Improved machine learning approaches for individualized human assistance, supervision, and behavior prediction

Von der Fakultät für Ingenieurwissenschaften,
Abteilung Maschinenbau und Verfahrenstechnik
der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades
einer
Doktorin der Ingenieurwissenschaften
Dr.-Ing.
genehmigte Dissertation

von

Qi Deng
aus
Guizhou, China

Gutachter: Univ.-Prof. Dr.-Ing. Dirk Söffker
Univ.-Prof. Dr. rer. nat. Johannes Gottschling

Tag der mündlichen Prüfung: 22.12.2020

To my husband and child

Acknowledgements

This thesis is the result of the research work I carried out at the Chair of Dynamics and Control (SRS) at the University of Duisburg-Essen during 2016 to 2020. I would like to thank all the people who have supported and guided me during this important period in my life.

First and foremost, I would like to express my special appreciation and thanks to my supervisor, Univ.-Prof. Dr.-Ing. Dirk Söffker, for offering the opportunity to be his student, for the continuous guidance and great support he has given me during my Ph.D. study. Without his valuable advices and constant feedback this work would not have been achievable.

I am also grateful to my second supervisor, Univ.-Prof. Dr. rer. nat. Johannes Gottschling, for his valuable feedback, scientific questions, and helpful advices to improve this work.

I would like to thank all my colleagues at the Chair of Dynamics and Control (SRS) for their kind support, encouragement, and cooperation.

A very special word of thanks goes for my parents and my sister, Yan Lu, Daju Deng, and Jie Deng, who always encourage and believe in me.

Most importantly, I would like to thank my beloved husband, Mengnan Yan, for always being there for me. Thank you for understanding and supporting me for everything.

The last word goes for Yiluo Yan, my baby girl, who has given me the extra strength and motivation to complete this work. Thank you for coming into my life, you are the best thing that has ever happened to me.

Dortmund, December 2020 Qi Deng

Kurzfassung

Seit einigen Jahren spielt die Forschung zur Vorhersage des Fahrverhaltens von Fahrerinnen und Fahrern eine wichtige Rolle bei der Entwicklung von Fahrerassistenzsysteme (eng: Advanced Driver Assistance System, ADAS). Aus diesem Grund wurden viele Methoden des Maschinellen Lernens entwickelt und in diesem Bereich angewendet. Aufgrund der Vorteile des Hidden-Markov-Modells (HMM) beim Umgang mit Zeitreihendaten sowie Zustandsübergangsbeschreibungen scheint das HMM ein geeigneter Algorithmus für die Vorhersage des Fahrverhaltens zu sein. Eines der Ziele dieser Arbeit ist es, verschiedene Fahrverhaltensmodelle und verwandte HMM-basierter Algorithmen zu analysieren.

Außer der Verwendung eines einzelnen HMM zur Erstellung eines Fahrverhaltensmodells, können zwei Entwurfsideen (HMM-abgeleitete und HMM-kombinierte Ansätze) aus den vorhandenen Forschungsergebnissen abgeleitet werden, um die HMM-Leistung zu verbessern. Basierend auf HMM-kombiniertem und HMM-abgeleitetem Entwurfsideen werden zwei neu Methoden namens Fuzzy Logic-basierte Hidden Markov-Modelle (FL-HMM) und Multi-Layer-HMM (ML-HMM) in dieser Arbeit entworfen.

Um das zukünftige Fahrverhalten zu bestimmen und vorherzusagen, besteht die Hauptidee darin, das historische Verhalten des Fahrers maschinell zu erlernen. Aus diesem Grund muss zuerst ein Modell erstellt und trainiert werden. Um den Trainingsprozess zu verbessern, wird in dieser Arbeit eine Strategie für eine höhere Zuverlässigkeit in Bezug auf die Genauigkeit, die Erkennungsrate und die Fehlalarmrate entwickelt. Die Strategie wird als Full-Scale-Trainingsschleife bezeichnet und kann zur Optimierung der Modellstruktur und des Modelltrainings verwendet werden. Basierend auf der vorgeschlagenen Strategie, werden fünf herkömmliche Methoden (HMM, Support Vector Machines (SVM), künstliche neuronale Netze (ANN), Convolutional Neural Network (CNN), Random Forest (RF)) und zwei neue Methoden (FL-HMM, ML-HMM) als Beispiele zur Identifizierung des Fahrverhaltens verwendet.

Die Designparameter sind unbekannt und müssen vor dem Training manuell eingestellt werden. Diese Parameter können verwendet werden, um die Struktur und den Trainingsprozess des Algorithmus zu bestimmen. Zur Verbesserung der Vorhersageleistung der zugehörigen Modelle, werden die Designparameter geändert, um geeignete Werte zu finden. Unter Verwendung der vorgeschlagenen Trainingsstrategie können die am besten geeigneten Designparameter automatisch bestimmt werden, um die Leistung der Algorithmen zu optimieren. In dieser Arbeit werden die Designparameter in zwei Kategorien unterteilt: Hyperparameter und Vorfilter. Der Schwerpunkt liegt auf der Demonstration der Fähigkeit der vorgeschlagenen Strategien, um die Vorhersageleistung der verschiedenen Methoden zu verbessern und die Auswirkungen von Hyperparametern und Vorfiltern zu diskutieren. Basierend auf den Daten von 17 Fahrern werden für jede Methode vier verschiedene Modelle entwickelt, um die Wirksamkeit von Hyperparametern und Vorfiltern zu validieren. Die erhaltenen Ergebnisse zeigen, dass die Vorhersageleistung unter Verwendung der vorgeschlagenen optimierten Trainingsstrategie verbessert werden kann.

Abstract

In recent years, research and development of predicting driving behaviors play an important role in the development of Advanced Driver Assistance Systems (ADAS). For this reason, many machine learning approaches have been developed and applied in this field. Due to the advantages of Hidden Markov Model (HMM) in dealing with time series data as well as state transition descriptions, the HMM seems to be a suitable algorithm in driving behavior prediction. Therefore, one of the aims of this thesis is to analyze the current state of various driving behavior models and related HMM-based algorithms.

Except for using a single HMM to establish a driving behavior model, two design ideas (HMM-derived or HMM-combined approaches) can be concluded from the existing research to improve the HMM performance. Based on the two design ideas two newly developed approaches named Fuzzy Logic-based Hidden Markov Models (FL-HMM) and Multi-Layer HMM (ML-HMM) are designed based on HMM-combined and HMM-derived approach in this thesis.

To determine and predict drivers behaviors in the future, the main idea is to learn the driver's historical behaviors. For this reason a model has to be established and trained first. To improve the training process, in this thesis a strategy is developed for higher reliability in terms of accuracy, detection rate, and false alarm rate. The strategy is named full scale training loop and can be used to optimize both model structure and model training. Based on the proposed approach, seven algorithms including five conventional algorithms (HMM, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Convolutional Neural Network (CNN), Random Forest (RF)) and two new approaches (FL-HMM, ML-HMM) are used as examples to identify driving behaviors.

To improve the prediction performance of the related models, design parameters, which are unknown and need to be set manually before training, are modified. Using the proposed training procedure the most suitable design parameters can be determined automatically to optimize the performance of the algorithms. In this thesis, design parameters are divided into two categories: hyperparameters and prefilter. The focus is to demonstrate the ability of the proposed approach to improve the prediction performance of different algorithms, and to discuss the effects of hyperparameters and prefilters. Based on the data achieved from 17 drivers, four different models are designed for each algorithm to validate the effectiveness of using hyperparameters and prefilters. The finally obtained results show that the prediction performance can be improved using the proposed optimized training procedure.

Contents

List of Figures

List of Tables

Nomenclature

Symbols

v_{ego}	Velocity of ego-vehicle
v_f	Velocity of vehicle in front
v_{fl}	Velocity of vehicle in left-front
v_{fr}	Velocity of vehicle in right-front
v_{bl}	Velocity of vehicle left-behind
v_{br}	Velocity of vehicle right-behind
v_b	Velocity of vehicle behind
d_f	Distance to vehicle in front
d_{fl}	Distance to vehicle in left-front
d_{fr}	Distance to vehicle in right-front
d_{bl}	Distance to vehicle left-behind
d_{br}	Distance to vehicle right-behind
d_b	Distance to vehicle behind
TTC_f	Time to Collision to vehicle in front: $d_f/(v_{ego} - v_f)$
TTC_{fl}	Time to Collision to vehicle in left-front: $d_{fl}/(v_{ego} - v_f l)$
TTC_{fr}	Time to Collision to vehicle in right-front: $d_{fr}/(v_{ego} - v_f r)$
TTC_{bl}	Time to Collision to vehicle left-behind: $d_{bl}/(v_{ego} - v_b l)$
TTC_{br}	Time to Collision to vehicle right-behind: $d_{br}/(v_{ego} - v_b r)$
TTC_b	Time to Collision to vehicle behind: $d_b/(v_{ego} - v_b)$
α	Heading angle of ego-vehicle
P_a	Accelerator pedal position
P_b	Brake pedal pressure
Ln	Current lane number
I	Indicator
G	Gearbox
$Saccade$	Saccade
$Blink$	Blink
F_{Blink}	Blink frequency
N_{Screen}	Screen number
x	Screen coordinate (x-axis)
y	Screen coordinate (y-axis)
Q	Hidden state symbols
V	Observation symbols
S	Hidden state sequence
O	Observation sequence
A	State transition probability matrix
B	Observation probability matrix
π	Initial probability distribution
N_{Est}	Number of estimated lane change maneuvers

N_{Act}	Number of actually measured lane change maneuvers
N_{Tree}	number of decision trees of random forest

Abbreviations

ADAS	Advanced Driver Assistance System
ANN	Artificial Neural Networks
CNN	Convolutional Neural Network
HMM	Hidden Markov Models
RF	Random Forest
SVM	Support Vector Machines
FL	Fuzzy Logic
ML	Machine learning
MF	Membership function
NSGA-II	Non-dominated Sorting Genetic Algorithm II
FL-HMM	Fuzzy Logic-based Hidden Markov Models
ML-HMM	Multi-Layer Hidden Markov Models
HHMM	Hierarchical Hidden Markov Models
AR-HMM	Auto-Regressive Hidden Markov Models
BP-AR-HMM	Beta Process Autoregressive Hidden Markov Models
HDP-HMM	Sticky Hierarchical Dirichlet Process Hidden Markov Models
ACC	Accuracy
DR	Detection Rate
FAR	False Alarm Rate
TP	True Positive
FN	False Negative
TN	True Negative
FP	False Positive
LK	Lane Keeping
LCL	Lane Changing to Left
LCR	Lane Changing to Right

1 Introduction

1.1 Motivation and objectives of the work

Every year, a significant amount of people died and are injured in traffic accidents. World health organization (WHO) provided a global status report on road safety in 2015 and showed that more than 1.2 million people are killed each year on the world's roads, the currently leading cause of death for people with age ranged from 15 to 29 years [Wor15]. Therefore, the driving safety problem has been paid more attention in the past years. Some institutions collect and summarize data describing causes of traffic accidents. Reports [Wor15] [Nat15] [Nat16a] state many different causes, such as weather, traffic environment, driving vehicles, and factors related to the individuality of drivers. The report conducted by National Highway Traffic Safety Administration (NHTSA) in 2015 [Nat15] assigns the most critical reason of traffic accidents to drivers as 94 %. Driver's individual factors in the driving process and road traffic accidents are mainly reflected in driver's own behavior. Therefore, the research of driving behavior is meaningful for traffic safety. However following general driving rules, drivers will usually choose the most appropriate operations based on their own driving experiences and habits. Drivers' driving behaviors are assumed as individual. If the driver's current behavior can be classified and the upcoming behavior can be correctly predicted, the driver can be better guided. Therefore, the integration of the driver's behaviors makes it possible, that an assistance system can help the driver to detect the improper behaviors more specifically and to indicate dangerous situations earlier. Driving assistance systems should be adjusted based on the analysis of individual driving behaviors to improve traffic safety as well as to realize intelligent driving. To establish a model, the common idea is through learning from the given driving behaviors to detect the driving intention or furthermore predict the driving behavior. When a similar driving situation occurs, a corresponding driving behavior can be predicted.

Nowadays many institutions have conducted the research of driving behavior prediction. Different machine learning algorithms like Artificial Neural Networks (ANN), Dynamic Bayesian Networks (DBN), Support Vector Machines (SVM), Fuzzy Logic (FL), Random Forest (RF), Convolutional neural network (CNN), and Hidden Markov Models (HMM) are applied for learning and modeling about driver's decision.

In fact, current research proposes new methods to realize and improve driving behaviors prediction. However, only a few articles concern the optimization of an established prediction model to improve the prediction efficiency. Therefore, one of the objectives of this thesis is to propose a new training strategy named full scale training loop to improve driving behaviors prediction model based on the known machine learning approaches or an established model. To accomplish this task, the known approaches like SVM, HMM, ANN, CNN, and RF are selected as example to study the effectiveness of the proposed training strategy.

Compared with other popular machine learning algorithms, the HMM and DBN are designed as a probabilistic graphical model. One advantage is that it is easier for a human to understand directly the probabilistic relationships between the nodes. However, the DBN is more complicated than HMM in terms of network definition. In addition, driver's driving behaviors are based on the driver's own experiences, habits, and the current traffic environment. During driving, driver's behaviors cannot be measured directly but can be inferred by analyzing measurable parameters described current driving situation. The upcoming behavior is stochastic and only depends on the present state. Therefore, driving behaviors can be described as a hidden Markov process [SBH08, JF15, JF16]. The HMM algorithm has an advantage for handling time series data and stochastic signal process. For these reasons, the HMM algorithm is suitably applied for driving behavior or other human behavior studies [LZT+14]. In 2016, the authors of [MT16] reviewed machine-learning techniques for statistical analysis and modeling of driver behavior. The authors also pointed out that HMM has been successfully applied to model driver behavior using large amounts of driving data. Additionally, only a few HMM-based approaches are summarized in [MT16]. In the review paper [LZT+14] the authors summarized the current researches of the identification of driver behaviors. They compared some related algorithms such as HMM, Neural Network (NN), Fuzzy rule-based classifier, and Gaussian Mixed Model (GMM). The authors listed the advantages and disadvantages of the four algorithms, and pointed out that the HMM algorithm demonstrates a high accuracy and a very good performance in real-time driving behavior prediction. However, the authors in [LZT+14] critically pointed out the disadvantage of HMM requiring manual definition of the sequence distribution of the current observation. To solve this problem and to improve the performance of a single HMM, many authors proposed different HMM-based approaches [NTY+12, NTH+14, TNH+16, JF15, JF16]. In general, the design ideas of these HMM-based approaches are roughly divided into two categories: HMM-derived and HMM-combined. However, papers focusing on introducing and comparing HMM and HMM-based methods in this field are not available yet. Therefore, popular HMM-based approaches applied in driving behavior studies will be summarized in this contribution. Based on the two design ideas, two novel approaches are developed in this thesis.

The contributions of this thesis could be summarized as following points.

- A new training strategy named full scale training loop is proposed to improve prediction performance of known machine learning approaches.
- Two new approaches based on HMM-derived and HMM-combined methods are developed to improve driving behaviors prediction.

In addition, some open questions listed in section 2.4 will also be answered in this thesis.

1.2 Outline of the thesis

The thesis consists of 8 chapters. Some parts of this thesis are published or prepared for journal papers [DWS19+] [DS20a] [DS20b][DS20c], or have been published in proceedings of conferences [DWS18] [DS18] [DS19a] [DS19b].

In chapter 2 the state-of-the-art regarding driver behavior prediction [DS20a] is presented. Nowadays many institutions have conducted the research and established various driver behavior models for driving. Popular approaches are summarized, especially HMM-based approaches applied to driving behavior studies. In addition, some open questions of existing researches are summarized.

In chapter 3 the fundamentals of five known machine learning approaches including Support Vector Machines (SVM), Artificial Neural Networks (ANN), Convolutional neural network (CNN), Hidden Markov Models (HMM), and Random Forest (RF) are introduced.

In chapter 4 a full scale training loop is proposed to optimize unknown parameters, and therefore to improve the prediction performance of the known machine learning approaches or an established model.

In chapter 5 two newly developed approaches named Fuzzy Logic-Hidden Markov Models (FL-HMM) and Multi-Layer HMM (ML-HMM) are presented, which are designed based on the HMM-combined / HMM-derived ideas and the proposed full scale training loop.

In chapter 6 the experiment design for obtaining driving data is described. Based on the data sets, the experimental results are given and the proposed approaches are validated.

In chapter 7 the conclusions about the full scale training loop as well as the newly developed approaches are provided.

In chapter 8 the approaches and discusses research contributions are summarized. Limitations and further work are also presented.

2 Literature review

This chapter consists of four sections. In section 2.1, the concept of human driver behavior recognition / prediction, a typical driving model structure, and the application status of driver behavior model are introduced. The state-of-the-art of driver behavior prediction and its related approaches are reviewed in section 2.2. In addition, the influencing factors on driving behaviors i.e. the main objects of current researches are summarized. A discussion on the applications, advantages, and disadvantages of the mentioned methods, limitations, as well as the development trends of driving behavior models are addressed in section 2.3. The open research questions are summarized in section 2.4.

The contents, figures, and tables presented in this chapter are prepared and submitted for publication of [DS20b].

2.1 Background

Nowadays many institutions have conducted the research and established various driver behavior models for driving. The studies reported that, the driver's character, gender, age, fatigue, driving experience, etc. will mainly affect the driver's behavior [Wor15] [Nat16a] [Nat16b]. Some review papers summarized already the current state of driving behavior studies with focus on the identification of driver behavior characteristics [LZT+14] [WXC14], the prediction of tactical driving behaviors (intent) [DT11], the detection of driver drowsiness and distraction [KGYK15], the analysis of driving styles [MM15] [MHW+18], and the recognition of human behavior through visual monitoring [CSG+10] or human emotional states [DKB20]. However, these review papers only summarize and discuss specific aspects of driving behaviors, like driving styles, drowsiness, etc. In addition, the authors often summarize popular algorithms. The derivation of popular algorithms or other approaches developed are not discussed. In [BDCK20] a taxonomy of 200 models is constructed around different modeling tasks including state estimation, intention estimation, trait estimation, and motion prediction. However, specific information like model input, output, and explanation/discussion of algorithms is not presented in this survey [BDCK20].

To study (and therefore to model) the driver's behavior, to copy the driver's behavior to machines (algorithms), and then to use the driver's model for predicting the driver's behavior in the future, different types of driving behavior models exist in current studies and will be summarized in this chapter.

2.1.1 Human driver behavior and intention

The main goal of this thesis is to study human driver behaviors. The driver behavior model depicts a theoretical framework of human cognition, maneuver, and control processes which plays an important role with respect to prediction, reducing driving risk, developing intelligent vehicles, and improving driving assistance system. The driving behavior in current research is not only a real action or a specific behavior, but also includes the reactions when driver realizes a driving task, such as the driver's driving patterns, driver's intentions, the driving maneuvers, the trajectories of the vehicle, etc.

Generally, driver behavior is what a driver actually does, and driver intention is what a driver intends for his/her own behavior to be. Therefore, driver behavior prediction (i.e. driver intention recognition) model is based on the real understanding of the ego-vehicle states and other events which occur in the environment, and then the model judges and estimates what will happen in the next step. Normally, the estimated behavior will be realized by the driver/vehicle in the next few seconds/minutes/hours. This process is defined as behavior prediction.

2.1.2 Prediction and recognition of human driver behaviors

In this section the modeling basics and the typical classifications of driver behavioral model are briefly introduced.

A driving behavior model consists of three aspects including input, information processing process, and output. The vehicle states and variables (position, speed, acceleration, steering wheel angle, etc.), situations of the surrounding vehicles (position, speed, acceleration, etc.), information on road conditions, traffic signals, and traffic information are generally given as input of a driving behavior model. These inputs will affect the driving behaviors, and change the driver's actions (output). In [Mic85] the author summarized types of driving behavior models, which are classified into two categories including trait models (psychological) and task analysis (behavioral). Trait models describe the relationship between the driver's characteristics mainly focusing on the perspective of accident proneness. Due to physical and psychological factors, drivers may have traffic accidents in some cases. In general, trait models will be used to study individual differences of drivers, which include external differences (such as driver's driving skills, age, gender, etc.) and internal differences (such as driver's fatigue, attention, emotions, etc.).

General modeling ideas of nowadays driving behavior models (task analysis) will consider specific observations as well as analysis of driving activities, which include driver's perception, decision making, navigation tasks, and attention regulation, etc. These driving activities will be analyzed as different types of driving sub-tasks and divided into different levels. Afterward the levels will be dynamically linked to a

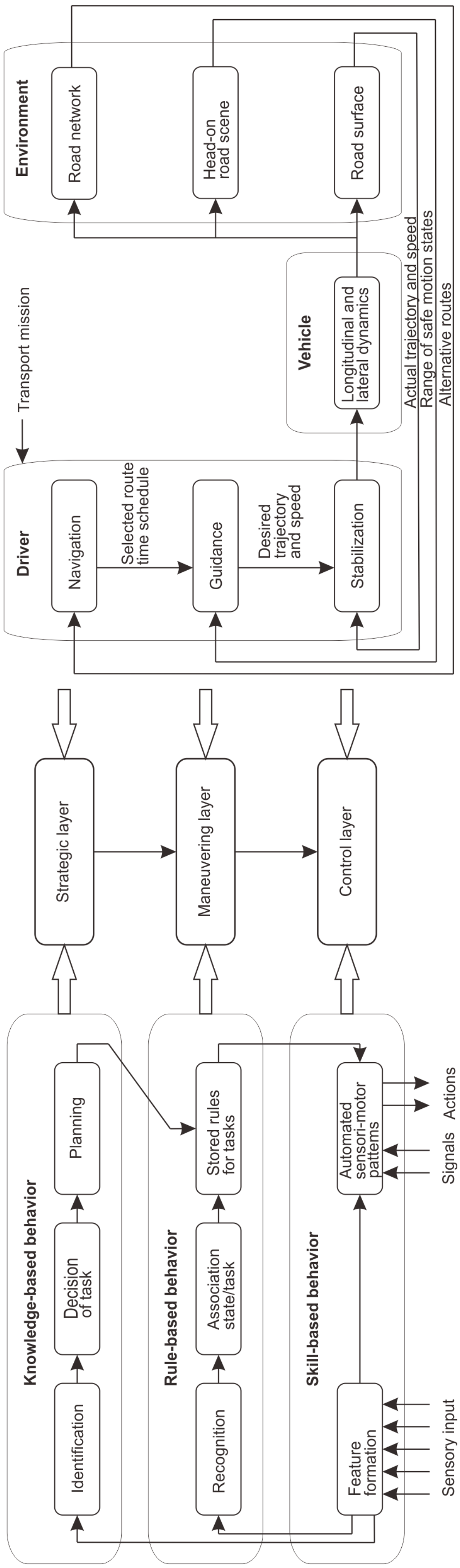


Figure 2.1: Three hierarchical model of driving task (modified based on [Don99])

complete driving task as well as used to build a hierarchical structure. In hierarchical models, task requirements, execution time, and cognitive processes at each level are not identical.

A well-known three-level model was proposed for task modeling by Rasmussen in 1983. Rasmussen described human behavior in a human-machine system stating three levels, which are denoted as skill-, rule-, and knowledge-based [Ras83]. In general, the skill-based performance denotes the driver's actions that are independent on the person's conscious attention. However the conscious is not the only basis for the driver's behavior. Sometimes the drivers drive according to the established rules or their own experiences, i.e. the rule-based level. If no clear rules or previous experiences can be applied in a driving situation, the driver's behaviors will be made based on driver's analysis, judgment, plan, and trials, in this situation the driving behavior is knowledge-based.

In 1985, Michon established a hierarchical model considering the problem-solving task of driver, which is structured at three levels: strategical (planning), tactical (maneuvering), and operational (control) [Mic85].

Ranney [Ran94] combines Michon's hierarchical control model and Rasmussen's three-level model, and describes the differences between the skills of novice drivers as well as experienced drivers, and the differences between familiar and unfamiliar situations. For experienced drivers, skill-based, rule-based, and knowledge-based behaviors are involved at the operational, tactical, as well as strategical level respectively. However, for novice drivers or driving in unfamiliar areas the drivers behavior may follow a different path in this model. For example, at the operational level, experienced drivers shift gears on a skill-based level, but the driving behaviors of novice drivers are on the knowledge-based level initially.

Donges introduced a three-level hierarchy of driving task in 1982 and delineated each level in detail based on Rasmussen's three-level model in 1992 [Don16] [Don99], as shown in Figure 2.1. The first level is a skill-based behavior process (control layer). Donges used the term "stabilization" instead of "control", in this process drivers use driving skills automatically. The second level describes a rule-based behavior process (maneuvering layer), in which drivers determine the maneuvers (such as overtaking, slowing down, changing lanes, etc.) according to the traffic rules and the current driving situation. The third level is denoted as the knowledge-based behavior process (strategic layer), drivers in this process make an entire driving plan and handle emergency based on their knowledges as well as experiences.

In 2003, Hollnagel et al. [HNL03] proposed a model for Driver in-Control (DiC). The model describes the driving tasks using four levels, which include Targeting, Monitoring, Regulating, and Tracking. Each level has its own control objectives. The control objective of the targeting level is the goal of a driving task, such as the choice of the destination. In the monitoring level it mainly monitors the driving environment, such as the state of traffic signs and signals. The regulating level controls the anticipatory risk and focus on the abnormal/dangerous driving. The last level (tracking) is concerned with vehicle speed, distance from the vehicle to front/behind vehicles, relative lateral position, etc. This model is not only used to describe the corresponding driving tasks of each control level, but also describes the dynamic interaction as well as the propagation of disturbances between the different levels. It provides a reasonable explanation for the interaction between driving behavior and traffic environment.

2.2 State-of-the-art

2.2.1 Influencing factors of driving behaviors

Driving behaviors are dynamic and individual. They are based on driver's own characters, experiences, habits, the current driving environment, etc. Driving behaviors can be influenced by many factors, which are also the main topics of the current research in this field. This section summarizes several common influencing factors of driving behavior, and introduces their current corresponding research.

Driving styles

Due to the driver's character, psychological status and other factors, driving behaviors could be categorized into many driving styles. Sagberg et al. [SSPE15] state that driving style depends on the individual driver and it is a habitual driving way. The authors point out that in the existing literature labeling of driving styles are commonly defined by common sense. In current driving style contributions [SSPE15,ZVL14,AAAD12], aggressive driving is a very common term. Aggressive driving behaviors include driving without obeying the traffic rules, such as over speed limit driving, sudden accelerating, sudden braking, abrupt lane changing, or sharp turning. These driving behaviors will lead the driver/vehicle to risks or even accidents. Therefore, one of the purposes of driving style studies is to sort out these aggressive driving. It is helpful to develop ADAS, because when a dangerous driving behavior is recognized, the driver can be warned immediately. At the same time, the driver's behaviors will be guided to improve traffic safety. To detect aggressive driving behaviors, in many driving style analysis studies, normal (safe/defensive) driving is given as a referent [ZVL14]. As explained in [SSPE15], dangerous and safe driving styles can be divided into different levels. Several different terms are used to label global driving styles, like calm, careful, aggressive [SSPE15]. However, it still lacks an unified conceptual standard to clearly distinguish these styles. In the existing contributions, the levels, the terms, and the concepts of driving styles depend on author's own definitions.

Aggressive driving can be classified based on physiological signals, biometric information, or vehicle driving state like vehicle velovity, acceleration, etc. In [DL15] Derbel et al. propose an approach for calculation of car insurance fee through estimating the driver aggressiveness. Using the vehicle velocity signal, vehicle acceleration, and vehicle jerk (i.e. derivative of acceleration) collected from a black box, a developed Fuzzy Inference System (FIS) model was designed to improve performance of the driving behavior recognition. By comparing the vehicle acceleration and the vehicle's comfort acceleration limit, the aggressiveness can be determined. For example, the authors point out that the values of vehicle's comfort acceleration limit in the case of acceleration/deceleration are equal to 4 m/s^2 and -3 m/s^2, respectively. If the value of the acceleration gets larger than 4 m/s^2 or less than -3 m/s^2, the driving will be considered as more aggressive. The same strategy was also reported in this research for vehicle jerk. Finally, aggressive and normal driving can be classified using fuzzy rules.

In [AAAD12] A. Aljaafreh et al. developed a method for detection and classification of aggressive driving. Using fuzzy logic, driving styles are classified into below normal (BN), normal (N), aggressive (A), and very aggressive (VA). The data were collected from a 2-axis accelerometer which is embedded in most of the GPS tracker devices. Longitudinal acceleration, lateral acceleration, and the speed of the vehicle are used as inputs respectively, afterwards their corresponding logical values need to be defined one by one. Based on fuzzy rules the output of the system is used to classify the individual driving behaviors into the different driving styles. Aggressive driving events could be detected from normal driving. In [ZVL14] aggressive driving style is classified using 3-axis (lateral, vertical, longitudinal) accelerometer data. The authors compare using one acceleration signal alone or combining two or three of them to recognize driving styles. The results are shown that using longitudinal acceleration signal the aggressive and safe driving style can be more effectively classified.

In addition to judge aggressive driving, driving style analysis is also used for reducing fuel consumption. In [BC16] Bao et al. proposed a method for predicting the driving style to search a personalized eco-friendly style. The drivers were divided into three classes including calm, normal, and aggressive driving based on Learning Vector Quantization (LVQ) neural network. Based on the predicted driving style, current traffic (congestion and average speed of each road), time, and road type, the fuel-consumption-minimizing route could be determined.

From these studies it can be concluded that the study of driving styles/patterns cannot only be used for warning drivers to avoid dangerous driving and related problems, but also for calculating car insurance fees, improving fuel economy, and other aspects. The information about driving styles is obviously helpful to develop driving assistant systems. Based on different types of drivers these systems can give drivers suitable suggestion to fit their driving habits.

Fatigue driving

Fatigue driving is another important driving style leading to traffic accidents. The drivers' inattentiveness, tiredness, drowsiness, or sleeping during the driving process refer to fatigue driving. The National Highway Traffic Safety Administration [Nat17] reported that about 90000 accidents involved fatigue driving in 2015. Driver fatigue detection research can be divided into two main categories: based on driver behavior and on vehicle behavior.

Based on driver behavior means measuring the driver's own characteristics, such as physiological parameters or biometric information. The driver's physiological parameters include electroencephalogram (EEG), electroencephalogram (EOG), electrocardiogram (ECG), etc., which can indicate driver's mental fatigue and psychical fatigue. Therefore, in some studies these parameters are used to determine whether drivers are fatigue. In [KKLD11] the authors propose a feature-extraction method to extract drowsiness-related features from the EEG, EOG, and ECG signals. These features will be used to classify the drivers fatigue into different levels. To simplify the calculation process and to form suitable feature sets to the classifier, the authors chose two dimensionality-reduction methods including spectral regression (SR)-based linear discriminant analysis (LDA) and its kernel-based version (KSR). The results show that the classification accuracy of using KSR is better than using SR, the percentages are 97 % and 95 % respectively. In addition, the authors compare the differences between using only one signal or using a combination of different signals. The conclusion of the research shows that the ideal results cannot be obtained with ECG or EOG alone. However, a high classification accuracy can be achieved using only EEG, or using a combination of EEG+ECG, or EEG+EOG. Other researches concluded the same outcomes. Using ECG-based Neural Network (NN) [PLKR11] the accuracy of the classification reaches 90 %, and using EEC- and ECG-based Support Vector Machine (SVM) [SLZ+11] the percentage of accuracy is larger than 87.5 %.

The other measures of driver's characteristics are through the analysis of eyelid blinking, eye movement, eye closure, head pose, etc. to detect fatigue driving. In [QLH12] Qin et al. focused on the analysis of eye closure of the driver. The authors extracted two-dimensional Discrete Cosine Transform (2D-DCT) feature of each eye images. Two HMMs were trained based on eye opening and closure images, respectively. The states of the two HMMs were calculated at the same time. The recognition result with the highest likelihood is used to determine the fatigue statues of the driver. Lee and Chung [LC12] use a dynamic Bayesian network framework to evaluate the driver fatigue. Two sensors are used to collect data including eye movement and photoplethysmograph (PPG) signal. If the calculated driver fatigue reaches a defined threshold, the drivers will be warned.

In addition to the aforementioned researches based on studying the driver's own characteristics, analyzing the vehicle situation is also used to detect driving fatigue.

The driver's maneuvers could be estimated to determine whether there is fatigue driving. Generally this method uses the current vehicle status including the distances between the ego-vehicle and other vehicles, deviations from lane position, steering wheel angle, velocity, acceleration, as well as other controller-area-network (CAN) signals. For example, in [CRKK16] an approach is given for detecting driver fatigue based on HMM. Signals are processed according to three independent modules including vision, audio, and other-signals module. The inputs of vision and audio modules are video and voice respectively. The module namely 'other-signals' uses heart rate, steering wheel position, gas, brake, and clutch pedal positions as inputs to detect driving fatigue. The three modules are independent from each other and final results were fused using the output of each module.

As shown in the mentioned researches, the fatigue driving behaviors can be determined from analysis of the driver and the vehicle states. Physiological parameters or biometric information are often used for fatigue driving detection. However, drivers are required to wear an appropriate equipment like helmet to collect data. It is impractical for drivers in real driving. To avoid this, possible solutions are through analyzing the state of the human eyes and the state of vehicle. In this case, drivers are not required to wear equipment, the data can be collected by eye trackers, camera, or the vehicle CAN bus. This can be achieved in driving assistance systems.

Drunk driving

The National Highway Traffic Safety Administration (NHTSA) reported that in 2014 the accidents due to drunk driving accounted for 31% of the total accidents in the United States [Nat16b]. Therefore, drunken driving is one of the major causes of traffic accidents. Therefore, the research of drunken driving is helpful for traffic safety.

A drunk driving recognition model based on Dynamic Bayesian Network (DBN) is proposed in [WY10]. The integrating multi-hysiological variables such as blood alcohol concentration, eye movement, and head movement are selected as inputs, which are collected by drunken breath analyzer and image capture devices. The authors proposed a simple graphical model integrating all the information to recognize the abnormal driving behaviors. The results show that the fatigue and drunk driving behavior can be detected in a simulated environment. In addition to the driver state, the vehicle state is also often used to determine drunk driving. In [MAA17] the authors select CAN bus data such as GPS, torque, engine RPM, vehicle speed, acceleration, etc. to detect drunk driving patterns. Using machine learning algorithm (Logistic Regression) the drunk driving patterns can be classified with an accuracy of 82 %. Dai et al. [DTB+10] proposed a system for detecting drunk driving only using a smart phone. Using smart phone the orientation angles and accelerations of the mobile phone are collected to determine the lateral and longitudinal acceleration of the vehicle. Through the both accelerations two behavioral clues including lane changing (drifting, swerving, etc.) and speed changing (suddenly accelerating and braking) can be detected. Finally, by considering these two information the model based on pattern matching techniques can judge whether the driver is a drunk driver. In [ASABZ13] the authors proposed a context-aware driver behavior system for detection of different behaviors, which include normal, drunk, reckless, and fatigue driving. By collecting contextual information about the driving environment, the abnormal behavior could be detected, in the meanwhile other vehicles on the road will be warned to avoid traffic accidents.

