictive Prevention of Loss of Vehicle Control for Roadway Departure Avoidance. In: *IEEE Transactions on Intelligent Transportation Systems*, 14 (1), S. 56–68

AL-KAISY, A.; Y. JUNG; H. RAKHA [2005]: Developing Passenger Car Equivalency Factors for Heavy Vehicles during Congestion. In: *Journal of Transportation Engineering*, 131 (7), S. 514–523

AL-KAISY, A.F.; F.L. HALL; E.S. REISMAN [2002]: Developing passenger car equivalents for heavy vehicles on freeways during queue discharge flow. In: *Transportation Research Part A: Policy and Practice*, 36 (8), S. 725–742

ALTCHÉ, F.; A. de LA FORTELLE [2017]: An LSTM Network for Highway Trajectory Prediction,*IEEE ITSC 2017.* Yokohama, Japan.

AMAL, I.T.; F.L. HALL [1994]: Effect of adverse weather conditions on speed-flow-occupancy relationships. In: *Transportation Research Record*

ANDERSEN, H.; W. SCHWARTING; F. NASER; Y.H. ENG; M.H. ANG; D. RUS; J. ALONSO-MORA [2017]: Trajectory optimization for autonomous overtaking with visibility maximization,*2017 IEEE 20th International Conference*

ANDERSON, S.J.; S.C. PETERS; T.E. PILUTTI; K. IAGNEMMA [2010]: An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In: *International Journal of Vehicle Autonomous Systems*, 8 (2/3/4), S. 190

AOUDE, G.S.; V.R. DESARAJU; L.H. STEPHENS; J.P. HOW [2011]: Behavior classification algorithms at intersections and validation using naturalistic data,*2011 IEEE Intelligent Vehicles Symposium.* Baden-Baden, Germany.

ARDELT, M.; C. COESTER; N. KAEMPCHEN [2012]: Highly Automated Driving on Freeways in Real Traffic Using a Probabilistic Framework. In: *IEEE Transactions on Intelligent Transportation Systems*, 13 (4), S. 1576–1585

ASSUM, T.; T. BJØRNSKAU; S. FOSSER; F. SAGBERG [1999]: Risk compensation—the case of road lighting. In: *Accident Analysis & Prevention*, 31 (5), S. 545–553

BABES-VROMAN, M.; V. N. MARIVATE; K. SUBRAMANIAN; M. LITTMAN [2011]: Apprenticeship Learning About Multiple Intentions. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*

BAGLOEE, S.A.; M. TAVANA; M. ASADI; T. OLIVER [2016]: Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. In: *Journal of Modern Transportation*, 24 (4), S. 284–303

BALAL, E.; R.L. CHEU; T. SARKODIE-GYAN [2016]: A binary decision model for discretionary lane changing move based on fuzzy inference system. In: *Transportation Research Part C: Emerging Technologies*, 67, S. 47–61

BALK, S.A.; K.S. MOORE; J.E. STEELE; W.J. SPEARMAN; A.T. DUCHOWSKI [2006]: Mobile phone use in a driving simulation task: Differences in eye movements. In: *Journal of Vision*, 6 (6), S. 872

BARCELO, J.; E. CODINA; J. CASAS; J.L. FERRER; D. GARCIA [2005]: Microscopic traffic simulation. A tool for the design, analysis and evaluation of intelligent transport systems. In: *Journal of Intelligent and Robotic Systems*, 41 (2-3), S. 173–203

BASU, C.; Q. YANG; D. HUNGERMAN; M. SINGHAL; A.D. DRAGAN [2017]: Do You Want Your Autonomous Car To Drive Like You?,*HRI '17 ACM/IEEE International Conference on Human-Robot Interactio.* Vienna, Austria.

BAYES; PRICE [1763]: An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S. In: *Philosophical Transactions of the Royal Society of London*, 53 (0), S. 370–418

BÉLISLE, F.; N. SAUNIER; G.-A. BILODEAU; S. LE DIGABEL [2017]: Optimized Video Tracking for Automated Vehicle Turning Movement Counts. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2645 (1), S. 104–112

BELLA, F.; A. CALVI [2013]: Effects of simulated day and night driving on the speed differential in tangent-curve transition: a pilot study using driving simulator. In: *Traffic injury prevention*, 14 (4), S. 413–423

BELLA, F.; A. CALVI; F. D'AMICO [2014]: Analysis of driver speeds under night driving conditions using a driving simulator. In: *Journal of safety research*, 49, S. 45–52

BELLMAN, R. [1952]: On the Theory of Dynamic Programming. In: *Proceedings of the National Academy of Sciences of the United States of America*, 38 (8), S. 716–719

BELLMAN, R. [1957]: Dynamic programming, Princeton University Press, Princeton.

BEN-AKIVA, M.; S. LERMAN [1985]: Reviews: Discrete Choice Analysis: Theory and Application to Travel Demand. In: *Transportation studies*, 20

BEN-AKIVA, M.E.; C. CHOUDHURY; T. TOLEDO [2006]: Lane changing models,*Proceedings of the International Symposium of Transport Simulation.* Lusanne, Switzerland.

BENER, A.; D. CRUNDALL [2008]: Role of gender and driver behaviour in road traffic crashes. In: *International Journal of Crashworthiness*, 13 (3), S. 331–336

BENGIO, Y. [2012]: Practical recommendations for gradient-based training of deep architectures. In: *CoRR*, abs/1206.5533

BERGSTRA, J.S.; R. BARDENET; Y. BENGIO; B. KÉGL [2011]: Algorithms for hyper-parameter optimization,*NIPS'11 Proceedings of the 24th International Conference on Neural Information Processing Systems.* Granada, Spain.

BERK, R.A. [2004]: Regression analysis: A constructive critique, Sage

BERKENKAMP, F.; M. TURCHETTA; A.P. SCHOELLIG; A. KRAUSE [2017]: Safe Model-based Reinforcement Learning with Stability Guarantees,*Proc. of Neural Information Processing Systems (NIPS)*

BERTINI, R.L.; M.T. LEAL [2005]: Empirical Study of Traffic Features at a Freeway Lane Drop. In: *Journal of Transportation Engineering*, 131 (6), S. 397–407

BHATTACHARYA, S.; H. IDREES; I. SALEEMI; S. ALI; M. SHAH [2011]: Moving Object Detection and Tracking in Forward Looking Infra-Red Aerial Imagery. In: HAMMOUD, R.I. (Eds.): *Machine vision beyond visible spectrum*, S. 221–252, Springer. Heidelberg, New York

BHATTACHARYYA, R.P.; D.J. PHILLIPS; B. WULFE; J. MORTON; A. KUEFLER; M.J. KOCHENDERFER [2018]: Multi-Agent Imitation Learning for Driving Simulation. In: *CoRR*, abs/1803.01044

BINGHAM, C.R.; B.G. SIMONS-MORTON; A.K. PRADHAN; K. LI; F. ALMANI; E.B. FALK; J.T. SHOPE; L. BUCKLEY; M.C. OUIMET; P.S. ALBERT [2016]: Peer Passenger Norms and Pressure: Experimental Effects on Simulated Driving Among Teenage Males. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 41 (A), S. 124–137

BOER, E.R.; M. HOEDEMAEKER [1998]: Modeling driver behavior with different degrees of automation: A hierarchical decision framework of interacting mental models: *Proceedings of the 17th European annual conference on human decision making and manual control*

BOLTZMANN, L. [1868]: Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten: vorgelegt in der Sitzung am 8. October 1868, k.k. Hof- und Staatsdruckerei

BORRELL, C.; A. PLASÈNCIA; M. HUISMAN; G. COSTA; A. KUNST; O. ANDERSEN; M. BOPP; J.-K. BORGAN; P. DEBOOSERE; M. GLICKMAN; S. GADEYNE; C. MINDER; E. REGIDOR; T. SPADEA; T. VALKONEN; J.P. MACKENBACH [2005]: Education level inequalities and transportation injury mortality in the middle aged and elderly in European settings. In: *Injury prevention : journal of the International Society for Child and Adolescent Injury Prevention*, 11 (3), S. 138–142

BRACKSTONE, M.; M. MCDONALD; B. SULTAN [1999]: Dynamic Behavioral Data Collection Using an Instrumented Vehicle. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1689, S. 9–16

BREIMAN, L. [1984]: Classification and regression trees, Chapman & Hall/CRC, New York, N.Y.

BREIMAN, L. [2001]: Random Forests. In: *Machine Learning*, 45 (1), S. 5–32

BROWN, T.; G. MILAVETZ; D.J. MURRY [2013]: Alcohol, Drugs and Driving: Implications for Evaluating Driver Impairment. In: *Annals of Advances in Automotive Medicine*, 57, S. 23–32

BUITINCK, L.; G. LOUPPE; M. BLONDEL; F. PEDREGOSA; A. MUELLER; O. GRISEL; V. NICULAE; P. PRETTENHOFER; A. GRAMFORT; J. GROBLER; R. LAYTON; J. VANDERPLAS; A. JOLY; B. HOLT; G. VAROQUAUX [2013]: API design for machine learning software: experiences from the scikit-learn project,*ECML PKDD Workshop 2013*

BURGHOUT, W. [2004]: A note on the number of replication runs in stochastic traffic simulation models.

CAMACHO, E.F.; C. BORDONS [2007]: Model Predictive control, Springer, London.

CANTIN, V.; M. LAVALLIÈRE; M. SIMONEAU; N. TEASDALE [2009]: Mental workload when driving in a simulator: effects of age and driving complexity. In: *Accident; analysis and prevention*, 41 (4), S. 763–771

CARMON, Y. [2007]: Markov Decision Processes with General Discount Functions. faculty of Electrical Engineering, Technion IIT.

CARTER, P.M.; C.R. BINGHAM; J.S. ZAKRAJSEK; J.T. SHOPE; T.B. SAYER [2014]: Social norms and risk perception: predictors of distracted driving behavior among novice adolescent drivers. In: *The Journal of adolescent health : official publication of the Society for Adolescent Medicine*, 54 (5 Suppl), S32-41

CASAS, J.; J.L. FERRER; D. GARCIA; J. PERARNAU; A. TORDAY [2010]: Traffic Simulation with Aimsun. In: BARCELÓ, J. (Eds.): *Fundamentals of Traffic Simulation*, S. 173–232, Springer New York. New York, NY

CASCETTA, E. [2001]: Transportation Systems Engineering: Theory and Methods, Springer, Boston, MA.

CASSIDY, M.J.; J. RUDJANAKANOKNAD [2005]: Increasing the capacity of an isolated merge by metering its on-ramp. In: *Transportation Research Part B: Methodological*, 39 (10), S. 896–913

CHAN, C.-Y.; B. BOUGLER [2005]: Evaluation of cooperative roadside and vehicle-based data collection for assessing intersection conflicts,*IEEE Intelligent Vehicles Symposium 2005.* Las Vegas, NV, USA.