The recognition of drunk driving is similar to fatigue driving, which can be analyzed through the driver's and vehicle's state. The difference is that physiological parameters of drunk driving recognition is based on blood alcohol concentration instead of using EEG, EOG, etc.

Driving skill

The researchers in [Nat15] state that most accidents are caused by driver errors. Therefore, a large amount of researches are studying the driver's driving behaviors with the goal of sending early warnings to the driver and assisting the human driver.

In general, driving skills can be defined as the drivers' actions that are independent on the drivers' conscious attention [Ras83]. Driving skills are reflected by human drivers manually controlling vehicles to achieve specific driving tasks, such as speed changing, steering, gear shifting, etc. However, each driver has own individual driving skills. To realize intelligent driving and improve driver assistance systems, it seems to be helpful to analyze these individual characteristics of driver. The main idea of driving skill prediction is that by learning a driver's historical driving behavior, to determine and to predict the behavior of the driver in the future for different driving situations. Before the driver is making decisions, advice will be given or the driver will be warned early enough before a risky action is taken.

In [DWWB13] C. Ding et al. focused on the prediction of lane-changing trajectory based on Back-Propagation (BP) neural network. One advantage of this BP neural networks is that the network learns the relationship of inputs and outputs through adapting its free parameters. This ability to learn the uncertainties of driver's driving behaviors by trained BP neural network could be adjusted to accommodate any driving. The inputs of this model consist of the prior position, velocity, acceleration, and time headway of the ego-vehicle. The effectiveness of BP neural network for predicting lane-changing trajectories is proven in [DWWB13]. To explore how long the predicted time is suitable, the authors selected 1 s and 2 s as examples. The result with 1 s prediction time is more accurate than with 2 s prediction time.

In [JF15] [JF16], a prediction method of vehicle speed was presented. By using neural network (NN) models the average traffic speeds will be predicted based on current and historical traffic data. However, the individual vehicle speeds are not only limited by traffic speed, but also influenced by other factors, such as vehicle type, road type, and lane change, etc. Using the average traffic speeds obtained by NN, the prediction of individual vehicle speeds were realized based on HMM. Kumagai et al. [KSOA03] focused on the prediction of driver intension of stopping the vehicle at an intersection with their current and historical maneuvers based on a simple DBN.

The goal of driving skill and behavior research in this section is mainly to predict the driver's next actions and also to avoid misoperations of the driver or give suggestion for the next step. However, the main purpose of driving styles/patterns, fatigue driving, and drunk driving research is to identify whether there is an abnormal drive. Thus, the driver will be suggested to change the driving style or to stop driving.

Traffic environment

Another important factor affecting driver behavior is related to different driving scenes, such as highway and inner-city scenarios. For different driving environments, driving behaviors are also different. There are relevant studies of driving behaviors for some typical traffic environment. The main cause of accidents in highway are speed and lane changes. The authors of [ANN12] [BWKS14] [KPLL13] focus on lane-change or speed-change [JF15] [JF16] [LKO14]prediction in highway scenarios. Another highway scenario discussed is at highway lane drops, such as in [XLW+14] [DYF16] the authors studied driving behaviors entering a highway. The driving behaviors in these scenarios are mainly considered whether the drivers need to change lane, i.e. merge and non-merge behaviors. Other studies like [AH16] [KKS15] discuss the behaviors at an intersection inner-city. The driving behaviors in the inner-city are complex, they mainly include acceleration, deceleration stopping, turning, driving through the intersections with or without traffic signals.

2.2.2 Machine learning based human driving behavior recognition and prediction

In this section popular algorithms evaluating and classifying driving behaviors will be discussed and summarized. The main focus is on HMM and other related algorithms based on combinations using HMM.

Popular algorithms

Establishing driving behavioral models, several approaches have been applied. Different kinds of machine learning algorithms like Artificial Neural Networks (ANN), Dynamic Bayesian Networks (DBN), Support Vector Machines (SVM), Fuzzy Logic (FL), Random Forest (RF), Convolutional neural network (CNN), and Hidden Markov Models (HMM) are used to establish driving behavioral models and therefore will be introduced here.

Artificial Neural Network (ANN), which is also called Neural Network (NN), is a computational model used in machine learning, and imitates of biological neural network work. The trained ANN is presented as a dataset. The goal of the ANN is to obtain a desired output according to the corresponding inputs. As a common machine learning algorithm, the ANN approach is usually applied in fields of human driving behavior problems. For example, ANN-based models have been used in predicting the acceleration distribution for vehicle following on highways [CAFH13], for lane changing prediction [DWWB13], vehicle speed prediction [JF15] [JF16], and drowsiness detection [DHB$^+$11].

Bayesian Networks (BN) are a probabilistic model which graphically represents a set of random variables and their conditional dependencies. In [GSZ$^+$17] a framework based on BN for estimation of driver's drowsiness is proposed. The framework combines different information, such as weather, sleep time, eye movement, vehicle movement, yawning position, head tilt, etc. to infer driver fatigue. The result of this research shows that, if only one certain factor is considered in this framework, the percentage of eyelid closure could be used to obtain the best inferring result. If the framework combines three features, the inferred probability is always greater than 95 %. However in a regular BN, a node representing a driving state influences other state without considering the change of time. The driver fatigue as well as the other driving behaviors are time-dependent. Therefore, Dynamic Bayesian Network (DBN) is applied more often to human behavior recognition and prediction.

The DBN algorithm is derived from BN and is designed to consider dynamic requirements. It describes that, the value of a variable of a BN at time point t can be calculated from the values at time point $t-1$. In [GBD13] DBN is used for estimation and prediction the acceleration as well as turn-rate for car-following and lane-change of a 4-way intersection. The algorithm of DBN was used in [LC12] for driver fatigue evaluation. Al-Sultan et al. [ASABZ13] proposed a method based on a combination of DBN and a five-layer context-aware architecture to detect four different driving styles, which include normal, drunk, reckless, and fatigue.

Support Vector Machine (SVM) is a supervised machine learning method and primary a binary class classifier developed by Vapnik in 1979 [Vap95] [Bur98]. It is a margin-based classifier to transform the data to a high dimensional space to separate data using a hyperplane. The process of SVM learning is trying to find an optimal hyperplane between data points of different classes to generate a maximal geometric margin [Vap95] [Bur98]. Kumar [KPLL13] focused on the prediction of lane change intention based on a combination of a multi-class support vector machine (SVM) classifiers and Bayesian filtering. In [PLCY14] Pan et al. used SVM to establish a driving behavior model based on multiple information (eg. steering wheel, brake throttle, and road conditions) to determine whether the current driving behavior belongs to good (safe) driving. In [WWW+14] SVM is applied for fatigue driving detection, the authors concluded that SVM performs well using the driver's EEG signal, the results show an accuracy rate of 88.62 %.

Fuzzy rule-based classifier is one of the most popular approaches used in classification problems. The structure of this model is easy to interpret by using IF-THEN rules. Fuzzy Logic (FL) approach is considered as an extension of Boolean logic. It is based on fuzzy sets and allows to model the truth of statements continuously between true (one) and false (zero) using membership functions. Common fuzzy sets are based on triangular, trapezoidal, or Gaussian membership functions [ZB02]. Therefore, the inside view of this model is comprehensible and its logic could be easily understood. The authors of [HWJ+12] used a FL model for prediction of driver behaviors of stopping the vehicle at an intersection, when the traffic signal turns yellow. Distance from the current vehicle position to the stop line will be categorized as close, medium, and far distance. Close and far distance respectively indicate possibility to go as well as possible to stop. Medium distance contains the largest uncertainty. Using the defined membership functions at different distances the drivers stopping probability could be calculated.

Other popular algorithms such as Random Forest (RF) is used in [CLZ+17], seven basic driving behaviors are recognized based on a 3-axis accelerometer with a high accuracy of 98.1 %, and the robustness of RF is verified. In recent years, many works have been published to implement deep learning in the field of computer vision, document/handwriting recognition, and also driving behaviors recognition tasks [GMZ18] [LKMH17]. In recent years, many works have been published to implement Convolutional neural network (CNN) in the field of driving behaviors recognition tasks. For example, in [GMZ18] physiological signals are used to predict lane change behaviors based on a novel Group-wise CNN. The authors of [LKMH17] proposed using CNN and images from radar and camera sensors to predict lane changing intention.

To evaluate popular algorithms, some studies use same datasets and select different algorithms to establish models and evaluate performance of different algorithms. For example, in [DWH+20] a driving behavior prediction system is accomplished based on HMM, SVM, CNN, and RF. In addition, eye-tracking information is integrated. The results show that the performance of RF algorithm is the best of all four algorithms tested. Especially combining environmental and eye-tracking data the RF algorithm achieved the best results with an accuracy more than 99 %. The authors in [DS19a] proposed a strategy for improved training of conventional algorithms. Four algorithms (SVM, ANN, HMM, and RF) are used as examples. The authors point out that usually a set of unknown parameters are needed to be set manually before training, when a conventional algorithm is used alone. With the proposed training procedure, the most suitable values of these unknown parameters are determined automatically to optimize the performance of the conventional algorithms. The authors compared the reliability of the algorithms with respect to the relevant accuracy, detection, and fault alarm parameters. The results show that, for the four conventional algorithms, RF and ANN have better prediction performance than HMM and SVM. Using the introduced training procedure in [DS19a], the prediction performance of the conventional algorithms is (partly strongly) improved.

HMM and HMM-derived approach

In the review paper [LZT$^+$14] the authors compare some related algorithms such as HMM, Neural Network (NN), Fuzzy rule-based classifier, and Gaussian Mixed Model (GMM). The authors listed the advantages and disadvantages of the four algorithms considered, and pointed out that the HMM algorithm demonstrates a high accuracy and a very good performance in real-time driving behavior prediction. Compared with other popular algorithms, the HMM and DBN are designed as a probabilistic graphical model. One advantage is that it is easier for a human to understand directly the probabilistic relationships between the nodes. However, the DBN is more complicated than HMM in terms of network definition. In addition, driver's driving behaviors are based on the driver's own experiences, habits, and the current traffic environment. During driving, driver's behaviors cannot be measured directly but can be inferred by analyzing measurable parameters described current driving situation. The upcoming behavior is stochastic and only depends on the present state. Therefore, driving behaviors can be described as a hidden Markov process [SBH08, JF15, JF16]. The HMM algorithm has an advantage for handling time series data and stochastic signal process. For these reasons, the HMM algorithm is suitably applied for driving behavior or other human behavior studies [LZT$^+$14]. In 2016, the authors of [MT16] reviewed machine-learning techniques for statistical analysis and modeling of driver behavior. The authors also pointed out that HMM has been successfully applied to model driver behavior using large amounts of driving data. Additionally, only a few HMM-based approaches are summarized in [MT16].

Nowadays, there are a large number of driving behavior researches developed by HMM-based approaches. In general, the design ideas of these HMM-based approaches are roughly divided into two categories: HMM-derived and HMM-combined. However, review papers focusing on introducing and surveying HMM and HMM-based methods in this field are not available yet. Therefore, this section aims to summarize popular HMM-based approaches applied in driving behavior studies.

- **Hidden Markov Model**

  Hidden Markov Model (HMM) is applied for estimation of unmeasured states, therefore, it is widely applied in fields of driving behavior estimation. The focus of this section is to summarize the driving behaviors recognition and prediction approaches using HMM and HMM-derived approaches. In addition, there are some algorithms based on HMM combined with many other algorithms like neural networks, SVM, etc. also mentioned in this subsection.

  In [TSLL15], the authors propose to use HMM in determining driver intention for a variety of vehicle maneuvers including stop/non-stop, change lane left/right and turn left/right. To predict a trajectory of a lane changing, Liu et al. [LKO14] established two HMMs including normal lane change model and dangerous change model, which were trained based on normal sample data and crash data respectively.

  In [DWS18], the HMM algorithm was applied for driving behaviors prediction, where a prefilter was used to process and combine signals to form features for the HMM recognition process. Three different driving intentions namely: lane change left, lane change right, and lane keeping are modeled as hidden states for the HMM. The results show that the evaluation metrics including all accuracy (ACC), detection rate (DR), and one minus false alarm rate (1-FAR) values are larger than 80 %.

- **Hierarchical HMM**

  The hierarchical HMM (HHMM) is a multi-level HMM derived from HMM [FST98]. Like HMM, the HHMM algorithm contains a set of hidden states and a set of observations. The difference from HMM is that the states of HHMM contains three different kinds including root states, internal states, and the production states. Root and production states indicate states of the highest and lowest levels HMM respectively. Only production states contain an observation probability distribution matrix, i.e. observations are generated directly from production states. Each state of high-level HMMs (root and internal states) could be considered as a low-level HMM, that means each root and internal state serves as a probabilistic model [FST98]. Therefore, HHMM can be used to describe the relationships between each HMMs.

  In [GKKO11] the authors proposed a system for estimation and prediction of driver/vehicle behaviors in autonomous vehicles. Four different HMMs are trained according to four different scenarios, which include turning left/right, going straight, and stopping at an intersection. The results show that using this method driver behaviors can be successfully predicted. The authors presented an extension through using HHMM for prediction process. The driver states are the low-level HMMs, so the relationship between them could be estimated by the high-level HMMs. In [ZKZ+15] an HHMM approach is used to develop a rollover warning system of heavy duty vehicle. The authors pointed out that using lateral acceleration and roll angle the lateral slip and rollover behaviors of heavy duty vehicle can successfully be detected with a high accuracy of 99.7 %.

Unlike common HHMM, in [DS19b] a Multi-Layer (3-layer) HMM approach is proposed and developed for predicting lane changing behaviors. The approach is based on situation-specific HMMs combined with thresholds, for which related parameters are adapted during a training phase. The first layer is considered to predict the driving behaviors using only one signal as input. The inferential results from the first layer are given to the second layer, and the second layer only considers some selected information, such as all velocities, all distances, etc. Only the third layer considers all information. All sub HMMs of each layer are calculated in parallel and all of them can be used to predict driving behaviors. The results show that the accuracies of lane changing to right and lane changing to left are more than 90 %.

- **Bayesian Nonparametric HMM: Hierarchical Dirichlet Process (HDP)-HMM**

  One main issue in HMM is that the number of assortment of hidden states must be set before training, so each hidden state must be defined before modeling. If the assortment of hidden states increases, the model complexity also increases. If any of the assortments of the hidden states has not been defined during the training phase, consequently the whole model is incomplete and incorrect. To solve this problem, Hierarchical Dirichlet Process (HDP)-HMM was proposed by [BGR01] [TJBB06]. As a Bayesian non-parametric alternative for standard HMM, it is used without fixing the number of assortments of hidden states. In 2007 Fox et al. [FSJS07] proposed a Sticky HDP-HMM, which is an extension of HDP-HMM. It's frequency of transition between hidden states is reduced compared to the HDP-HMM model.

  In [NTY+12] [NTH+14] [TNH+16] the authors assumed that contextual information of driving behavior has a double articulation structure, which is similar to language, i.e., the driving behavior is a sequence of "driving words". A "driving word" is a sequence of "driving letters". In [TNH+16] steering angle, brake pressure, and accelerator signals are selected as input. Different segments of input signals are generated as "driving letters", which are considered as short-term behavior unit. A long meaningful behavior unit is named as a "driving word", such as "start", "turning right", "following a leading vehicle", etc. Here, the sticky HDP-HMM was used to find meaningful segmentations ("driving letters") from driving behavior. Nested Pitman-Yor language model (NPYLM) [MYU09] was used to combine and sequence meaningful chunks ("driving word"). Based on these chunks the driver's intention can be estimated. It is worth to mention that sticky HDP-HMM with NPYLM is a development of an unsupervised learning method, i.e. “driving letters” and “driving words” are unknown before training. Therefore, the evaluation method is different from common methods that use accuracy or detection rate to evaluate the prediction performance. In [TNH+16] three experiments are used to verify the model performance. The results of the first experiment indicate that more than two next “driving letters” are correctly predicted using a developed NPYLM with sticky HDP-HMM method. The results of the second and third experiments show that the average prediction time are 17 s and 8.9 s respectively.

The sticky HDP-HMM approach is also used to develop a general framework to learn and recognize lane-change interactions of the ego-vehicle with its surrounding vehicles on highways [ZZWX20].

In [WXZ18a] a new framework for driving style analysis is developed by combining Hierarchical Dirichlet Process and Hidden Semi-Markov Model (HDP-HSMM) derived from HDP-HMM. After comparing with HDP-HMM and sticky HDP-HMM, the authors in [WXZ18a] find that HDP-HSMM is able to segment driving patterns as expected, but HDP-HMM cannot learn driving patterns as expected, and the sticky HDP-HMM method is sensitive to data fluctuation. According to [WXZ18a], HDP-HSMM performs best among them.

- **Auto-Regressive HMM (AR-HMM) and Beta Process (BP)-AR-HMM**

  Auto-Regressive HMM (AR-HMM) is similar to standard HMM, but it has one more weight matrix W which consists of probabilities of moving from one observation to another. Abe et al. [AMO07] applied AR-HMM for modeling and predicting driving trajectory behaviors. Different driving behavior models could be switched by analyzing gas pedal stroke and brake pedal stroke.

  Similar to an HMM, the AR-HMM needs to determine the number of choosing hidden states (driving behaviors), i.e. the number of classes. To avoid this problem, Fox et al. [FSJW09] proposed the Beta Process AR-HMM (BP-AR-HMM), which combines the nonparametric Bayesian technique and AR-HMM. Therefore, this BP-AR-HMM model can produce infinite state. The total number of states can be determined in theory, but cannot be defined before training. In [HKI+16] the author applied BP-AR-HMM to predict the driving behavior, historical driving behaviors will be segmented into discrete states, which are produced by BP-AR-HMM. Each discrete state corresponds to an AR model. The observations in [HKI+16] consider accelerator opening rate, brake pressure, and the steering angle signals. Using the BP-AR-HMM, driving behaviors including brake pressure and steering angle are predicted. The results show that, compared with HMM, AR-HMM, and HDP-HMM, the

BP-AR-HMM has the smallest mean absolute error (MAE) which is about 0.05-0.2 MPa between the measured and predicted brake pressure values.

- **Summary**

  It can be concluded that the approaches derived from HMM are based on similar ideas, the HMM's characteristic of time series is mainly considered and used in these algorithms. Driving behaviors will be decomposed into multiple layers tasks. The lowest level task is to recognize each specific operation, such as acceleration, deceleration, and steering wheel signals. Obtained results of the lowest level will be given as inputs to higher level to identify driving behaviors like go straight, turn left/right. It is worth pointing out that these methods are proven to be effective in predicting driving behaviors. One possible reason is that signals and driving behaviors change always over time, and the current driving behavior is always affected by the previous one.

HMM-combined approach

Except for using HMM-derived approaches, HMM is often combined with other algorithms to improve the performance. Usually in this case, HMM and other algorithms are used to complete different tasks respectively.

- **Artificial neural network (ANN)-HMM**

  In addition, HMM is often combined with other algorithms. Different from HMM's derivative algorithm, in combination methods, HMM and other algorithms are used to complete different tasks respectively.

  For example in [AK07] Boyraz et al. proposed a method to determine a driving maneuver in an urban road scenario. An ANN was used to recognize and classify driving maneuvers based on different signals, such as steering wheel angle and speed. These labels were classified by ANN and then used to train the HMM. In the final phase, driving maneuvers of Right Turn, Left Turn, UTurn, Roundabout, Emergency Brake, and Reversing could be predicted based on HMM. In addition driver performance is also classified from 1 (best) to 8 (worst) using HMM. In [JF15] [JF16], a prediction method of vehicle speed was presented. By using neural network (NN) models the average traffic speeds can be predicted, afterwards the estimated traffic speeds are given as inputs to predict individual vehicle speeds based on HMM. Zhang et al. [ZWL15] proposed a deep neural networks (DNN)-HMM approach. Acceleration data are collected as inputs. The DNN approach is used to extract features from the row sensor data, by solving the observation probability distribution of HMM can be modeled automatically. This solves a disadvantage of HMM that the observations need to be defined manually relying on the experience of researchers. In [ZWYY09] an approach combining HMM and ANN is constructed to identify driving intention and to predict maneuvering behaviors on cornering, where HMM is used to predict three driving behaviors including emergency steering, normal cornering, and straight line driving. Then HMM prediction results are used as a guideline to train ANN, so specific steering angle is obtained by ANN. This solves a disadvantage of ANN which needs a lot of training samples. The results show that the steering angle can be successfully predicted, where the result has a low absolute deviation of real and predicted steering angles.

- **Support Vector Machine (SVM)-HMM**

  As a two-class classifier, a SVM is a supervised machine learning method. As known the approach is transforming data into a suitable space divided by hyperplanes. It's pattern classification is based on current observations, but not on context. If the current analyzed observations are interference signals, wrong results will be obtained. In addition, driving behaviors are dynamic, the decisions of the drivers at each time point will be affected by driving behaviors at the last time point. The HMM approach has an advantage of being able to analyze dynamic data and the temporal evolution of states. Due to the driving behaviors in different driving styles are not the same, in the same driving environment different drivers make different decisions. Using one HMM it is difficult to classify these driving behaviors of different drivers. Also here HMM results are depending on which hidden state has the largest output probability, i.e. the maximum log-likelihood. However, when the input features are not obvious, it may lead to small differences between the log-likelihoods. Therefore, it is difficult to distinguish some easily confused driving behaviors using HMM. To avoid this problem, a SVM-HMM approach is proposed in current research. For example in [TSL14], the SVM is used to distinguish different driving styles like normal and fatigue driving styles. For each driving style a corresponding HMM is used representing the upcoming driving behaviors.

  The SVM-HMM based model is usually applied to predict or recognize the driving behavior of different driving types/patterns. The general flowchart of the system based on SVM-HMM is shown in Figure 2.2. Here a SVM is used to distinguish different driving patterns. For each driving pattern a corresponding HMM is trained with respective observation sequences (i.e. training samples). The whole model including SVM and all HMMs is trained and saved in training phase. In test/application phase, the SVM can determine which driving pattern a test data set belongs to, and then switch to the corresponding HMM.

  In [TSL14] the authors choose SVM-HMM for detection of driver drowsiness. Here two different HMMs were trained for drowsy or non-drowsy. The SVM is used for determining which HMM should be used. Similarly, Aoude [ADSH12]

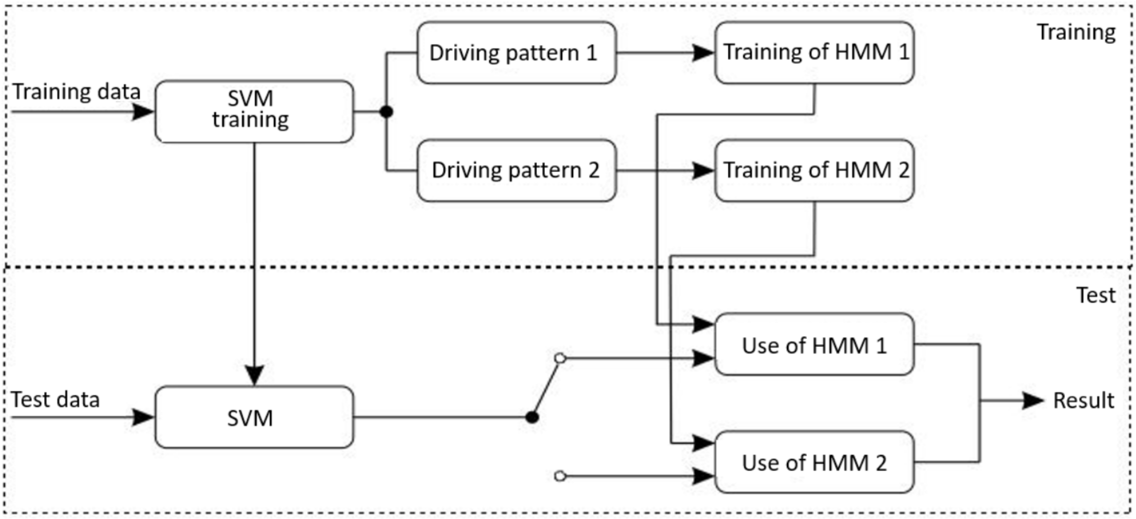


Figure 2.2: Flowchart using SVM-HMM [DS20b]

applied SVM-HMM for estimating driver behavior at intersections. The drivers were classified into compliant or violating type. In [XCL17] the authors proposed a framework to predict accident of vehicle collision on a straight two-lane highway. The SVM was used to classify a Leaving Lane scene (LL) and a Remaining in Lane scene (RL) based on the vehicle's trajectory. The HMMs were trained for each lane scene respectively and predicting whether the driver will have an accident.

- **Fuzzy logic (FL)-HMM** Fuzzy logic (FL) is an extension of Boolean logic (classical logic), in which the degrees of truth may be any real number between zero and one defined by related membership functions labeled and denoted with linguistic variables. The approach is used to present vague estimations and verbal descriptions, based on experiences. For this reason, in [DDWL14] Ding et al. introduced a lane-change intention recognition method based on FL and HMM. Here a Comprehensive Decision Index (CDI) is designed using FL to represent the driver's estimations about the current surrounding traffic. The CDI is calculated through three parameters, which include the ratio of the average traffic speed of original lane and target lane, Enhanced-Time-to-Collision, and the ratio of the real as well as the ideal distance from the ego-vehicle to the vehicle in front. Afterward, estimated CDI values can be used as input to train HMM. Finally, driver's intention including lane keeping, transition state, and lane change can be recognized through the trained HMM. By analyzing the test data sets including 69 lane change intention, in total 65 intentions are correctly identified with a short delay gap about 1.67 s. The authors in [DS18] proposed a newly developed approach Fuzzy Logic-based Hidden Markov Models (FL-HMM). The FL approach is used for additional distinction of driving scenes into very safe, safe, and dangerous driving scenarios. Afterwards, a corresponding HMM is trained for each driving scenes respectively and predicting the driving behaviors. Three different driving behaviors including left/right lane change and lane keeping are modeled as hidden states for these HMMs. High accuracies of 93 % and 91 % for lane changing to right and lane changing to left are observed respectively. Therefore, the design idea of FL-HMM can be categorized into two common forms. One of them is that the obtained outputs of FL (HMM) are given as the input to train HMM (FL) [DDWL14]. The other design idea is using FL to distinguish different driving scenes, to determine which driving scenes a data set belongs to, and then to decide and switch to the corresponding HMM [DS18].

- **Gaussian Mixture Model (GMM)-HMM**

  In [WXZ18b] a Gaussian mixture model (GMM) combined with HMM (GMM-HMM) is proposed to predict drivers braking behaviors. The GMM is used to model stochastic relationships between driving situations and braking actions. After learning the GMM parameters, HMM is applied to estimate drivers braking behaviors based on the mixture components of GMM. The obtained results show that the accuracy, sensitivity, and specificity reach 89.41 %, 83.42 %, and 97.41 % respectively. Lefevre et al. [LCG+15] develop a driver model based on GMM-HMM and its two application examples. One is used to predict lane departures on the highway and the other is to predict acceleration while lane keeping. The obtained results show that the proposed driver model can successfully predict and therefore avoid all 65 lane departure instances. In addition, the acceleration is also estimated correctly. By comparing the results of predicted acceleration, the author pointed out that the performance of a personalized/individualized model is always better than an average/general model. Similarly, in [WZHX18] Gaussian mixture regression (GMR)-HMM is applied to develop a lane-departure warning system. For each driver, an individualized model is established to predict the upcoming lateral vehicle trajectory. The authors also discussed some influencing factors, some of them depend on the design of the system and can be tuned according to different design requirements. Other factors like vehicle dynamics, road curvature, and driver state depend on the design of experiments and the states of vehicles/drivers, which do not affect the algorithm/system. Based on [WZHX18], the authors further propose a new Bounded Generalized GMM-HMM method derived from GMM-HMM [WXH19], which performs better than GMM-HMM. However, the authors point out that the structure of the Bounded Generalized GMM-HMM is more complex and it causes more computational costs than GMM-HMM.

- **Summary** The modeling ideas of HMM combined with other algorithms can be concluded using three common forms. First, the classification result of HMM (/other algorithms) can be used as input of other algorithms (/HMM), such as in [ZWYY09] results of HMM are guiding to train ANN, and in [DDWL14] results of FL are given as inputs to HMM. Second, parameters like observation probability distribution of HMM can be modeled by other algorithm [ZWL15] [LCG+15]. Third, other algorithms are used to distinguish different driving styles/patterns/scenarios, then HMMs are trained and used to recognize driving behaviors for different situations [XCL17]. Using the combined approaches, it is possible to utilize both of the advantages of HMM and the respective other algorithms. In [ZWL15] [DS18], it was proven that the HMM combined with other algorithms have better performance than a common HMM or a conventional algorithm used alone.

2.3 Summary and discussion

In this contribution different types of driving behavioral research and related typical research objects are introduced. In existing studies, the various algorithms were proposed to recognize and predict human driving behaviors. Popular algorithms are briefly summarized in this contribution.

2.3.1 Comparison of popular algorithms

Each algorithm has its characteristics and therefore advantages and disadvantages. They may perform differently in diverse domains or using different data sets.. In this section, a brief comparison between different algorithms is given to explain which algorithm is suitable in which context.

Application

In Table 2.1 the major application fields of HMM and other algorithms based on combinations with HMM are shown. It can be seen that HMM and methods based on combinations with HMM are used in various areas related to driving behavior mentioned in this article.

Related features as well as the application fields and a brief comparison of the data collection approaches are summarized in Table 2.2. According to the summary in this table, conventional machine learning algorithms like ANN, SVM, FL, RF, HMM are used in recent years for research related to driving behavior recognition and prediction. As the most popular method for deep learning, the CNN algorithm is not commonly used for this field because in most of the cases the inputs for CNN are images. It's worth mentioning that DBN and HMM are capable to handle temporal data. In comparison to HMM, DBN requires complex definition of the network, and perform poorly on high dimensional inputs. However HMM is not able to utilize raw data directly and requires data processing upfront. The detailed strength and weakness of all algorithms are discussed in the below subsection.

Advantages and disadvantages

It was proved that ANN can handle the classification of the signals with large variants, and in many cases using ANN models the result can reach a high accuracy, e.g. in [ZSF14] the prediction results are about 99 %, 94 %, and 74 % for non, left, and right lane-changing respectively. Similarly, the ANN model in [DYF16] was used to predict whether the driver merges to left lane at highway lane drops, the prediction accuracy for merge and non-merge behaviors is 85 % and 89 % respectively. In [DHS11] the authors proposed a fatigue (drowsiness) detection model based on ANN in simulative environment, and a high detection rate at 98.65 % was obtained. In addition, the ANN algorithm has a self-learning ability, it can be quickly adjusted to accommodate new problems. Another advantage of this algorithm is that the predefined assumptions for problem solving are not needed. As described in [DHS11], the temporal aspects are not concerned in the ANN model. The authors suggest to use HMM or DBN. According to the principle of ANN, this algorithm has some known disadvantages which are also proved by many studies. For example, the training time is too long, training data are relatively large. Besides, the deviation between the calculated and the desired result depends strongly on the weights. Therefore, the weights need to be adjusted to minimize the difference.

Actually, the application of SVM algorithm also shows a high accuracy in many linear/non-linear cases. For linear cases, data is classified by a linear hyperplane. For non-linear cases, kernels are applied to convert non-separable data into separable data and then hyperplane can be used to divide different classes. For example, in [DYF16] a high accuracy for merge (91 %) and non-merge cases (84 %) at highway lane drops can also be achieved using SVM. The SVM algorithm generates a linear hyperplane and divides the two classes with the maximal margin between the two categories within this hyperplane. Through selection of an appropriate kernel, SVM can be used in nonlinear separated problems and it could work well. As mentioned in [LRL07], the SVM method has two main advantages. First, only few samples are needed for training of SVM in high-dimensional spaces. Second, it can minimize upper bound of the generalization errors rather than training errors. For this reason, over-fitting is avoided and model performance is improved. However, several binary classifiers of SVM are required to analyze multiclass problems. Therefore, the computational complexity of SVM increases while the number of classes increases [HL02].