CHANDRASHEKAR, G.; F. SAHIN [2014]: A survey on feature selection methods. In: *Computers & Electrical Engineering*, 40 (1), S. 16–28

CHAWLA, N.V. [2005]: Data Mining for Imbalanced Datasets: An Overview. In: Maimon, Oded and Rokach, Lior (Eds.): *Data Mining and Knowledge Discovery Handbook*, S. 853–867, Springer US. Boston, MA

CHAWLA, N.V.; K.W. BOWYER; L.O. HALL; W.P. KEGELMEYER [2002]: SMOTE: Synthetic Minority Over-sampling Technique. In: *Journal of Artificial Intelligence Research*, 16, S. 321–357

CHEN, W.-Y.; D.-w. HUANG; W.-n. HUANG; W.-l. HWANG [2004]: TRAFFIC FLOW ON A 3-LANE HIGHWAY. In: *International Journal of Modern Physics B*, 18 (31n32), S. 4161–4171

CHEN, Y. [2018]: Learning-based Lane Following and Changing Behaviors for Autonomous Vehicle, Master's thesis, Pennsylvania, PA, USA.

CHEUNG, E.; A. BERA; E. KUBIN; K. GRAY; D. MANOCHA [2018]: Identifying Driver Behaviors using Trajectory Features for Vehicle Navigation. http://arxiv.org/pdf/1803.00881v2.

CHOUDHURY, C.F. [2007]: Modeling driving decisions with latent plans, PhD Thesis, Massachusett.

CHOVAN, J.D.; L. TIJERINA; G. ALEXANDER; D.L. HENDRICKS [1994]: Examination of lane change crashes and potential IVHS countermeasures final report. U.S. Department of Transportation, Cambridge, MA, USA.

CHRISTOPH, M.; N. VAN NES; J.J.A. PAUWELUSSEN; R. MANSVELDERS; A.R.A. VAN DER HORST; D.M. HOEDEMAEKER [2010]: In-vehicle and site-based observations of vehicles and cyclists. A small-scale ND study in the Netherlands. PROLOGUE. TNO Defensie en Veiligheid, Soesterberg, Netherlands.

CLARET, P.L.; J.d.D.L.d. CASTILLO; J.J.J. MOLEÓN; A.B. CAVANILLAS; M.G. MARTÍN; R.G. VARGAS [2003]: Age and sex differences in the risk of causing vehicle collisions in Spain, 1990 to 1999. In: *Accident Analysis & Prevention*, 35 (2), S. 261–272

CŒUGNET, S.; J. NAVETEUR; P. ANTOINE; F. ANCEAUX [2013]: Time pressure and driving: Work, emotions and risks. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 20, S. 39–51

COMBS, T.; L. SANDT; N. MCDONALD; M. CLAMANN [2018]: Limitations in Detection Technologies for Automated Driving Systems and Implications for Pedestrian Safety.

CORTES, C.; V. VAPNIK [1995]: Support-vector networks. In: *Machine Learning*, 20 (3), S. 273–297

CORTÉS-BERRUECO, L.E.; C. GERSHENSON; C.R. STEPHENS [2016]: Traffic Games: Modeling Freeway Traffic with Game Theory. In: *PloS one*, 11 (11), e0165381

CURRY, A.E.; J.H. MIRMAN; M.J. KALLAN; F.K. WINSTON; D.R. DURBIN [2012]: Peer passengers: how do they affect teen crashes? In: *The Journal of adolescent health : official publication of the Society for Adolescent Medicine*, 50 (6), S. 588–594

CYBENKO, G. [1989]: Approximation by superpositions of a sigmoidal function. In: *Mathematics of control, signals and systems*, 2 (4), S. 303–314

D'ORAZIO, T.; M. LEO; C. GUARAGNELLA; A. DISTANTE [2007]: A visual approach for driver inattention detection. In: *Pattern Recognition*, 40 (8), S. 2341–2355

DAAMEN, W.; M. LOOT; S.P. HOOGENDOORN [2010]: Empirical Analysis of Merging Behavior at Freeway On-Ramp. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2188 (1), S. 108–118

DAGANZO, C.F.; M.J. CASSIDY; R.L. BERTINI [1999]: Possible explanations of phase transitions in highway traffic. In: *Transportation Research Part A: Policy and Practice*, 33 (5), S. 365–379

DARIO, P.; G. SANDINI; P. AEBISCHER; G. BENI; J. WANG [1993]: Swarm Intelligence in Cellular Robotic Systems,*Robots and Biological Systems: Towards a New Bionics?* Berlin, Heidelberg.

DAS, S.; B.A. BOWLES [1999]: Simulations of highway chaos using fuzzy logic,*18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS (Cat. No.99TH8397).* New York, NY, USA.

DEFAZIO, A.; F. BACH; S. LACOSTE-JULIEN [2014]: SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. http://arxiv.org/pdf/1407.0202v3.

DING, C.; W. WANG; X. WANG; M. BAUMANN [2013]: A Neural Network Model for Driver's Lane-Changing Trajectory Prediction in Urban Traffic Flow. In: *Mathematical Problems in Engineering*, 2013 (1), S. 1–8

DO, T.-N.; P. LENCA; S. LALLICH; N.-K. PHAM [2010]: Classifying Very-High-Dimensional Data with Random Forests of Oblique Decision Trees. In: GUILLET, F.; G. RITSCHARD; D.A.

ZIGHED; H. BRIAND (Eds.): *Advances in knowledge discovery and management*, S. 39–55, Springer-Verlag. Berlin

DOWLING, R.; A. SKABARDONIS; V. ALEXIADIS [2004]: Traffic Analysis Toolbox Volume III. Guidelines for Applying Traffic Microsimulation Modeling Software

DRAPER, N.R.; H. SMITH [2014]: Applied regression analysis, John Wiley & Sons

DUAN, Y.; M. ANDRYCHOWICZ; B.C. STADIE; J. HO; J. SCHNEIDER; I. SUTSKEVER; P. ABBEEL; W. ZAREMBA [2017]: One-Shot Imitation Learning. http://arxiv.org/pdf/1703.07326v3.

DUAN, Y.; J. SCHULMAN; X. CHEN; P.L. BARTLETT; I. SUTSKEVER; P. ABBEEL [2016]: RL^2: Fast Reinforcement Learning via Slow Reinforcement Learning. http://arxiv.org/pdf/1611.02779v2.

DUMETZ, J. [2016]: Road Behavior and Culture: A Statistical Review. In: *Business Perspectives and Research*, 4 (2), S. 111–117

EDWARDS, H.; A. STORKEY [2016]: Towards a Neural Statistician. In: *arXiv e-prints*

EHMANNS, D. [2003]: Modellierung des taktischen Fahrerverhaltens bei Spurwechselvorgängen. In: *Schriftenreihe Automobiltechnik, ika, RWTH Aachen*, 63, S. 1–134

EITER, T.; H. MANNILA [1994]: Computing Discrete Fréchet Distance. Technical Report CD-TR 94/64, Vienna.

ELFAOUZI, E.; R. BILLOT; P. NURMI; B. HEILMANN [2010]: Effects of adverse weather on traffic and safety: State-of-the-art and a European initiative,*15th International Road Weather Conference.* Québec City, Canada.

ERDMANN, J. [2014]: Lane-changing model in SUMO,*SUMO2014.* Berlin.

ERDMANN, J. [2015]: SUMO's Lane-Changing Model. In: BEHRISCH, M. (Eds.): *Modeling Mobility with Open Data*. *2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014*, S. 105–123, Springer-Verlag. s.l.

ESSA, M.; T. SAYED [2016]: A comparison between PARAMICS and VISSIM in estimating automated field-measured traffic conflicts at signalized intersections. In: *Journal of Advanced Transportation*, 50 (5), S. 897–917

ESTER, M.; H.-P. KRIEGEL; J. SANDER; X. XU [1996]: A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*

EUROPEAN TRANSPORT SAFETY COUNCIL (ETSC) [2016]: Prioritising the Safety Potential of Automated Driving in Europe. https://etsc.eu/wp-content/uploads/2016_automated_driving_briefing_final.pdf.

FAGNANT, D.J.; K. KOCKELMAN [2015]: Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. In: *Transportation Research Part A: Policy and Practice*, 77, S. 167–181

FALCONE, P.; F. BORRELLI; J. ASGARI; H.E. TSENG; D. HROVAT [2007]: Predictive Active Steering Control for Autonomous Vehicle Systems. In: *IEEE Transactions on Control Systems Technology*, 15 (3), S. 566–580

FAN, J.; R. LI [2001]: Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. In: *Journal of the American Statistical Association*, 96 (456), S. 1348–1360

FARAH, H. [2016]: When Do Drivers Abort an Overtaking Maneuver on Two-Lane Rural Roads? In: *Transportation Research Record: Journal of the Transportation Research Board*, 2602 (1), S. 16–25

FARAH, H.; H.N. KOUTSOPOULOS [2014]: Do cooperative systems make drivers' car-following behavior safer? In: *Transportation Research Part C: Emerging Technologies*, 41, S. 61–72

FARAH, H.; E. YECHIAM; S. BEKHOR; T. TOLEDO; A. POLUS [2008]: Association of risk proneness in overtaking maneuvers with impaired decision making. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 11 (5), S. 313–323

FAWCETT, T. [2006]: An introduction to ROC analysis. In: *Pattern recognition letters*, 27 (8), S. 861–874

FAYYAD, U.; G. PIATETSKY-SHAPIRO; P. SMYTH [1966]: From Data Mining to Knowledge Discovery in Databases. In: *AI Magazine*, 17 (3)

FEDERAL HIGHWAY ADMINISTRATION (FHWA) [2006]: Next Generation SIMulation (NGSIM) Fact Sheet. https://www.fhwa.dot.gov/publications/research/operations/its/06135/index.cfm (10.04.2019)

FEDERAL HIGHWAY ADMINISTRATION (FHWA) [2017]: TRAFFIC ANALYSIS TOOLBOX VOLUME VI. https://ops.fhwa.dot.gov/publications/fhwahop08054/sect7.htm (September 9, 2017)

FEDERAL HIGHWAY RESEARCH INSTITUTE (BASt) [2016]: Rechtsfolgen zunehmender Fahrzeugautomatisierung - Legal consequences of increasing vehicle automation,*Forschung Kompakt 11/12,* Bergisch Gladbach, germany.

FEDERAL STATISTICAL OFFICE OF GERMANY (Destatis) [2018]: Traffic Accidents,*Traffic*.

FELLENDORF, M.; P. VORTISCH [2001]: Validation of the Microscopic Traffic Flow Model VISSIM in Different Real-World Situations,*80th Annual Meeting of the Transportation Research Board.* Washington, DC.

FELLENDORF, M.; P. VORTISCH [2010]: Microscopic Traffic Flow Simulator VISSIM, Springer, Newyork, NY.

FERGUSON, D.I.; D.J. BURNETTE [2013]: Mapping active and inactive construction zones for autonomous driving. US9141107B2

FILDES, B.; G. RUMBOLD; A. LEENING [1991]: Speed Behaviour and Drivers' Attitudes to Speeding.

FIX, E.; J.L. HODGES [1989]: Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. In: *International Statistical Review / Revue Internationale de Statistique*, 57 (3), S. 238

FORSCHUNGSGESELLSCHAFT FÜR STRAßEN- UND VERKEHRSWESEN (FGSV) - THE COMMITTEES OF THE ROAD AND TRANSPORTATION RESEARCH ASSOCIATION [2015b]: Handbuch für die Bemessung von Straßenverkehrsanlagen (HBS), Edition 2015, (German Highway Capacity Manual). Forschungsgesellschaft für Straßen- und Verkehrswesen (FGSV), Cologne. Germany.

FRANÇOIS-LAVET, V.; R. FONTENEAU; D. ERNST [2015]: How to Discount Deep Reinforcement Learning: Towards New Dynamic Strategies. http://arxiv.org/pdf/1512.02011v2.

FRÉCHET, M.M. [1906]: Sur quelques points du calcul fonctionnel. In: *Rendiconti del Circolo Matematico di Palermo*, 22 (1), S. 1–72

FRIEDRICH, B. [2016]: The Effect of Autonomous Vehicles on Traffic. In: MAURER, M.; J.C. GERDES; B. LENZ; H. WINNER (Eds.): *Autonomous Driving: Technical, Legal and Social Aspects*, S. 317–334, Springer Berlin Heidelberg. Berlin, Heidelberg

FRITZSCHE, H.-T. [1994]: A model for traffic simulation. In: *Traffic Engineering+ Control*, 35 (5), S. 317–321

FULLER, R.; J.A. SANTOS [2007]: Human factors for highway engineers, Emerald, Bingley.

GAO, Y.; A. GRAY; J.V. FRASCH; T. LIN; E.K. TSENG; J.K. HEDRICK; F. BORRELLI [2012]: Spatial Predictive Control for Agile Semi-Autonomous Ground Vehicles,*American Control Conference (AVEC ' 12).* Montreal,Canada.

GEISTEFELDT, J.; S. GIULIANI [2015]: HBS-konforme Simulation des Verkehrsablaufs auf Autobahnen,*Final Report-Forschungsauftrag FE03.0460/2009/OGB.* Bundesanstalt für Strassenwesen (BASt).

GEISTEFELDT, J.; S. GIULIANI; P. VORTISCH; U. LEYN; R. TRAPP; F. BUSCH; A. RASCHER; N. CELIKKAYA [2014]: Assessment of Level of Service on Freeways by Microscopic Traffic Simulation. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2461, S. 41–49

GENDERS, W.; S. RAZAVI [2016]: Using a Deep Reinforcement Learning Agent for Traffic Signal Control. In: *CoRR*, abs/1611.01142

GEURTS, P.; D. ERNST; L. WEHENKEL [2006]: Extremely randomized trees. In: *Machine Learning*, 63 (1), S. 3–42

GHAHRAMANI, Z. [1998]: Learning dynamic Bayesian networks. In: GILES, C.L.; M. GORI (Eds.): *Adaptive Processing of Sequences and Data Structures: International Summer School on Neural Networks "E.R. Caianiello" Vietri sul Mare, Salerno, Italy September 6-13, 1997 Tutorial Lectures*, S. 168–197, Springer Berlin Heidelberg. Berlin, Heidelberg

GINDELE, T.; S. BRECHTEL; R. DILLMANN [2015]: Learning Driver Behavior Models from Traffic Observations for Decision Making and Planning. In: *IEEE Intelligent Transportation Systems Magazine*, 7 (1), S. 69–79

GIPPS, P.G. [1981]: A behavioural car-following model for computer simulation. In: *Transportation Research Part B: Methodological*, 15 (2), S. 105–111

GIPPS, P.G. [1986]: A model for the structure of lane-changing decisions. In: *Transportation Research Part B: Methodological*, 20 (5), S. 403–414

GONZALEZ, D.S.; J.S. DIBANGOYE; C. LAUGIER [2016]: High-speed highway scene prediction based on driver models learned from demonstrations,*2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC).* Rio de Janeiro, Brazil.

GRAY, A.; Y. GAO; T. LIN; J.K. HEDRICK; H.E. TSENG; F. BORRELLI [2012]: Predictive control for agile semi-autonomous ground vehicles using motion primitives,*2012 American Control Conference (ACC).* Montreal, QC, Canada.

GROEGER, J.A.; I.D. BROWN [1989]: Assessing one's own and others' driving ability: Influences of sex, age, and experience. In: *Accident Analysis & Prevention*, 21 (2), S. 155–168

GRÖMPING, U. [2009]: Variable Importance Assessment in Regression: Linear Regression versus Random Forest. In: *The American Statistician*, 63 (4), S. 308–319

GUGERTY, L.; M. RAKAUSKAS; J. BROOKS [2004]: Effects of remote and in-person verbal interactions on verbalization rates and attention to dynamic spatial scenes. In: *Accident Analysis & Prevention*, 36 (6), S. 1029–1043

GUYON, I.; J. WESTON; S. BARNHILL; V. VAPNIK [2002]: Gene Selection for Cancer Classification using Support Vector Machines. In: *Machine Learning*, 46 (1), S. 389–422

GUZMAN, H.; M. LARRAGA; L. ALVAREZ-ICAZA; F. HUERTA [2014]: A Realistic Two-Lanes Traffic Simulation Model Based on Cellular Automata,*2014 UKSim-AMSS 8th European Modelling Symposium*

HABEL, L.; M. SCHRECKENBERG [2014]: Asymmetric Lane Change Rules for a Microscopic Highway Traffic Model. In: WĄS, J.; G.C. SIRAKOULIS; S. BANDINI (Eds.): *Cellular Automata*, S. 620–629, Springer International Publishing. Cham

HAKAMIES-BLOMQVIST, L.; S. MYNTTINEN; M. BACKMAN; V. MIKKONEN [1999]: Age-related Differences in Driving: Are Older Drivers More Serial? In: *International Journal of Behavioral Development*, 23 (3), S. 575–589

HALATI, A.; H. LIEU; S. WALKER [1997]: CORSIM - CORRIDOR TRAFFIC SIMULATION MODEL,*Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*

HALE, A.R.; J. STOOP; J. HOMMELS [1990]: Human error models as predictors of accident scenarios for designers in road transport systems. In: *Ergonomics*, 33 (10-11), S. 1377–1387

HALKIDI, M.; Y. BATISTAKIS; M. VAZIRGIANNIS [2001]: On Clustering Validation Techniques. In: *Journal of Intelligent Information Systems*, 17 (2/3), S. 107–145

HAMDAR, S.H.; L. QIN; A. TALEBPOUR [2016]: Weather and road geometry impact on longitudinal driving behavior: Exploratory analysis using an empirically supported acceleration modeling framework. In: *Transportation Research Part C: Emerging Technologies*, 67, S. 193–213

HAN, J.; M. KAMBER; J. PEI [2012]: Data mining,*Concepts and techniques,* Elsevier, Morgan Kaufmann, Amsterdam, Netherlands.

HAN, J.; C. MORAGA [1995]: The influence of the sigmoid function parameters on the speed of backpropagation learning. In: MIRA, J.; F. SANDOVAL (Eds.): *From natural to artificial neural computation*. *International Workshop on Artificial Neural Networks, Malaga-Torremolinos, Spain, June 7-9, 1995 : proceedings*, S. 195–201, Springer-Verlag. Berlin, New York

HARDING, J. [2008]: New Behavioral Model for Microscopic Freeway Traffic-Flow Simulation. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2088 (1), S. 10–17

HARTLEY, R.; A. ZISSERMAN [2015]: Multiple view geometry in computer vision, Cambridge Univ. Press, Cambridge.

HARTMANN, M.; N. MOTAMEDIDEHKORDI; S. KRAUSE; S. HOFFMANN; P. VORTISCH; F. BUSCH [2017]: Impact of Automated Vehicles on Capacity of the German Freeway Network,*ITS World Congress.* Montreal.

HASTIE, T.; R. TIBSHIRANI; J.H. FRIEDMAN [2009]: The elements of statistical learning,*Data mining, inference, and prediction,* Springer, New York, NY.

HE, H.; J. BOYD-GRABER; K. KWOK; H.C. DAUMÉ [2016]: Opponent Modeling in Deep Reinforcement Learning,*ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning.* New York, NY, USA.

HE, H.; Y. SHE; J. XIAHOU; J. YAO; J. LI; Q. HONG; Y. JI [2018]: Real-Time Eye-Gaze Based Interaction for Human Intention Prediction and Emotion Analysis: *Proceedings of Computer Graphics International 2018*

HELLINGA, B. [1998]: Requirements for the Calibration of Traffic Simulation Models.

HIDAS, P. [2002]: Modelling lane changing and merging in microscopic traffic simulation. In: *Transportation Research Part C: Emerging Technologies*, 10 (5-6), S. 351–371

HIDAS, P. [2004]: Evaluation of lane changing and merging in microsimulation models,*27th Australasian Transport Research Forum.* Adelaide.

HIDAS, P. [2005]: A functional evaluation of the AIMSUN, PARAMICS and VISSIM microsimulation models. In: *Road & transport research*, 14 (4), S. 45–59

HIDAS, P.; K. BEHBAHANIZADEH [1999]: Microscopic simulation of lane changing under incident conditions,*International Symposium on Transportation and Traffic Theory.* Jerusalem, Israel.

HO, J.; S. ERMON [2016]: Generative Adversarial Imitation Learning. http://arxiv.org/pdf/1606.03476v1.

HOEL, C.-J.; K. WOLFF; L. LAINE [2018]: Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning. http://arxiv.org/pdf/1803.10056v1.

HONG, J.-H.; B. MARGINES; A.K. DEY [2014]: A Smartphone-based Sensing Platform to Model Aggressive Driving Behaviors: *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*

HOOGENDOORN, R.; B. VAN ARERM; S. HOOGENDOOM [2014]: Automated driving, traffic flow efficiency, and human factors: Literature review. In: *Transportation Research Record*, 2422 (1), S. 113–120

HOSMER JR, D.W.; S. LEMESHOW; R.X. STURDIVANT [2013]: Applied logistic regression, John Wiley & Sons

HOU, J.; G.F. LIST; X. GUO [2014a]: New algorithms for computing the time-to-collision in freeway traffic simulation models. In: *Computational intelligence and neuroscience*, 2014, S. 761047

HOU, Y.; E. PRAVEEN; S. CARLOS [2014b]: Modeling Mandatory Lane Changing Using Bayes Classifier and Decision Trees. In: *IEEE Transactions on Intelligent Transportation Systems*, 15 (2), S. 647–655

HUANG, S.H.; D. HELD; P. ABBEEL; A.D. DRAGAN [2018]: Enabling robots to communicate their objectives. In: *Autonomous Robots*, 113 (3), S. 329

HUIKUN BI; TIANLU MAO; ZHAOQI WANG; ZHIGANG DENG [2016]: A Data-driven Model for Lane-changing in Traffic Simulation, Zurich, Switzerland

INFANTES, G.; M. GHALLAB; F. INGRAND [2011]: Learning the behavior model of a robot. In: *Autonomous Robots*, 30 (2), S. 157–177

IPG AUTOMOTIVE GMBH [2015]: CarMaker, IPG Automotive GmbH

IVERS, R.; T. SENSERRICK; S. BOUFOUS; M. STEVENSON; H.-Y. CHEN; M. WOODWARD; R. NORTON [2009]: Novice drivers' risky driving behavior, risk perception, and crash risk: findings from the DRIVE study. In: *American journal of public health*, 99 (9), S. 1638–1644

IVERSEN, H. [2004]: Risk-taking attitudes and risky driving behaviour. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 7 (3), S. 135–150

JÄGERBRAND, A.K.; J. SJÖBERGH [2016]: Effects of weather conditions, light conditions, and road lighting on vehicle speed. In: *SpringerPlus*, 5, S. 505

JAIN, A.K.; R.C. DUBES [1988]: Algorithms for clustering data, Prentice Hall, Englewood Cliffs.