One advantage of FL is that it's easier to understand the internal learning process and the logic of FL, which could be interpreted by using IF-THEN rules. The FL

Table 2.1: Overview of the major applications of HMM

Algorithm	**Typical application field**	**Considered traffic**	**Data used environment**	**Sample references**
HMM	Drivers intention prediction (stop/non-stop behavior)	Intersection	Sensor data	[KSOA03]
HMM	Drivers intention prediction (stop/non-stop, lane-change, left/right-turn behavior)	Highway and urban/inner-city	Driving simulation data	[TSLL15]
HMM	Driving styles recognition	Urban/inner-city	Driving simulation data	[AK07]
HMM	Driving styles recognition	Highway and urban/inner-city	Sensor data (GPS, camera, and accelerometer)	[WCY14]
HHMM	Recognition and prediction of driving behavior (lane-change behavior or intersection precedence/access)	Intersection	Driving simulation data	[GKKO11]
SVM-HMM	Driving behavior recognition	Intersection	Sensor data	[ADSH12]
SVM-HMM	Drowsiness recognition		Driving simulation data	[TSL14]
NN-HMM	Driving behavior prediction (vehicle speed prediction)	Highway	Historical traffic data and driving simulation data	[JF15] [JF16]
ANN-HMM	Driving behavior recognition and safety risk prediction	Urban/inner-city	Driving simulation data	[AK07]
FL-HMM	Drivers intention recognition (lane-change behavior)		CAN-Bus data and sensor data	[DDWL14]
Fuzzy Inference System (FIS)-HM	Driving behavior estimation and driving styles recognition (aggressiveness estimation)	Urban/inner-city	GPS and sensor data	[DL15]
HSS-HMM	Driver's intention estimation	Entrance on a highway	Driving simulation data	[XLW+14]
Sticky HDP-HMM	Driver's intention prediction	Urban/inner-city	CAN-Bus data and sensor data	[TNH+16]
AR-HMM	Driving behavior prediction	Oval circuit course	Driving simulation data	[AMO07]
BP-AR-HMM	Driving behavior prediction	Urban/inner-city	CAN-Bus data and sensor data	[HKI+16]

Table 2.2: Comparision of the different approaches (modified after [DS20b])

Algorithm	Popular algorithms							HMM-derived methods					HMM-combined methods		
	ANN	DBN	SVM	RF	FL	CNN	HMM	HHMM	HDP-HMM	AR-HMM	BP-AR-HMM	ML-HMM	ANN-HMM	SVM-HMM	FL-HMM
Sample references	[CAFH13, DWWB13, BC16, PLKR11, DYF16, JF15, JF16, DHB+11]	[ASABZ13, GBD13, GSZ+17, KKS15, LC12, WY10, KSOA03]	[KPLL13, WWW+14, PLCY14, DYF16, SLZ+11]	[CLZ+17, ZVL14, DWH+20, DSTS20]	[DL15, AAAD12, HWJ+12]	[GMZ18, LKMH17, DWH+20]	[QLH12, CRKK16, JF15, JF16, LKO14, XLW+14, AH16, DWS18]	[GKKO11, ZKZ+15, DS19b]	[NTY+12, NTH+14, TNH+15, TNH+16, HKI+16, WXZ18a, ZZWX20]	[FSJW09, HKI+16]	[HKI+16]	[GKKO11, ZKZ+15, DS19b]	[AK07, ZWYY09, JF15, JF16, ZWL15]	[ADSH12, TSL14, XCL17]	[DS18, DDWL14]
Typical application field															
Driving styles/patterns	yes	yes	yes	yes	yes	-	yes	-	-	-	-	-	yes	yes	yes
Fatigue driving	yes	yes	yes	yes	yes	-	yes	-	-	-	-	-	yes	yes	yes
Drunk driving	yes	yes	yes	yes	yes	-	yes	-	-	-	-	-	yes	yes	yes
Driving behavior and intention	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Data used															
Physiological variables (recorded by Bio-sensor)	yes	yes	yes	yes	yes	-	yes	-	-	-	-	-	-	-	-
Images (recorded by camera or eye tracker)	yes	yes	yes	yes	yes	yes	yes	-	-	-	-	-	-	-	-
Vehicle operation signals (recorded by CAN-Bus, driving simulator, or other sensors)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Vehicle movement (recorded by smart-phone, driving simulator, or other sensors)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Traffic information (recorded by GPS, driving simulator, or other sensors)	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Other ability															
Temporal pattern recognition	-	yes	-	-	-	-	yes	yes	yes	yes	yes	yes	yes	yes	yes
Results	ACC: 73 % - 95 %	DR: 82 % - 95 %	ACC: 84 % - 91 %	ACC: 98 %	ACC: 83 %	ACC: 82 % - 86 %	ACC: 82 % - 90 %	ACC: 99.7 %	MAE of brake pressure: 0.5-0.8 MPa	MAE of brake pressure: 0.1-0.5 MPa	MAE of brake pressure: 0.1-0.2 MPa	ACC: > 90 %	RMSE of speed: < 4 % MAPE of speed: < 5 %	ACC: 85 % FAR: 3.8 %	ACC: 85 % - 95 % DR: > 82 % FAR: < 10 %
Results based on real / experimental driving	real	real	real	real	exp.	real	exp.	exp.	real	real	real	exp.	real	exp.	exp.
Online / Offline test	Offline	Online	Offline	Offline	Offline	Offline	Offline	Online	Offline	Offline	Offline	Offline	Offline	Online	Offline
Related reference	[ZSF14]	[LC12]	[DYF16]	[CLZ+17]	[BNKZ11]	[GMZ18]	[TSLL15]	[ZKZ+15]	[HKI+16]	[HKI+16]	[HKI+16]	[DS19b]	[JF16]	[XCL17]	[DS18]

yes: available
-: not available yet
ACC: Accuracy
DR: Detection rate
False alarm rate
MAE: Mean absolute error
RMSE: Root mean square error
MAPE: Mean absolute percentage errors

approach has the ability to deal with nonlinearities and uncertainties. As described in [MLP12], the presented model based on FL and NN is able to classify different driving patterns like aggressive and moderate driving behaviors. It has a high reliability and feasibility with simulated data. In [BNKZ11] it can be seen that a high accuracy (83 %) of driving style recognition can also be obtained using FL. The authors propose a model to determine the individual driving styles from aggressive, anxious, economical, keen, and sedate driving. The authors pointed out that this approach is valuable for future driving assistant system for each individual driver. The disadvantage of FL is that it requires high computation time and easily over-fitting to the traininig data.

As a probabilistic and graphical model, one advantage of DBN is that it is also easier for a human to understand directly the probabilistic relationships between the nodes [ASABZ13]. As stated in [LC12] the DBN algorithm has the ability to integrate different categories of parameters, even the collection or measurement methods of these parameters are different.

The CNN approach is widely used in computer vision and pattern classification tasks [GMZ18] [LKMH17]. In terms of performance, CNNs typically outperform artificial neural networks. The approach can extract relevant features from images and videos in a more detailed manner. However, it requires more training data and it may lead to over-fitting imbalanced class labels.

In comparison to other algorithms like ANN, HMM, CNN, SVM, etc. RF uses multiple models at the same time to calculate the results. Each tree is trained by a subset of features, which are selected randomly. The results (predictions) of all the trees combine into a final result, which is obtained through the majority voting result. This means that avoiding over-fitting is possible. the tree-like structure of RF, the RF algorithm is a suitable solution for multi-class classification.

As HMM is designed as a probabilistic graphical model, one advantage of HMM is that it is easier to understand directly the probabilistic relationships between the nodes. Based on the principle of HMM, the current state also depends on the state at the previous moment [LZT+14]. Therefore, another advantage of HMM compared to SVM and RF is that it has the ability to handle dynamic data and temporal pattern recognition. Using HMM the class label is determined by calculating the probability, rather than obvious boundaries. In addition, through the summarization and comparison of various studies, the authors in [LZT+14] conclude that the HMM algorithm has a high accuracy and a very good performance in real-time driving behavior prediction. It was also proven in other studies, eg. in [TSLL15] that high accuracies between 82-90 % can be achieved when predicting lane changing, turning, and stopping behaviors. The major disadvantages of HMM are that the number of assortment of hidden states must be known before training, therefore this algorithm is not suitable for long-term forecasting systems [LZT+14]. However, some studies have shown that HMM-derived algorithms could effectively solve these problems, such as Sticky HDP-HMM [FSJS07] [TNH+16] and BP-AR-HMM [FSJW09]. Other algorithms [JF15] [JF16] [TSL14] [XCL17] [DDWL14] based on a combination with HMM were proposed to improve the performance of driving behavior model, such as NN-HMM, SVM-HMM, FL-HMM, and other similar algorithms.

2.3.2 Data collection and variable selection

As mentioned, modeling of regular driving behavior includes input, information processing, and output processes. Information processing (using various algorithms) and output (the different research objectives) have already been summarized in section 2.2. Another important aspect is about collection of input data including the selection of suitable features. The major research objectives, data collection methods, and the corresponding variables in the fields of driving styles/patterns, fatigue/drunk driving, and driving behavior recognition are listed in Table 2.3 to Table 2.5 respectively.

Data collection

Three major options to collect the data for modeling processes can be distinguished:

- Simulate all scenarios in the driving environments using driving simulator. Various software are available to run a driving simulation. During the driving the related data can also be collected and saved [TSLL15] [AK07] [BNKZ11] [HES12] .
- While driving in the real world, related data (signals) can be collected by various hardwares/sensors. These hardwares/sensors are used to analyze the movement of the vehicle and infer the driving behavior accordingly. In general, the sensors can be divided into two types.

  The first class of sensors are directly installed on the vehicle, which can be further divided according to the integration level to the vehicle, and it usually contains two kinds. One kind of sensors is fully embedded with the vehicle and uses CAN Bus to process collected data (eg. vehicle speed, strokes, and press of acceleration/brake pedals) [CRKK16] [MAA17] [LMT13] [GKO14]. Another kind of sensors is installed extra on the vehicle to measure the data. For example, like the sensor used to measure the distance between the vehicles, the cameras use to record eye/face activities [QLH12] [LC12] [WY10] [ASABZ13].

  A second class of sensors are installed on the side of a road and can be used to record the information of driving environment [HWJ+12].

- Integrated sensors on mobiles also allow collecting data. Multiple sensors of the smartphone including GPS, accelerometer, gyroscope, and magnetometer can be directly used to detect the current position, speed, acceleration in 2 or 3 dimensions [DTB+10] [EMAY12] [KB14]. One of the main advantages of using smartphone to collect data is that they are ubiquitous and convenient.

Variable selection

As shown in Table 2.3, driving styles/patterns recognition model is mainly used to recognize the normal (safe) and abnormal (aggressive, risky, anxious, etc.) driving styles. The most commonly selected variables for driving styles recognition are vehicle speed, acceleration, engine speed (RPM), stock of pedal, and press of pedal [AAAD12] [DL15] [BC16] [ASABZ13] [BNKZ11] [EMAY12].

The typically used variables to analyze whether drivers are fatigue or drunken are shown in Table 2.4, it can be summarized into three major categories:

- Using a bio-signal sensor the physiological variables can be measured to recognize the fatigue/drunken driving behavior, eg. EEG, ECG [PLKR11] [SLZ+11], blood alcohol concentration [WY10].
- Images of face/eye, movement trajectory of eye, and blinking frequency recorded by camera or eye tracker are usually given as input data [QLH12] [LC12].
- Through analyzing lateral movement (drifting and swerving) and longitudinal movement (suddenly accelerating and braking) of vehicles, the fatigue and drunkenness of driver can be detected. The relevant variables could be collected by CAN bus [CRKK16] [MAA17] and smartphone [DTB+10].

As shown in Table 2.5, the driving skill research mainly includes the recognition and prediction of lane-changing and speed-changing behaviors in highway [LKO14] [CAFH13] [HES12] and urban environment [DWWB13] [KSOA03] [AK07]. The lane-changing behaviors consists of three aspects: merge and non-merge behaviors at highway lane drops [XLW+14] [DYF16] [HES12], turning behavior at an intersection [KPLL13] [KKS15] [GKKO11] [LMT13], performing overtaking maneuver in highway or urban city [BWKS14] [LKO14] [HES12]. The speed-changing behaviors mainly include stop/non-stop at an intersection and acceleration/deceleration in highway or urban city [LMT13]. The input data differs according to the research objectives, but they are mainly used to describe the vehicle movement and can be collected by driving simulator, sensors, and smartphone. Listed examples are given in Table 2.5.

Table 2.3: Collected data for driving style recognition

Driving styles/patterns	**Data used**	**Selected variables**	**Applied algorithm**	**Sample references**
Below normal, normal, aggressive, and very aggressive	Sensor data (vehicle GPS tracker)	Longitudinal acceleration, lateral acceleration, and vehicle speed	FL	[AAAD12]
Normal, drunk, reckless, and fatigue	Sensor data (Onboard Unit OBU)	Vehicle speed, acceleration, the position in lane, direction of drivers eyes, level of alcohol in the drivers blood, other vehicle's speed, current position, and direction	DBN	[ASABZ13]
Risky and safe	Smartphone (accelerometer, gyroscope, and magnetometer)	Position, speed, acceleration, deceleration, and deflection angle	Dynamic Time Warping (DTW), Bayesian classification	[EMAY12]
Calm, normal, and aggressive	Sensor data (OBD and GPS data)	Vehicle speed, acceleration, engine speed (RPM), spent time in different speed interval, in different acceleration interval, and in different engine speed, etc.	NN	[BC16]
Aggressiveness for calculation of car insurance fee	Sensor data	vehicle acceleration and it's comfort acceleration limit	Kalman Filtering (KF) HMM	[DL15]
Aggressive, anxious, economical, keen, and sedate	Driving simulation data	velocity, acceleration, and the angular point of the curve	FL	[BNKZ11]

Table 2.4: Collected data for fatigue and drunk driving recognition

Fatigue/drunk driving	**Data used**	**Selected variables**	**Applied algorithm**	**Sample references**
Fatigue driving	Sensor data	EEG and ECG	SVM (accuracy 87.5 %)	[SLZ+11]
Fatigue driving	Sensor data	Heart rate, variability (HRV), ECG	NN (accuracy 90 %)	[PLKR11]
Fatigue driving	Sensor data	Eye image	Embedded HMM (accuracy 91.6 %)	[QLH12]
Fatigue driving	Sensor data	eye movement and photoplethysmograph (PPG) signal	DBN (true detection rate 95 %) (false detection rate 10 %)	[LC12]
Fatigue driving	Sensor data (CAN bus)	distances between the vehicle and other vehicles, deviations from lane position, steering wheel angle, velocity, acceleration, etc.	HMM (accuracy 88.77 %)	[CRKK16]
Fatigue and drunk driving	Sensor data	blood alcohol concentration, eye movement and head movement	DBN	[WY10]
Drunk driving	Sensor data (CAN bus)	torque, engine RPM, vehicle speed, acceleration, etc.	Logistic Regression (accuracy 82 %)	[MAA17]
Drunk driving	Smartphone	orientation angles, the lateral and longitudinal acceleration	pattern matching techniques (false negative rate 0 %) (false positive rate 0.49-2.39 %)	[DTB+10]

Table 2.5: Collected data for driving behavior recognition

Driving behavior	Data used	Selected variables	Applied algorithm	Sample references
Right Turn/ Left Turn/ UTurn	Driving simulation data	Vehicle speed and steering wheel angle	HMM	[AK07]
Acceleration, braking, turning recognition	Sensor data (CAN bus)	Brake light, indicator, acceleration, steering angle, and vehicle speed	SVM, k-mean clustering	[LMT13]
Driving decision recognition	Sensor data	vehicle speed, relative distance to the leading vehicle, relative speed of the leading vehicle, previous acceleration, previous yaw angle	NN	[CAFH13]
Stop probability at an intersection	Sensor data	Vehicle speed, pedal strokes of the acceleration and brake pedals	DBN, HMM	[KSOA03]
Lane-changing trajectory prediction	Sensor data	Vehicle offset from the lane mark, yaw angle, and the width of the lane changing vehicle	HMM	[LKO14]
Lane changing prediction	Sensor data	Time-to-Collision (TTC), time to reach the left gap (TTG), time before reaching the end of the entrance of highway, difference of velocity, indicator, accelerate, ect	Scenario model tree (SMT)	[BWKS14]
Lane-changing trajectory prediction	Driving simulation data	Vehicle speed, speed difference between the lead/ lag vehicle, distance between the lead/lag vehicle, remaining distance from the merging vehicle to the end of the merge lane	Fuzzy logic (FL)	[HES12]
Lane-changing trajectory prediction	Sriving simulation data	Position, velocity, acceleration, and time headway of the ego-vehicle	Back-Propagation (BP) Neural Network	[DWWB13]

Table 2.6: Common variables for driving behavior research

Feature	Prediction and recognition of driving skill			Driving styles, fatigue driving, and drunk driving
	Lane change, turn, brake, stop	Prediction of vehicle speed	Prediction of trajectory	
Environmental (ENV) information - State of ego-vehicle or surrounding vehicles				
Vehicle speed	1,2,3	2	1,2	1,2,3
Acceleration	1,2,3	2	1,2,3	1,2,3
Jerk	-	-	2	-
Yaw angle	1,2	-	2	-
Deflection angle	-	1	1	2,3
Vehicle position	1,2,3	2	1,2	2,3
Relative distance	1,2	2	2	2
Relative speed	1,2	2	-	-
Vehicle trajectory data	-	-	1,2	-
Vehicle gap	1,2	-	-	-
Distance to the end of the merge lane	1,2	-	-	-
Time to collision (TTC)	1,2	-	2	-
Time headway	-	-	2	-
Road condition	2	2	-	-
Weather condition	-	2	-	-
Environmental (ENV) information - Driver's operation information				
Engine speed (RPM)	-	-	-	1,2
Steering wheel angle	1,2	-	-	1,2
Indicator	1,2	-	-	-
Brake light	1,2	-	-	-
Stoke, force, or position of acceleration / brake pedal	1,2	2	-	-
Gearbox	1	-	-	-
Eye-tracking (ET) information				
Eye movement	2	-	-	2
Eye blinking frequency	2	-	-	2
Eye image	2	-	-	2
Saccade	2	-	-	2
Physiological information				
EEG	-	-	-	2
ECG	-	-	-	2
EOG	-	-	-	2
Alcohol in blood	-	-	-	2
Photoplethysmography (PPG)	-	-	-	2
Heart movement	-	-	-	2
Sample references	[AK07, LMT13, CAFH13, KSOA03, BWKS14, KPLL13, TSLL15, XCL17, XLW+14, DYF16, ADSH12, DWH+20, DSTS20]	[JF15, MPG+14, CCJJ17, JF16, ZGL+17, LM15]	[LKO14, HES12, DWWB13, LKO14, YZBZ13]	[AAAD12, ASABZ13, EMAY12, BC16, DL15, BNKZ11, SLZ+11, PLKR11, QLH12, LC12, CRKK16, WY10, MAA17, DTB+10]

Data collected using:
1. Driving simulator
2. Real vehicle with sensors
3. Smartphone
-: Not selected

Some popular selected features for driving behavior models and their collecting methods are shown in Table 2.6. The input features are divided into three categories including physiological, eye-tracking (ET), and environmental (ENV) information. In addition, environmental information contains two variable types state of ego-vehicle / surrounding vehicles and driver's operation information. It could be found that, speed, acceleration, and position data can be collected by smartphone. Using these variables it is sufficient to infer the lateral and longitudinal movement of the vehicle. Different sensors are used to record different categories of variables such as signal data (CAN bus), image (camera), video (video sensor), as well as physiological signal (bio-signal sensor).

2.4 Open research questions

As introduced in the previous sections, several different machine learning approaches that can be used to establish driving behaviors model and furthermore to assist the drivers to increase driving efficiency. Each of them has its own strengths and weaknesses. According to the 'No Free Lunch' (NFL) theorem [WM97], no single machine learning algorithm is suitable in every situation. To build a suitable system (model) for driving behavior prediction, the below questions are summarized and will be answered in this thesis.

- In fact, current research proposes new methods to realize and improve driving behaviors prediction. However, only a few articles concern the optimization of an established prediction model to improve the recognition efficiency. Is there an approach that can be used directly to improve the performance of the known machine learning approaches or an established model?
- Many known machine learning approaches are applied in the field of human behaviors prediction. Which known approach performs better? Which parameters affect the performance of a known algorithms? How can they be optimized?
- Many authors suggested that HMM has an advantage for handling time series data, and it is suitably applied for driving behavior or other human behavior studies. In the existing research, different HMM-based approaches are proposed. In general, the design ideas of these HMM-based approaches are roughly divided into two categories: HMM-derived and HMM-combined. What are the differences, advantages, and disadvantages between the two design ideas?

  The HMM is commonly combined with other algorithms or derived into new approaches to improve and to achieve the desired performance of the driving behavior model. However, it would cause an increased complexity and computational cost of the model. Therefore, the reduction of model complexity need to be considered before the design.

- Input parameter can be roughly divided into environmental information including situation of ego-vehicle and the surrounding vehicles, physiological information, and eye-tracking information. What are their respective effects on model performance? Which parameters are suitable for selection as input?

- The aim of this thesis is to predict human driving behavior, so the driving behaviors should be predicted before the actual actions. To train a model, training data need to be labeled and defined as driving behaviors. Is the prediction time influenced by the definition of driving behaviors? How to determine a suitable definition of driving behavior?

- Each driver has own individual characteristics. The development of the driving behavior model for unique driver is helpful for the vehicle to become more human friendly. However, datasets performed by different drivers require models with different predefined parameters (design parameters). Manually setting the values of these parameters with better performance will be very tedious. Therefore, it is necessary to find an effective way to determine these design parameters.

- Hyperparameters defined structure of HMM-based or other ML-based model need to be preset, e.g. the number of hidden states of conventional HMM, the number of hidden layers of HHMM, etc. In addition, for Bayesian Nonparametric HMM, the number of assortment of hidden states depends on the size of the training data. How to adjust these parameters will be discussed in this work.

3 Fundamentals - Known machine learning approaches

Currently there is an abundance of machine learning algorithms and they have been used extensively in the field of classifying human behaviors. Different kinds of machine learning algorithms like Hidden Markov Models (HMM), Support Vector Machines (SVM), Artificial Neural Network (ANN), Convolutional neural network (CNN), Fuzzy Logic (FL), and Random Forest (RF) are popularly used to establish human behavioral models and therefore will be introduced in this chapter.

Assuming a set of training data is given as $\{(X_1, y_1), (X_2, y_2), ..., (X_T, y_T)\}$, where $X \in R^n$ with n as the number of the selected input variables, $y \in \{Class1, Class2, ..., Classl\}$ indicate in total l different class labels, and T is the length of the training data set. Through a classification model, the current input X_t is mapped to a predefined class label y_t, where $t \in T$.

The contents and figures presented in this chapter are modified after previous publications [DWS18][DSW⁺19][DS19a]. Part of the contents are prepared for publication of [DS20c].

3.1 Hidden Markov Model

An HMM describes the relationship between two stochastic processes [Rab89]. One consists of a set of unobserved (hidden) states $Q = \{Q_1, Q_2, ...Q_N\}$, with N as the number of hidden state which cannot be measured directly. The other stochastic process is denoted by a set of M observable symbols $V = \{V_1, V_2, ...V_M\}$. The hidden state and observation symbol at time t are defined as S_t and O_t respectively, that means $S_t = Q_i,\ i \in [1, N]$ and $O_t = V_l,\ l \in [1, M]$. Thus a hidden state sequence is $S = \{S_1, S_2, ...S_T\}$ and an observation sequence is $O = \{O_1, O_2, ...O_T\}$, where T is the length of the sequence. Usually the sample time t is considered as discrete time. The notation time step t is given according to [RJ86]. It is identical to the event discrete step k, usually used in event discrete notations. In each step, a hidden state can be switched to another with a state transition probability $a_{ij} = P(S_t = Q_j \mid S_{t-1} = Q_i),\ i,\ j \in [1, N]$ which means the probability of moving from one state Q_i to another state Q_j. All transition probabilities can constitute a state transition probability matrix

$$A_{N*N} = \{a_{ij}\},\ i,\ j \in [1, N]. \tag{3.1}$$

An observation probability $b_{j(l)}$ defines the probability of an observation V_l being generated from a state Q_j at time t, that means $b_{j(l)} = P(O_t = V_l \mid S_t = Q_j)$. The corresponding observation probability distribution matrix is denoted as

$$B_{N*M} = \{b_{j(l)}\},\ j \in [1, N] \text{ and } l \in [1, M]. \tag{3.2}$$

To describe an HMM it is necessary to use an initial probability distribution

$$\pi_{1*N} = \{\pi_i\} = P(S_1 = Q_i), i \in [1, N], \tag{3.3}$$

which indicates the probability of starting in state Q_i. Using above definitions, a complete HMM is generally defined by $\lambda = (A_{N*N}, B_{N*M}, \pi_{1*N})$. However, an HMM needs to determine the number of classes N and the number of assorted observations M in advance. The larger the value of N or M, the more complicated the model.

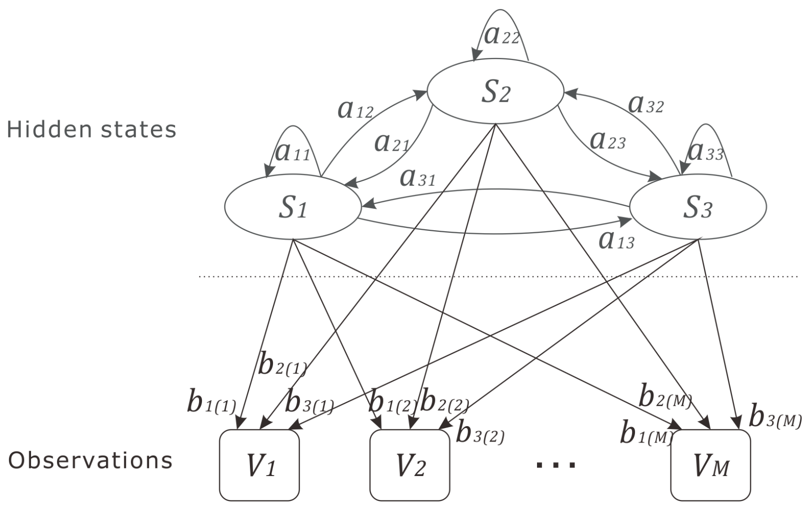


Figure 3.1: Structure of HMM with 3 states ($N = 3$, M = number of observation options) [DWS18]

To apply HMM-based human driver behaviors recognition first the model has to be defined by training. Using the model the most probable state sequence can be estimated. To train HMM the Baum-Welch algorithm (also called expectation maximization) [Rab89] [Moo96] can be used to estimate the maximum likelihood model parameters $\lambda = (A_{N*N}, B_{N*M}, \pi_{1*N})$. So from a given observation sequence O and its corresponding hidden state sequence S, the parameters of the HMM λ can be computed and adjusted to best fit the both sequences. Based on the saved HMM λ, the most probable sequence of driver's behaviors which has the highest probability, can be calculated by using Viterbi algorithm. Here the hidden state sequence and the observation sequence are expressed as a function of simulated time. That is, at time t an hidden state Q_t is equal to y_t and an observation O_t is a vector consisting of observation variables X_t, where y and X are explained at the beginning of this chapter.

3.2 Support Vector Machine

Support Vector Machine (SVM) as a supervised machine learning method is a prominent binary class classifier developed by Vapnik [Bur98]. As known the approach is transforming input data X into a suitable space divided by hyperplane. The process of SVM learning is trying to find an optimal hyperplane $w * X + b = 0$ between data points of different classes to generate a maximal geometric margin $\frac{2}{\|w\|}$ [Bur98], where w is the normal vector to a hyperplane. As shown in Figure 3.2, in which the hyperplane is represented as a separation line, and category label contains only $Class1$ and $Class2$.

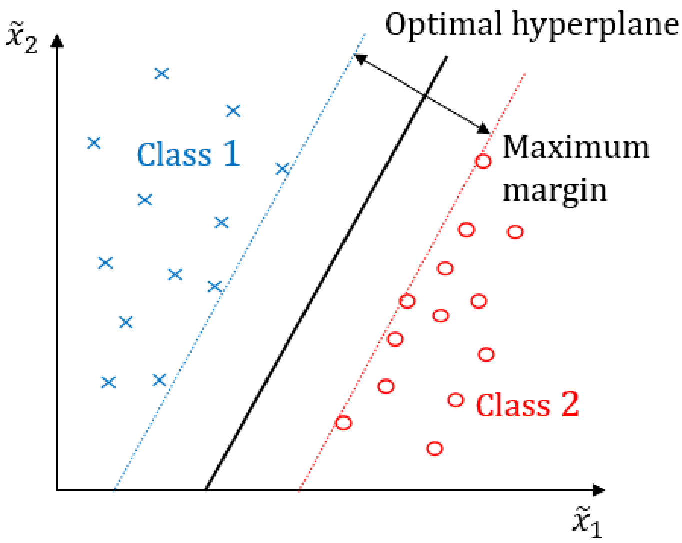


Figure 3.2: Margin of decision boundaries using SVM

According to [SS02], the SVM can be basically divided into two cases: linear and nonlinear SVM. For the linearly separable case, a linear classifier is supposed to make all the training data to satisfy the constraints:

$$\begin{cases} w * X_t \geq 1, & if \quad y_t = Class1, \\ w * X_t \leq -1, & if \quad y_t = Class2, \quad t \in T. \end{cases} \tag{3.4}$$

As explained in [SS02], an optimal hyperplane is placed in a position, for which it has the largest distance to the nearest points of the classes to be distinguished. Finding an optimal hyperplane is to determine a maximum value of margin $\frac{2}{\|w\|}$, which can be written to solve an optimization problem

$$\begin{cases} min \ \frac{1}{2} * \| \ w \ \|^2, \\ s.t. \quad y_t * (w * X_t + b) \geq 1, \quad t \in T. \end{cases} \tag{3.5}$$

However, when the training dataset is non-separable data, direct usage of the above algorithm is not feasible. In this case, some of the training points are not satisfy $y_t * (w * X_t + b) \geq 1$. Therefore, the slack variable $\xi_t \geq 1$ is introduced for each training sample point (X_t, y_t) (Figure 3.3). Combining the slack variables [SS02], the objective function and the margin constraints can be rewritten as

$$\begin{cases} min \ \frac{1}{2} * \parallel w \parallel^2 + C * \sum_{t=1}^{T} \xi_t, & C > 0, \\ s.t. \ \ y_t * (w * X_t + b) \geq 1 - \xi_t, & t \in T. \end{cases} \tag{3.6}$$

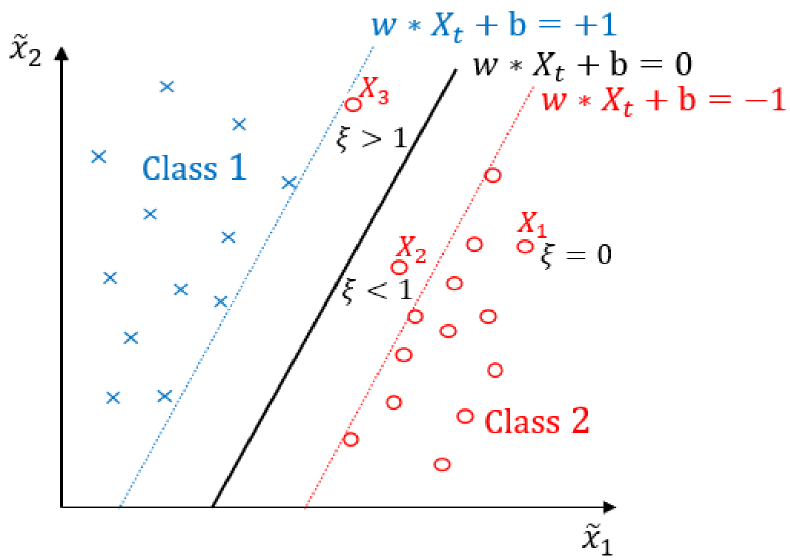


Figure 3.3: Slack variables

As shown in Figure 3.3, there are two cases.

- If $0 \leq \xi_t \leq 1$, the classification is correct, e.g. points X_1 and X_2 are inside the margin.
- If $\xi_t < 1$, the point X_t is misclassified, such as X_3.

Therefore, the value of $\sum_{t=1}^{T} \xi_t$ indicates an upper bound on the number of training errors. The smaller the value $\sum_{t=1}^{T} \xi_t$, the better the classification of the training data. Parameter C named as cost parameter controls the tolerate on the misclassified points and it is chosen by the user.

- When C tends to positive infinity, the SVM classifier does not allow misclassified samples, which may lead to over-fitting.
- When C tends to 0, the SVM classifier only requires that the geometric margin is as large as possible, which will cause some classification errors.

In cases where the given data is nonlinearly separable, the linearization of the hyperplane requires a transformation of the input data into a higher dimensional space where they could be separated linearly. This transformation can be achieved by using the kernel functions, such as the Gaussian kernel function, sigmoidal, RBF mappings [SS02]. During the linearization of the hyperplane, it is additionally ensured to find the maximum margin. After these steps, a linear optimal separating hyperplane will be found.

However, the SVM was originally designed only for two classes. To analyze multiclass problems, several binary classifiers of SVM are required. The most popular solutions of multiclass SVM are presented in [SS02] like one-against-all and one-against-one.

3.3 Artificial neural network

As a common machine learning approach, Artificial Neural Network (ANN) is a computational approach used in machine learning. The idea is inspired from the animal's central nervous systems and applied in fields of human behavior studies. An ANN model is composed of many artificial neurons linked together.

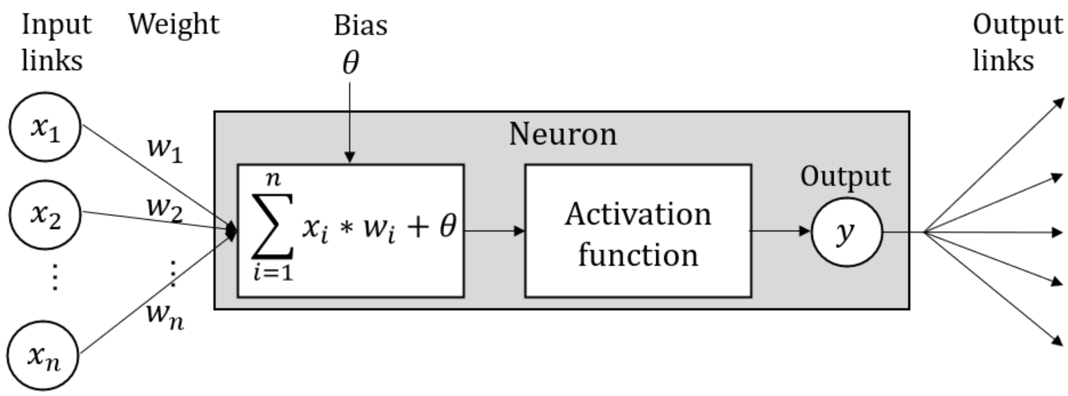


Figure 3.4: Mathematical model of neuron

Each Neuron receives inputs x from many other neurons, then it changes its internal state which is calculated by a weighted sum of its inputs $\sum_{t=1}^{n} w_i * x_i$ and further adds a threshold (bias) θ. An activation function is used to decide whether a neuron should be activated or not. When the value of the weighted sum exceeds a given threshold, a neuron is activated and then sends one output signal to other neurons. Nowadays, many different activation functions are developed in ANN, such as sigmoid, tanh, ReLU, etc. [Gur]. As shown in Figure 3.4 the output of ANN can be expressed as

$$y = F(\sum_{i=1}^{n} w_i * x_i + \theta), \tag{3.7}$$

where F denotes a selected activation function.

The trained ANN network is presented as a dataset. The goal of the ANN is to obtain a desired output according to the corresponding inputs. Typically, ANN contains many layers, the first and the last layers are the input and output respectively. Signals travel from the first to the last layer in the ANN network. Layers between input and output layer are called hidden layers. The number of hidden layers and the number of nodes in each hidden layer are hyperparameters of ANN.

3.4 Convolutional Neural Network

Deep learning is a machine learning technique inspired by the human brain. Convolutional Neural Network (CNN) [Fuk80] is a specialized type of artificial neural network and belongs to the subgroup of deep learning techniques. The algorithm CNN is generally used as feature learning and extraction of various types of data. In recent years, many works have been published to implement CNN in the field of computer vision, document/handwriting recognition, and also driving behaviors recognition tasks [LKMH17] [GMZ18].

The authors in [LBH15] stated that four key ideas behind CNN are local connections, shared weights, pooling, and the use of multiple layers. A typical CNN architecture is formed by a series of different layers including multiple convolutional layers, multiple pooling layers, one or multiple fully-connected layers, and one output layer. The convolutional layer and the pooling layer are stacked together in an alternating mode to form the first stages, i.e. a convolutional layer is connected to a pooling layer, and then this pooling layer is connected to a new convolutional layer, and so on.

A convolutional layer transforms its inputs to multiple feature maps and pass its result to the next layer. This transformation is performed by sliding a window over the input and calculating its output by convolving the local input region with the filter bank of the feature map. The filter in CNN refers to a vector of weights and a bias, which are applied in the equation 3.7 (explained in section 3.3) to determine an output value of a neuron in ANN. Unlike the ANN approach, a distinguishing feature of CNNs is that many neurons can share the same filter. A pooling layer was first introduced in [LGN09] and it is applied to reduce the dimensions of the input data of the next layer, in which the outputs of neuron clusters at one layer are combined into a single neuron in the next layer. Typically there are the maximum pooling and average pooling using the maximum/average value from each neuron cluster at the prior layer respectively. Finally, one or multiple fully connected layers accepts all inputs from the previous layer, and its output is an N-dimensional vector, where N is the total number of all possible classes.

The algorithm CNN is mainly used in image processing applications. For example, the authors of [LKMH17] proposed using CNN and images from radar and camera sensors to predict lane changing intention. In [GMZ18] physiological signals are used to predict lane change behaviors based on a novel Group-wise CNN. However, CNN has not been used to train signals, such as velocities, accelerations, distances, etc. Therefore, in this work it will be demonstrated whether CNN is suitable for identifying driving behavior with these variables. However, inputs for a CNN typically need to be the same size. As assumed at the beginning of chapter 2, a training dataset $\{X_1, X_2, ..., X_T\}$ contains T samples, and each sample X_t is a vector including all the selected input variables. According to the principle of CNN, the training dataset should be formed into 3-dimensional array, each array is composed by all the elements of a vector and transformed into a same-sized matrix, as shown in Figure 3.5. Input variables in the vector are put into a same-sized matrix in sequence, and the remaining vacancies in the matrix are set to zero.

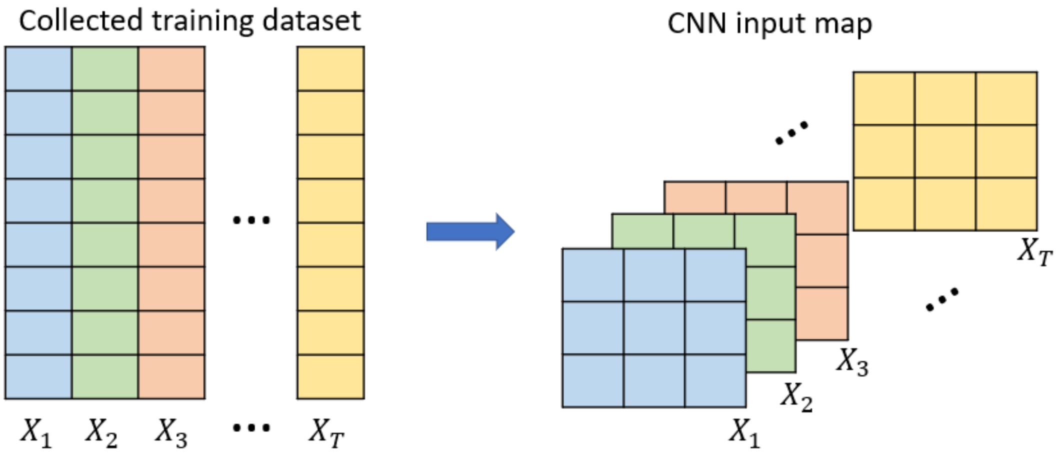


Figure 3.5: Collected variable transformed into CNN inputs [DSW+19]

3.5 Random Forest

Random Forest (RF) was firstly proposed by Breiman [Bre01]. As an extension of decision tree it is used to solve classification or regression problems. A decision tree poses a series of selection problems, and each final answer to these questions is represented by leaves. Each leaf corresponds to a category in the classification problem.

The structure of a decision tree is divided into several stages. Each non-leaf node represents a question that needs to be answered by making a decision between two or more selections. After each selection, the question of the next node becomes more specific. This process is considered as feature extraction, which are evaluated by each node and passed to one branch until finally the level is reached and thus a classification is determined.

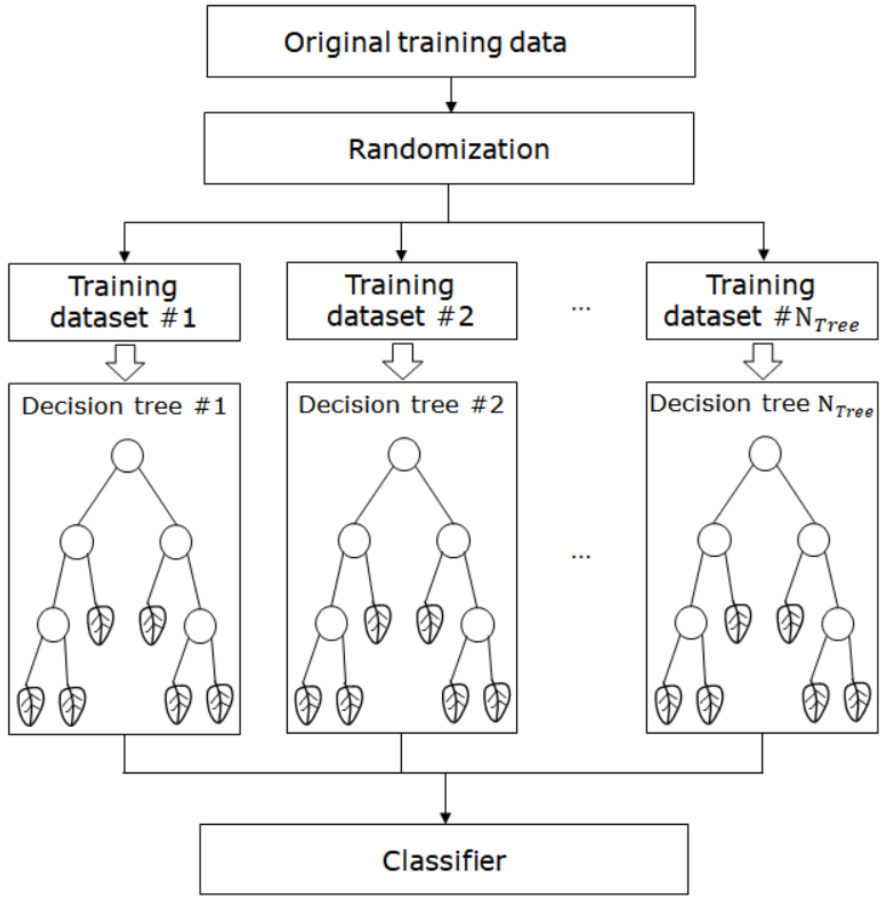


Figure 3.6: Random forest [DSW+19]

The algorithm RF includes two aspects: "random" and "forest". The term "forest" denotes a combination of a set of randomized decision trees, here each decision tree is independent from others. "Random" means the random selection of Bootstrap samples [Bre96], which is generated from the training data set with replacement. The procedure is illustrated in Figure 3.6. First, the number of decision trees to construct N_{Tree} should be defined. For each tree a new bootstrap sample will be selected from training data set. A corresponding decision tree is grown on each bootstrap by recursively repeating the following steps:

(a) At each internal node, selecting m groups of variables at random from the variables X, where $X \in R^n$ and $m < n$

(b) Determining the best split-point among each group of variables

(c) Splitting the node into two or more daughter nodes

After these N_{Tree} decision trees are generated, the output result of the RF is obtained through the voting results of these decision trees. This procedure is called random forest.

3.6 Comparison of algorithms

Each algorithm has its characteristics and therefore advantages and disadvantages. In this section, a brief comparison among different algorithms is given to further understand and analyze the mentioned machine learning methods.

The SVM algorithm generates a linear hyperplane and divides the two classes with the maximal margin between the two categories within this hyperplane. As mentioned in [LRL07], the SVM method has two main advantages. First, only few samples are needed for training of SVM in high-dimensional spaces. Second, it can minimize upper bound of the generalization errors rather than training errors. For this reason, over-fitting is avoided and model performance is improved. However, the accuracy of SVMs results decreases while the number of classes increases. Thus, the SVM is not suitable to solve the classification problem with a larger number of variant classes.

As HMM is designed as a probabilistic graphical model, one advantage of HMM is that it is easier to understand directly the probabilistic relationships between the nodes. Based on the principle of HMM, the current state also depends on the state at the previous moment [LZT+14]. Therefore, another advantage of HMM compared to SVM and RF is that it has the ability to handle dynamic data and temporal pattern recognition. Using HMM the class label is determined by calculating the probability, rather than obvious boundaries.

It was proved that ANN can handle the classification of the signals with large variants, and in many cases using ANN models the result can reach a high accuracy, e.g. in [ZSF14] the prediction results are about 99 %, 94 %, and 74 % for non, left, and right lane-changing respectively. As described in [DHS11], the temporal aspects are not concerned in the ANN model. The authors suggest to use HMM or DBN. According to the principle of ANN, this algorithm has some known advantages and disadvantages which are also proved by many studies. For example, the ANN algorithm has a self-learning ability, it can be quickly adjusted to accommodate new problems. However, the training time is too long, training data are relatively large.

The CNN approach is widely used in computer vision and pattern classification tasks [GMZ18] [LKMH17]. In terms of performance, CNNs typically outperform artificial neural networks. It can extract relevant features from images and videos in a more detailed manner. However, it requires more training data and it may lead to over-fitting due to imbalanced class labels.

Similarly, the RF can be applied for human behavior classification. In comparison to HMM, CNN, and SVM, RF uses multiple models at the same time to calculate the results. Each tree is trained by a subset of features, which are selected randomly. The results (predictions) of all the trees are combined into a final result, which is obtained through the majority voting result. This means that avoiding over-fitting is possible. Due to the tree-like structure of RF, the RF algorithm is a suitable solution for multi-class object classification. It works well with large training datasets rather than small training sets.

4 Improved known machine learning approaches by introducing full scale training loop

To improve the performance of driving behaviors prediction using single known machine learning algorithm, many approaches have been proposed. In general, two approaches are introduced: one is defining suitable input features, another is optimizing hyperparameters of known machine learning approaches.

To define suitable input features, a prefilter is proposed to process and combine signals to form features in this work. However, different datasets require model with different prefilters and hyperparameters. Manually setting prefilters and hyperparameters values with better performance manually will be very tedious. Therefore, an optimized training procedure named a full scale training loop is proposed in this work, in which all the unknown parameters including prefilter and some hyperparameters are denoted as design parameters, and the most suitable design parameters can be determined automatically.

The contents, figures, and tables presented in this chapter are modified after previous publications [DWS18][DS19a]. Part of the contents, figures, and tables are prepared for publication of [DS20c].

4.1 Full scale training loop

Optimization means finding a suitable solution by maximizing or minimizing objective functions. As shown in Figure 4.1, the design of the full scale training loop consists of two important parts including result computation (driving behavior prediction) and model optimization, which are described in the following sub-sections.

4.1.1 Driving behavior prediction model

The driving behavior prediction model and related training are shown in Figure 4.1. During driving, all input parameters are assumed as measurable. Signals are dynamic and therefore changing over time. To transform signals into an input vector and thus forming features for model training, a prefilter is applied in this contribution, which is typical within the automotive field using related electronic equipment with limited accuracy. For each signal, a set of thresholds is used to divide related signal into different segments containing certain information. These segments will replace specific values of the signal and be used to form a new input vector, i.e. signals are quantized and combined. Therefore, the thresholds' values are important to define the input features and to affect the model performance. The proposed

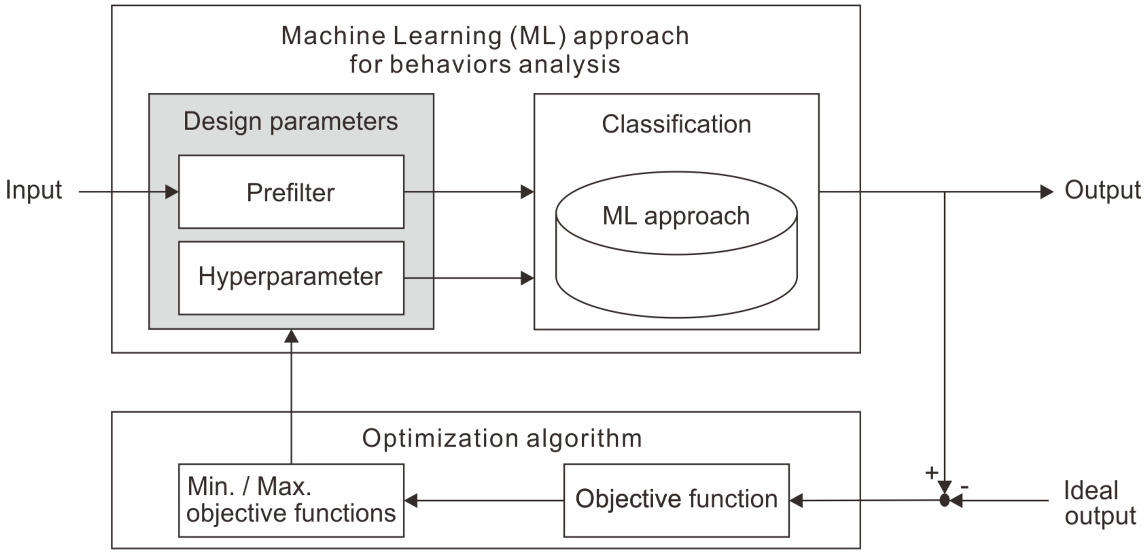


Figure 4.1: Illustration of a full scale training loop

prefilter is composed of all thresholds. Suitable thresholds' values for each input are determined automatically during the training process.

The core of prediction process is realized by a machine learning approach. After selecting a set of design parameters, driving behavior prediction model can be trained using a given input dataset and its corresponding class labels. With this saved model the driving behaviors can be calculated. In the next step, the actual measured driving behaviors from the training data, and the driving behaviors which are calculated by the model will be compared to check the model effectiveness.

4.1.2 Optimization

As shown in Figure 4.1, the second step is to determine the most suitable design parameters. To achieve this, Non-dominated Sorting Genetic Algorithm II (NSGA-II) [DPAM02] is used. The NSGA-II is derived from the NSGA and used to solve Multi-objective Optimization problems (MOPs). With NSGA-II, the most suitable design parameters can be determined to minimize the objective functions which describe the targets of the optimization. Considering possible values of design parameters, each design parameter changes from the minimum to the maximum value of this parameter. Finally, a set of optimal design parameters is determined to minimize the objective functions.

To evaluate the performance of classifiers, accuracy is one of the most commonly used metric due to its simplicity [HS15], however using accuracy is not enough to identify performance of a classifier when dealing with unbalanced data. As explained in [HS15], a high accuracy can be achieved by correctly classifying the majority class while neglecting the minority class. To avoid this issue, three evaluation metrics in cluding Accuracy (ACC), detection rate (DR), and false alarm rate (FAR) are selected, which are widely used for evaluating classifiers [Chu99]. They are calculated based on True Positive (TP), False Positives (FP), True Negative (TN), as well as False Negative (FN) numbers. For explanation a confusion matrix is illustrated (Figure 4.2) as example to describe the parameters for changing lane to the right. True Positive (TP) denotes the number of the events when the estimated maneuver is

		Is changing to the right lane the actual state?	
		Yes	No
Is changing to the right lane the estimated state?	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Figure 4.2: Explanation of confusion matrix [Lane changing to right] (modified after [DWS18])

positive (changing lane to the right) and the actual one is also positive, contrastively False Positive (FP) denotes the number of the events when the estimated maneuver is positive and the actual value not, similarly for True and False Negatives (TN/FN). By comparing the degree of coincidence between the actual state and the estimated state at each moment, the values of ACC, DR, and FAR can be calculated for the complete driving sequence applying the well-known formulas (cf. [Chu99]).

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$DR = \frac{TP}{TP + FN} \tag{4.2}$$

$$FAR = \frac{FP}{TN + FP} \tag{4.3}$$

As previously discussed, the design parameters determine the model training, and thus affect the estimated results. The values of TP, FP, TN, and FN will be defined by the estimated results and finally affect the ACC, DR, and FAR values. In this contribution the design parameters will be selected with respect to the improvement of the aforementioned ACC, DR, and FAR parameters.

As described in [HS15], a combination of ACC, DR, and FAR can be selected for dealing with imbalanced classes, where DR and FAR measures are used to stabilize and optimize ACC performance. To discriminate an optimal solution from various generated solutions, the objective functions in this work are designed and expressed by

$$f_{1-3} = (1 - ACC) + (1 - DR) + FAR, \text{ and} \tag{4.4}$$

$$f_4 = abs(N_{Est} - N_{Act}), \tag{4.5}$$

where N_{Est} and N_{Act} denote number of estimated and actually measured lane change maneuvers respectively, i.e. the equation (4.5) determines the number of miscalculated driving maneuvers. The f_{1-3} represents equation (4.4) for different predictions including left/right lane change and lane keeping. In this equation, values of $1 - ACC$, $1 - DR$, and FAR are combined symmetrically, which means the optimal result will be determined by simultaneously selecting the minimum $1 - ACC$, $1 - DR$, and FAR of each driving behavior.

4.2 Improved approaches

In this contribution, five known machine learning approaches introduced in chapter 3 are selected to verify the performance of the proposed full scale training loop. The selected algorithms combined with the proposed training loop are named as improved models.

The proposed full scale training loop can be used to optimize both model structure and model training. However, the respective effects of prefilter and hyperparameters are not in-depth discussed in the existing researches. This issue will be further discussed and analyzed in this contribution. Therefore, each selected algorithm will be trained with default / optimized prefilters and hyperparameters respectively. As shown in Table 4.1, it can be divided into four groups. Model group M1 is used as a

Table 4.1: Description of model group for driving behavior prediction [DS20c]

Model group	Hyperparameter	Prefilter
M1 (Baseline)	default	without / default
M2	default	optimized
M3	optimized	without / default
M4	optimized	optimized

reference, which is trained by default parameters and raw data. Group M2 also uses default hyperparameters, but prefilter is optimized. Contrastively M3 and M4 are trained with optimized hyperparameters and with (/without) optimized prefilter. It is worth noting that HMM needs data processing to generate sequences. Therefore, a default prefilter is used for models HMM-M1 and HMM-M3, and the related sequence process method is referred to the previous work [DS19a]. In contrast to HMM, other mentioned algorithms of model group M1 and M3 are trained with raw data.

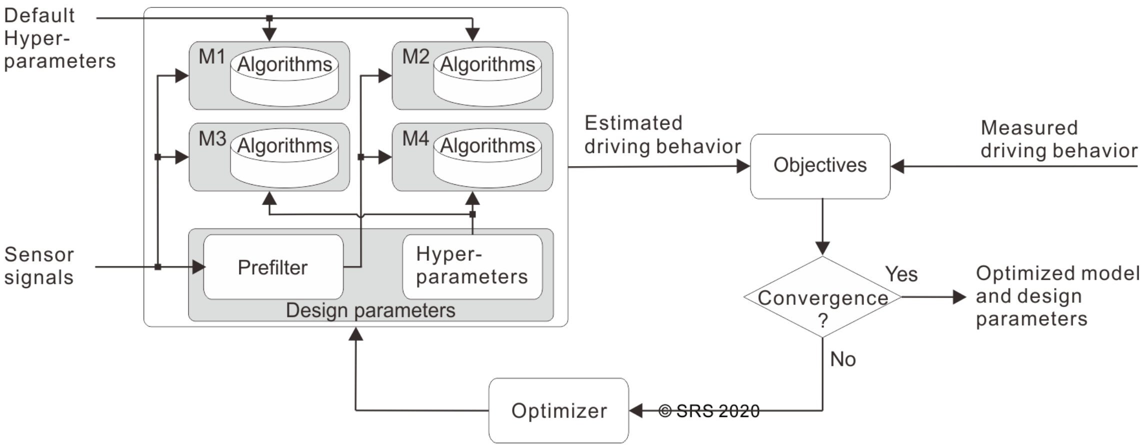


Figure 4.3: Illustration of optimized training procedure [DS20c]

It can be summarized that group M1 is the reference, group M2, M3, and M4 are different improved models. For each algorithm, four different models will be trained to evaluate their prediction performance. Therefore, 20 models are established for known ML approaches. The driving behavior prediction model and related training are shown in Figure 4.3.

4.2.1 Improved Support Vector Machine

In this work different classes refer to different driving behaviors. Driving behaviors are classified by transforming input variables into an input vector and thus forming a distribution in a high dimensional space. The observation vectors of different classes are on different sides of the hyperplane. Using a number of different hyperplanes the training sample can be completely separated.

Obviously, the driving behaviors prediction model in this work is a multiclass problem (LK, LCL, LCR). For this reason a multiclass model has to be chosen. As explained in section 3.2, the most popular solutions of multiclass SVM are one-against-all and one-against-one. Several binary classifiers of SVM are required to analyze multiclass problems. In this work the second one-against-one approach is used to establish the model. For every two classes out of m classes, one binary classifier is trained, so the total number of classifiers is $m(m-1)/2$. Therefore, three classifiers are required for the driving behaviors prediction model. They are used to distinguish every two driving behaviors, such as LK-against-LCL, LK-against-LCR, and LCR-against-LCL. After the three classifiers are trained, they are all applied to classify the driving behaviors for each input data. The final class label is determined by the voting results of these three classifiers.

As summarized in [DKP03], the SVM hyperparameters include regularization parameter (i.e. Cost parameter) C and parameter of kernel function. To simplify the modeling process, linear kernel (without any parameters) is considered in this contribution. Therefore, only one hyperparameter C needs to be optimized. As introduced in section 3.2, hyperparameter C is used to trade off between minimizing training error and minimizing model complexity. The larger the value of C, the less the classifier make classification errors, which may lead to over-fitting. The smaller the value of C, the worse the classification performance. Therefore, hyperparameter C has to be chosen carefully to obtain a good performance.

4.2.2 Improved Hidden Markov Model

As an alternative algorithm Hidden Markov Model (HMM) is used to predict human driver behaviors in this work. The model based on HMM can be defined as a system in which a driving behavior is switched to another with a state transition probability. The driving behaviors performed are the hidden states including LK, LCL, and LCR, so $N = 3$. Thus, the sizes of π and matrix A are 1×3 and 3×3 respectively. The size of the matrix B depends on the number of observation vector choices M, which is unknown and needs to be defined in advance. A prefilter applied in a previous study [DWS18] is used to process and combine the observation variables (signals) to form features. If all observation variables are considered as inputs in an HMM, and each variable is divided into M_l segments. The total number of input variables is denoted as N_{input}, the value of M can be calculated by

$$M = \prod_{l=1}^{N_{input}} M_l, \ l \in [1, N_{input}]. \tag{4.6}$$

A larger number of N_{input} will lead to higher value of M and increase the complexity of the observation probability matrix B (with size $N \times M$) and the HMM. To avoid this problem and to simplify the modeling process, in the previous work [DWS18] two different types of input (driving environment information and operation signal) were proposed to train two sub HMMs, and final results were fused using the results obtained by the sub HMMs. In this work, similar idea is applied, where four different sub HMMs are trained. Each type of input is used to train a sub HMM, i.e. HMM1 (with all velocities), HMM2 (with all distances), HMM3 (with all TTC values), and HMM4 (with all operation signals and lane number). For the next step the probabilities P_i of the four sub HMMs are calculated, the final probability is fused using the weights w_i, expressed as

$$P = \sum_{i=1}^{4} w_i * P_i,\ i \in [1,4], \sum_{i=1}^{4} w_i = 1. \tag{4.7}$$

The hidden state with the highest final probability is predicted as next driving behavior.

4.2.3 Improved Artificial Neural Networks

Typically, ANN contains many layers, the first and the last layers are the input and output respectively. Signals travel from the first to the last layer in the ANN network. Layers between input and output layer are called hidden layers. The number of hidden layers and the number of nodes in each hidden layer are hyperparameters of ANN. In [AYKG14] the authors point out that ANN with high neuron numbers will take considerable time and may lead over-fitting.

In addition, the output node of ANN is a decimal value, i.e. if the trained labels are 0 and 1, the calculated label values are a decimal value between 0 and 1. To determine final results, usually a cut-off threshold is used to distinguish the decimal values into two classes. For example, if the result value is more than 0.5, then it belongs to label 1. Since three labels are used in this contribution, two cut-off thresholds (x_1 and x_2) will be defined.

4.2.4 Improved Convolutional Neural Network

In [AD19] hyperparameters of CNN is optimized using Genetic Algorithms, and the authors summarize CNN hyperparameter contains two types, one type determines the network structure, such as kernel size and number of hidden layer. The other determines the network training, such as learning rate, number of epochs, and batch size. The author designed several possible combinations of hyperparameters and analyzed their impact on CNN. In contrast, in this contribution possible combinations of hyperparameters will not be set in advance, but suitable hyperparameter combinations will be automatically defined through training loop.

In this work, the proposed CNN architecture consists of 6 layers including 1 normalization layer, 2 convolutional layers, 2 pooling layers, and 1 fully connected layer. Hyperparameters like kernel size of each layer, learning rate, number of epochs, and batch size are considered as design parameters.

4.2.5 Improved Random Forest

The algorithm RF contains a set of randomized decision trees, all the decision trees are independent from others. Each decision tree is trained by a randomly selected Bootstrap sample [Bre96], which is generated from the training data set with replacement. The total number of decision trees N_{Tree} should be defined before the training process. After these N_{Tree} decision trees are generated, the output result of the RF is obtained through the voting results of all relevant decision trees. This procedure is called random forest.

It is important that the number of N_{Tree} is unknown. In [OPB12], the authors discussed the effect of the number of random forest trees. It can be concluded that, the prediction performance increases when more trees are used in the model until a certain point is reached, afterwards the result from learning more trees decreases. Therefore, the value of N_{Tree} is also worth optimizing.

4.3 Summary

As previously described, design parameters are a set of unknown parameters needed to be set manually before training. To get a model with better results, a full scale training loop is proposed to automatically determine the most suitable design parameters and then to optimize the performance of the known machine learning approaches. In this work, design parameters consists of two important parts including hyperparameters and a prefilter (Figure 4.3), which effect model structure and model training respectively.

The proposed prefilter is defined by thresholds of each input variable. Based on the prefilter thresholds, signal data will be divided into segments containing certain information. Obviously, threshold parameters are very important, defining implicitly the features for model training and finally affecting the accuracy.

Hyperparameters are parameters whose values need to be set manually prior to the training and usually using default values. It has been proved in [AD19] that optimizing hyperparameters is useful to improve model performance. According to the principles of each algorithm, the hyperparameters of each algorithm are different.

- SVM: Hyperparameters of SVM include cost parameter C and parameter of kernel function. Since linear kernel (without any parameters) is considered in this contribution, only parameter C is considered as hyperparameter and its default value is 1.

Table 4.2: Descriptions of design parameters of different models based on known approaches (modified after [DS20c])

Approach	Design parameters	M1	M2	M3	M4
SVM	Prefilter thresholds	-	-	$\surd$	$\surd$
	Hyperparameter: cost parameter C	-	$\surd$	-	$\surd$
HMM	Prefilter thresholds	-	-	$\surd$	$\surd$
	Hyperparameter: weights $[w_1 ... w_4]$	-	$\surd$	-	$\surd$
ANN	Prefilter thresholds	-	-	$\surd$	$\surd$
	Hyperparameter: - number of hidden nodes - cut-off thresholds $[x_1\ x_2]$	-	$\surd$	-	$\surd$
CNN	Prefilter thresholds	-	-	$\surd$	$\surd$
	Hyperparameter: - kernel size - learning rate - epochs - batch size	-	$\surd$	-	$\surd$
RF	Prefilter thresholds	-	-	$\surd$	$\surd$
	Hyperparameter: number of decision trees N_{Tree}	-	$\surd$	-	$\surd$

$\surd$: Selected
- : Not selected as design parameters
M1 - M4: Described in Table 4.1

- HMM: Due to the reason explained in section 4.2.1, four different sub HMMs are trained in this work. The values of w_i represent the impacts of each HMM and affect the final results, so they are all considered as hyperparameter of HMM. The default values of w_i are all set to 0.25.

- ANN: The number of hidden layers and the number of nodes in each hidden layer are hyperparameters of ANN. This contribution is carried out using Matlab and the default hidden layer is only one layer. To simplify the complexity of comparison, the same hidden layer as the default ANN is considered and the focus is on comparing the impact of different hidden nodes. According to the setting of Matlab, the default number of nodes is 10. In addition to the number of hidden notes, the cut-off thresholds also affect the prediction accuracy of ANN. Thus, the prefilter parameters and two cut-off thresholds (x_1, x_2) are also considered as hyperparameters of ANN. As previously described, the middle value of the values of labels will be selected as the default value (i.e. 1.5 and 2.5).

- CNN: In this contribution, the CNN architecture including one normalization layer, two convolutional layers, two pooling layers, and one fully connected layer. Kernel size of each layer, learning rate, number of epochs, and batch size are unknown and are considered as hyperparameters. The default hyperparameters of CNN are not set in Matlab. Normally, researchers in different studies designed a suitable set of hyperparameters manually, and this process is a tedious problem for many researchers. Similarly, through experience a set of CNN hyperparameters was configured manually in the previous work [DWH+20], and it is implemented as default values in this contribution.

- RF: According to the conclusion of [OPB12], the number of trees N_{Tree} plays an important role on the prediction performance of RF. Therefore, the hyperparameter of RF is N_{Tree} in this contribution. The default number of trees is 30, The default tree number is 30, which is referred to the previous work [DS19a].

In Table 4.2 design parameters of each approach are shown in details. In this contribution, the number of design parameters is fixed for the training.

5 A new approach: Fuzzy Logic (FL)-HMM - a new multi model based machine learning approach

As previously introduced in chapter 2, compared with other machine learning approaches, the approach HMM has an advantage for handling time series data, and it is suitably applied for driving behavior or other human behavior studies. The authors in [LZT+14] pointed out that the HMM algorithm demonstrates a high accuracy and a very good performance in real-time driving behavior prediction. However, the HMM approach requires manual definition of the sequence distribution of the current observation. To solve this problem and to improve the performance of a single HMM, many authors proposed different HMM-based approaches. In general, the design ideas of these HMM-based approaches are roughly divided into two categories: HMM-derived and HMM-combined. According to the two design ideas and the introduced full scale training loop, two new approaches named Fuzzy Logic-based Hidden Markov Models (FL-HMM) and Multi-Layer HMM (ML-HMM) are developed in this thesis.

The contents, figures, and tables presented in this chapter are modified after previous publications [DS18][DS19b]. Part of the contents, figures, and tables are prepared for publication of [DS20a][DS20c].

5.1 Fuzzy Logic (FL)-HMM based on HMM-combined approach

A newly developed Fuzzy Logic-based Hidden Markov Models (FL-HMM) approach is proposed in the previous publication [DS18]. The design idea is similar to some HMM-combined approaches like SVM-HMM [XCL17], in which SVM is used to distinguish different driving scenarios (leaving lane and remaining in lane scene). In the previous publication [DS18] Fuzzy Logic (FL) is used for additional distinction of driving scenes into Very Safe (VS), Safe (S), and Dangerous (D) driving scenarios. This is based on the assumption that different driving scenarios will affect driver behaviors. For example, the drivers need to take a long/short time to change lanes in relatively safe/dangerous driving scenes. Afterwards, a corresponding HMM is trained for each driving scenes respectively and predicting the driving behaviors.

The driving behaviors prediction model based on FL-HMM is shown in Figure 5.1. It is realized in four steps described in the following sub-sections.

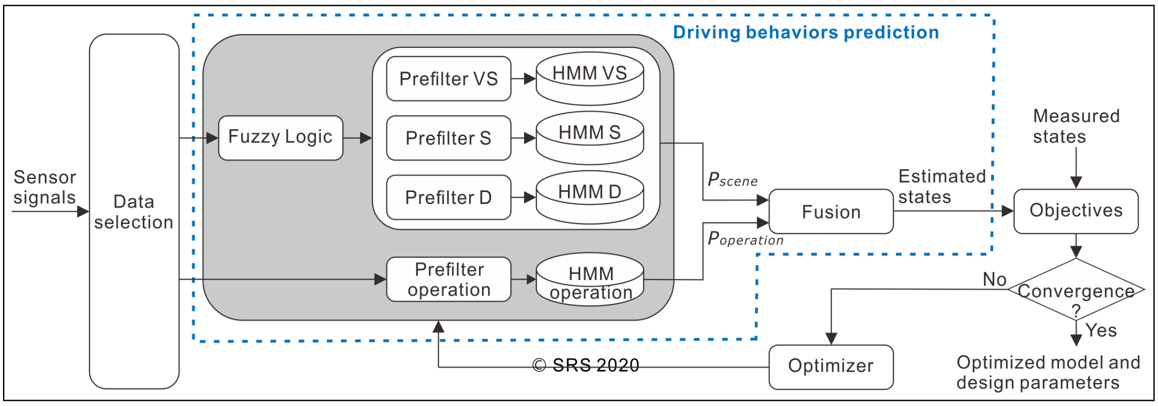


Figure 5.1: Illustration of FL-HMM-based driving behaviors prediction model (modified after [DS18])

5.1.1 Prediction based on driving scenes

Driving on the highway, the relationships between the ego-vehicle and the other surrounding vehicles are the main influences affecting the driver making decisions. In this step, the current driving situation will be mainly discussed.

Fuzzy Logic (FL) is a popular approach used for modeling vagueness introducing many-valued logic. Based on this a classification task can also be realized. It does not require to model all classifications mathematically. The structure of FL is easy to interpret by using IF-THEN rules. The logic of FL-based model can be easily implemented. The FL approach is considered as an extension of Boolean logic, it is based on fuzzy sets and allows to model the truth of statements continuously between true (one) and false (zero) using membership functions [HWJ+12]. Common fuzzy sets are based on triangular, trapezoidal, or Gaussian membership functions [ZB02]. In this contribution, trapezoidal membership function will be used to describe different driving situations.

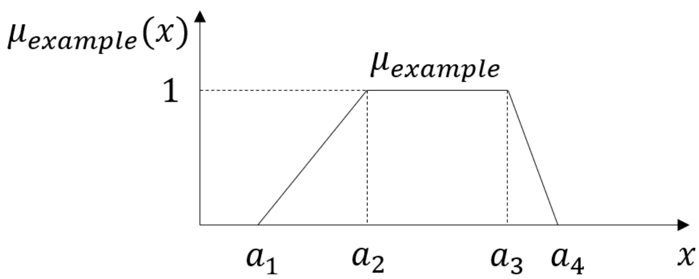


Figure 5.2: Trapezoidal membership function [DS18]

As shown in Figure 5.2, x as input variable so $\mu_{example}(x)$ denotes the degree of membership. A trapezoidal membership function can be described by four parameters

a_1, a_2, a_3, and a_4 as

$$\mu_{example}(x) = \begin{cases} 0, & x < a_1 \\ (x - a_1)/(a_2 - a_1), & a_1 \leq x < a_2 \\ 1, & a_2 \leq x < a_3 \\ (a_4 - x)/(a_4 - a_3), & a_3 \leq x < a_4 \\ 0, & x \geq a_4 \end{cases} \tag{5.1}$$

These parameters a_1, a_2, a_3, and a_4 are four threshold values for the input variable.