JAPKOWICZ, N. [2000]: The Class Imbalance Problem: Significance and Strategies. In: *Proceedings of the 2000 International Conference on Artificial Intelligence ICAI*

JASKOWIAK, P.A.; D. MOULAVI; A.C.S. FURTADO; R.J.G.B. CAMPELLO; A. ZIMEK; J. SANDER [2016]: On strategies for building effective ensembles of relative clustering validity criteria. In: *Knowledge and Information Systems*, 47 (2), S. 329–354

JAYNES, E.T. [1957]: Information Theory and Statistical Mechanics. In: *Physical Review*, 106 (4), S. 620–630

JIN, C.-J.; V.L. KNOOP; D. LI; L.-Y. MENG; H. WANG [2018]: Discretionary lane-changing behavior: empirical validation for one realistic rule-based model. In: *Transportmetrica A: Transport Science*, 16, S. 1–19

JIN, W.-L. [2013]: A multi-commodity Lighthill–Whitham–Richards model of lane-changing traffic flow. In: *Transportation Research Part B: Methodological*, 57, S. 361–377

JODOGNE, S.; C. BRIQUET; J.H. PIATER [2006]: Approximate Policy Iteration for Closed-Loop Learning of Visual Tasks,*Machine Learning: ECML 2006*

JOLLIFFE, I.T.; J. CADIMA [2016]: Principal component analysis: a review and recent developments. In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 374 (2065)

JURGEN, R. [2012]: V2V/V2I Communications for Improved Road Safety and Efficiency, SAE International, Warrendale.

KAELBLING, L.P.; M.L. LITTMAN; A.W. MOORE [1996]: Reinforcement Learning: A Survey. In: *Journal of Artificial Intelligence Research*, 4, S. 237–285

KAN, S.; L. SUN; X. ZHANG [2009]: Analysis of Drivers' Motivations during Discretionary Lane Changes on a Multi-Lane Freeway: *Logistics*

KANG, K.; H.A. RAKHA [2017]: Game Theoretical Approach to Model Decision Making for Merging Maneuvers at Freeway On-Ramps. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2623, S. 19–28

KANG, K.; H.A. RAKHA [2018]: Modeling Driver Merging Behavior: A Repeated Game Theoretical Approach. In: *Transportation Research Record: Journal of the Transportation Research Board*, 14 (4), 036119811879298

KARJANTO, J.; N.M. YUSOF [2015]: Comfort Determination in Autonomous Driving Style,*AutomotiveUI'15.* Nottingham, UK.

KASPER, D.; G. WEIDL; T. DANG; G. BREUEL; A. TAMKE; A. WEDEL; W. ROSENSTIEL [2012]: Object-Oriented Bayesian Networks for Detection of Lane Change Maneuvers. In: *IEEE Intelligent Transportation Systems Magazine*, 4 (3), S. 19–31

KATHS, H. [2017]: Development of tactical and operational behaviour models for bicyclists based on automated video data analysis, Ph.D., Munich, Germany.

KESKINEN, E.; M. HATAKKA; S. LAAPOTTI; A. KATILA; M. PERÄAHO [2004]: Driver behaviour as a hierarchical system. In: *Traffic and Transport Psychology*, S. 9–29

KESTING, A.; M. TREIBER; D. HELBING [2007]: General Lane-Changing Model MOBIL for Car-Following Models. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1999 (1), S. 86–94

KHETERPAL, N.; K. PARVATE; C. WU; A. KREIDIEH; E. VINITSKY; A. BAYEN [2018]: Flow: Deep Reinforcement Learning for Control in SUMO.

KHOSROSHAHI, A.; E. OHN-BAR; M.M. TRIVEDI [2016]: Surround vehicles trajectory analysis with recurrent neural networks,*2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC).* Rio de Janeiro, Brazil.

KIKUCHI, S.; P. CHAKROBORTY [1992]: CAR-FOLLOWING MODEL BASED ON FUZZY INFERENCE SYSTEM. In: *Transportation Research Record*, no. 1365

KILPELÄINEN, M.; H. SUMMALA [2007]: Effects of weather and weather forecasts on driver behaviour. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 10 (4), S. 288–299

KIM, B.; C.M. KANG; S.H. LEE; H. CHAE; J. KIM; C.C. CHUNG; J.W. CHOI [2017]: Probabilistic Vehicle Trajectory Prediction over Occupancy Grid Map via Recurrent Neural Network. http://arxiv.org/pdf/1704.07049v2.

KIM, S.-J. [2006]: Simultaneous calibration of a microscopic traffic simulation model and OD matrix, Doctoral dissertation

KIM, BEOMJOON AND PINEAU, JOELLE [2016]: Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning. In: *International Journal of Social Robotics*, 8 (1), S. 51–66

KITA, H. [1993]: Effects of merging lane length on the merging behavior at expressway on-ramps,*International Symposium on Transportation and Traffic Theory.* Berkeley, California, USA.

KITA, H. [1999]: A merging–giveway interaction model of cars in a merging section: a game theoretic analysis. In: *Transportation Research Part A: Policy and Practice*, 33 (3-4), S. 305–312

KNOOP, V.L.; C. BUISSON [2014]: Calibration and Validation of Probabilistic Discretionary Lane-Change Models. In: *IEEE Transactions on Intelligent Transportation Systems*, S. 1–10

KNOOP, V.L.; S.P. HOOGENDOORN; Y. SHIOMI; C. BUISSON [2012a]: Quantifying the Number of Lane Changes in Traffic. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2278 (1), S. 31–41

KNOOP, V.L.; R.E. WILSON; C. BUISSON; B. VAN AREM [2012b]: Number of Lane Changes Determined by Splashover Effects in Loop Detector Counts. In: *IEEE Transactions on Intelligent Transportation Systems*, 13 (4), S. 1525–1534

KNOSPE, W.; L. SANTEN; A. SCHADSCHNEIDER; M. SCHRECKENBERG [2002]: A realistic two-lane traffic model for highway traffic. In: *Journal of Physics A: Mathematical and General*, 35 (15), S. 3369–3388

KNOSPE, W.; L. SANTEN; A. SCHADSCHNEIDER; M. SCHRECKENBERG [2004]: Empirical test for cellular automaton models of traffic flow. In: *Phys. Rev. E*, 70 (1), S. 16115

KOBER, J.; J.A. BAGNELL; J. PETERS [2013]: Reinforcement learning in robotics: A survey. In: *The International Journal of Robotics Research*, 32 (11), S. 1238–1274

KOCH, G.R. [2015]: Siamese Neural Networks for One-Shot Image Recognition.

KOHAVI, R. [1995]: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: *International Joint Conference on Artificial Intelligence*, 14

KONDYLI, A.; L. ELEFTERIADOU [2010]: Driver Behavior at Freeway-Ramp Merging Areas. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2124

KONIDARIS, G.D.; G.M. HAYES [2005]: An Architecture for Behavior-Based Reinforcement Learning. In: *Adaptive Behavior*, 13 (1), S. 5–32

KOSONEN, I. [1999]: HUTSIM,*Urban traffic simulation and control model : principles and applications,* Helsinki University of Technology, Espoo.

KOTSIANTIS, S. [2007]: Supervised Machine Learning: A Review of Classification Techniques. In: *Informatica (Ljubljana)*, 31

KOVVALI, V.; V. ALEXIADIS; L. ZHANG [2007]: Video-Based Vehicle Trajectory Data Collection,*86th Transportation Research Board 86th Annual Meeting.* Washington DC, United States.

KRAJZEWICZ, D.; J. ERDMANN; M. BEHRISCH; L. BIEKER [2012]: Recent Development and Applications of SUMO - Simulation of Urban MObility. In: *International Journal On Advances in Systems and Measurements*, 5 (3&4), S. 128–138

KRAJZEWICZ, D.; GEORG HERTKORN; C. RÖSSEL; PETER WAGNER [2002]: SUMO (Simulation of Urban MObility),*An open-source traffic simulation,Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002).* Sharjah, United Arab Emirates.

KRAUSS, S. [1997]: Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics, PhD Thesis, Cologne, Germany.

KRETZSCHMAR, H.; M. KUDERER; W. BURGARD [2014]: Learning to predict trajectories of cooperatively navigating agents,*2014 IEEE International Conference on Robotics and Automation (ICRA).* Hong Kong, China.

KROGSCHEEPERS, C.; K. KACIR [2001]: Latest Trends in Micro Simulation: an Application of the Paramics Model,*20th South African Transport Conference:'Meeting the Transport Challenges in Southern Africa'.* South Africa.

KUANG, Y.; X. QU; J. WENG; A. ETEMAD-SHAHIDI [2015]: How Does the Driver's Perception Reaction Time Affect the Performances of Crash Surrogate Measures? In: *PloS one*, 10 (9), S. e0138617

KUDERER, M.; S. GULATI; W. BURGARD [2015]: Learning driving styles for autonomous vehicles from demonstration,*2015 IEEE International Conference on Robotics and Automation (ICRA).* Seattle, WA.

KUDERER, M.; H. KRETZSCHMAR; C. SPRUNK; W. BURGARD [2012]: Feature-Based Prediction of Trajectories for Socially Compliant Navigation: *Robotics: Science and Systems 2012*

KUEFLER, A.; J. MORTON; T. WHEELER; M. KOCHENDERFER [2017]: Imitating Driver Behavior with Generative Adversarial Networks. http://arxiv.org/pdf/1701.06699v1.

KUGE, N.; T. YAMAMURA; O. SHIMOYAMA; A. LIU [2000]: A driver behavior recognition method based on a driver model framework. SAE Technical Paper.

KUMAGAI, T.; Y. SAKAGUCHI; M. OKUWA; M. AKAMATSU [2003]: Prediction of driving behavior through probabilistic inference: *Proc. 8th Intl. Conf. Engineering Applications of Neural Networks*

LAJUNEN, T.; H. SUMMALA [1995]: Driving experience, personality, and skill and safety-motive dimensions in drivers' self-assessments. In: *Personality and Individual Differences*, 19 (3), S. 307–318

LAROSE, D.T.; C.D. LAROSE [2015]: Data mining methods and models, Wiley-Interscience, Hoboken, NJ.

LAUDE, J.R.; M.T. FILLMORE [2015]: Simulated driving performance under alcohol: Effects on driver-risk versus driver-skill. In: *Drug and alcohol dependence*, 154, S. 271–277

LAURESHYN, A. [2010]: Application of automated video analysis to road user behaviour. Lund University, Lund, Sweden.

LAVAL, J.A.; C.F. DAGANZO [2006]: Lane-changing in traffic streams. In: *Transportation Research Part B: Methodological*, 40 (3), S. 251–264

LAVAL, J.A.; L. LECLERCQ [2008]: Microscopic modeling of the relaxation phenomenon using a macroscopic lane-changing model. In: *Transportation Research Part B: Methodological*, 42 (6), S. 511–522

LAWITZKY, A.; D. ALTHOFF; C.F. PASSENBERG; G. TANZMEISTER; D. WOLLHERR; M. BUSS [2013]: Interactive scene prediction for automotive applications,*2013 IEEE Intelligent Vehicles Symposium (IV)*

LECLERCQ, L.; N. CHIABAUT; J. LAVAL; C. BUISSON [2007]: Relaxation Phenomenon after Lane Changing. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1999 (1), S. 79–85

LEE, S.J.; Z. POPOVIĆ [2010]: Learning Behavior Styles with Inverse Reinforcement Learning,*ACM Trans. Graph. (ACM Transactions on Graphics).* Los Angeles, California.