Driving in large, middle, and small distance respectively indicates very safe, safe, as well as dangerous scenes for lane change scenario. In addition a Time to Collision (TTC) statement first suggested by Hayward in 1972 [Hay72] is used to determine the safety of lane changes. The value of TTC refers to the time for two vehicles to collide on the same path. Lower TTC values correspond to higher dangerous levels. In the design of Driver Assistance Systems, the use of TTC values for classifying the safety of lane changing maneuvers strongly depends on the speed of the vehicle. In [Win16] TTC values were calculated to prevent forward collisions and reduce the damage caused by the crash. It shows that when the speed is around 130 km/h, the drivers will be warned if the TTC value is less than 3 s, and the drivers need to fully brake if the value under 2 s. However, in reality the drivers often successfully change lanes with low TTC values. In [CKG15], the authors analyzed the TTC values for lane change based on data from the "100-Car naturalistic driving study" collected by Virginia Tech Transportation Institute (VTTI). The results show that the minimum TTC value for lane change is between 2.1-2.7 s, when the speed is ranged from 70-90 mph (i.e. 113-145 km/h). A smaller TTC value denotes that the drivers are in a dangerous scene and need to change lanes as soon as possible if they want to overtake. Therefore, these two variables including the TTC and distance to vehicle in front will be considered as inputs for classification of driving scenes. The first input is the distance to vehicle in front. The corresponding fuzzy values are close, middle, and far. Similarly, the value of TTC to the vehicle in front will be considered as a second input, and the corresponding fuzzy values are low, middle, and high. Finally the output of the fuzzy model are three different driving evaluations denoted as Very Safe (VS), Safe (S), and Dangerous (D). The fuzzy rules are summarized in Table 5.1. In very safe scenes, the drivers possibly take a long time for changing lanes. However, in dangerous scenes the drivers will change lane in a short time or hard brake. Safe scenes contains the largest uncertainty.

The trained FL can determine which driving scene the current situation belongs to, and then switch to a corresponding model of the scene. For each driving scenes a corresponding HMM (HMM VS, HMM S,or HMM D (in Figure 5.1)) will be used to represent the upcoming driving behaviors. To improve the performance of HMM, a prefilter proposed in [DWS18] is used to process and combine signals to form features for the HMM recognition process. The application of this prefilter can

Table 5.1: Fuzzy rules used in driving situation recognition [DS18]

Distance \ TTC	Low	Middle	High
Close	D	D	S
Middle	D	S	VS
Far	S	VS	VS

effectively improve the performance of HMM, and this has been confirmed in the previous publications [DS19a] [DWS18]. To simplify the modeling process, in this contribution a prefilter using five different thresholds is defined. Each observation variable is divided into six segments. However, these thresholds are unknown, so they are also defined as design parameters.

5.1.2 Prediction based on drivers operation

Normal driving behaviors can be predicted through the driving environment. However, sometimes the drivers may make exceptional decisions like changing lanes with sudden acceleration or keeping lane during deceleration. As a supplement to the model based on the driving environment, another HMM will be established based on the driver's operation signals to predict these exceptional driving behaviors.

Therefore, the indicator signal, the steering wheel angle, the accelerator pedal position, and the brake pedal pressure are selected as observation variables of HMM-operation (in Figure 5.1). Similarly, the corresponding prefilter of this HMM is defined by using two different thresholds for each observation variable.

5.1.3 Fusion

As previously mentioned both methods are combined in this work. One model considers the relationships with other vehicles (driving scene), and the other considers the driver's operation. As shown in Figure 5.1, using the two models the probabilities of the next driving behaviors are calculated separately. The final probabilities are fused using the weight w, expressed as

$$P = w * P_{scene} + (1 - w) * P_{operation}, w \in [0, 1]. \tag{5.2}$$

Finally, the hidden state with the highest probability is predicted as next driving behaviors.

5.1.4 Optimization

The last part of the modeling is related to the definition of parameters, here connected with optimization. As previously described, the thresholds of FL, the prefilters of HMMs and w are defined as design parameters affecting the prediction capability:

- FL thresholds definition (prediction of driving scene, selecting HMM and prefilter)
- Prefilter tresholds (defining observation sequence)
- w (affecting driving scene prediction)

To define the best fitting model parameters during the optimization process, suitable objective functions (equations 4.4 and 4.5) introduced in section 4.1 have to be chosen.

5.2 Multi-Layer (ML)-HMM based on HMM-derived approach

The second new approach is denoted as Multi-Layer HMM (ML-HMM) as proposed in [DS19b] and based on the idea of HMM-derived approaches. One of the inspirations comes from an Embedded HMM derived from hierarchical HMM in [QLH12]. The method is used to analyze the eye closure of drivers. The author divided each eye image into three parts, for each part a corresponding HMM is trained as sub-HMM. The obtained results of sub-HMMs are given as inputs to train the major HMM and to estimate the final results.

The second inspiration comes from the RF approach, in which each tree is trained by a subset of features selected randomly. Results of the all trees are combined into a final result, which is obtained through a majority voting result. The effectiveness of the RF approach has been successfully proved in the previous publication [DWH+20] stating that the performance of RF algorithm is the best comparing with ANN, SVM, CNN, and HMM.

The important ideas from the Embedded HMM and RF can be concluded as:

- Not all the features are necessary to be considered as inputs in each single model.
- Each single model is considered to recognize behaviors only for a particular working case.

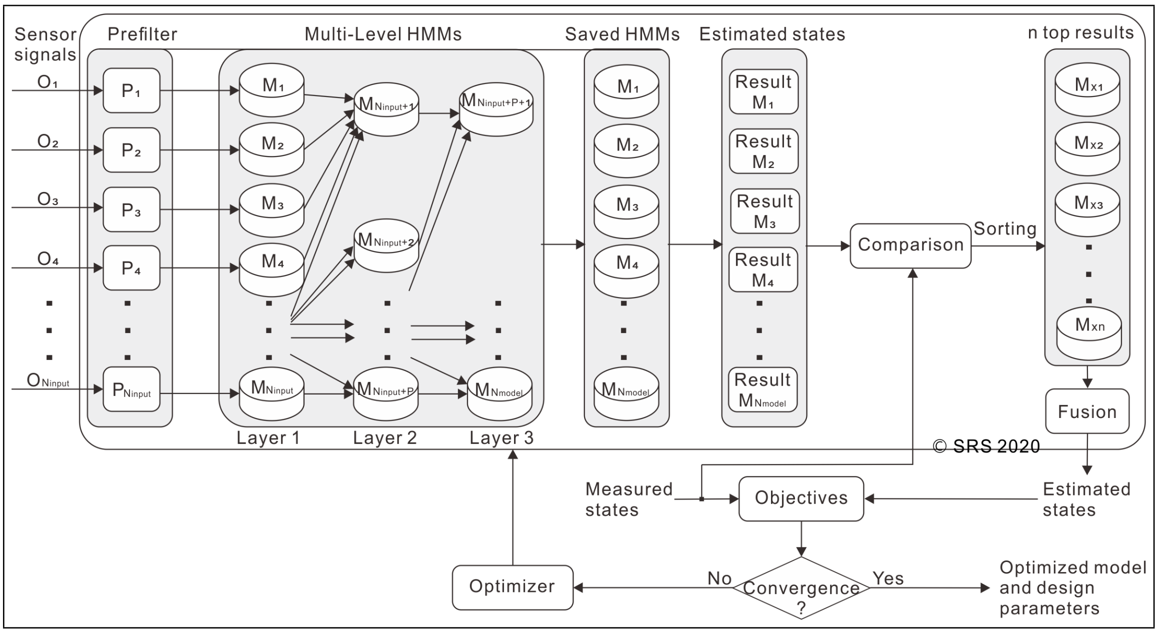


Figure 5.3: Illustration of ML-HMM-based driving behaviors prediction model (modified after [DS19b])

- Results of the all sub-models are combined into a final result.

Based on these ideas, a 3-layer ML-HMM approach is developed for predicting lane changing behaviors [DS19b]. As shown in Figure 5.3, ML-HMM consists of three important processes, which are described in the following sub-sections.

5.2.1 Data preprocessing based on prefilter

Similar to FL-HMM, a prefilter [DWS18] is applied for each observation variable (O_1-O_{Ninput}). However, the prefilters (P_1-P_{Ninput}) are not only used to grade the variable O, but also to classify the derivative of this variable $\dot{O}$. That means, the value and the trend of the variable are considered at the same time. For example, if an input is given as velocity, the segment of the current velocity and the segment of the current acceleration/deceleration should be distinguished by using the prefilter.

To simplify the modeling process, for each observation variable a prefilter is defined by using 20 different thresholds for each O and 5 different thresholds for each $\dot{O}$. Similarly, these thresholds are unknown and defined as design parameters.

5.2.2 Prediction based on Multi-Layer HMMs

However, if all observation variables are considered as inputs in an HMM, and each variable is divided into M_l segments. These segments are combined to form observation vectors which represent different driving situations. If the total number of observation variables (inputs) is denoted as N_{input}, the number of observation vector choices M can be calculated by

$$M = \prod_{l=1}^{N_{input}} M_l,\ l \in [1, N_{input}]. \tag{4}$$

If the values of N_{input} increase, the value of M becomes large, and finally the observation probability matrix B (with size $N \times M$) becomes complicated. This will affect the calculation speed of HMM. Therefore, only one observation variable is considered as the input of the first layer HMM. Then outputs of first layer HMMs are combined into different models containing different information in the second and third layers. As shown in Figure 5.3, the driving behaviors prediction model is realized in three layers.

- First layer HMMs: each HMM is considered to predict the driving behaviors in certain single working cases. Only one observation variable is given as input, and this makes it possible to add more inputs in the model. All HMMs are calculated in parallel, and this helps to reduce the complexity of the second inputs. The obtained results from the first layer are given to the second layer.
- Second layer HMMs: each HMM is established for combined working cases but only selected information is considered. Environmental information and ego-vehicle state are roughly divided into several categories, such as information about all operational signals of ego-vehicle, information about all velocities, all distances, or all TTC values of ego and surrounding vehicles, information about front, behind, left front/behind, or right front/behind driving environment, etc. The results of the first layer HMMs constituting the corresponding information will be used to train a second layer HMM. For example, if a second layer HMM consists the information about all distances, its inputs should be all results of the first layer HMMs using distance.
- Third layer HMMs: similarly, the inputs of the third layer are the inferential results from the second layer. Each third layer HMM represents driving behavior in a combined working case and all information is considered. That is, it combines several second layer results that contain different information, and ultimately results are obtained considering all observable variables.

It is important to know that all sub HMMs of each layer are calculated in parallel and all of them can be used to predict driving behavior. As mentioned in the second section, if the outputs of each layer are used as the inputs to the next layer, this can significantly reduce the value of M for each HMM and simplify observation probability matrix B. This makes the approach being simpler and faster.

5.2.3 Fusion

As shown in Figure 5.3, all trained HMMs from the previous step are saved, and for each HMM a respective result will be calculated. The results from all HMMs will be compared with the actual behavior, and by doing that, the performance of all HMMs are evaluated, then the best performed n HMMs (M_{x1}-M_{xn}) will be sorted out, where the subscripts $x1$-xn indicate the serial numbers of top n HMMs, i.e. $xi \in [1, Nmodel]$ and $i \in [1, n]$.

For the next step the probabilities of all n HMMs are calculated, the final probability is fused using the weights w, expressed as

$$P = \sum_{i=1}^{n} w_i \cdot P_{xi}, \ i \in [1, n], \tag{5.3}$$

where w_i represents the impact of each n top HMM to the driving behavior prediction. Finally, the hidden state with the highest final probability is predicted as next driving behavior.

5.2.4 Optimization

Similar to FL-HMM, the proposed full scale training loop is applied in ML-HMM to define the suitable design parameters, and therefore to optimize the performance of driving behaviors prediction. As previously described, the prefilters of HMMs, the number of the best performed HMMs n, and the weights w are unknown and affect the prediction capability:

- Prefilters thresholds (i.e. thresholds of observation segment) define the observation sequence and finally affect the prediction performance of each HMM.
- Value of n determines how many HMMs are selected to calculate the final result.
- Values of w represent the impacts of each top HMM and affect the final results.

Therefore, these parameters are defined as design parameters of ML-HMM. By using NSGA-II the most suitable design parameters will be determined to minimize the objective functions (equations 4.4 and 4.5 introduced in section 4.1) which describe the targets of the optimization.

5.3 Summary

Except for using a single HMM to establish a driving behavior model, two design ideas (HMM-derived or HMM-combined approaches) can be concluded from the existing researches to improve the HMM performance. Based on the two design ideas two newly developed approaches (FL-HMM and ML-HMM) are considered. Using the proposed full scale training loop introduced in chapter 4, the two considered approaches are trained to establish driving behaviors prediction models.

To discuss the effects of hyperparameters and prefilters, for each approach four different models are established. The detailed explanation of the four model group are listed in Table 4.1 of chapter 4. Design parameters of each model group of FL-HMM and ML-HMM are shown in Table 5.2.

Table 5.2: Descriptions of design parameters of FL-HMM and ML-HMM (modified after [DS20b])

Approach	Design parameters	M1	M2	M3	M4
FL-HMM	Prefilter thresholds	-	-	√	√
	Hyperparameter: - FL thresholds - Weights $[w_{scene}\ w_{operation}]$	-	√	-	√
ML-HMM	Prefilter thresholds	-	-	√	√
	Hyperparameter: - Number of the best performed HMMs n - Weights $[w_1\ ...\ w_n]$	-	√	-	√

√: Selected
- : Not selected as design parameters
M1 - M4: Described in Table 4.1

- FL-HMM: As introduced in [DS18], hyperparameters of FLHMM are the thresholds of FL and the weights w. The default value of w is defined as $[0.5, 0.5]$.
- ML-HMM: Similarly, the related hyperparameters of MLHMM are summarized as the number of the best performed HMMs n, and the weights w [DS19b]. The default values are $n = 10$ and $w = [1/4, 1/4, 1/4, 1/4]$.

6 Experimental results

This chapter presents the experimental results of the driving behavior prediction model based on different approaches introduced in chapter 3, 4, and 5. First the experiment setup is described. Then, training and test as well as the suitable NSGA-II design parameters are used to develop the models. Finally experimental results will be presented.

The contents, figures, and tables presented in this chapter are modified after previous publications [DS18a][DS19a][DS19b][DSW+19]. Part of the contents, figures, and tables are prepared for publication of [DS20a][DS20c].

6.1 Experimental design

As concluded in chapter 2, machine learning algorithms can used to establish driving behaviors model and furthermore to assist the drivers to increase driving efficiency. However, there are two questions about how to build a suitable system (model).

- Input parameter can be roughly divided into three categories including physiological, eye-tracking, and environmental information. As summarized in chapter 2, physiological data are usually used to predict fatigue or drunk driving, but not to predict driving behaviors. Environmental (ENV) data is widely used in the field of driving behavior prediction. In contrast, only a few studies consider eye-tracking (ET) data as input. Therefore, in the first experiment ENV and ET data are selected as input to discuss what are the roles of using ENV data and ET data?
- Which algorithm performs better and should be selected?

To answer the above questions, first an experiment abbreviated as "E1" is designed. Only the five mentioned known machine learning algorithms including SVM, HMM, ANN, CNN, and RF are used. Furthermore, the performance of these algorithms will also be evaluated, the role of using eye-tracking (ET) data and the definition of lane changing behavior will be discussed. The conclusion of the first experiment (E1) including input and algorithm selection will be applied to the second experiment.

The second experiment (E2) consists of three steps, with which the following questions will be discussed and answered.

- Can the proposed training loop be used to improve the performance of the known machine learning approaches? (Section 6.3.2)

- Can the newly developed HMM-based approaches (FL-HMM and ML-HMM) successfully predict driver behaviors? (Section 6.3.3)

- The design parameters include hyperparameters and prefilters, what are their respective effects for improving model performance? Which approach performs best among all considered approaches? (Section 6.3.4)

In the E2 experiment, all the mentioned approaches in this thesis will be used to establish driving behaviors models. The impact of the proposed training process on each approach will be discussed.

6.1.1 Laboratory configuration

Driving simulator is a dedicated engineering tool developed to accurately reproduce the driving scenario in a virtual environment. A driving simulator provides a realistic environment and different situations in which allows people to drive without the risks and restrictions in real life. A professional driving simulator SCANeR^{TM}studio as shown in Figure 6.1 is applied at the Chair of Dynamics and Control (SRS) to perform driving simulation for training and test of the proposed approaches. The simulator is equipped with five monitors with 180 degree field of view, base-fixed driver seat, steering wheel, and pedals. The three rear-view mirrors, which are essential for a driver to decide to change lane, are displayed on the corresponding positions of the monitors. The data acquiring frequency of the driving simulator is 20 Hz.

Figure 6.1: Driving simulator, Chair Dynamics and Control, U DuE

To understand what the simulated world looks like, the simulator also provides virtual sensors to collect data, such as camera, radar, lasers, and GPS. Therefore, a comprehensive understanding of the vehicle's environment can be built based on these collected data. As shown in Figure 6.2, a driving environment will be created by using function scenario of SCANeR^{TM}studio. While the simulation is running, the signals will be collected by different sensors. For example, radar sensor can detect targets such as other vehicle's type, name, absolute and relative positions, speed, etc. With GPS sensor, position of a vehicle is observed and transformed as coordinates. Vehicles can use camera images to find information about the road markings such as lane lines, or track other vehicles on the road. Like how humans see the world, computer vision is necessary for vehicles provided by the simulator to recognize the traffic lights and signs. With all the information collected, a driving assistant system (human behavior prediction model) can be established and simulated in Matlab, and finally suggestions/warnings are given to driver to control the vehicle's direction, speed and so on.

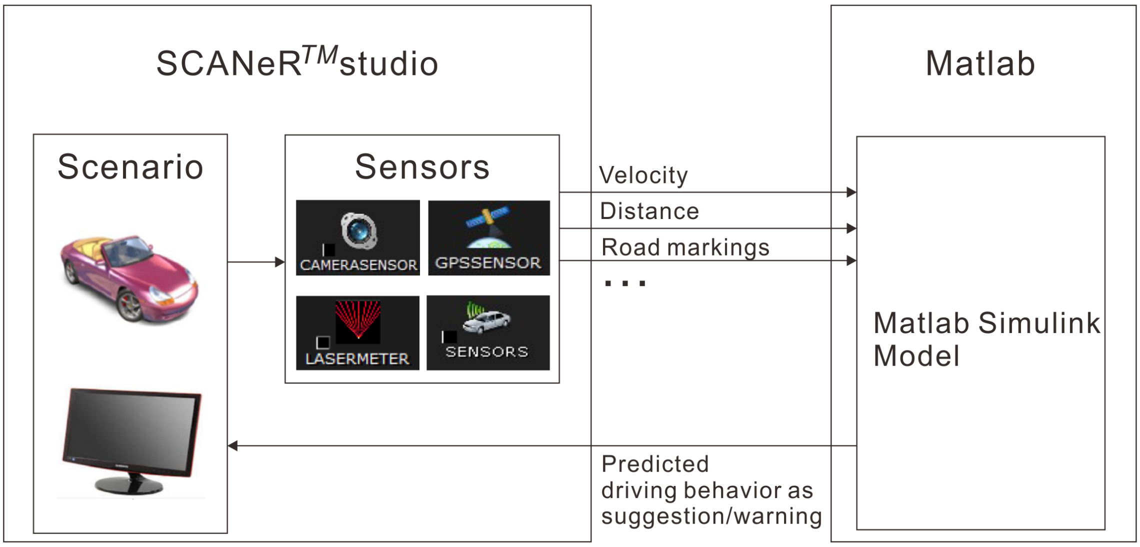


Figure 6.2: Illustration of Working Structure

6.1.2 First experiment (E1) - Experimental design to find out suitable definition

The driving simulator can be additionally coupled with an Eye-Tracker. The Eye-Tracker is placed between the screens and the driver's seat and faces to the driver's face. The cameras of the Eye-Tracker determine the driver's facial features positioning, so the head rotation and the direction of gaze can be determined.

Data collection

A highway with six lanes for two directions as well as simulated traffic environment is used for the experiment. During driving, the participant could perform overtaking maneuver when the preceding vehicle drives slowly. After overtaking the participant could also drive back to the initial lane. The time points of changing lane to left and right were decided by the participant. Following the german traffic rules, it is only allowed to overtake on the left lane. In total 10 participants were recruited. They all held valid driving licenses. The original training dataset is related to each participant performing a 30 minutes drive. Data from another 10 minutes drive are used for test.

In this experiment, the continuous environmental (ENV) and eye-tracking (ET) data collected from SCANeR^{TM}studio and faceLAB consist of 51 variables, each variable denotes one feature of the recorded data. To simplify the modeling process, only a part of the variables need to be taken into consideration of the prediction model. Based on experience, the ENV data available for observations in Table 6.1 are considered to describe the current driving situation. The selected variables from

Table 6.1: Descriptions of selected input variables (E1) [DSW+19]

Data collection	Symbol	Definition
Environmental data (ENV)	v	Velocity of ego-vehicle
	d_f	Distance to vehicle in front
	d_{fl}	Distance to vehicle in left-front
	d_{fr}	Distance to vehicle in right-front
	d_{bl}	Distance to vehicle left-behind
	d_{br}	Distance to vehicle right-behind
	TTC	Time to Collision: $d_f/(v - v_f)$, where v_f denotes velocity of vehicle in front
	α	Heading angle of ego-vehicle
	i	Number of the current lane
Eye-tracking data (ET)	$Saccade$	Saccade
	$Blink$	Blink
	F_{Blink}	Blink frequency
	N_{Screen}	Screen number
	x	Screen coordinate (x-axis)
	y	Screen coordinate (y-axis)

ET are used to detect the saccadic eye movements of the driver. Details of all the selected variables are given in Table 6.1. In the real world, these variables will be taken from different sensors, such as camera, radar, and lidar [VB15] [Fle08]. In [FWL+18] the authors introduced to use a front RGB camera of a smartphone to capture eye and face images. This can solve the problem of integrating Eye-Tracker hardware into the existing on-vehicle systems.

The purpose of this contribution is not only to analyze the performance of different classification methods, but also to evaluate the influence of ET data for driving behavior prediction. For one driver, one original dataset will be processed into three different training datasets:

- Case 1: Only ET data (6 inputs)
- Case 2: Only ENV data (9 inputs)
- Case 3: Combination of ET+ENV data (15 inputs)

For each algorithm, the three different training datasets are used to train three different models to evaluate their prediction performance.

Data processing

During driving simulation, the current lane i can be determined using the vehicles center point. Therefore, by comparing the value of lane i at different times, the lane changing of the vehicle can be determined at time t_{lane} (Figure 6.3).

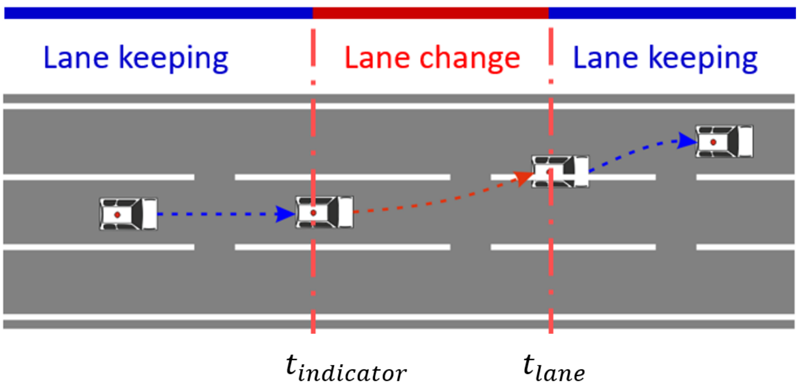


Figure 6.3: Illustration of lane changing behavior (E1)

In the experiment, the driver is expected to activate turn indicator before lane changing. The moment of turning on lights indicates the starting time of the lane changing intention and therefore it is defined as the latest moment in time when the driver intends to pass the lane.

The interval from the beginning of lane changing $t_{indicator}$ to the completion of lane changing t_{lane} is the total required time for the lane changing behavior. It can be expressed as $t_{change} = t_{lane} - t_{indicator}$. However, the driver sometimes changed lanes

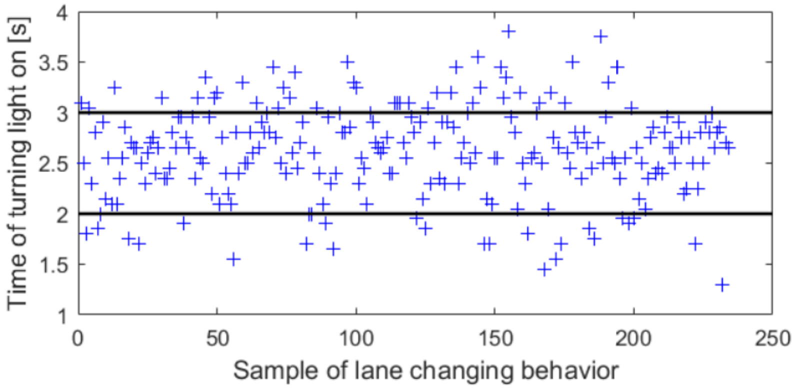


Figure 6.4: Starting time of the lane changing behavior of driver #1 defined by $t_{indicator}$ (E1) [DSW+19]

and forgot to turn on the light. In this case, the average value of t_{change} is set as the time, at which the driver decided to change the lane. For example, the time of lane changing behavior of driver #1 is recorded in Figure 6.4. The x-axis and y-axis represent the samples of lane changing behaviors and the time required (t_{change}) for each lane change respectively. It can be clearly seen that the time of driver #1 deciding to change the lane is always between 2 s and 3 s before the lane change is completed. The average value of t_{change} of driver #1 is 2.609 s. Thus, the decision time of lane changing of the driver #1 will be considered as 2.609 s before the action when the lane changing behavior is made without turning on the signal.

6.1.3 Second experiment (E2) - Experimental design to study the strategy of full scale training loop

Similar to the first experimental setup, while driving the driver is interacting with a simulated traffic environment. Following the german traffic rules, the participants could decide to change driving behaviors according to their own willingness. The differences are in total 17 participants were recruited to take part in the study. They all held valid driving licenses. In addition, the driving scenario for 9 drivers is a highway with four lanes in two directions. Another 8 drivers drove on a six-lane highway driving scene. The original training dataset is related to each participant performing a 30-40 minutes drive. Data from another 10-15 minutes drive are used for test.

Data collection

Driver's driving behaviors depend on the current environment conditions and the individual driver's characteristics. On the highway, the relationships between the

Table 6.2: Descriptions of selected input variables (E2)

Input	Definition	Range	Unit	Data type
		Environmental information		
v_{ego}	Velocity of ego-vehicle	[0 220]	km/h	Real
v_f	Velocity of vehicle in front	[0 220]	km/h	Real
v_{fl}	Velocity of vehicle in left-front	[0 220]	km/h	Real
v_{fr}	Velocity of vehicle in right-front	[0 220]	km/h	Real
v_{bl}	Velocity of vehicle left-behind	[0 220]	km/h	Real
v_{br}	Velocity of vehicle right-behind	[0 220]	km/h	Real
v_b	Velocity of vehicle behind	[0 220]	km/h	Real
d_f	Distance to vehicle in front	[0 250]	m	Real
d_{fl}	Distance to vehicle in left-front	[0 250]	m	Real
d_{fr}	Distance to vehicle in right-front	[0 250]	m	Real
d_{bl}	Distance to vehicle left-behind	[0 250]	m	Real
d_{br}	Distance to vehicle right-behind	[0 250]	m	Real
d_b	Distance to vehicle behind	[0 250]	m	Real
TTC_f	TTC to vehicle in front	[0 12]	s	Real
TTC_{fl}	TTC to vehicle in left-front	[0 12]	s	Real
TTC_{fr}	TTC to vehicle in right-front	[0 12]	s	Real
TTC_{bl}	TTC to vehicle left-behind	[0 12]	s	Real
TTC_{br}	TTC to vehicle right-behind	[0 12]	s	Real
TTC_b	TTC to vehicle behind	[0 12]	s	Real
		Drivers operation information		
α	Heading angle of ego-vehicle	[-3.14 3.14]	rad	Real
S	Steering wheel angle	[-3.14 3.14]	rad	Real
P_a	Accelerator pedal position	[0 1]	-	Real
P_b	Brake pedal pressure	[0 400]	N	Real
Ln	Current lane number	[1, 2]	-	Integer
I	Indicator	[0, 1, 2, 3]	-	Integer
G	Gearbox	[1, ...5]	-	Integer

ego-vehicle and other surrounding vehicles are the main factors effecting the decision making of the driver. In this contribution, the feasibility of data collection must be considered while defining input parameters. As shown in Table 6.2, in total 26 observation variables are selected as input, which belong to two aspects including information about surrounding vehicles and states of the ego-vehicle. All input variables are assumed to be measurable (for example by driving simulator). In the real world, data of these parameters will be collected through different sensors, such as camera, radar, and lidar [VB15].

Data processing

To label the data as driving behaviors, the signal data need to be classified and processed. As explained in section 6.1.1, the current lane i can be determined through the position of the vehicle's center point in the driving simulation. Therefore, the lane changing behavior at time t_{lane} can be recognized when the value of lane i is changed. However, the drivers sometimes changed lanes and forgot to turn on the light in the experiment. To accurately define lane changing behavior, as shown in Figure 6.5 the starting time of the lane changing behavior can be determined by detecting the last significant change of steering wheel angle at time t_{angle}. The time interval in between t_{angle} and t_{lane} is defined as lane changing behavior.

Table 6.3: Average $t_{indicator}$ and t_{angle} (E2)

Driver	Average $t_{indicator}$ [s]	Average t_{angle} [s]	Difference [s]
1	2.77	2.57	0.19
2	2.35	2.00	0.35
3	2.87	1.91	0.96
4	2.53	2.40	0.13
5	2.70	2.00	0.70
6	2.36	1.99	0.37
7	2.63	2.00	0.63

The time of lane changing behavior of driver #1 defined by t_{angle} is recorded in Figure 6.6. It can be seen that the time when drivers change the lane is always between 1 s and 2.5 s before the lane change is completed. As mentioned in Figure 6.4 (E1), most of the lane changing intentions occurred within 3.5 s, i.e. 1 s earlier than 2.5 s (max. lane changing time in Figure 6.6). To further determine the driving intention, the average t_{angle} and $t_{indicator}$ of $\#1-\#7$ drivers are listed in Table 6.3. It can be found that the difference between the t_{angle} and $t_{indicator}$ is no more than 1 s, which means that the driving intentions occur within 1 s before the driving behavior. Therefore, the lane changing intentions in E2 are considered occurring 1 s before the behaviors, i.e. the time interval in between $t_{angle}-1$ and t_{lane} (Figure 6.5). To evaluate the prediction performance of the proposed approaches, the class labels of training data and test data are defined using driving intention and driving behavior (real action) respectively.

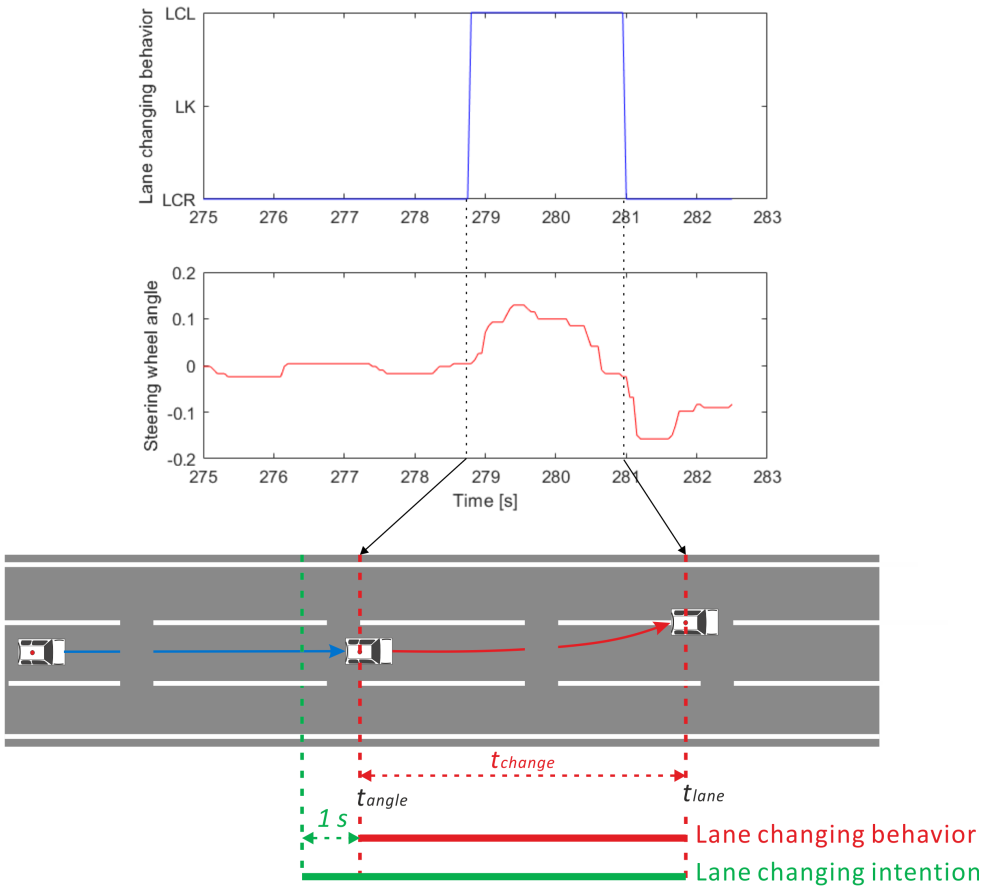


Figure 6.5: Illustration of lane changing behavior and intention (E2)

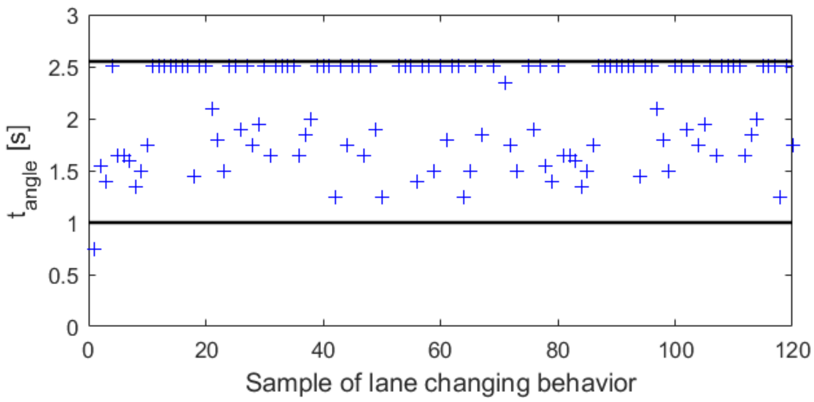


Figure 6.6: Starting time of the lane changing behavior of driver #1 defined by t_{angle} (E2)

6.2 Training and test procedures

6.2.1 Training phase

The main goal of the training phase is to establish driving behavior prediction models based on related approaches.

E1 - Training

The purpose of the first experiment is to discuss and analyze the impact of input parameters (ENV and ET data) and effectiveness of different known machine learning approaches. In the training phase, not only the proposed models are established, but also the correctness and the performance of the models are evaluated. Therefore the training phase contains model training and model validation. The training and test

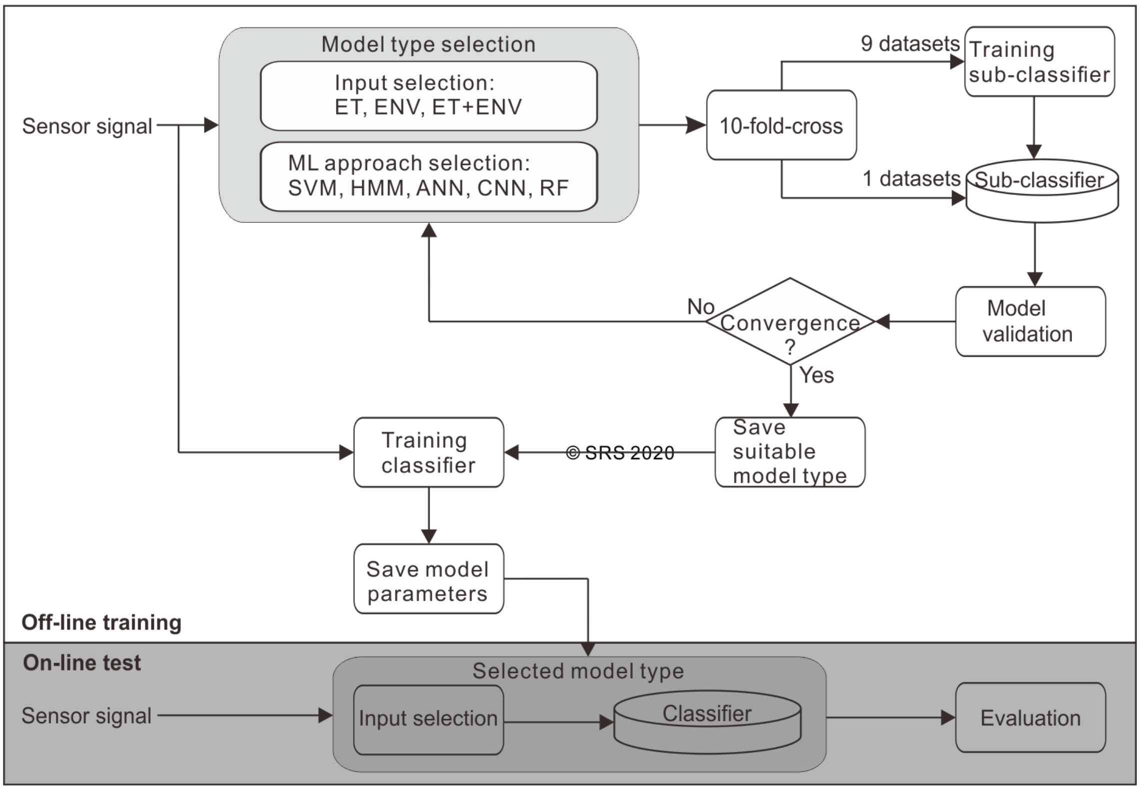


Figure 6.7: Illustration of training and test phase (E1)

phase of E1 are illustrated in Figure 6.7. To verify selected model type, data used for training and for validation should be different and both datasets must contain different lane changing behaviors. Therefore, the 10-fold-Cross-Validation [BD15] technique is applied. This method divides a dataset into 10 sub-datasets. For each time one sub-dataset will be selected for validation, and other sub-datasets for sub-model training. This process will be repeated 10 times until all the sub-dataset have been selected for validation.

E2 - Training

The purpose of the second experiment is to establish the proposed models and then to predict the driving behaviors in real time. In the training phase, for each driver a training dataset is given as input and the output is a trained model for individualized driver. As described in section 4.2, for each selected algorithm, four different models (M1-M4) will be established, which are trained with default / optimized prefilters and hyperparameters respectively. As shown in Figure 6.8, the proposed improved models are trained through the following steps.

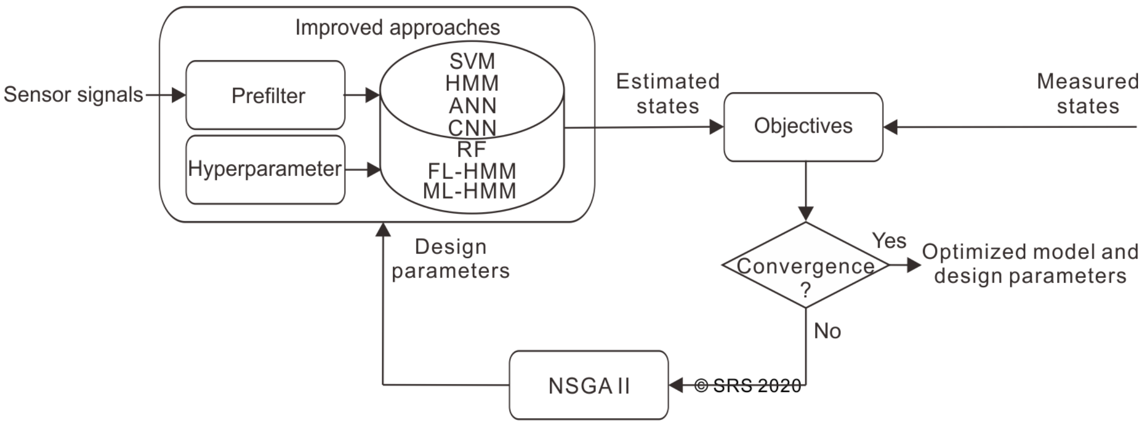


Figure 6.8: Illustration of optimized model and design parameter definition

(a) According to the principle of NSGA-II methodology, first a set of design parameters is generated randomly by NSGA-II.

(b) Using the defined design parameters a training data set is processed and combined to form observation sequence. Then the observation sequence and its actual labels can be used to estimate each HMM parameter, with these HMMs the hidden state could be calculated.

(c) The actual hidden state sequence and the hidden state sequence calculated by the proposed models will be compared to check the values of the corresponding ACC, DR, and FAR. Afterwards, the objective functions 4.4 and 4.5 in section 4.1.2 could be calculated.

(d) Process is repeated from (a) to (d) until convergence.

(e) Through the comparison of the objective functions results for each model, the optimal design parameters are found.

6.2.2 Test phase

The proposed model (based on driver-specific parameters) is applied for driver behaviors prediction. The predicted behaviors and the real behaviors can be compared for evaluation.

E1 - Test

The test phase of E1 is shown in Figure 6.7. After training data collection and model training, the drivers were required to drive again for each on-line model separately, each time is about 10 minutes and and was realized in a driving scenario different from the training scenario. The estimated behaviors will be calculated and saved in real time. Through the comparison between the estimated driving behaviors and the actual driving behaviors, the veracity of the prediction could be evaluated.

E2 - Test

The models based on known approaches or the improved models with the corresponding optimized design paramters for each test data set are already calculated in the training phase (Figure 6.9). Because for each driver there are 7*4=28 (7 approaches and 4 model types) different models, on-line test is not applied for each driver like E1. Instead, the test data set from each driver are saved for models' test and evaluation. Based on these established models, the driving behaviors in the upcoming driving processes could be determined. The measured and estimated driving behaviors are compared to check the correspondence.

6.3 Evaluation results

To verify the effectiveness of the models the actual and estimated driving behaviors are compared, and the values of ACC, DR, and FAR are calculated for the complete driving sequence. The calculation equations are expressed as equation 4.1 - 4.3 in section 4.1.2 .

6.3.1 Known machine learning approaches (E1)

As mentioned previously, the first experiment is designed to find the answers of the following questions:

- What are the roles of using ENV data and ET data?
- Which algorithm performs best and should be selected?

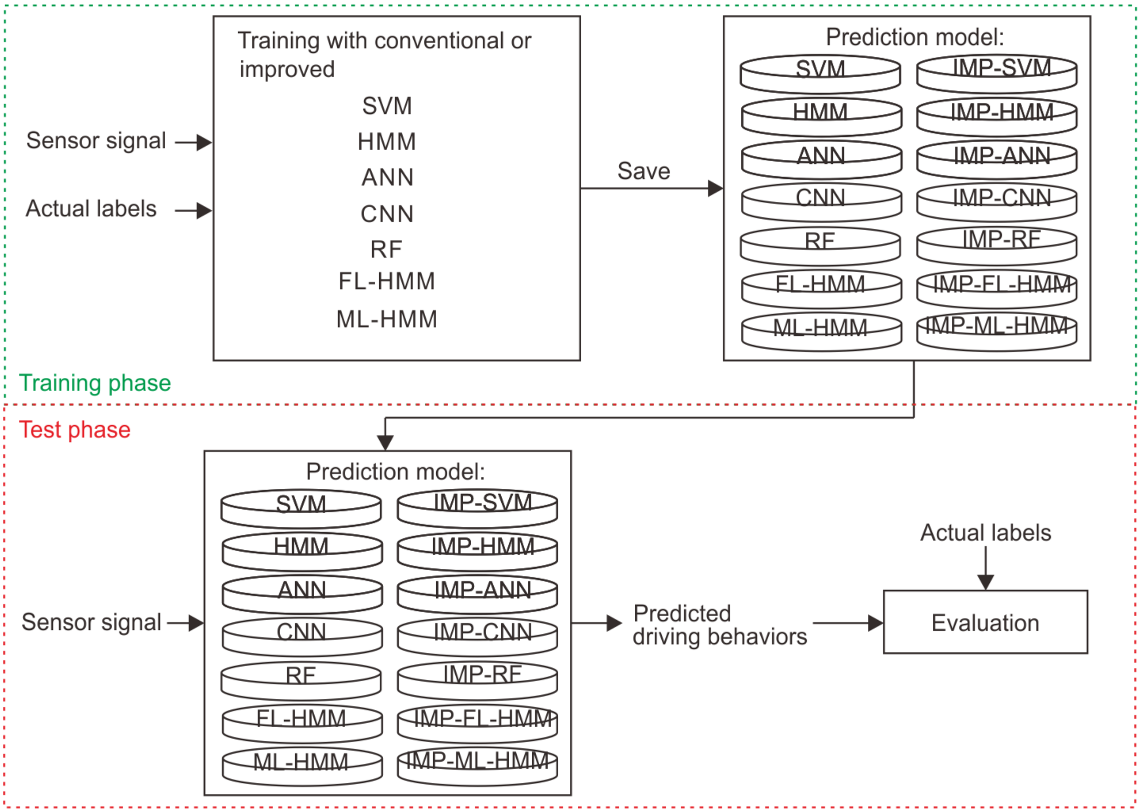


Figure 6.9: Illustration of training and test phase (E2)

Off-line validation

The prediction models are established by using SVM, HMM, ANN, CNN, and RF. For each algorithm three models will be built up based on the ET data, the ENV data, as well as the ET+ENV data. Therefore, fifteen models are established in the experiment, details are shown in Table 6.4.

The aim of off-line training is to select suitable model type according to input and approach selection. To verify the models, 10-cross-validation is used in this experiment. After training, the fifteen models are established for driving behavior prediction. Based on the established models, the driving behaviors in the upcoming driving processes could be determined. The measured driving behaviors and the estimated driving behaviors which are calculated by the model will be compared to check the correspondence. Then, the ACC, DR, and FAR of each model for each driver are calculated. Finally, the average ACC, DR, and FAR from all the ten drivers for each model are calculated. Finally, the average ACC, DR, and FAR from all the ten drivers for each model are calculated. The results of the ACC from different models are shown in Figure 6.10. The following conclusions can be made according to the results.

Table 6.4: Driving behavior prediction models (E1)

Model number	Algorithm	Training data
1	SVM	ET data
2		ENV data
3		ET+ENV data
4	HMM	ET data
5		ENV data
6		ET+ENV data
7	ANN	ET data
8		ENV data
9		ET+ENV data
10	CNN	ET data
11		ENV data
12		ET+ENV data
13	RF	ET data
14		ENV data
15		ET+ENV data

(a) Using only ET data all models show the worse results (the ACC values of LK using different algorithms are 56.60 % (SVM), 52.01 % (HMM), 79.29 % (CNN), and 90.75 % (RF)). Compared to the result using ET data, the ACC values of HMM using the ENV data are increased from 52.01 % (LK), 67.63 % (LCL), and 75.20 % (LCR) to 90.76 % (LK), 94.76 % (LCL), and 95.47 % (LCR) respectively. Similarly, using the ENV data the ACC of SVM, ANN, CNN, and RF algorithms are higher than using the ET data. The reason is that the driving environment information is predominating drivers decision making, and it is not able to be collected through eye-tracker.

(b) Using ET+ENV the results of HMM, ANN, CNN, and RF can be marginally improved in comparison to the results using ENV alone. The ET data can help the machine to learn the sight focus (gaze) of the driver. However, the result of SVM algorithm shows that, the ET+ENV data is worse than ENV data. For example, using SVM the ACC of ENV data is better than using the ET+ENV data (about 20 % higher). Therefore, introducing ET data will not certainly improve the prediction performance. As known from classifier and fusion research [RS16], classifiers combined with a lower accuracy classifier may provide worse results than the mean results of individual classifier. This is the possible reason why ET data does not lead to an improvement of the

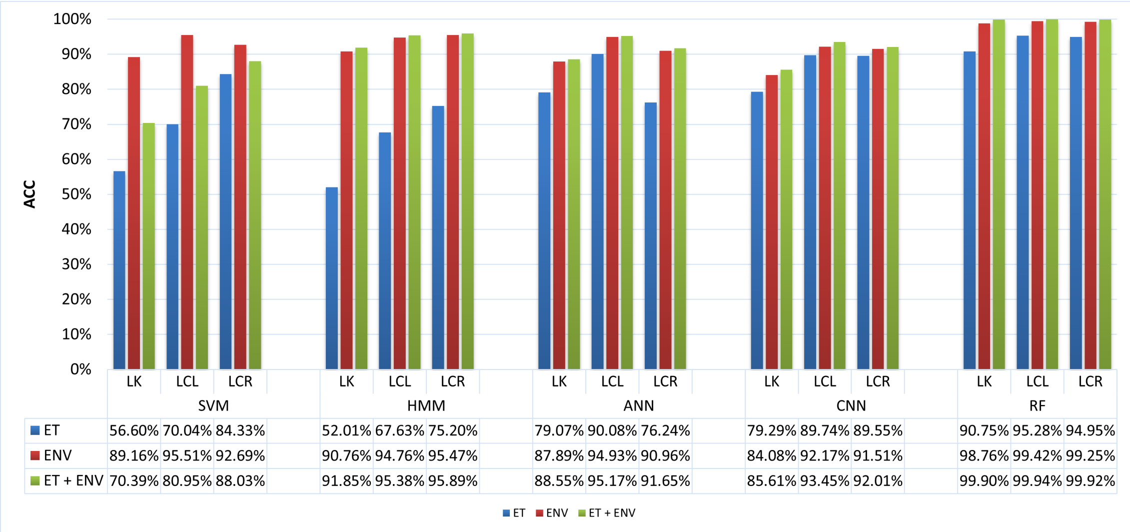


Figure 6.10: Average ACC of 10-cross-validation achieved by known ML models (E1)

prediction results for SVM. Of course this theoretically may also be caused by non-suitable choice of classifiers.

(c) The models based on RF have the highest ACC values compared to the models that use the same data type but different algorithms. It can be concluded from the results that RF algorithm in this case provides better performance than SVM, HMM, and CNN algorithms. Among the three models using RF algorithm, the ET+ENV data achieved the best result in the experiment, in which all ACC values are larger than 99 %.

To further evaluate the performance of driving behaviors prediction, the Receiver Operating Characteristic (ROC) results are shown in Figure 6.11. To compare the average DR and FAR values for each model, the ROC graph is used instead of the ROC curve. From the obtained results it could be detected, that four of the fifteen models show good performance: HMM (with ENV), HMM (with ET+ENV), RF (with ENV), and RF (with ET+ENV). Their DR values are higher than 85 % and FAR values are lower than 10 %. The other eleven models have unsatisfactory ACC or FAR values, which are the models SVM, HMM, ANN, and RF (only using ET), SVM (with ENV, ET+ENV), as well as CNN (with ET, ENV, ET+ENV). The reason for worst performance of CNN is due to the imbalance within training labels. Based on normal driving conditions, the number of LK behaviors are more than the number of lane changing behaviors. Data augmentation (DA) is often used for images and videos to solve this problem. However, the prediction of driving behaviors with ENV and ET features is different from computer vision. Therefore, in the future new DA algorithms should be considered to process EVN and ET data.

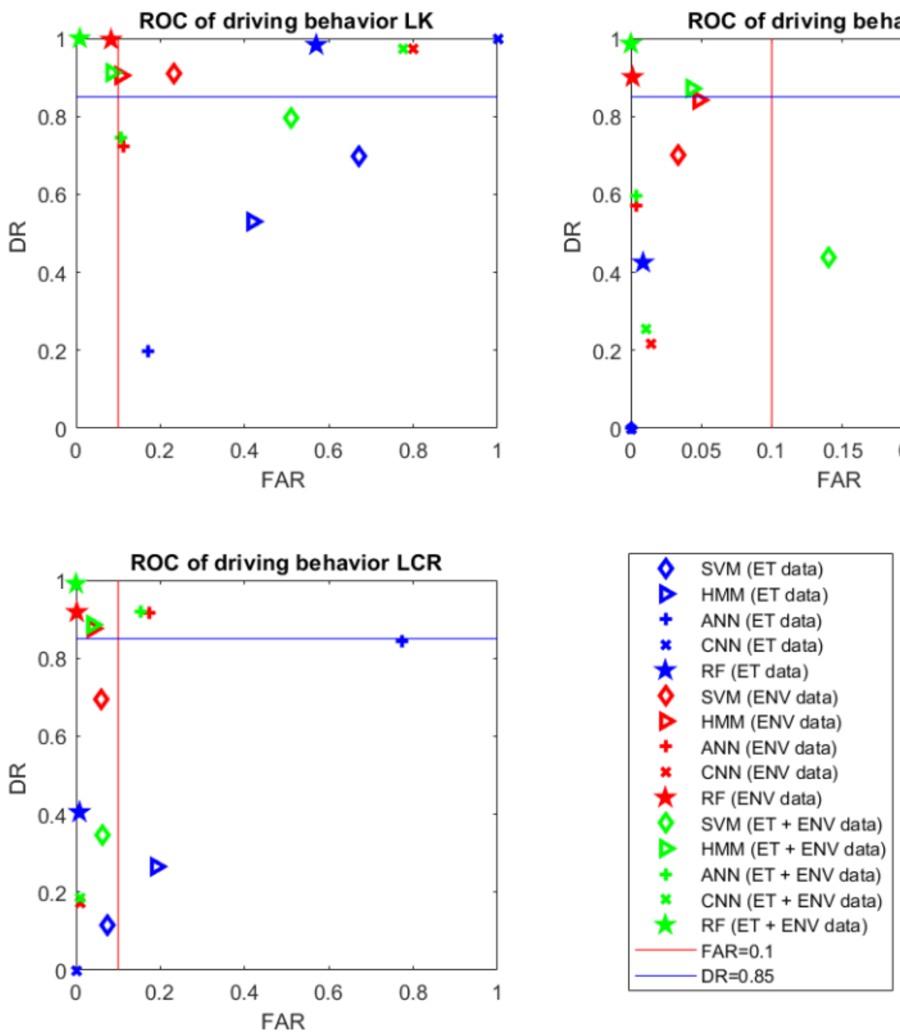


Figure 6.11: ROC graph for different models based on known ML approaches (offline)

Table 6.5: Training time of five known ML approaches [Training dataset #1] (E1)

Data type	Driving duration [s]	Number of data	Algorithm	Training time [s]
ENV	1829.9	36598	SVM	777.240
			HMM	62.010
			ANN	52.934
			CNN	177.756
			RF	11.989

The required training times of the four algorithms for the same training data are recorded (Table 6.5). The training data are obtained by a 1829.9 s drive and only ENV data are selected. It can be observed that, for HMM, ANN, and RF, the training can be completed in a few seconds about 62 s, 53 s and 12 s respectively. In contrast, the training for CNN and SVM model requires long time (more than 177 s and 777 s respectively).

From the results it can also be concluded that using the RF / HMM (with ENV) and RF / HMM (with ET+ENV) the ACC and DR values are higher, in the meanwhile the FAR values are lower than other approaches. Thus, the RF and HMM have better performance within all models. In addition, the RF and HMM algorithms require shorter training time than other approaches. However, since the principle of the RF approach is random selection of inputs and decision trees, even if the same training dataset is used, the RF model obtained every time is different. In contrast, the HMM approach can get a stable model.

In this section two RF-based behavior prediction models using the ENV and ET+ENV data, are developed. Each driver specific test dataset must be related to the data, which are used in training. The task of on-line test is to implement prediction on-line and evaluate the performance of the driving behaviors prediction. The prediction ability of the model can be demonstrated in terms of prediction time and some evaluation metrics like ACC, DR, etc. In the following sub-sections, the experimental results will be described.

Prediction time

An ideal model should be able to predict the driving behaviors before the actual lane change. As described in section "Data processing", the lane changing as the driving behavior is considered starting from $t_{indicator}$ to t_{lane}. The total required time is defined as t_{change}. When the drivers change the lane without turning signals while lane changing, the average value of all t_{change} will be used to determine the lane changing behavior. However, the following two points should be discussed:

(a) Whether the prediction time is influenced by this preset time t_{change}?

(b) How to determine the value of t_{change}?

As mentioned in Figure 6.4 the collected value of t_{change} in this experiment is always between 2 s and 3 s. Therefore, 2 s, 2.5 s, and 3 s are selected as the preset t_{change} to analyze the impact on the behavior prediction. From the training results, the performance of model based on RF using ENV data and using the ET+ENV data has been successfully proven. For the two models, the mentioned three preset t_{change} are used respectively. Thus, there are six different models. The details and the results of the six models are given in Table 6.6. The total number of lane changing maneuver is 28. Among the six models, the correct number of maneuver predictions for Model #3 is highest in all models. However, three wrong predictions can still be found.

Table 6.6: On-line prediction results of RF [Test dataset #1] (E1) [DSW+19]

Model number	#1	#2	#3	#4	#5	#6
Preset t_{change} [s]	2	2.5	3	2	2.5	3
Training data	ENV			ET+ENV		
Number of Lane changing maneuver	28			28		
Number of correctly prediction	20	24	27	20	24	25
Number of incorrectly prediction	1	3	3	0	3	3

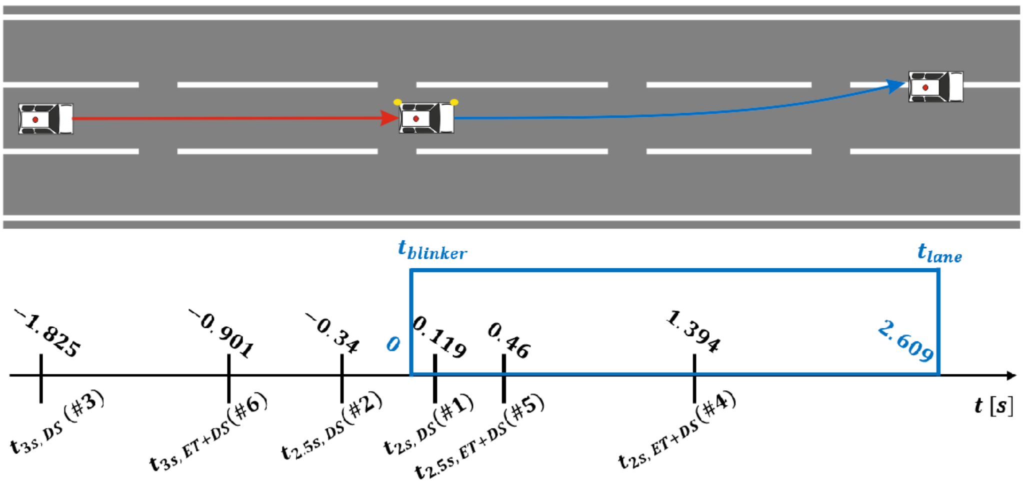


Figure 6.12: Average on-line prediction times of different models [Test Dataset #1] (E1) [DSW+19]

In addition, the average time of predicted lane changing for different models are shown in Figure 6.12, Here, the value of the starting time of the lane changing $t_{indicator}$ is defined as 0. As mentioned, the average value of lane changing time is $t_{change} = 2.609$ s for driver #1. Therefore, the blue box indicates the duration of lane changing behavior. It is clear that, when the value of t is less than 0, the estimated behavior of the model occurs before the drivers turn on the signal. In contrast, when the value of t is larger than 0, the estimated behavior occurs after the indicator signal. It is also obvious from Figure 6.12, that the predicted lane changing behaviors of all models can be realized before the drivers lane changing behavior occurs (t_{lane}). Three of them are prior than the time when the drivers turn signal $t_{indicator}$, and the earliest lane changing is predicted by model #3 (about 1.825 s before $t_{indicator}$). The same comparison is made in the other test datasets. The results are confirmed that with model #3 the best results can be obtained. Model #3 is based on the RF algorithm using ENV data with preset time ($t_{change} = 3\ s$). Due to the delay in data transition from the Eye-Tracker sensor to the program in the on-line test, the results of the model using ET+ENV data are not as good as in the off-line training. Therefore, the model RF (with ENV data) is considered for driving behavior prediction of a real on-line test, here 3 s is chosen as preset time.

On-line evaluation

The RF (ENV data) models for each test dataset are already calculated and saved in the previous section. The driving behaviors will be determined by using corresponding models. Through the comparison between calculated and actual driving behaviors, accuracy can be evaluated.

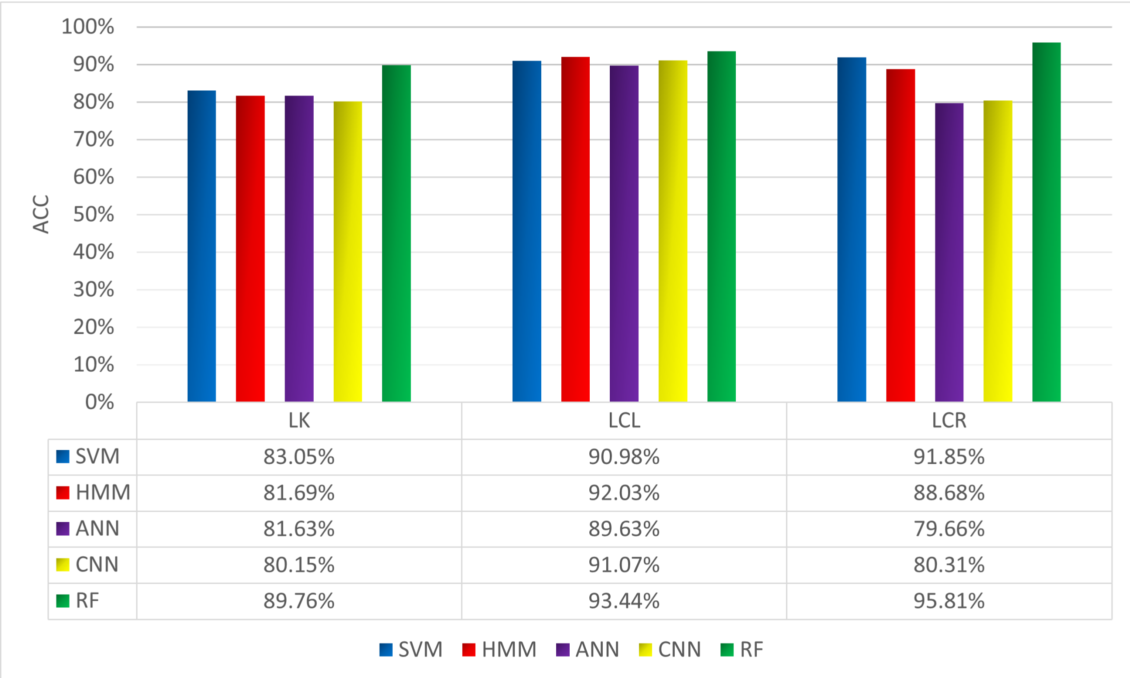


Figure 6.13: Average ACC achieved by different models for 10 on-line test datasets

To verify the effectiveness of the model in terms of driving behaviors prediction, other related known approaches are used for comparison. The percentage of ACC for each group is shown in Figure 6.13. Based on the average value of DR and FAR calculated by different models for ten test datasets, the ROC graph could be drawn, which is shown in Figure 6.14. Similarly, RF and HMM show the more stable capability for driving behavior prediction. As conclusion based on the analysis from Figure 6.13 and Figure 6.14, the ACC value and the ROC result of RF and HMM are

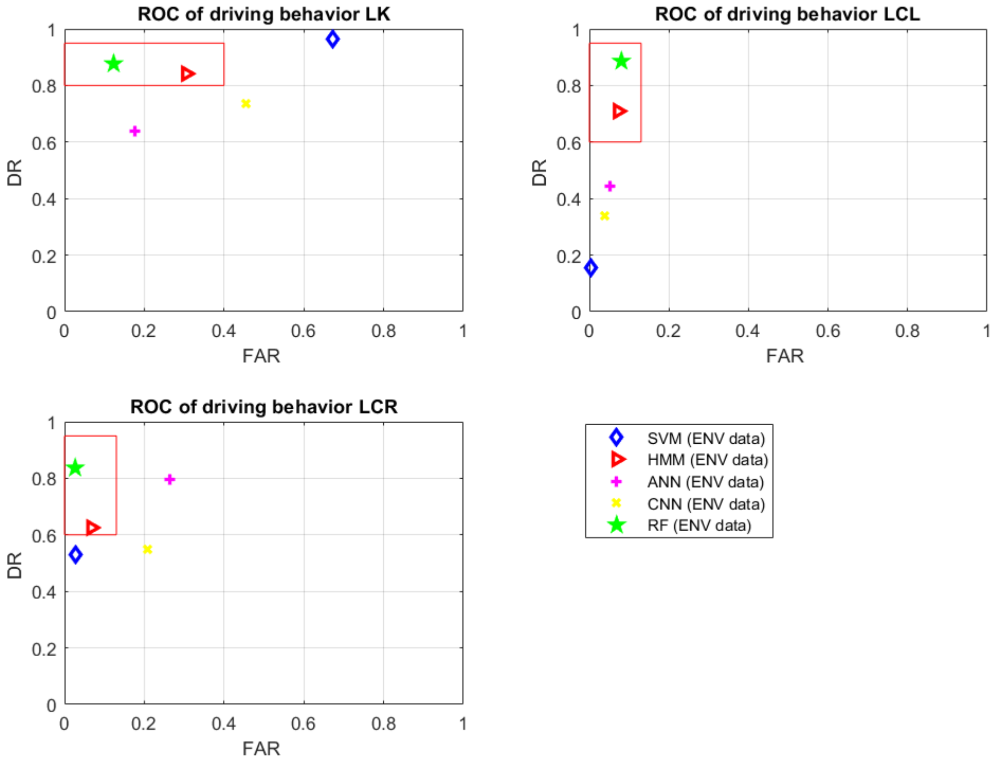


Figure 6.14: ROC graph for different models based on known ML approaches (online)

significantly better than the other three algorithms. In addition, the model based on SVM has the worst performance among these five algorithms.

The conclusion can answer the questions raised at the beginning of this section. The prediction performance using ET+ENV data depends on the applied algorithm. Better results can be observed using HMM, ANN, CNN, and RF. For SVM, better performance can be obtained when only ENV data is used. Therefore, the integration of ET data will not necessarily improve the prediction performance. In the following experiment, only ENV data will be selected. Among all prediction results for all algorithms, the RF and HMM algorithms using ENV data present better performance. Considering that using HMM can get a stable model, the new approaches will be developed based on HMM in this thesis.

6.3.2 Known machine learning approaches with improved training loop (E2)

The first step of the E2 experiment is designed to answer the question:

- Can the proposed training loop be used to improve the performance of the known machine learning approaches?

Therefore, the known machine learning approaches with the default and optimized design parameters will be compared in this section, i.e. known machine learning approaches based on M1 (baseline: default prefilter and hyperparameter) and M4 (optimized prefilter and hyperparameter) are compared (cf. section 4.1).

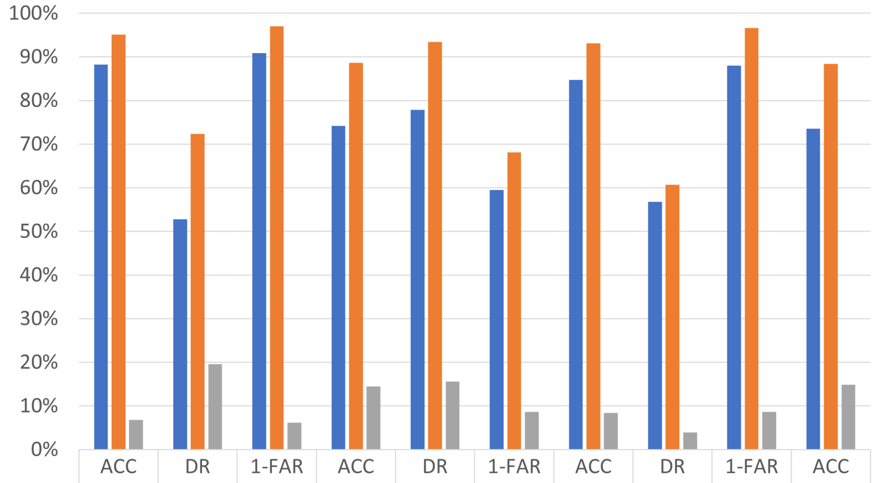
As described in chapter 4, five conventional and extensively used known approaches (SVM, HMM, ANN, CNN, and RF) are selected to verify the performance of the proposed training strategy. For model group M1, the related sequence process method of conventional HMM is referred to the previous publication [DWS18]. Other approaches using M1 model are trained with raw data. The hyperparameters that need to be set are selected as default values summarized in section 5.2. For improved model M4, using the proposed training procedure the most suitable design parameters can be determined automatically to optimize the performance of the known approaches. All mentioned models use the same observation variables (total 26 inputs) mentioned in section 6.1.3.