LEVINE, S.; V. KOLTUN [2012]: Continuous Inverse Optimal Control with Locally Optimal Examples,*ICML'12 Proceedings of the 29th International Coference on International Conference on Machine Learning.* Edinburgh, Scotland.

LEVINE, S.; Z. POPOVIC; V. KOLTUN [2010]: Feature Construction for Inverse Reinforcement Learning. In: J. D. Lafferty; C.K.I. WILLIAMS; J. Shawe-Taylor; R. S. Zemel; A. Culotta (Eds.): *Advances in Neural Information Processing Systems 23*, S. 1342–1350, Curran Associates, Inc

LEVINE, S.; Z. POPOVIC; V. KOLTUN [2011]: Nonlinear inverse reinforcement learning with gaussian processes. In: *Advances in Neural Information Processing Systems*

LI, J.; D.-y. QU; C. LIU; J.-z. WANG; X.-h. XU [2018]: Study on Vehicle Lane-Changing Behavior Based on Cellular Automaton. In: *Journal of Highway and Transportation Research and Development (English Edition)*, 12 (1), S. 75–80

LI, S.; Y. WU; Z. XU; X. LIN [2014]: Improved lane-changing model for vanets in SUMO,*The 7th IEEE/International Conference on Advanced Infocomm Technology.* Fuzhou, China.

LI, X.-G.; B. JIA; Z.-Y. GAO; R. JIANG [2006]: A realistic two-lane cellular automata traffic model considering aggressive lane-changing behavior of fast vehicle. In: *Physica A: Statistical Mechanics and its Applications*, 367, S. 479–486

LIGHTHILL, M.J.; G.B. WHITHAM [1955]: On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 229 (1178), S. 317–345

LIMA AZEVEDO, C. [2014]: Probabilistic Safety Analysis Using Traffic Microscopic Simulation, PhD thesis, Lisbon, Portugal.

LINIGER, A.; A. DOMAHIDI; M. MORARI [2015]: Optimization-based autonomous racing of 1:43 scale RC cars. In: *Optimal Control Applications and Methods*, 36 (5), S. 628–647

LITTMAN, M.L.; T.L. DEAN; L.P. KAELBLING [1995]: On the Complexity of Solving Markov Decision Problems,*Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI1995)*

LIU, P.; A. KURT; K. REDMILL; U. OZGUNER [2016]: Classification of Highway Lane Change Behavior to Detect Dangerous Cut-in Maneuvers,*The Transportation Research Board (TRB) 95th Annual Meeting*

LIU, X.; J. WU; Z. ZHOU [2009]: Exploratory Undersampling for Class-Imbalance Learning. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39 (2), S. 539–550

LOHNINGER, H. [1999]: Teach/Me data analysis,*Single user edition ; [examples from all fields of science, 25 fully interactive applets, 25 animations and slide shows,* Springer, Berlin.

LOPES, M.; F. MELO; L. MONTESANO [2009]: Active Learning for Reward Estimation in Inverse Reinforcement Learning. In: BUNTINE, W.; M. GROBELNIK; D. MLADENIĆ; J. SHAWE-TAYLOR (Eds.): *Machine Learning and Knowledge Discovery in Databases*, S. 31–46, Springer Berlin Heidelberg. Berlin, Heidelberg

LUNENFELD, H. AND G. J. ALEXANDER [1990]: A User's Guide to Positive Guidance (3rd Edition),*FHWA SA-90-017.* Federal Highway Administration (FHWA), Washington, DC, USA.

LUSMANN, J.; D. NEUNZIG; M. WEILKES [1997]: Traffic Simulation with Consideration of Driver Models, Theory and Examples. In: *Vehicle System Dynamics*, 27 (5-6), S. 491–516

LÜTZENBERGER, M. [2014]: A Driver's Mind. In: JANSSENS, D.; A.-U.-H. YASAR; L. KNAPEN (Eds.): *Data science and simulation in transportation research*, S. 182–205, IGI Global. Hershey, Pennsylvania (701 E. Chocolate Ave., Hershey, PA, 17033)

LÜTZENBERGER, M.; S. ALBAYRAK [2014]: Current frontiers in reproducing human driver behavior,*Society for Computer Simulation International 2014 – Proceedings of the 2014 Summer Simulation Multiconference.* Monterey, California.

MACADAM, C.C.; G.E. JOHNSON [1996]: Application of Elementary Neural Networks and Preview Sensors for Representing Driver Steering Control Behaviour. In: *Vehicle System Dynamics*, 25 (1), S. 3–30

MACQUEEN, J. [1967]: Some methods for classification and analysis of multivariate observations: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*

MAHUT, M. [1999]: Behavioural Car Following Models.

MANGAROSKA, K.; K. SHARMA; M. GIANNAKOS; H. TRÆTTEBERG; P. DILLENBOURG [2018]: Gaze Insights into Debugging Behavior Using Learner-centred Analysis: *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*

MARSLAND, S. [2014]: Machine Learning,*An Algorithmic Perspective, Second Edition,* CRC Press, Hoboken.

MARTIUS, G.; C.H. LAMPERT [2016]: Extrapolation and learning equations,*29th Conference on Neural Information Processing Systems (NIPS 2016).* Barcelona, Spain.

MATSUMOTO, M.; T. NISHIMURA [1998]: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. In: *ACM Transactions on Modeling and Computer Simulation*, 8 (1), S. 3–30

MAUCH, M.; M.J. CASSIDY [2002]: Freeway Traffic Oscillations Observations and Predictions,*15th Int. Symp. on Traffic and Transportation Theory*

MCALLISTER, R.; Y. GAL; A. KENDALL; M. VAN DER WILK; A. SHAH; R. CIPOLLA; A. WELLER [2017]: Concrete Problems for Autonomous Vehicle Safety: Advantages of Bayesian Deep Learning,*Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence AI and autonomy track*

MCCARTT, A.T.; V.I. SHABANOVA; W.A. LEAF [2003]: Driving experience, crashes and traffic citations of teenage beginning drivers. In: *Accident Analysis & Prevention*, 35 (3), S. 311–320

MCCULLOCH, W.S.; W. PITTS [1943]: A logical calculus of the ideas immanent in nervous activity. In: *The bulletin of mathematical biophysics*, 5 (4), S. 115–133

MCDONALD, M.; J. WU; M. BRACKSTONE [1997]: Development of a fuzzy logic based microscopic motorway simulation model: *IEEE Conference on Intelligent Transportation Systems*

MCGWIN, G.; C. OWSLEY; K. BALL [1998]: Identifying crash involvement among older drivers: agreement between self-report and state records. In: *Accident Analysis & Prevention*, 30 (6), S. 781–791

MECHANICAL SIMULATION [2015]: CarSim, Mechanical Simulation

MEESMANN, U.; H. MARTENSEN; E. DUPONT [2015]: Impact of alcohol checks and social norm on driving under the influence of alcohol (DUI). In: *Accident; analysis and prevention*, 80, S. 251–261

MENDEL, J.M. [1995]: Fuzzy logic systems for engineering: a tutorial. In: *Proceedings of the IEEE*, 83 (3), S. 345–377

MENG, Q.; J. WENG [2012]: Classification and Regression Tree Approach for Predicting Drivers' Merging Behavior in Short-Term Work Zone Merging Areas. In: *Journal of Transportation Engineering*, 138 (8), S. 1062–1070

MEYER-DELIUS, D.; M. BEINHOFER; W. BURGARD [2011]: Grid-based models for dynamic environments. Dept. of Computer Science, University of Freiburg, Tech. Rep.

MIAO, J.; L. NIU [2016]: A Survey on Feature Selection. In: *Procedia Computer Science*, 91, S. 919–926

MICHIE, D.; M. BAIN; J. HAYES-MICHIE [1990]: Knowledge-based Systems for Industrial Control, IET, Stevenage.

MICHON, J.A. [1985]: A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In: MICHON, J.A. (Eds.): *A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? Human behavior and traffic safety*, S. 485–524, Springer US. New York

MOGHADDAM, Z.R.; M. JEIHANI [2017]: The Effect of Travel Time Information, Reliability, and Level of Service on Driver Behavior Using a Driving Simulator. In: *Procedia Computer Science*, 109, S. 34–41

MORIDPOUR, S.; G. ROSE; M. SARVI [2010a]: Effect of Surrounding Traffic Characteristics on Lane Changing Behavior. In: *Journal of Transportation Engineering*, 136 (11), S. 973–985

MORIDPOUR, S.; M. SARVI; G. ROSE [2010b]: Lane changing models: a critical review. In: *Transportation Letters*, 2 (3), S. 157–173

MORIDPOUR, S.; M. SARVI; G. ROSE; E. MAZLOUMI [2012]: Lane-Changing Decision Model for Heavy Vehicle Drivers. In: *Journal of Intelligent Transportation Systems*, 16 (1), S. 24–35

MOTAMEDIDEHKORDI, N.; S. AMINI; S. HOFFMANN; F. BUSCH; M.R. FITRIYANTI [2017]: Modeling tactical lane-change behavior for automated vehicles: A supervised machine learning approach,*5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*

MOTAMEDIDEHKORDI, N.; M. MARGREITER; T. BENZ [2016]: Effects of Connected Highly Automated Vehicles on the Propagation of Congested Patterns on Freeways,*TRB 95th Annual Meeting.* Washington DC, United States.

MUKADAM, M.; A. COSGUN; A. NAKHAEI; K. FUJIMURA [2017]: Tactical Decision Making for Lane Changing with Deep Reinforcement,*NIPS 2017 Workshop*

MUÑOZ, J.C.; C.F. DAGANZO [2002]: Moving Bottlenecks: A Theory Grounded on Experimental Observation. In: *Transportation and Traffic Theory in the 21st Century*, S. 441–461, Emerald Group Publishing Limited

MUNZER, T.; B. PIOT; M. GEIST; O. PIETQUIN; M. LOPES [2015]: Inverse Reinforcement Learning in Relational Domains: *Proceedings of the 24th International Conference on Artificial Intelligence*

MURPHY, K.P. [2002]: Dynamic Bayesian Networks: Representation, Inference and Learning, Ph.D. thesis, Berkeley, Calif.

NABATILAN, L.B.; F. AGHAZADEH; A.D. NIMBARTE; C.C. HARVEY; S.K. CHOWDHURY [2012]: Effect of driving experience on visual behavior and driving performance under different driving conditions. In: *Cognition, Technology & Work*, 14 (4), S. 355–363

NAGATANI, T. [1993]: Self-organization and phase transition in traffic-flow model of a two-lane roadway. In: *Journal of Physics A: Mathematical and General*, 26 (17), L781

NAGATANI, T. [1994a]: Dynamical jamming transition induced by a car accident in traffic-flow model of a two-lane roadway. In: *Physica A: Statistical Mechanics and its Applications*, 202 (3-4), S. 449–458

NAGATANI, T. [1994b]: Traffic Jam and Shock Formation in Stochastic Traffic-Flow Model of a Two-Lane Roadway. In: *Journal of the Physical Society of Japan*, 63 (1), S. 52–58

NAGEL, K.; M. SCHRECKENBERG [1992]: A cellular automaton model for freeway traffic. In: *Journal de Physique I*, 2 (12), S. 2221–2229

NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA) [2004]: A Comprehensive Examination of Naturalistic Lane-Changes. U.S. Department of Transportation.

NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA) [2013]: Traffic Safety Facts: Distracted Driving 2011. US Department of Transportation, Washington D.C.

NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA) [2015]: Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. NHTSA's National Center for Statistics and Analysis, Washington, DC, USA.

NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION (NHTSA) [2017]: Automated Vehicles for Safety. US Department of Transportation.

NEUMANN, J. von; O. MORGENSTERN [2007]: Theory of games and economic behavior, Princeton Univ. Press, Princeton, NJ.

NG, A.Y.; S.J. RUSSEL [2000]: Algorithms for Inverse Reinforcement Learning,*Proceedings of the seventeenth International Conference on Machine Learning (ICML-2000)*

NG, R.T.; J. HAN [2002]: CLARANS: a method for clustering objects for spatial data mining. In: *IEEE Transactions on Knowledge and Data Engineering*, 14 (5), S. 1003–1016

NIEHAUS, A.; R.F. STENGEL [1991]: Probability-based decision making for automated highway driving,*Vehicle Navigation and Information Systems.* Troy, MI, USA.

NISHIWAKI, Y.; C. MIYAJIMA; N. KITAOKA; R. TERASHIMA; T. WAKITA; K. TAKEDA [2008]: Generating lane-change trajectories of individual drivers,*2008 IEEE International Conference on Vehicular Electronics and Safety*

OGDEN, E.J.D.; H. MOSKOWITZ [2004]: Effects of alcohol and other drugs on driver performance. In: *Traffic injury prevention*, 5 (3), S. 185–198

OSSIANDER, E.M.; P. CUMMINGS [2002]: Freeway speed limits and traffic fatalities in Washington State. In: *Accident Analysis & Prevention*, 34 (1), S. 13–18

OWENS, D.A.; J.M. WOOD; J.M. OWENS [2007]: Effects of age and illumination on night driving: a road test. In: *Human factors*, 49 (6), S. 1115–1131

PADEN, B.; M. CAP; S.Z. YONG; D. YERSHOV; E. FRAZZOLI [2016]: A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles. http://arxiv.org/pdf/1604.07446v1.

PANDE, A.; M. ABDEL-ATY [2006]: Assessment of freeway traffic parameters leading to lane-change related collisions. In: *Accident Analysis & Prevention*, 38 (5), S. 936–948

PANWAI, S.; H. DIA [2005]: Comparative evaluation of microscopic car-following behavior. In: *IEEE Transactions on Intelligent Transportation Systems*, 6 (3), S. 314–325

PAPATHANASOPOULOU, V.; C. ANTONIOU [2015]: Towards data-driven car-following models. In: *Transportation Research Part C: Emerging Technologies*, 55, S. 496–509

PAPATHANASOPOULOU, V.; C. ANTONIOU [2016]: Flexible car-following models incorporating information from adjacent lanes,*2016 IEEE 19th International Conference.* Rio de Janeiro, Brazil.

PARK, M.; K. JANG; J. LEE; H. YEO [2015]: Logistic regression model for discretionary lane changing under congested traffic. In: *Transportmetrica A: Transport Science*, 11 (4), S. 333–344

PASETTO, M.; S.D. BARBATI [2011]: How the Interpretation of Drivers' Behavior in Virtual Environment Can Become a Road Design Tool: A Case Study. In: *Advances in Human-Computer Interaction*, 2011 (1), S. 1–10

PATIRE, A.D.; M.J. CASSIDY [2011]: Lane changing patterns of bane and benefit: Observations of an uphill expressway. In: *Transportation Research Part B: Methodological*, 45 (4), S. 656–666

PATTEN, C.J.D.; A. KIRCHER; J. OSTLUND; L. NILSSON; O. SVENSON [2006]: Driver experience and cognitive workload in different traffic environments. In: *Accident Analysis & Prevention*, 38 (5), S. 887–894

PEARSON, K. [1895]: Note on Regression and Inheritance in the Case of Two Parents. In: *Proceedings of the Royal Society of London (1854-1905)*, 58 (-1), S. 240–242

PEARSON, K. [1901]: LIII. On lines and planes of closest fit to systems of points in space. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2 (11), S. 559–572

PÊCHER, C.; C. LEMERCIER; J.-M. CELLIER [2011]: The Influence of Emotions on Driving Behavior, Nova Science Pub Inc

PEDREGOSA, F.; G. VAROQUAUX; A. GRAMFORT; V. MICHEL; B. THIRION; O. GRISEL; M. BLONDEL; P. PRETTENHOFER; R. WEISS; V. DUBOURG; J. VANDERPLAS; A. PASSOS; D. COURNAPEAU; M. BRUCHER; M. PERROT; E. DUCHESNAY [2011]: Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research*, 12, S. 2825–2830

PELLEGRINI, S.; A. ESS; K. SCHINDLER; L. VAN GOOL [2009]: You'll never walk alone: Modeling social behavior for multi-target tracking. In: *IEEE 12th International Conference on Computer Vision*, S. 261–268

PIERGIOVANNI, A.J.; A. WU; M.S. RYOO [2018]: Learning Real-World Robot Policies by Dreaming. http://arxiv.org/pdf/1805.07813v3.

PIOT, B.; M. GEIST; O. PIETQUIN [2017]: Bridging the Gap Between Imitation Learning and Inverse Reinforcement Learning. In: *IEEE transactions on neural networks and learning systems*, 28 (8), S. 1814–1826

POURABDOLLAH, M.; E. BJARKVIK; F. FURER; B. LINDENBERG; K. BURGDORF [2017]: Calibration and evaluation of car following models using real-world driving data,*2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC).* Yokohama, Japan.

PRAKASH, Y.; K. PRABHU; S. KAMTEKAR; S. GADHE [2014]: Incorporation of Swarm Intelligence in Autonomous Cars. In: *International Journal of Computer Science and Information Technologies*, 5 (5)

PREVITALI, F.; A. BORDALLO; S. RAMAMOORTHY [2015]: IRL-based Prediction of Goals for Dynamic Environments,*IEEE International Conference on Robotics and Automation (ICRA), Workshop on Machine Learning for Social Robotics.* Seattle, Washington.

PTV AG [2016]: VISSIM 9 Handbuch.

PTV GROUP [2018]: PTV VISSIM 10 - Introduction to COM API.

QIAN, X.; F. ALTCHE; P. BENDER; C. STILLER; A. de LA FORTELLE [2016]: Optimal trajectory planning for autonomous driving integrating logical constraints: An MIQP perspective,*2016 IEEE 19th International Conference*

QUADSTONE LTD. [2018]: PARAMICS User Guide v21. Quadstone LTD., Edinburgh, UK.

QUINLAN, J.R. [1993]: C4.5: programs for machine learning, Kaufmann, San Mateo, Calif.

RABINER, L.R.; B.H. JUANG [1986]: An introduction to hidden Markov models. In: *IEEE ASSP Magazine*, 3, S. 4–16

RAHMAN, M.; M. CHOWDHURY; Y. XIE; Y. HE [2013]: Review of Microscopic Lane-Changing Models and Future Research Opportunities. In: *IEEE Transactions on Intelligent Transportation Systems*, 14 (4), S. 1942–1956

RASMUSSEN, J. [1987]: New technology and human error, Wiley, Chichester.

RATLIFF, N.D.; J.A. BAGNELL; M. ZINKEVICH [2006]: Maximum margin planning,*International Conference on Machine Learning*

RATNER, E.; D. HADFIELD-MENELL; A.D. DRAGAN [2018]: Simplifying Reward Design through Divide-and-Conquer. http://arxiv.org/pdf/1806.02501v1.

RCE SYSTEMS [2018]: DataFromSky, RCE systems

REDSHAW, S. [2011]: Driving Cultures: Cars, Young People and Cultural Research. In: *Cultural Studies Review*, 12 (2), S. 74

REGHELIN, R.; L.V.R. de ARRUDA [2012]: A centralized traffic controller for intelligent vehicles in a segment of a multilane highway,*2012 IEEE Intelligent Vehicles Symposium*

REICHEL, M.; M. BOTSCH; R. RAUSCHECKER; K.-H. SIEDERSBERGER; M. MAURER [2010]: Situation Aspect Modelling and Classification Using the Scenario Based Random Forest Algorithm for Convoy Merging Situations,*2010 13th International IEEE Conference on Intelligent Transportation Systems - (ITSC 2010).* Funchal, Madeira Island, Portugal.

REICHEL, M.; M. BOTSCH; R. RAUSCHECKER; K.-H. SIEDERSBERGER; M. MAURER [2012]: Situation aspect modelling and classification using the Scenario Based Random Forest algorithm for convoy merging situations,*Situation Aspect Modelling and Classification*

RENDON-VELEZ, E.; P.M. VAN LEEUWEN; R. HAPPEE; I. HORVÁTH; W.F. VAN DER VEGTE; J.C.F. de WINTER [2016]: The effects of time pressure on driver performance and physiological activity: A driving simulator study. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 41, S. 150–169

REZENDE, D.J.; S. MOHAMED; I. DANIHELKA; K. GREGOR; D. WIERSTRA [2016]: One-Shot Generalization in Deep Generative Models. http://arxiv.org/pdf/1603.05106v2.

RICKERT, M.; K. NAGEL; M. SCHRECKENBERG; A. LATOUR [1996]: Two lane traffic simulations using cellular automata. In: *Physica A: Statistical Mechanics and its Applications*, 231 (4), S. 534–550

RIVOIRARD, L.; M. WAHL; P. SONDI; M. BERBINEAU; D. GRUYER [2016]: Using Real-World Car Traffic Dataset in Vehicular Ad Hoc Network Performance Evaluation. In: *International Journal of Advanced Computer Science and Applications*, 7 (12)

ROBERT TIBSHIRANI [1994]: Regression Shrinkage and Selection Via the Lasso. In: *Journal of the Royal Statistical Society, Series B*, 58, S. 267–288

ROCHEFORT, Y.; S. BERTRAND; H. PIET-LAHANIER; D. BEAUVOIS; D. DUMUR [2012]: Cooperative Nonlinear Model Predictive Control for Flocks of Vehicles. In: *IFAC Proceedings Volumes*, 45 (1), S. 169–174

ROCK, S.M. [1995]: Impact of the 65 mph speed limit on accidents, deaths, and injuries in illinois. In: *Accident Analysis & Prevention*, 27 (2), S. 207–214

ROSS, S.; J.A. BAGNELL [2010]: Efficient Reductions for Imitation Learning: *AISTATS*

ROUSSEEUW, P.J. [1987]: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. In: *Journal of Computational and Applied Mathematics*, 20, S. 53–65

RUNDMO, T.; H. IVERSEN [2004]: Risk perception and driving behaviour among adolescents in two Norwegian counties before and after a traffic safety campaign. In: *Safety Science*, 42 (1), S. 1–21

SADIGH, D.; S. SASTRY; S. A. SESHIA; A. D. DRAGAN [2016a]: Planning for Autonomous Cars that Leverage Effects on Human Actions: *Robotics: Science and Systems XII*

SADIGH, D.; S.S. SASTRY; S.A. SESHIA; A. DRAGAN [2016b]: Information gathering actions over human internal state,*2016 IEEE/RSJ International Conference*

SAE INTERNATIONAL [2016]: Automated Driving.

SAHOO, S.S.; C.H. LAMPERT; G. MARTIUS [2018]: Learning Equations for Extrapolation and Control,*Proceedings of the 35 th International Conference on Machine Learning.* Stockholm, Sweden.

SALLAB, A.; M. ABDOU; E. PEROT; S. YOGAMANI [2017]: Deep Reinforcement Learning framework for Autonomous Driving. In: *Electronic Imaging*, 2017 (19), S. 70–76

SALVENDY, G. [2012]: Handbook of Human Factors and Ergonomics, John Wiley & Sons, Inc, Hoboken, NJ, USA.

SALVUCCI, D.; E. BOER; A. LIU [2001]: Toward an Integrated Model of Driver Behavior in Cognitive Architecture. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1779, S. 9–16

SALVUCCI, D.D. [2006]: Modeling driver behavior in a cognitive architecture. In: *Human factors*, 48 (2), S. 362–380

SÁNCHEZ-MAROÑO, N.; A. ALONSO-BETANZOS; M. TOMBILLA-SANROMÁN [2007]: Filter Methods for Feature Selection - A Comparative Study: *Intelligent Data Engineering and Automated Learning - IDEAL 2007*

SANTORO, A.; S. BARTUNOV; M. BOTVINICK; D. WIERSTRA; T. LILLICRAP [2016a]: One-shot Learning with Memory-Augmented Neural Networks. In: *CoRR*, abs/1605.06065

SANTORO, A.; S. BARTUNOV; M. BOTVINICK; D. WIERSTRA; T. P. LILLICRAP [2016b]: Meta-Learning with Memory-Augmented Neural Networks: *ICML*

SASOH, A.; T. OHARA [2002]: Shock Wave Relation Containing Lane Change Source Term for Two-Lane Traffic Flow. In: *Journal of the Physical Society of Japan*, 71 (9), S. 2339–2347

SAYER, J.; J. DEVONSHIRE; C. FLANNAGAN [2007]: Naturalistic Driving Performance During Secondary Tasks: *Driver Behavior at Freeway-Ramp Merging Areas*

SCHAKEL, W.J.; V.L. KNOOP; B. VAN AREM [2012]: Integrated Lane Change Model with Relaxation and Synchronization. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2316 (1), S. 47–57

SCHLECHTRIEMEN, J.; A. WEDEL; J. HILLENBRAND; G. BREUEL; K.-D. KUHNERT [2014]: A lane change detection approach using feature ranking with maximized predictive power. In: *2014 IEEE Intelligent Vehicles Symposium Proceedings*, S. 108–114

SCHLECHTRIEMEN, J.; F. WIRTHMUELLER; A. WEDEL; G. BREUEL; K. KUHNERT [2015]: When will it change the lane? A probabilistic regression approach for rarely occurring events,*2015 IEEE Intelligent Vehicles Symposium (IV)*

SCHMIDT, M.; N. LE ROUX; F. BACH [2017]: Minimizing finite sums with the stochastic average gradient. In: *Mathematical Programming*, 162 (1-2), S. 83–112

SCHUBERT, R. [2012]: Evaluating the Utility of Driving: Toward Automated Decision Making Under Uncertainty. In: *IEEE Transactions on Intelligent Transportation Systems*, 13 (1), S. 354–364

SCHUBERT, R.; K. SCHULZE; G. WANIELIK [2010]: Situation Assessment for Automatic Lane-Change Maneuvers. In: *IEEE Transactions on Intelligent Transportation Systems*, 11 (3), S. 607–616

SCHWARTING, W.; P. PASCHEKA [2014]: Recursive conflict resolution for cooperative motion planning in dynamic highway traffic,*17th International IEEE Conference*

SHALEV-SHWARTZ, S.; S. SHAMMAH; A. SHASHUA [2016]: Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. http://arxiv.org/pdf/1610.03295v1.

SHARIFZADEH, S.; I. CHIOTELLIS; R. TRIEBEL; D. CREMERS [2016]: Learning to Drive using Inverse Reinforcement Learning and Deep Q-Networks. http://arxiv.org/pdf/1612.03653v2.

SHIMOSAKA, M.; J. SATO; K. TAKENAKA; K. HITOMI [2017]: Fast Inverse Reinforcement Learning with Interval Consistent Graph for Driving Behavior Prediction: *AAAI*

SHINAR, D. [2017]: Traffic Safety and Human Behavior, Emerald Publishing Limited

SHINAR, D.; N. TRACTINSKY; R. COMPTON [2005]: Effects of practice, age, and task demands, on interference from a phone task while driving. In: *Accident Analysis & Prevention*, 37 (2), S. 315–326

SHTEINGART, H.; Y. LOEWENSTEIN [2014]: Reinforcement learning and human behavior. In: *Current opinion in neurobiology*, 25, S. 93–98

SHYHALLA, K. [2014]: Alcohol involvement and other risky driver behaviors: effects on crash initiation and crash severity. In: *Traffic injury prevention*, 15 (4), S. 325–334

SIMONS-MORTON, B.G.; M.C. OUIMET; Z. ZHANG; S.E. KLAUER; S.E. LEE; J. WANG; R. CHEN; P. ALBERT; T.A. DINGUS [2011]: The effect of passengers and risk-taking friends on risky driving and crashes/near crashes among novice teenagers. In: *The Journal of adolescent health : official publication of the Society for Adolescent Medicine*, 49 (6), S. 587–593

SIROKY, D.S. [2009]: Navigating Random Forests and related advances in algorithmic modeling. In: *Statist. Surv.*, 3, S. 147–163

SOKOLOVA, M.; G. LAPALME [2009]: A systematic analysis of performance measures for classification tasks. In: *Information Processing & Management*, 45 (4), S. 427–437

SORIGUERA, F.; I. MARTÍNEZ; M. SALA; M. MENÉNDEZ [2017]: Effects of low speed limits on freeway traffic flow. In: *Transportation Research Part C: Emerging Technologies*, 77, S. 257–274

SPARMANN, U. [1978]: Spurwechselvorgänge auf zweispurigen BAB-Richtungsfahrbahnen, Forschung Straßenbau und Straßenverkehrstechnik, Bonn.

SPEARMAN, C. [1904]: The proof and measurement of association between two things. In: *The American journal of psychology*, 15 (1), S. 72–101

ST-AUBIN, P.; N. SAUNIER; L. MIRANDA-MORENO [2015]: Large-scale automated proactive road safety analysis using video data. In: *Transportation Research Part C: Emerging Technologies*, 58, S. 363–379

STEINHAUSER, K.; F. LEIST; K. MAIER; V. MICHEL; N. PÄRSCH; P. RIGLEY; F. WURM; M. STEINHAUSER [2018]: Effects of emotions on driving behavior. In: *Transportation Research Part F: Traffic Psychology and Behaviour*, 59, S. 150–163

STUTTS, J.; D. REINFURT; L. STAPLIN; E. RODGMAN [2001]: The role of driver distraction in traffic crashes. AAA Foundation for Traffic Safety, Washington D.C.

SUMMALA, H. [1996]: Accident risk and driver behaviour. In: *Safety Science*, 22 (1-3), S. 103–117

SUMMALA, H. [1997]: Hierarchical model of behavioural adaptation and traffic accidents. In: *Traffic and transport psychology. Theory and application*

SUTTON, R.S.; A.G. BARTO [1998]: Reinforcement Learning: An Introduction. In: *IEEE Transactions on Neural Networks*, 9 (5), S. 1054

SWOV INSTITUTE FOR ROAD SAFETY RESEARCH [2013]: UDrive Project.

TALEBPOUR, A.; H.S. MAHMASSANI; S.H. HAMDAR [2015]: Modeling Lane-Changing Behavior in a Connected Environment. A Game Theory Approach. In: *Transportation Research Procedia*, 7, S. 420–440

TAN, P.-N.; M. STEINBACH; V. KUMAR [2005]: Intro to Data Mining, P.Ed Australia, New Jersey.

TANG, J.; S. ALELYANI; H. LIU [2014]: Feature Selection for Classification: A Review: *Data Classification: Algorithms and Applications*

TANZMEISTER, G.; J. THOMAS; D. WOLLHERR; M. BUSS [2014]: Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation: *IEEE International Conference on Robotics and Automation (ICRA), 2014*

TAO, D.; R. ZHANG; X. QU [2017]: The role of personality traits and driving experience in self-reported risky driving behaviors and accident risk among Chinese drivers. In: *Accident; analysis and prevention*, 99 (Pt A), S. 228–235

TAUBMAN-BEN-ARI, O.; M. MIKULINCER; O. GILLATH [2004]: The multidimensional driving style inventory—scale construct and validation. In: *Accident Analysis & Prevention*, 36 (3), S. 323–332

TESIS [2018]: DYNA4, TESIS

THARWAT, A. [2018]: Classification assessment methods. In: *Applied Computing and Informatics*

THE ROYAL SOCIETY [2017]: Machine learning,*The power and promise of computers that learn by example*.

THEODORIDIS, S.; K. KOUTROUMBAS [2009]: Pattern recognition, Elsevier/Acad. Press, Amsterdam [u.a.].

THIEMANN, C.; M. TREIBER; A. KESTING [2008]: Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data. In: *Transportation Research Record: Journal of the Transportation Research Board*, 2088, S. 90–101

THORNDIKE, R.L. [1953]: Who belongs in the family? In: *Psychometrika*, 18 (4), S. 267–276

TIELERT, T.; M. KILLAT; H. HARTENSTEIN; R. LUZ; S. HAUSBERGER; T. BENZ [2010]: The impact of traffic-light-to-vehicle communication on fuel consumption and emissions,*2010 Internet of Things IOT*

TOLEDO, T.; H. KOUTSOPOULOS [2004]: Statistical Validation of Traffic Simulation Models. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1876, S. 142–150

TOLEDO, T.; H. KOUTSOPOULOS; M. BEN-AKIVA [2003]: Modeling Integrated Lane-Changing Behavior. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1857, S. 30–38

TOLEDO, T.; D. ZOHAR [2007]: Modeling Duration of Lane Changes. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1999, S. 71–78

TREIBER, M. [2002]: Realistische Mikrosimulation von Straßenverkehr mit einem einfachen modell,*Proc. 16th Symp. Simulationstechnik ASIM 2002 Rostock*

TREIBER, M.; A. KESTING [2009]: Modeling Lane-Changing Decisions with MOBIL. In: APPERT-ROLLAND, C.; F. CHEVOIR; P. GONDRET; S. LASSARRE; J.-P. LEBACQUE; M. SCHRECKENBERG (Eds.): *Traffic and Granular Flow '07*, S. 211–221, Springer Berlin Heidelberg. Berlin, Heidelberg

TURRI, V.; A. CARVALHO; H.E. TSENG; K.H. JOHANSSON; F. BORRELLI [2013]: Linear model predictive control for lane keeping and obstacle avoidance on low curvature roads,*16th International IEEE Conference*

ULBRICH, S.; M. MAURER [2015]: Situation Assessment in Tactical Lane Change Behavior Planning for Automated Vehicles,*IEEE 18th international conference*

US DEPARTMENT OF TRANSPORTATION : A Comprehensive Examination ofNaturalistic Lane-Changes.