The average values of evaluation metrics (ACC, DR, 1-FAR) of each group are shown in Figure 6.15 to Figure 6.19. The following conclusions can be made according to the results.

(a) The difference value greater than 0 means that using the M4 model can get better results. It can be found that the results for all algorithms using M4 model are better than using M1 model. The overall ACC values using M1 model are 73.52 % (SVM), 74.97 % (HMM), 83.26 % (ANN), 83.54 % (ANN), and 88.23 % (RF) respectively. The overall ACC results of SVM-M4, HMM-M4, ANN-M4, CNN-M4, and RF-M4 are increased to 88.42 %, 85.46 %, 84.09 %, 89.12 %, and 88.44 %. .

(b) The improvement (positive value of difference) greater than 5 % denote as significant improvement. The results of SVM-M4, HMM-M4, ANN-M4, and CNN-M4 are significantly improved in comparison to the results using M1 model. For example, the DR value of LCL using CNN-M4 is increased from 32.41 % to 67.98 % (about 35 % higher). The maximum increase of HMM-M4 and ANN-M4 are also the DR value of LCL, which are improved by 13 % and 10 % respectively. The DR of LCR is the most improved using SVM-M4, which achieves 38 % increment.

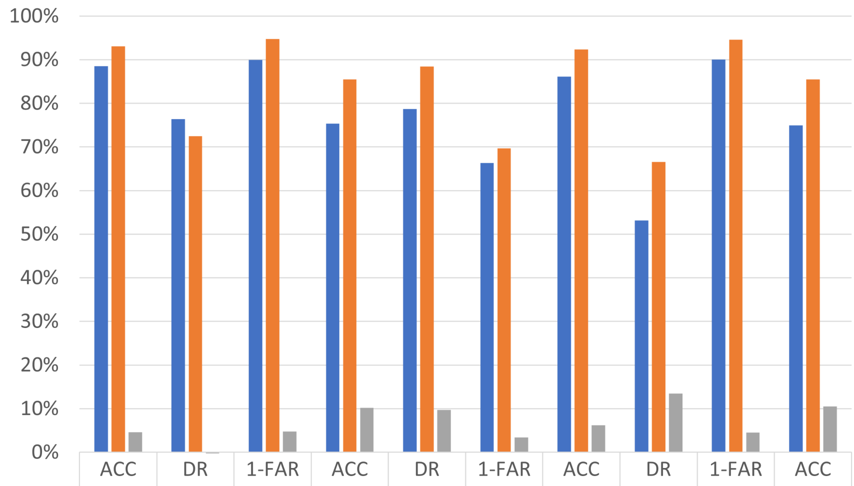
(c) for the algorithm ANN, CNN, and RF, some exceptions can still be found. For example the 1-FAR value of LCR (using ANN-M4) is worse than the DR value using ANN-M1 (about 0.42% lower). However, these values are marginally



	LCR			LK			LCL			Overall
	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC
SVM_M1	88.27%	52.81%	90.87%	74.16%	77.87%	59.51%	84.70%	56.76%	88.00%	73.56%
SVM_M4	95.10%	72.38%	97.00%	88.65%	93.42%	68.10%	93.09%	60.66%	96.64%	88.42%
Difference: SVM_M4 - SVM_M1	6.83%	19.57%	6.13%	14.48%	15.55%	8.59%	8.40%	3.90%	8.64%	14.85%

SVM_M1 SVM_M4 Difference: SVM_M4 - SVM_M1

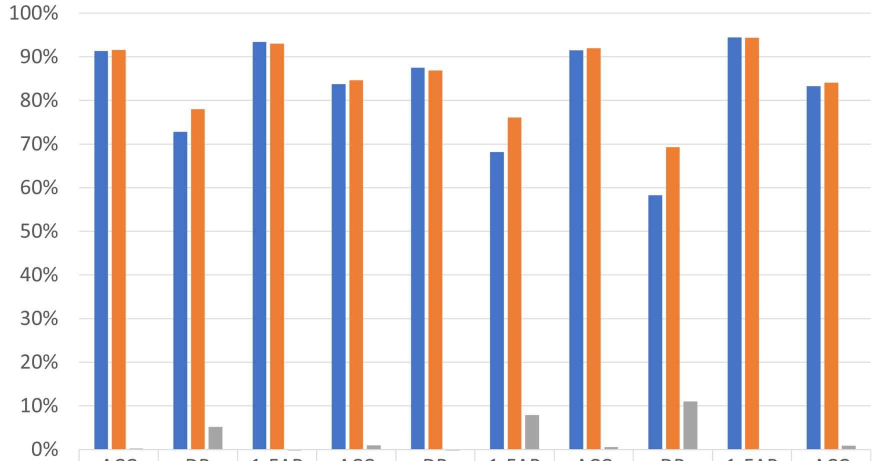
Figure 6.15: Evaluation metrics of conventional SVM (M1) and improved SVM (M4)



	LCR			LK			LCL			Overall
	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC
HMM_M1	88.51%	76.41%	89.94%	75.35%	78.72%	66.31%	86.08%	53.15%	90.01%	74.97%
HMM_M4	93.10%	72.49%	94.72%	85.51%	88.40%	69.68%	92.31%	66.57%	94.56%	85.46%
Difference: HMM_M4 - HMM_M1	4.59%	-3.92%	4.79%	10.16%	9.68%	3.36%	6.23%	13.42%	4.55%	10.49%

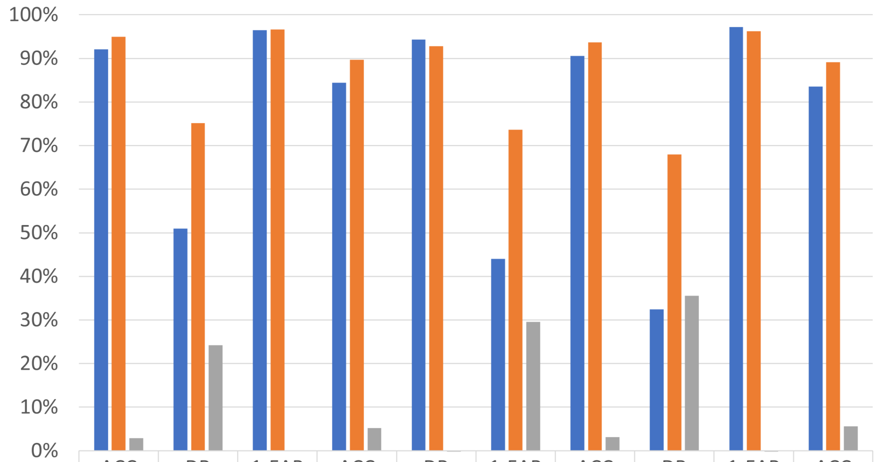
HMM_M1 HMM_M4 Difference: HMM_M4 - HMM_M1

Figure 6.16: Evaluation metrics of conventional HMM (M1) and improved HMM (M4)



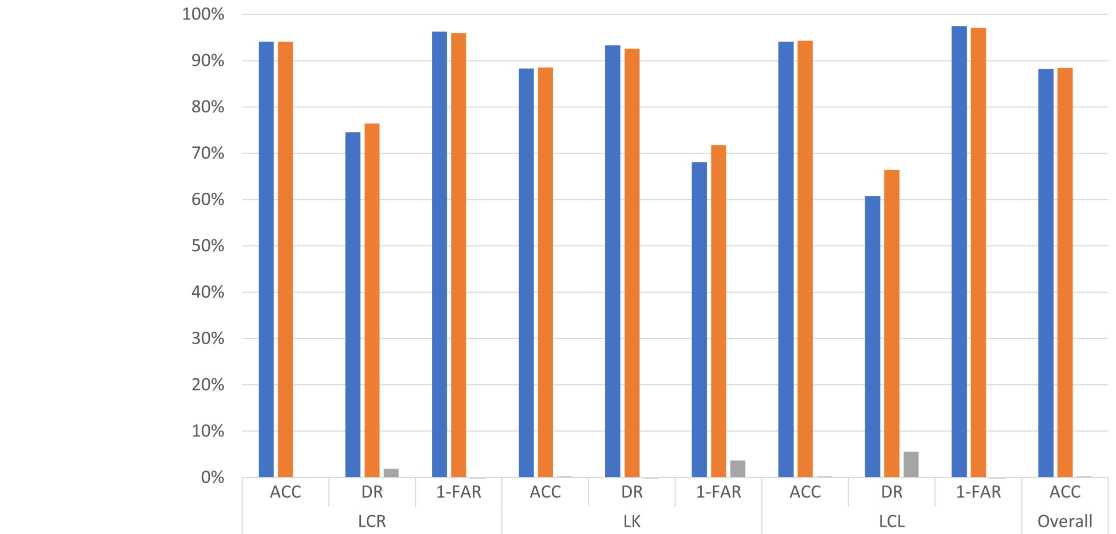
	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC
		LCR			LK			LCL		Overall
ANN_M1	91.36%	72.78%	93.39%	83.70%	87.48%	68.15%	91.46%	58.30%	94.43%	83.26%
ANN_M4	91.58%	77.97%	92.97%	84.62%	86.88%	76.07%	91.97%	69.29%	94.39%	84.09%
Difference: ANN_M4 - ANN_M1	0.22%	5.18%	-0.42%	0.92%	-0.60%	7.92%	0.51%	10.98%	-0.04%	0.83%

Figure 6.17: Evaluation metrics of conventional ANN (M1) and improved ANN (M4)



	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC	DR	1-FAR	ACC
		LCR			LK			LCL		Overall
CNN_M1	92.06%	50.96%	96.47%	84.44%	94.28%	44.01%	90.59%	32.41%	97.15%	83.54%
CNN_M4	94.91%	75.14%	96.60%	89.65%	92.82%	73.61%	93.69%	67.98%	96.24%	89.12%
Difference: CNN_M4 - CNN_M1	2.86%	24.17%	0.12%	5.21%	-1.46%	29.59%	3.09%	35.57%	-0.91%	5.58%

Figure 6.18: Evaluation metrics of conventional CNN (M1) and improved CNN (M4)



	LCR ACC	LCR DR	LCR 1-FAR	LK ACC	LK DR	LK 1-FAR	LCL ACC	LCL DR	LCL 1-FAR	Overall ACC
RF_M1	94.07%	74.56%	96.27%	88.30%	93.35%	68.08%	94.08%	60.82%	97.42%	88.23%
RF_M4	94.04%	76.45%	95.97%	88.52%	92.59%	71.75%	94.32%	66.39%	97.10%	88.44%
Difference: RF_M4 - RF_M1	-0.03%	1.89%	-0.30%	0.22%	-0.76%	3.67%	0.24%	5.57%	-0.33%	0.21%

Figure 6.19: Evaluation metrics of conventional RF (M1) and improved RF (M4)

reduced and the overall result considering all situations are still improved using M4 model.

(d) For RF-M4 the results are marginally improved in comparison to the results using RF-M1. The models based on RF-M1 have the highest ACC values among all the M1 models.

It can be concluded that the prediction performance of the algorithms considered is (partly marginally) improved by the introduced approach.

To further evaluate the performance of the improved models, two methods proposed in [DYF16] are used as reference. The authors design two combined ANN and SVM methods named ANN-SVM (Conservative / Aggressive), and the results show that the stability and accuracy of the combined methods are better than default ANN and SVM.

The evaluation metrics are shown in Figure 6.20. It can be found that, the improved models perform better considering all situations in comparison with ANN-SVM. Except for three cases including DR of LCR, DR of LCL, and 1-FAR of LK, the performance of the improved models at other (seven of ten) evaluation metrics are greater than 80 %. In addition, the conventional algorithm SVM, HMM, and CNN present weaker performance than conventional ANN and RF, and have been improved significantly (over 10 %) through the proposed training procedure. The proposed full scale training loop is able to strongly improve the algorithms with poor prediction performance.

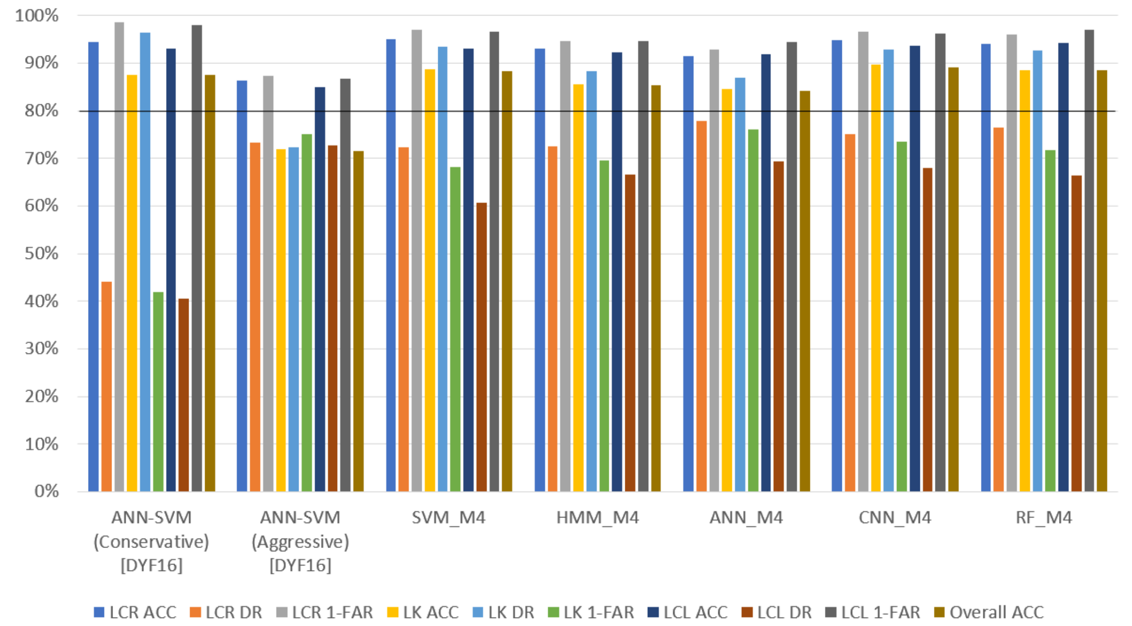


Figure 6.20: Evaluation metrics of different models for 17 test datasets

6.3.3 FL-HMM and ML-HMM approach (E2)

The second step of the second experiment (E2) is to answer the question

- Can the newly developed HMM-based approaches (FL-HMM and ML-HMM) successfully predict driver behaviors?

Therefore, the two newly developed approaches trained by the full scale training loop will be verified in this experiment. For comparison alternative advanced classification algorithms (SVM, NN, CNN, HMM, and RF) are applied. All mentioned models use the same observation variables (total 26 inputs). Based on these models, the driving behaviors in the upcoming driving processes could be determined. The measured and estimated driving behaviors are compared to check the correspondence.

To verify the effectiveness of the models in terms of driving behaviors prediction, the actual driving behaviors are compared to the estimated driving behaviors for all data sets. The percentage of the ACC, DR, and FAR for each group is calculated. Finally, the average evaluation metrics by using different models are shown in a boxplot Figure 6.21. Each box displays a distribution of a metric from 17 drivers.

The line and insert the symbol x in the middle of each box represent the median and average value respectively. From the obtained results it is clear that, FL-HMM and ML-HMM are relatively better than other methods to identify the driving behaviors. In most cases, the FL-HMM and ML-HMM approaches have a high ACC (larger than 90 %), a high DR (larger than 80 %), and a low FAR (less than 12 %). In addition, the median line and average value of ML-HMM are marginally better than FL-HMM.

The aim of this contribution is to predict driving behavior, so the driving behaviors should be predicted before the actual actions. The numbers of behaviors correctly predicted by using different algorithms are summarized in Table 6.7. In comparison to other machine learning approaches, FL-HMM and ML-HMM can correctly predict more upcoming behaviors, the numbers are 206/203 and 200/201 for LCL/LCR out of 218/213 real maneuvers respectively.

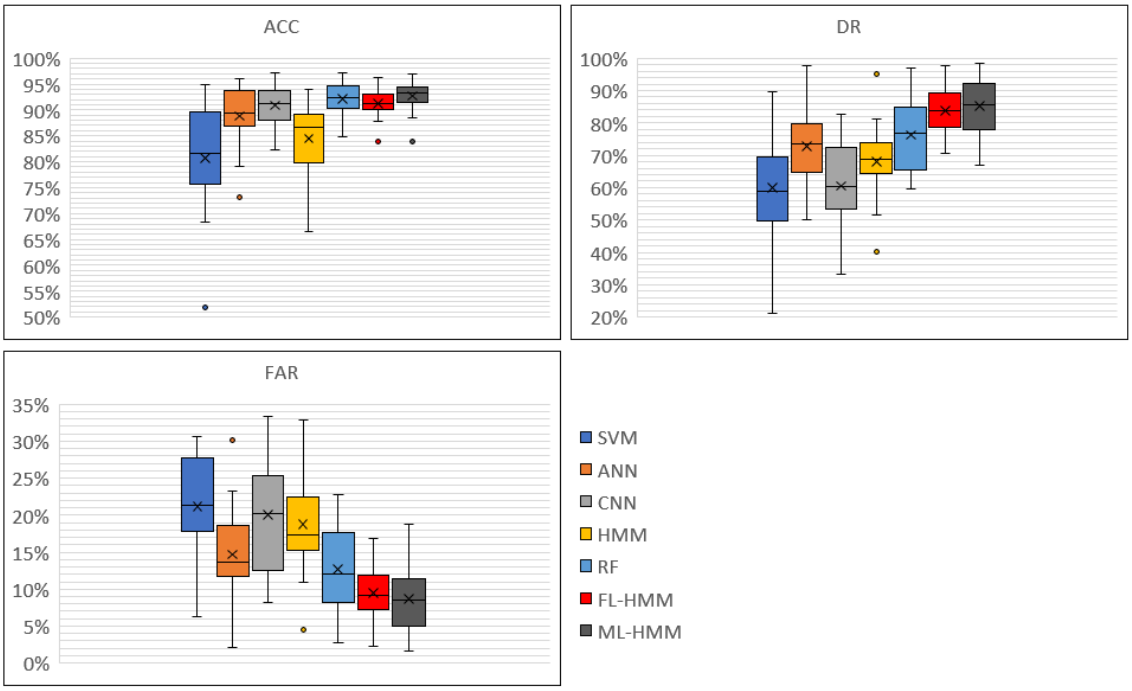
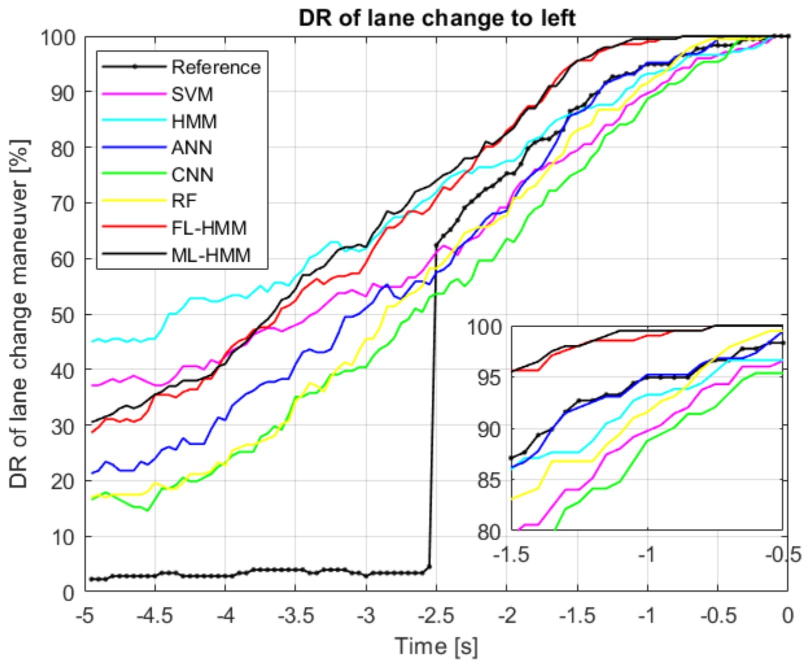
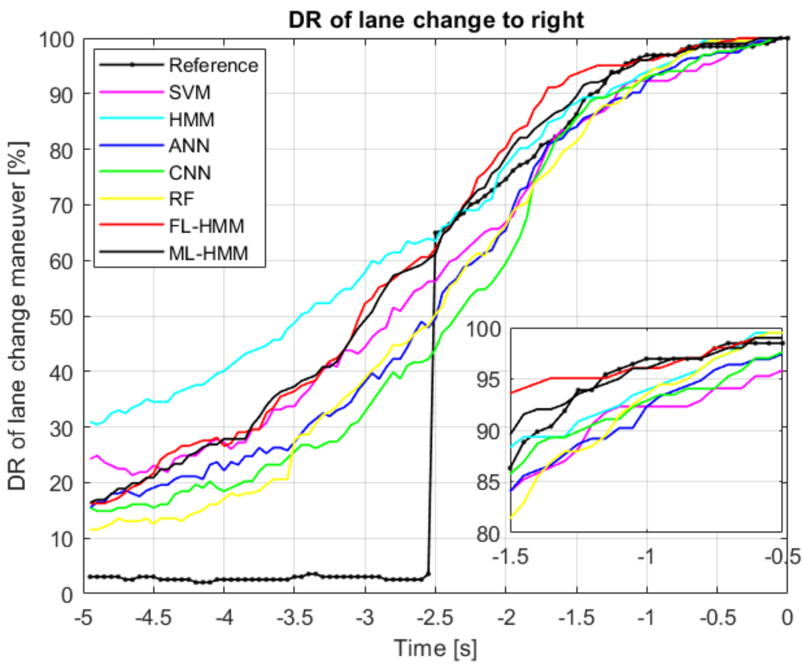


Figure 6.21: Boxplot of ACC, DR, and FAR for 17 test datasets

To further evaluate the performance of driving behaviors prediction, a method [ADS19] is used in this contribution. Here, each lane change maneuver is defined as a separate event, and DR is used to evaluate the classifiers. As shown in Figure 6.22, the x-axis refers to lane change time, and the black dotted curve indicates the actual time of all lane change maneuvers. It can be seen that more than 60 % maneuvers start to change lanes from 2.5 s, and 100 % maneuvers change lanes at 0 s. Therefore, the value of 0 s represents the time for all the actual lane change time. From 5 s before actual lane change, up to the time of actual lane change a



(a) DR of LCL maneuver



(b) DR of LCR maneuver

Figure 6.22: Prediction results of different models for 17 test datasets

Table 6.7: Number of correct predicted maneuver of different algorithms [Test datasets #1 - #17] (E2)

	Number of maneuver	
	LCL	LCR
Real	218	213
Approach	Number of correct predicted maneuver	
SVM	175	169
NN	188	194
CNN	151	168
HMM	178	197
RF	189	199
FL-HMM	206	203
ML-HMM	200	201

DR value will be calculated for performance evaluation. The time interval is divided into 100 time points, i.e. every 0.05 s, these time points are defined as "recognition time points", and for each time point a DR value will be calculated for performance evaluation. The earlier recognition time point reaches, the higher DR and the better the performance. The results (Figure 6.22) state that FL-HMM and ML-HMM can predict the upcoming maneuver with a high DR (larger than 95 %) about 1.6 s (LCL) and 1.2 s (LCR) before actually changing lanes. For other algorithms, the time are less than 1 s, or even less than 0.6 s.

Therefore, it can be concluded that FL-HMM and ML-HMM perform better than the conventional known approaches. In addition, the average training times of the seven approaches for the same training data are recorded in Table 6.8, the training of ML-HMM can be completed in a few seconds about 16.077 s, which is faster than FL-HMM.

6.3.4 Overall comparison (E2)

It is proved that model performance can be improved by using the proposed full scale training loop in the previous section. However, respective effects of prefilter and hyperparameters are not discussed in detail. Therefore, this issue will be further discussed and analyzed in this section. The third step of the second experiment is designed to answer the following two questions:

- What are the respective effects of hyperparameters and prefilter for improving model performance?

Table 6.8: Average training time of different algorithms [Training datasets #1 - #17] (E2)

Driver	Average data size	Algorithm	Average training time [s]
#1-#17	29012	SVM	835.687
		HMM	46.075
		ANN	40.387
		CNN	22.265
		RF	6.225
		FL-HMM	148.569
		ML-HMM	16.077

- Which approach performs best among all considered approaches?

Based on the proposed full scale training loop, seven algorithms including five conventional algorithms (SVM, HMM, ANN, CNN, RF) and two new approaches (FL-HMM, ML-HMM) are used to develop driving behaviors prediction models. As mentioned in section 4.1, for each algorithm four different models (M1-M4) will be trained to evaluate their prediction performance. Therefore, in total 7*4*17=476 models are established based on the data achieved from 17 drivers.

The actual driving behaviors are compared to the estimated driving behaviors for all models. The percentages of the ACC, DR, and FAR for each group are calculated. Finally, the average evaluation metrics by using different models are shown in boxplot figures Figure 6.23 to Figure 6.25, each box displays a distribution of a metric for 17 drivers. In addition, the purpose of this contribution is not only to discuss the effectivenesses of the mentioned algorithms, but also to discuss the influence of the prefilter and hyperparameter respectively. Therefore, different algorithms and model groups will be compared separately. To clearly observe the influences of prefilter and hyperparameter, the differences between the average metrics of different groups are listed in Table 6.9. The following conclusions can be made according to the results from Figure 6.23 to Figure 6.25 and Table 6.9.

(a) **Effect of overall optimization** (M4 vs. M1 & M2 & M3):
For all algorithms, the M4 models have the best prediction performance among all model groups. Except algorithm RF, the M4 models of other algorithms are significantly improved. For example in Figure 6.23, the median ACC values using CNN are 84.8 % (M1), 83.6 % (M2), and 90.6 % (M3) respectively. The median ACC value of HMM-M4 is increased to 91.0 %. Among the four model groups, HMM-M4 has the highest DR value and the lowest FAR value. The same conclusion can be obtained for other algorithms. However, the improvement is not obvious for some evaluation metrics. For example, in Figure 6.24,

the DR median value of SVM-M4 is even lower than SVM-M2. However, when comparing the variance of DR or comparing the lowest DR value, SVM-M4 is better than SVM-M1, SVM-M2, and SVM-M3. Therefore, the general conclusion can be stated that the use of optimized prefilter and hyperparameters (i.e. M4 models) can improve the performance of all mentioned algorithms.

(b) **Effect of prefilter** (M1 vs. M2, M3 vs. M4):
Similar to M1 and M2, the difference between M3 and M4 is whether using the optimized prefilter or not. The related hyperparameters are identical. Therefore, the effect of prefilter on the algorithms can be found by comparing the two pairs of models. As shown in Figure 6.23 to Figure 6.25, all the algorithms have better results after using optimized prefilters. This can also be concluded from Table 6.9 that the performance of most of the algorithms (such as SVM, HMM, CNN, FL-HMM, ML-HMM) has been significantly improved. The bold numbers in Table IV indicate a model with worse performance after using a prefilter. Some exceptions can still be found from the results, for example, the DR value of RF-M4 (using optimized prefilter) is worse than the ACC value of RF-M2 (about 0.73 % lower). However, these values are marginally reduced and the overall result considering all situations are still improved using optimized prefilter.

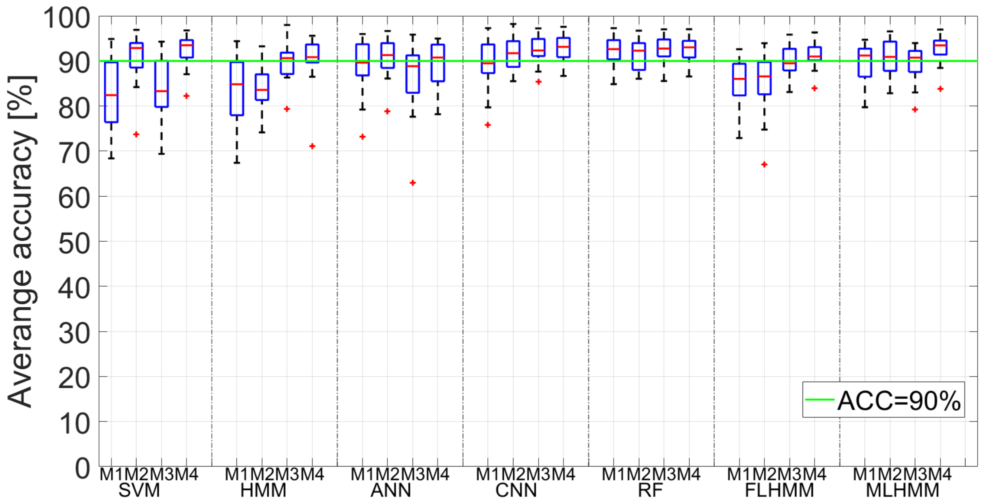


Figure 6.23: Boxplot of averange ACC achieved by 476 models

(c) **Effect of hyperparameter** (M1 vs. M3, M2 vs. M4):
The difference in the two groups is using different hyperparameters to establish driving behaviors models. Similar to the previous conclusion (b), the model performance can be improved by using optimized hyperparameters (Figure 6.24). As shown in Table 6.9, the prediction ability for some algorithms

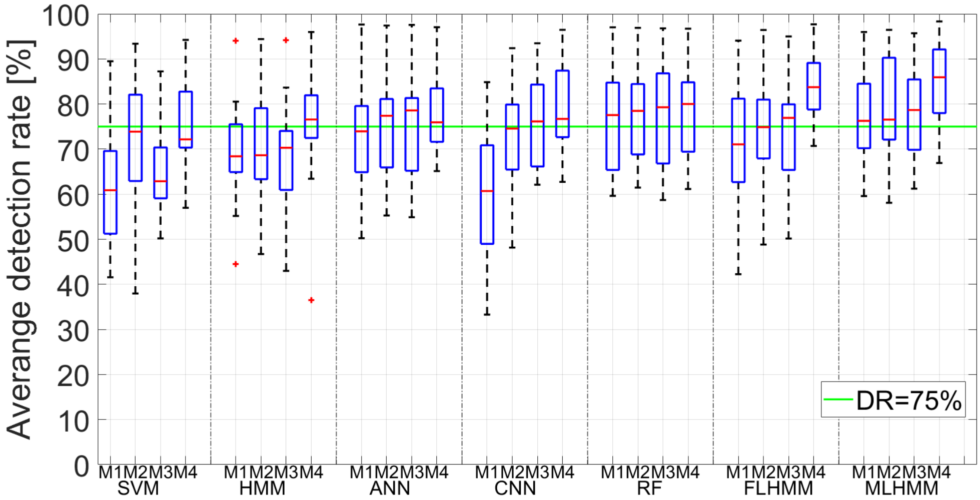


Figure 6.24: Boxplot of averange DR achieved by 476 models

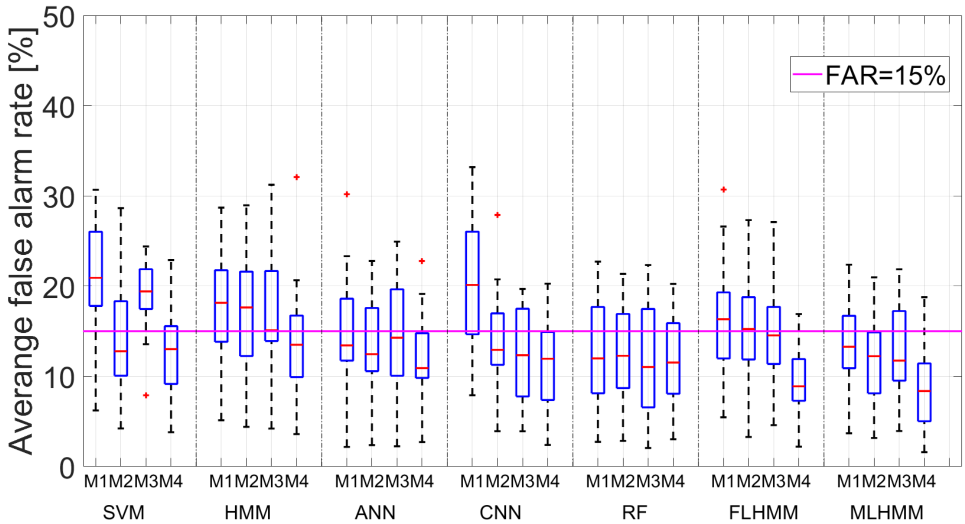


Figure 6.25: Boxplot of averange FAR achieved by 476 models

Table 6.9: Differences calculated by subtracting results from different model group [Test datasets #1 - #17] (E2)

	Difference	Metrics	SVM	HMM	ANN	CNN	RF	FL-HMM	ML-HMM
Effect of prefilter	M2-M1	ACC	8.73 %	0.83 %	1.87 %	2.73 %	**-0.73 %**	0.33 %	1.19 %
		DR	9.67 %	0.62 %	1.97 %	13.18 %	1.10 %	3.31 %	2.09 %
		FAR	-6.47 %	-0.39 %	-1.19 %	-6.61 %	-0.23 %	-1.72 %	-1.35 %
	M4-M3	ACC	7.99 %	0.55 %	2.71 %	0.24 %	**-0.07 %**	1.29 %	3.31 %
		DR	10.81 %	7.78 %	2.89 %	1.89 %	0.45 %	10.04 %	7.63 %
		FAR	-6.35 %	-3.36 %	-1.95 %	-0.98 %	-0.08 %	-4.98 %	-4.31 %
	Bold numbers: performance reduced after using optimized prefilter								
Effect of hyperparameter	M3-M1	ACC	1.91 %	6.44 %	**-2.16 %**	3.49 %	0.22 %	4.93 %	**-0.34 %**
		DR	2.19 %	**-1.39 %**	2.30 %	17.53 %	1.78 %	2.54 %	1.17 %
		FAR	-1.44 %	-0.87 %	-0.54 %	-8.62 %	-0.93 %	-2.43 %	-0.49 %
	M4-M2	ACC	1.18 %	6.16 %	**-1.32 %**	0.99 %	0.87 %	5.89 %	1.78 %
		DR	3.34 %	5.77 %	3.22 %	6.25 %	1.14 %	9.27 %	6.71 %
		FAR	-1.31 %	-3.84 %	-1.29 %	-3.00 %	-0.79 %	-5.69 %	-3.45 %
	Bold numbers: performance reduced after using optimized hyperparameters								
Hyperparameter - Prefilter	M3-M2	ACC	**-6.82 %**	5.61 %	**-4.03 %**	0.76 %	0.95 %	4.60 %	**-1.53 %**
		DR	**-7.47 %**	**-2.01 %**	0.33 %	4.35 %	0.68 %	**-0.77 %**	**-0.93 %**
		FAR	**5.03 %**	-0.48 %	**0.66 %**	-2.02 %	-0.70 %	-0.71 %	**0.86 %**
	Bold numbers: prefilters are more advantageous than hyperparameters.								

is significantly improved, such as the DR values using CNN-M3, CNN-M4, FL-HMM-M4, and ML-HMM-M4 are increased by 17.53 %, 6.25 % , 9.27 %, and 6.71 %.