VALLON, C.; Z. ERCAN; A. CARVALHO; F. BORRELLI [2017]: A machine learning approach for personalized autonomous lane change initiation and control,*2017 IEEE Intelligent Vehicles Symposium*

VAN AERDE, M.; B. HELLINGA; M. BAKER; H. RAKHA [1992]: INTEGRATION: An overview of traffic simulation features.

VAN BEINUM, A.; M. HOVENGA; V. KNOOP; H. FARAH; F. WEGMAN; S. HOOGENDOORN [2018]: Macroscopic traffic flow changes around ramps. In: *Transportmetrica A: Transport Science*, 14 (7), S. 598–614

VAN DER MOLEN, H.H.; A.M.T. BÖTTICHER [1988]: A hierarchical risk model for traffic participants. In: *Ergonomics*, 31 (4), S. 537–555

VAN HUYSDUYNEN, H.H.; J. TERKEN; J.-B. MARTENS; B. EGGEN [2015]: Measuring driving styles: *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI '15*

VAN SCHAGEN, I.; R. WELSH; A. BACKER-GRØ NDHAL; M. HOEDEMAEKER; T. LOTAN; A. MORRIS [2011]: owards a large-scale European Naturalistic Driving study : final report of PROLOGUE. SWOV Institute for Road Safety Research, Netherlands.

VASQUEZ, D.; Y. YU; S. KUMAR; C. LAUGIER [2014]: An Open Framework for Human-Like Autonomous Driving Using Inverse Reinforcement Learning,*2014 IEEE Vehicle Power and Propulsion Conference (VPPC).* Coimbra, Portugal.

VERHULST, P.F. [1838]: Notice sur la loi que la population poursuit dans son accroissement. In: *Correspondance mathématique et physique*, 10, S. 113–121

VINKHUYZEN, E.; M. CEFFKIN [2016]: Developing Socially Acceptable Autonomous Vehicles. In: *Ethnographic Praxis in Industry Conference Proceedings*, 2016 (1), S. 522–534

VINYALS, O.; C. BLUNDELL; T. LILLICRAP; K. KAVUKCUOGLU; D. WIERSTRA [2016]: Matching Networks for One Shot Learning. http://arxiv.org/pdf/1606.04080v2.

WAGNER, P.; K. NAGEL; D.E. WOLF [1997]: Realistic multi-lane traffic rules for cellular automata. In: *Physica A: Statistical Mechanics and its Applications*, 234 (3-4), S. 687–698

WAN, E.A. [1990]: Neural network classification: a Bayesian interpretation. In: *IEEE Transactions on Neural Networks*, 1 (4), S. 303–305

WANG, M.; S.P. HOOGENDOORN; W. DAAMEN; B. VAN AREM; R. HAPPEE [2015]: Game theoretic approach for predictive lane-changing and car-following control. In: *Transportation Research Part C: Emerging Technologies*, 58, S. 73–92

WANG, M.S.; N.T. JEONG; K.S. KIM; S.B. CHOI; S.M. YANG; S.H. YOU; J.H. LEE; M.W. SUH [2016]: Drowsy behavior detection based on driving information. In: *International Journal of Automotive Technology*, 17 (1), S. 165–173

WANG, P.; C.-Y. CHAN [2017]: Formulation of Deep Reinforcement Learning Architecture Toward Autonomous Driving for On-Ramp Merge,*IEEE International Conference on Intelligent Transportation Systems*

WANG, P.; C.-Y. CHAN; A. de LA FORTELLE [2018]: A Reinforcement Learning Based Approach for Automated Lane Change Maneuvers,*2018 IEEE Intelligent Vehicle Symposium*

WANG, P.; C.-Y. CHAN; H. LI [2019]: Automated Driving Maneuvers Under Interactive Environment Based on Deep Reinforcement Learning.

WANG, X. [2015]: A State Dependent Lane-Changing Model for Urban Arterials with Hidden Markov Model Method, Master's thesis

WEGENER, A.; M. PIÓRKOWSKI; M. RAYA; H. HELLBRÜCK; S. FISCHER; J.-P. HUBAUX [2008]: TraCI:An Interface for Coupling Road Traffic and Network Simulators,*Proceedings of the 11th Communications and Networking Simulation Symposium.* Ottawa, Canada.

WHEELER, T.R. [2014]: Probabilistic Driving Models and Lane Change Prediction, Stanford University

WIEDEMANN, R. [1974]: Simulation des Verkehrsflusses. Shriftenreihe des IfV, Universität Karlsruhe (TH)-IfV

WILLIAMS, A.F. [2003]: Teenage drivers: patterns of risk. In: *Journal of safety research*, 34 (1), S. 5–15

WISCONSIN DOT [2002]: Freeway System Operational Assessment. Paramics Calibration and Validation Guidelines

WULFMEIER, M.; P. ONDRUSKA; I. POSNER [2015]: Maximum Entropy Deep Inverse Reinforcement Learning. http://arxiv.org/pdf/1507.04888v3.

XU, H.; M. XU [2016]: A cellular automata traffic flow model based on safe lane-changing distance constraint rule,*2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC).* Xi'an, China.

XU, L.; J. HU; H. JIANG; W. MENG [2015]: Establishing Style-Oriented Driver Models by Imitating Human Driving Behaviors. In: *IEEE Transactions on Intelligent Transportation Systems*, 16 (5), S. 2522–2530

YANG, H.; X. WANG; Y. YIN [2012]: The impact of speed limits on traffic equilibrium and system performance in networks. In: *Transportation Research Part B: Methodological*, 46 (10), S. 1295–1307

YANG, L.; X. LI; W. GUAN; H.M. ZHANG; L. FAN [2018]: Effect of traffic density on drivers' lane change and overtaking maneuvers in freeway situation-A driving simulator-based study. In: *Traffic injury prevention*, 19 (6), S. 594–600

YANG, Q.I.; H.N. KOUTSOPOULOS [1996]: A Microscopic Traffic Simulator for evaluation of dynamic traffic management systems. In: *Transportation Research Part C: Emerging Technologies*, 4 (3), S. 113–129

YOO, J.H. [2014]: A game theory based model of human driving with application to autonomous and mixed driving, PhD thesis, Texas.

ZHANG, Y.; L. OWEN; J. CLARK [1998]: Multiregime Approach for Microscopic Traffic Simulation. In: *Transportation Research Record: Journal of the Transportation Research Board*, 1644, S. 103–114

ZHAO, X.; X. ZHANG; J. RONG [2014]: Study of the Effects of Alcohol on Drivers and Driving Performance on Straight Road. In: *Mathematical Problems in Engineering*, 2014 (1), S. 1–9

ZHENG, Z.; S. AHN; D. CHEN; J. LAVAL [2011]: Freeway traffic oscillations: Microscopic analysis of formations and propagations using Wavelet Transform. In: *Transportation Research Part B: Methodological*, 45 (9), S. 1378–1388

ZIEBART, B.D. [2010]: Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy, Pittsburgh.

ZIEBART, B.D.; A. MAAS; J.A. BAGNELL; A. K. DEY [2008]: Maximum Entropy Inverse Reinforcement Learning. In: *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*

ZOU, K.H. [2002]: Receiver operating characteristic (ROC) literature research. In: *On-line bibliography available from:< http://splweb. bwh. harvard. edu*, 8000

List of Figures

List of Tables

Glossary

Automated vehicle	The vehicle that is capable of sensing its environment and moving with little or no human input.
Feature	An attribute or group of attributes that constitute a characteristic, property or set of properties which is measurable and differentiable.
Mixed-traffic	Traffic flow in which the automated vehicles operate in conjunction with human-driven vehicles.
Operational behavior	At this level, maneuvers are converted to control operations and the basics of steering and braking come directly into the play.
Principal component	Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables in principle component analysis.
Socially acceptable driving behavior	Socially acceptable driving behavior is the behavior in which automated vehicles, when interacting with other road users, operate smoothly and in an appropriate interactional manner.
Tactical behavior	Behavior at tactical level deals with choices and actions related to the interaction with the road and other road users, such as keeping the distance, changing the lane, and taking a turn.

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BC	Behavioral Cloning
BPNN	Back Propagation Neural Network
DBN	Dynamic Bayesian Network
DRL	Deep Reinforcement Learning
DLC	Discretionary Lane Change
FPR	False Positive Rate
HGV	Heavy Goods Vehicle
HMM	Hidden Markov Model
IDM	Intelligent Driver Model
IL	Imitation Learning
IRL	Inverse Reinforcement Learning
kNN	k-Nearest Neighbour
LMRS	Lane changing Model with Relaxation and Synchronization
LSTM	Long Short-Term Memory
MaxEnt IRL	Maximum Entropy Inverse Reinforcement Learning
MAE	Mean Absolute Error
MDP	Markov Decision Process
ML	Machine Learning
MLC	Mandatory Lane Change
MOBIL	Minimizing Overall Braking Induced by Lane change
MSE	Mean Squared Error

NGSIM	Next Generation SIMulation
POMDP	Partially Observable Markov Decision Process
RBF	Radial Basis Function
RF	Random Forest
RFE	Recursive Feature Elimination
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
SUMO	Simulation of Urban Mobility
SVM	Support Vector Machine
TRP	True Positive Rate
TraCI	Traffic Control Interface
TTC	Time-To-Collision
VISSIM	Verkehr In Städten - SIMulationsmodell

Appendix 1

model	Attribute	Description	Unit	Parameter Range
Car-following	Acceleration	The acceleration ability of vehicles	$\frac{m}{s^2}$	[2.5 , 4]
	Deceleration	The deceleration ability of vehicles	$\frac{m}{s^2}$	[-4 , -1.5]
	Emergency Deceleration	The maximum deceleration ability of vehicles in case of emergency	$\frac{m}{s^2}$	[5 , 8]
	Sigma	The driver imperfection (between 0 and 1)	-	[0.3 , 0.8]
	Tau	The driver's desired minimum time headway. Exact interpretation varies by model.	s	[0.8 , 2]
	Minimum Gap	The offset to the leading vehicle when standing in a jam.	m	[1.5 , 3.5]
	Maximum Speed	The maximum speed that a vehicle will travel	$\frac{m}{s}$	[15 , 50]
	Speed Factor	The vehicles expected multiplicator for lane speed limits	-	[1 , 1.4]
	Speed Deviation	The deviation of the speed factor	-	[0.1 , 0.3]
Lane change	Lane change Strategic	The eagerness for performing strategic lane changing. Higher values result in earlier lane-changing.	-	[0 , 3]
	Lane change Cooperative	The willingness for performing cooperative lane changing. Lower values result in reduced cooperation.	-	[0.3 , 1]
	Lane change Speed Gain	The eagerness for performing lane changing to gain speed. Higher values result in more lane-changing.	-	[0 , 3]
	Lane change Keep Right	The eagerness for following the obligation to keep right. Higher values result in earlier lane-changing.	-	[0 , 3]
	Lane change Look ahead Left	Factor for configuring the strategic lookahead distance when a change to the left is necessary.	-	[0 , 3]
	Lane change Speed Gain Right	Factor for configuring the threshold asymmetry when changing to the left or to the right for speed gain. By default the decision for changing to the right takes more deliberation.	-	[0 , 0.5]
	Lane change Assertive	Willingness to accept lower front and rear gaps on the target lane. The required gap is divided by this value.	-	[1 , 3]

Tab. 7.1 List of parameters in SUMO default models and range used for model calibration