(d) **Hyperparameter vs. prefilter** (M2 vs. M3):
The influences of using prefilter or using hyperparameter are not the same for different algorithms. As shown in Table 6.9, for CNN and FL-HMM, hyperparameters are more decisive than prefilters. In contrast, prefilters are more advantageous for SVM, ANN, and ML-HMM. However, for other algorithms the difference is not obvious.

(e) **Algorithms**:
As shown in Figure 6.23 to Figure 6.25, consider the case of conventional algorithms, algorithm CNN-M4 is better than other conventional algorithms. In most cases, the ACC and DR values of CNN-M4 are larger than 90.0 % and 75.0 %̃, in the meanwhile the FAR values are lower than 15.0 %. Compared with all the models and all situations, the new approaches FL-HMM and ML-HMM are the best models to identify the driving behaviors. Their median ACC, DR, and FAR values are 91.0 %, 83.7 %, 8.9 %, and 93.5 %, 85.9 %, 8.4 % respectively.

6.4 Impact of input parameters

It can be seen from the conclusion in the previous section, by selecting suitable prefilters the prediction performance of SVM are significantly improved. among the five known ML approaches, the difference of all the evaluation measurements of (SVM-M2 - SVM-M1) and (SVM-M4 - SVM-M3) are higher than 5 %. The possible reason could be the great impact of input paramters to the SVM model, appropriate

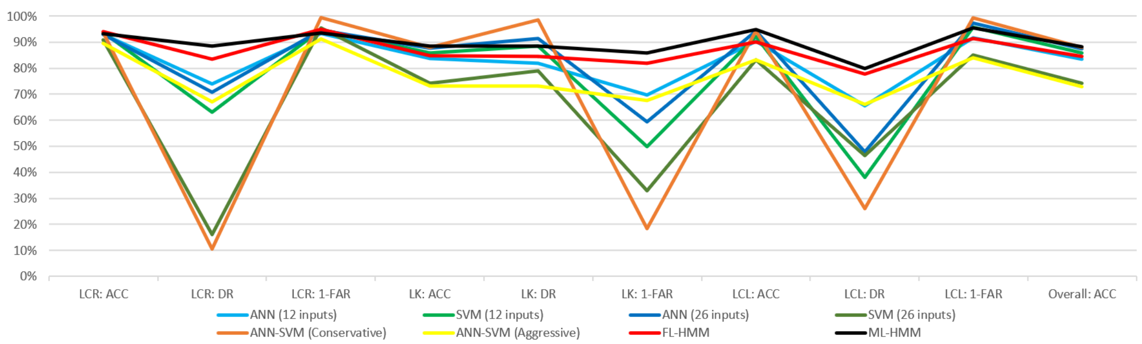


Figure 6.26: Average ACC, DR, and FAR achieved by different models for 7 test data sets [DS19b]

input features may lead to better prediction performance. This was also confirmed in the previous publication [DS19b].

As a single model, SVM and ANN with only 12 inputs are considered as reference, here related input variables [DS18] are used. The obtained resutls are shown in Figure 6.26. It can be observed that, using more inputs the results of ANN has not been improved significantly, but the performance of SVM has been significantly reduced. For example, the DR of LCR is decreased from 63.2% (SVM with 12 inputs) to 13.1% (SVM with 26 inputs). It can be concluded that in the case of a single model, using more inputs does not improve the performance. Similarly, from the obtained results it is clear that, FL-HMM and ML-HMM are good to identify the driving behaviors.

It can be concluded that, input selection plays an important role in some known ML approach like SVM and ANN. The reasons why the new FL-HMM and ML-HMM approaches developed in this thesis can get better results may be

- both FL-HMM and ML-HMM consist of different sub-models, and
- the final results are fused using the weights w.

This idea is similar to choosing a more suitable input parameter set from the all input parameters to calculate the final result.

7 Conclusion - What can be learned to realize improved machine learning approaches

The open research questions from chapter 2 are answered in the conclusion of this thesis.

A new training strategy named full scale training loop is proposed which can be used directly to improve the performance of the known machine learning approaches or an established model. In the training loop, all unknown parameters are denoted as design parameters, and the most suitable design parameters can be determined automatically. The parameters which are important to affect the model performance should be set as design parameters.

To simplify the modeling process, a prefilter is originally designed to process and combine signals to describe observations for the HMM prediction process. The finally obtained results show that, through selecting optimal prefilter parameters the ability of the HMM to predict driver behaviors is significantly improved. Therefore, the proposed prefilter is applied to the full scale training loop to define suitable input features.

Hyperparameters are parameters whose values need to be set manually prior to the training and usually using default values. To get a better model, optimizing hyperparameters is proven to be useful in improving model performance. Therefore, prefilter and hyperparameters are considered as design parameter in this thesis. The chapter 4 and chapter 5 list the design parameters of each approach in detail.

According to the open research questions described in section 2.4, two experiments are designed and described in chapter 6. The purpose of the first experiment is to explore which known approach performs better and should be selected. The purpose of the second experiment is to prove that the proposed training loop can be used to improve the performance of the known approaches, and discuss the effects of prefilter and hyperparameters respectively. The results can be concluded as the following points.

- Among all prediction results from all algorithms, the RF and HMM algorithms presents the best performance. However, the principle of the RF approach is random selection of inputs and decision trees. Even if the same training dataset is used, the RF model obtained every time is different. In contrast, a stable model can be generated by using HMM approach. For this reason, the approach HMM is selected to develop new approaches in this thesis.

- Existing researches based on HMM are roughly divided into two categories: HMM-derived and HMM-combined approaches. Based on the two modeling ideas two new approaches (FL-HMM and ML-HMM) are developed. The finally obtained results show that the prediction performance of FL-HMM and ML-HMM are significantly better than other common machine learning approaches.

- Based on the proposed training loop, seven algorithms including five known approaches (SVM, HMM, ANN, CNN, RF) and two new approaches (FLHMM, MLHMM) are used to develop driving behaviors prediction models. For each approach, four different models (M1 - M4) are trained to evaluate their prediction performance. The finally obtained results prove the optimization and selection of prefilter and hyperparameters can significantly improve the performance of the driving behavior prediction models. The group with the best performance in the contribution is M4 group which selects both optimized prefilter and hyperparameters. The influences of using prefilter or using hyperparameter are not the same for different algorithms. For CNN, hyperparameters are more decisive than prefilters. In contrast, prefilters are more advantageous for FL-HMM and ML-HMM. However, for other algorithms the difference is not obvious.

In addition, the impact of input parameters are discussed in chapter 6. It can be concluded that the integration of ET data will not necessarily improve the prediction performance. The prediction performance of using ET+ENV data depends on the selected algorithm. For HMM, ANN, CNN, and RF better results can be observed. For SVM using the ENV data alone shows the better performance. In the case of a single model, using more inputs does not improve the performance. Input selection plays an important role in some single known approach like SVM and ANN. The reasons why the new FL-HMM and ML-HMM approaches developed in this thesis can get better results may be

- both FL-HMM and ML-HMM consist of different sub-models, and
- the final results are fused using the weights w.

Finally, all points that can be learned from the thesis are summarized as below:

- The proposed full scale training loop can be applied to improve model performance.
- The proposed prefilter can effectively extract signal features and therefore to improve the performance of the approaches.
- The hyperparameters affecting model performance should be set as design parameters. Their suitable values can be determined through the training loop.

- There are two approaches (HMM-derived and HMM-combined approaches) for designing a new HMM-based model. Both of them provide improved prediction results.

- Not all the available variables are necessary to be considered as inputs in a single model. Using more inputs do not improve the performance.

- To get better prediction performance, a model can consist of different sub-models. Each sub-model is considered to predict human behaviors only for a particular working case. Results of the all sub-models can be combined into a final result.

8 Summary and outlook

8.1 Summary

Due to the importance of driving safety and efficiency, the research of human driving behaviors prediction has been focused in recent years. In this thesis a new strategy named full scale training loop is proposed for training improvement of existing classifiers. Based on the proposed approach, seven algorithms including five conventional algorithms (SVM, HMM, ANN, CNN, RF) and two new approaches (FLHMM, MLHMM) are used to develop driving behaviors prediction models. The focus is to demonstrate the ability of the proposed approach to improve the prediction performance of different algorithms, and to discuss the effects of hyperparameters and prefilters.

Three lane changing behaviors (LCR, LK, LCL) are modeled as classifications. A highway scenario with traffic designed to enable overtaking maneuvers is realized in a driving simulator to collect driving data for training and testing the models. The prediction performance of improved models by finding optimal design parameters are considered and improved. Based on data achieved from 17 different drivers the proposed approaches are validated. For each algorithm, four different models are trained to evaluate their prediction performance.

The finally obtained results prove the optimization and selection of prefilter and hyperparameters can significantly improve the performance of the driving behavior prediction models. The group with the best performance in the contribution is M4 group which applies both optimized prefilter and hyperparameters. The influences of using prefilter or using hyperparameters are not the same for different algorithms. For CNN, hyperparameters are more decisive than prefilters. In contrast, prefilters are more advantageous for FL-HMM and ML-HMM. However, for other algorithms the difference is not obvious.

Furthermore, compared with all the mentioned algorithms, FL-HMM and ML-HMM demonstrate better results in this contribution. The upcoming maneuver can be predicted with a high DR (larger than 95 %) about 1.6 s (LCL) and 1.2 s (LCR) before actual lane change actions.

8.2 Outlook

The human behaviors discussed in this thesis are lane changing behaviors. In future work, other human behaviors can be used to test and validate the proposed approach.

The driving scenarios are built based on the driving process on a highway. Other driving environment or complex real-world applications can be considered in the future.

In addition to the prediction of ego-vehicle behavior (a single driver behavior), the prediction and recognition of multi-vehicle interaction is also an important topic of current researches. Reliable predicting the movement of surrounding vehicles plays an important role in the development of autonomous vehicles. This thesis does not detail this point which can be considered as a relevant influencing factor in future work.

In the full scale training loop, a quantized prefilter is applied for mapping the vehicles environment to quantized states. Thus, other different types of prefilters can be studied in future work.

Bibliography

[AAAD12] Aljaafreh, A. ; Alshabatat, N. ; Al-Din, M.S. N.: Driving Style Recognition Using Fuzzy Logic. In: *IEEE International Conference on Vehicular Electronics and Safety*, 2012, pp. 460 – 463

[AD19] Aszemi, N. M. ; Dominic, P. D. D.: Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms. In: *International Journal of Advanced Computer Science and Applications (IJACSA)* 10 (2019), no. 6, pp. 269 – 278

[ADS19] Ameyaw, D. A. ; Deng, Qi. ; Söffker, D.: Probability of Detection (POD)-based metric for evaluation of Classifiers used in Driving Behavior Prediction. In: *Annual Conference of the Prognostics and Health Management (PHM) Society*, 2019, pp. 1 – 7

[ADSH12] Aoude, G. S. ; Desaraju, V. R. ; Stephens, L. H. ; How, J. P.: Driver Behavior Classification at Intersections and Validation on Large Naturalistic Data Set. In: *IEEE Transactions on Intelligent Transportation Systems* 13 (2012), pp. 724 – 736

[AH16] Amsalu, S. B. ; Homaifar, A.: Driver Behavior Modeling Near Intersections Using Hidden Markov Model Based on Genetic Algorithm. In: *IEEE International Conference on Intelligent Transportation Engineering*, 2016, pp. 193 – 200

[AK07] Acar, P. Boyraz M. ; Kerr, D.: Signal Modelling and Hidden Markov Models for Driving Manoeuvre Recognition and Driver Fault Diagnosis in an urban road scenario. In: *IEEE Intelligent Vehicles Symposium*, 2007, pp. 987 – 992

[AMO07] Abe, K. ; Miyatake, H. ; Oguri, K.: A Study on switching AR-HMM driving behavior model depending on drivers states. In: *IEEE Intelligent Transportation Systems Conference*, 2007, pp. 806 – 811

[ANN12] Agamennoni, G. ; Nieto, J. I. ; Nebot, E. M.: Estimation of Multivehicle Dynamics by Considering Contextual Information. In: *IEEE Transactions on Robotics* 28 (2012), no. 4, pp. 855 – 870

[ASABZ13] Al-Sultan, S. ; Al-Bayatti, A. H. ; Zedan, H.: Context-aware driver behavior detection system in intelligent transportation systems. In: *IEEE Transactions on Vehicular Technology* 62 (2013), no. 9, pp. 4264 – 4275

[AYKG14] Aydn, M. M. ; Yldrm, M. S. ; Karpuz, O. ; Ghasemlou, K.: Modling of Driver Lane Choice Behavior with Artificial Neural Networks (ANN) and Linear Regression (LR) Analysis on Deformed Roads. In: *Computer Science and Engineering: An International Journal (CSEIJ)* 4 (2014), no. 1, pp. 47 – 57

[BC16] Bao, Y. ; Chen, W.: A Personalized Route Search Method Based on Joint Driving and Vehicular Behavior Recognition. In: *Wireless Symposium (IWS), IEEE MTT-S International Wireless Symposium (IWS)*, 2016, pp. 1 – 6

[BD15] BragaNeto, U. M. ; Dougherty, E. R.: Error Estimation for Pattern Recognition. In: *IEEE Press Series on Biomedical Engineering* (2015), pp. 48 – 55

[BDCK20] Brown, K. ; Driggs-Campbell, K. ; Kochenderfer, M. J.: Modeling and Prediction of Human Driver Behavior: A Survey. In: *arXiv preprint arXiv* (2020), pp. 1 – 20

[BGR01] Beal, M. J. ; Ghahramani, Z. ; Rasmussen, C. E.: The Infinite Hidden Markov Model. In: *in Proceedings of the Advances in Neural Information Processing Systems, MIT Press*, 2001, pp. 577 – 584

[BNKZ11] Baer, T. ; Nienhueser, D. ; Kohlhaas, R. ; Zoellner, J. M.: Probabilistic Driving Style Determination by means of a Situation Based Analysis of the Vehicle Data. In: *IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 1698 – 1703

[Bre96] Breiman, L.: Bagging predictors. In: *Machine Learning, Springer* 24 (1996), pp. 123 – 140

[Bre01] Breiman, L.: Random forests. In: *Machine learning* 45 (2001), pp. 5 – 32

[Bur98] Burger, C. J. C.: A tutorial on support vector machines for pattern recognition. In: *Data Mining and Knowledge Discovery, Springer* 2 (1998), pp. 121 – 167

[BWKS14] Bonnin, S. ; Weisswange, T. H. ; Kummert, F. ; Schmuedderich, J.: General Behavior Prediction by a Combination of Scenario-Specific Models. In: *IEEE Transactions on Intelligent Transportation Systems* 15 (2014), no. 4, pp. 1478 – 1488

[CAFH13] Chong, L. ; Abbas, M. M. ; Flintsch, A. M. ; Higgs, B: A rule-based neural network approach to model driver naturalistic behavior in traffic. In: *Transportation Research Part C* 32 (2013), pp. 207 – 223

[CCJJ17] Cheng, Z. ; Chow, M. Y. ; Jung, D. ; Jeon, J.: A big data based deep learning approach for vehicle speed prediction. In: *IEEE International Symposium on Industrial Electronics* (2017), pp. 389 – 394

[Chu99] Chu, K.: An introduction to sensitivity, specificity, predictive values and likelihood ratios. In: *Emergency Medicine* 3 (1999), pp. 175 – 181

[CKG15] Chen, R. ; Kusano, K.D. ; Gabler, H.C.: Driver Behavior During Overtaking Maneuvers from the 100-Car Naturalistic Driving Study. In: *NCBI Journals Traffic Injury Prevention* 15 (2015), pp. 1 – 10

[CLZ+17] Cao, W. ; Lin, X. ; Zhang, K. ; Dong, Y. ; Huang, S. ; Zhang, L.: Analysis and Evaluation of Driving Behavior Recognition Based on a 3-axis Accelerometer Using a Random Forest Approach. In: *ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2017, pp. 303 – 304

[CRKK16] Craye, C. ; Rashwan, A. ; Kamel, M. S. ; Karray, F.: A Multi-Modal Driver Fatigue and Distraction assessment System. In: *International Journal of Intelligent Transportation Systems Research* 14 (2016), no. 3, pp. 173 – 194

[CSG+10] Candamo, J. ; Shreve, M. ; Goldgof, D. B. ; Sapper, D. B. ; Kasturi, R.: Understanding transit scenes: A survey on human behavior recognition algorithms. In: *IEEE transactions on intelligent transportation systems* 11 (2010), no. 1, pp. 206 – 224

[DDWL14] Ding, J. ; Dang, R. ; Wang, J. ; Li, K.: Driver Intention Recognition Method Based on Comprehensive Lane-Change Environment Assessment. In: *Intelligent Vehicles Symposium Proceedings*, 2014, pp. 214 – 220

[DHB+11] Daza, I. G. ; Hernandez, N. ; Bergasa, L. M. ; Parra, I. ; Yebes, J. J. ; Gavilan, M. ; Quintero, R. ; Llorca, D. F. ; Sotelo, M. A.: Drowsiness monitoring based on driver and driving data fusion. In: *Proceedings of the 2011 14th International IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 1199 – 1204

[DHS11] Daza, L. G. ; Hernandez, N. ; Sotelo, L. M.: Drowsiness monitoring based on driver and driving data fusion. In: *IEEE Conference on Intelligent Transportation Systems*, 2011, pp. 1199 – 1204

[DKB20] Dzedzickis, A. ; Kaklauskas, A. ; Bucinskas, V.: Human Emotion Recognition: Review of Sensors and Methods. In: *Sensors* 20 (2020), no. 3, pp. 1 – 40

[DKP03] Duan, K. ; Keerthi, S. S. ; Poo, A. N.: Evaluation of simple performance measures for tuning SVM hyperparameters. In: *Neurocomputing* 51 (2003), no. 2, pp. 41 – 59

[DL15] Derbel, O. ; Landry, R.: Driving Style Assessment Based On the GPS Data and Fuzzy Inference Systems. In: *12th International Multi-Conference on Systems, Signals and Devices*, 2015, pp. 1 – 8

[Don99] Donges, E.: A Conceptual Framework for Active Safety in Road Traffic. In: *International Journal of Vehicle Mechanics and Mobility Vehicle System Dynamics* 32 (1999), pp. 113 – 128

[Don16] Donges, E.: Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort. Springer International Publishing Switzerland, 2016, pp. 19–34

[DPAM02] Deb, K. ; Pratap, A. ; Agarwal, S. ; Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. In: *IEEE Transactions on Evolutionary Computation* 6 (2002), no. 2, pp. 182 – 197

[DS18] Deng, Q. ; Söffker, D.: Improved driving behaviors prediction based on Fuzzy Logic-Hidden Markov Model (FL-HMM). In: *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 2003 – 2008

[DS19a] Deng, Q. ; Söffker, D.: Classifying Human Behaviors: Improving Training of Conventional Algorithms. In: *IEEE Transactions on Intelligent Transportation Systems Conference (ITSC 2019)*, 2019, pp. 1060 – 1065

[DS19b] Deng, Q. ; Söffker, D.: Modeling and Prediction of Human Behaviors based on Driving Data using Multi-Layer HMMs. In: *IEEE Transactions on Intelligent Transportation Systems Conference (ITSC 2019)*, 2019, pp. 2014 – 2019

[DSTS20] Deng, Q. ; Saleh, M. ; Tanshi, F. ; Söffker, D.: Online Intention Recognition Applied to Real Simulated Driving Maneuvers. In: *IEEE Conference on Cognitive and Computational Aspects of Situation Management* (2020), pp. 1 – 6

[DT11] Doshi, A. ; Trivedi, M. M.: Tactical Driver Behavior Prediction and Intent Inference: A Review. In: *IEEE Conference on Intelligent Transportation Systems* (2011), pp. 1892 – 1897

[DTB+10] Dai, J. ; Teng, J. ; Bai, X. ; Shen, Z. ; Xuan, D.: Mobile Phone based Drunk Driving Detection. In: *IEEE International Conference on Pervasive Computing Technologies for Healthcare*, 2010, pp. 1 – 8

[DWH+20] Deng, Q. ; Wang, J. ; Hillebrand, K. ; Benjamin, C.R. ; Söffker, D.: Prediction performance of lane changing behaviors: a study of combining environmental and eye-tracking data in a driving simulator. In: *Transactions on Intelligent Transportation Systems (ITS)* vol. 21, 2020, pp. 3561 – 3570

[DWS18] Deng, Q. ; Wang, J. ; Söffker, D.: Prediction of human driver behaviors based on an improved HMM approach. In: *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 2066 – 2071

[DWWB13] Ding, C. ; Wang, W. ; Wang, X. ; Bauman, M.: A Neural Network Model for Drivers Lane-Changing Trajectory Prediction in Urban Traffic Flow. In: *Hindawi Publishing Corporation, Mathematical Problems in Engineering* 2013 (2013), pp. 1 – 8

[DYF16] Dou, Y. ; Yan, F. ; Feng, D.: Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. In: *IEEE International Conference on Advanced Intelligent Mechatronics*, 2016, pp. 901 – 906

[EMAY12] Eren, H. ; Makinist, S. ; Akin, E. ; Yilmaz, A.: Estimating Driving Behavior by a Smartphone. In: *IEEE Intelligent Vehicles Symposium*, 2012, pp. 234 – 239

[Fle08] Fleming, W. J.: New Automotive Sensors - A Review. In: *IEEE Sensors Journal* 8 (2008), no. 11, pp. 1900 – 1921

[FSJS07] Fox, E. B. ; Sudderth, E. B. ; Jordan, M. l. ; Sky, A. S. W.: The sticky HDP-HMM: Bayesian Nonparametric Hidden Markov Models with Persistent States. In: *MIT Laboratory for Information and Decision Systems, MIT Press*, 2007, pp. 1 – 60

[FSJW09] Fox, E. B. ; Sudderth, E. B. ; Jordan, M. I. ; Willsky, A. S.: Sharing features among dynamical systems with beta processes. In: *in Proc. Adv. Neural Inf. Process. Syst., Vancouver, BC, Canada* 22 (2009), pp. 549 – 557

[FST98] Fine, S. ; Singer, Y. ; Tishby, N.: The hierarchical hidden Markov model: Analysis and applications. In: *Machine Learning* 32 (1998), no. 1, pp. 41 – 62

[Fuk80] Fukushima, K.: Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. In: *Biological Cybernetics* 36 (1980), pp. 193 – 202

[FWL+18] Fan, X. ; Wang, F. ; Lu, Y. ; Song, D. ; Liu, J.: Eye Gazing Enabled Driving Behavior Monitoring and Prediction. In: *IEEE International Conference on Multimedia Expo Workshops* (2018), pp. 1 – 4

[GBD13] Gindele, T. ; Brechtel, S. ; Dillmann, R.: Learning Context Sensitive Behavior Models from Observations forPredicting Traffic Situations. In: *16th International IEEE Annual Conference on Intelligent Transportation Systems*, 2013, pp. 1764 – 1771

[GKKO11] Gadepally, V. ; Kurt, A. ; Krishnamurthy, A. ; Özgüner, Ü.: Driver/vehicle state estimation and detection. In: *14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2011, pp. 582 – 587

[GKO14] Gadepally, V. ; Krishnamurthy, A. ; Özgüner, Ü.: A Framework for Estimating Driver Decisions Near Intersections. In: *IEEE Transactions on Intelligent Transportation Systems* 15 (2014), no. 2, pp. 637 – 646

[GMZ18] Gao, J. ; Murphey, Y. L. ; Zhu, H.: Multivariate time series prediction of lane changing behavior using deep neural network. In: *The International Journal of Research on Intelligent Systems for Real Life Complex Problems, Moonis Ali, Springer* 48 (2018), no. 10, pp. 3523 – 3537

[GSZ+17] Gang, L. ; Song, X. ; Zhang, M. ; Yao, B. ; Zhou, L.: Analysis of driver fatigue causations based on the Bayesian network model. In: *Simulation: Transactions of the Society for Modeling and Simulation International* 93 (2017), no. 7, pp. 553 – 565

[Gur] Gurney, K.: An Introduction to Neural Networks. In: *British Library Cataloguing in Publication Data* , pp. 34 – 37

[Hay72] Hayward, J.C.: Near-Miss Determination through Use of a Scale of Danger. In: *Highway Research Record* 384 (1972), pp. 24 – 35

[HES12] Hou, Y. ; Edara, P. ; Sun, C.: A Genetic Fuzzy System for Modeling Mandatory Lane Changing. In: *Intelligent Transportation Systems (ITSC)*, 2012, pp. 1044 – 1048

[HKI+16] Hamada, R. ; Kubo, T. ; Ikeda, K. ; Zhang, Z. ; Shibata, T. ; Bando, T. ; Hitomi, K. ; Egawa, M.: Modeling and Prediction of Driving Behaviors Using a Nonparametric Bayesian Method With AR Models. In: *IEEE Transactions on Intelligent Vehicles* 1 (2016), no. 2, pp. 131 – 138

[HL02] Hsu, C. W. ; Lin, C. J.: A Comparison of Methods for Multi-class Support Vector Machines. In: *IEEE Transactions on neural networks* 13 (2002), no. 2, pp. 415 – 425

[HNL03] Hollnagel, E. ; Nabo, A. ; Lau, I. V.: A systemic Model for Driver-in-Control. In: *Proceedings of the Second International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 2003, pp. 86 – 91

[HS15] Hossin, M. ; Sulaiman, M. N.: A review on evaluation metrics for data classification evaluations. In: *International Journal of Data Mining & Knowledge Management Process* 5 (2015), no. 2, pp. 1 – 11

[HWJ+12] Hurwitz, D. S. ; Wang, H. ; Jr., M. A. K. ; Ni, D. ; Moore, D.: Fuzzy sets to describe driver behavior in the dilemma zone of high-speed signalized intersections. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 15 (2012), no. 2, pp. 132 – 143

[JF15] Jiang, B. ; Fei, Y.: Traffic and Vehicle Speed Prediction with Neural Network and Hidden Markov Model in Vehicular Networks. In: *IEEE Intelligent Vehicles Symposium*, 2015, pp. 1082 – 1087

[JF16] Jiang, B. ; Fei, Y. S.: Vehicle Speed Prediction by Two-Level Data Driven Models in Vehicular Networks. In: *IEEE Transaction on Intelligent Transportation Systems* 18 (2016), no. 7, pp. 1793 – 1801

[KB14] Kalra, N. ; Bansal, D.: Analyzing Driver Behavior using Smartphone Sensors: A Survey. In: *International Journal of Electronic and Electrical Engineering* 7 (2014), no. 7, pp. 697 – 702

[KGYK15] Kaplan, S. ; Guvensan, M. A. ; Yavuz, A. G. ; Karalurt, Y.: Driver behavior analysis for safe driving: A survey. In: *IEEE Transactions on Intelligent Transportation Systems* 16 (2015), no. 6, pp. 3017 – 3032

[KKLD11] Khushaba, R. N. ; Kodagoda, S. ; Lal, S. ; Dissanayake, G.: Driver drowsiness classification using fuzzy wavelet-packet-based featureextraction algorithm. In: *IEEE Transactions on Biomedical Engineering* 58 (2011), no. 1, pp. 121 – 131

[KKS15] Kye, D. K. ; Kim, S. W. ; Seo, S. W.: Decision Making for Automated Driving at Unsignalized Intersection. In: *International Conference on Control, Automation and Systems*, 2015, pp. 522 – 525

[KPLL13] Kumar, P. ; Perrollaz, M. ; Lefevre, S. ; Laugier, C.: Learning-Based Approach for Online Lane Change Intention Prediction. In: *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 797 – 802

[KSOA03] Kumagai, T. ; Sakaguchi, Y. ; Okuwa, M. ; Akamatsu, M.: Prediction of Driving Behavior through Probabilistic Inference. In: *Proceedings of the Eighth International Conference on Engineering Applications of Neural Networks*, 2003, pp. 117 – 123

[LBH15] Lecun, Y. ; Bengio, Y. ; Hinton, G.: Deep learning. In: *In: Nature* 521 (2015), pp. 436 – 444

[LC12] Lee, B. G. ; Chung, W. Y.: Driver Alertness Monitoring Using Fusion of Facial Features and Bio-Signals. In: *IEEE Sensors Journal* 12 (2012), no. 7, pp. 2416 – 2422

[LCG+15] Lefevre, S. ; Carvalho, A. ; Gao, Y. ; Tseng, H. E. ; Borrelli, F.: Driver models for personalised driving assistance. In: *Vehicle System Dynamics* 53 (2015), no. 12, pp. 1705 – 1720

[LGN09] Lee, H. ; Grosse, R. ; Ng, R. Ranganathand A. Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: *Proceedings of the 26th Annual International Conference on Machine Learning - ICML*, 2009, pp. 1 – 8

[LKMH17] Lee, D. ; Kwon, Y. P. ; McMains, S. ; Hedrick, J. K.: Convolution neural network-based lane change intention prediction of surrounding vehicles for ACC. In: *IEEE International Conference on Intelligent Transportation Systems*, 2017, pp. 1 – 6

[LKO14] Liu, P. ; Kurt, A. ; Özgüner, Ü.: Trajectory Prediction of a Lane Changing Vehicle Based on Driver Behavior Estimation and Classification. In: *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 942 – 947

[LM15] Lemieux, J. ; Ma, Y.: Vehicle Speed Prediction Using Deep Learning. In: *IEEE Vehicle Power and Propulsion Conference* (2015), pp. 1 – 5

[LMT13] Ly, M. V. ; Martin, S. ; Trivedi, M. M.: Driver Classification and Driving Style Recognition using Inertial Sensors. In: *IEEE Intelligent Vehicles Symposium*, 2013, pp. 1040 – 1045

[LRL07] Liang, Y. ; Reyes, M. L. ; Lee, J. D.: Real-time detection of driver cognitive distraction using support vector machines. In: *IEEE Transactions on Intelligent Transportation Systems* 8 (2007), no. 2, pp. 340 – 350

[LZT+14] Lin, N. ; Zong, C. ; Tomizuka, M. ; Song, P. ; Zhang, Z. ; Li, G.: An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control. In: *Hindawi Publishing Corporation - Mathematical Problems in Engineering* (2014), pp. 1 – 15

[MAA17] Masri, A. E. B. E. ; Artail, H. ; Akkary, H.: Toward Self-Policing: Detecting Drunk Driving Behaviors through Sampling CAN Bus Data. In: *IEEE International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, 2017, pp. 1 – 5

[MHW+18] Martinez, C. M. ; Heucke, M. ; Wang, F. Y. ; Gao, B. ; Cao, D.: Driving style recognition for intelligent vehicle control and advanced driver assistance: A survey. In: *IEEE Transactions on Intelligent Transportation Systems* 19 (2018), no. 3, pp. 666 – 676

[Mic85] Michon, J. A.: A Critical view of driver behavior models: What do we know what should we do? In: *Human behavior and traffic safety*, 1985, pp. 485 – 520

[MLP12] M., C. G. Q. ; Lpez, J. O. ; Pinilla, A. C. C.: Driver Behavior Classification Model based on an Intelligent Drivin Diagnosis System. In: *IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 894 – 899

[MM15] Meiring, G. A. M. ; Myburgh, H. C.: A Review of Intelligent Driving Style Analysis Systems and Related Artificial Intelligence Algorithms. In: *Sensors* 15 (2015), no. 12, pp. 30653 – 30682

[Moo96] Moon, T. K.: The expectation-maximization algorithm. In: *IEEE Signal Processing Magazine* 13 (1996), no. 6, pp. 47 – 60

[MPG+14] Montella, A. ; Pariota, L. ; Galante, F. ; Imbriani, L. L. ; Mauriello, F.: Prediction of Drivers' Speed Behavior on Rural Motorways Based on an Instrumented Vehicle Study. In: *Transportation Research Record: Journal of the Transportation Research Board* (2014), no. 2434, pp. 52 – 62

[MT16] Miyajima, C. ; Takeda, K.: Driver-behavior modeling using on road driving data: A new application for behavior signal processing. In: *IEEE Signal Processing Magazine* 33 (2016), no. 6, pp. 14 – 21

[MYU09] Mochihashi, D. ; Yamada, T. ; Ueda, N.: Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore*, 2009, pp. 100 – 108

[Nat15] National Highway Traffic Safety Adaministration : Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. In: *NHTSAs National Center for Statistics and Analysis*, 2015

[Nat16a] National Highway Traffic Safety Adaministration : 2015 Motor Vehicle Crashes: Overview. In: *NHTSAs National Center for Statistics and Analysis*, 2016

[Nat16b] National Highway Traffic Safety Administration : Traffic Safety Fact Sheet Summary of Motor Vehicle Crashes. In: *NHTSAs National Center for Statistics and Analysis*, 2016

[Nat17] National Highway Traffic Safety Adaministration: Drowsy Driving 2015. In: *NHTSAs National Center for Statistics and Analysis*, 2017

[NTH+14] Nagasaka, S. ; Taniguchi, T. ; Hitomi, K. ; Takenaka, K. ; Bando, T.: Prediction of Next Contextual Changing Point of Driving Behavior Using Unsupervised Bayesian Double Articulation Analyzer. In: *IEEE Intelligent Vehicles Symposium (IV)*, 2014, pp. 924 – 931

[NTY+12] Nagasaka, S. ; Taniguchi, T. ; Yamashita, G. ; Hitomi, K. ; Bando, T.: Finding meaningful robust chunks from driving behavior based on double articulation analyzer. In: *International Symposium on System Integration (SII)*, 2012, pp. 535 – 540

[OPB12] Oshiro, T.M. ; Perez, P.S. ; Baranauskas, J.A.: How Many Trees in a Random Forest? In: *International Conference on Machine Learning and Data Mining in Pattern Recognition,*, 2012, pp. 154 – 168

[PLCY14] Pan, J. S. ; Lu, K. ; Chen, S.H. ; Yan, L.: Driving Behavior Analysis of Multiple Information Fusion Based on SVM. In: *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer*, 2014, pp. 60 – 69

[PLKR11] Patel, M. ; Lal, S. ; Kavanagh, D. ; Rossiter, P.: Applying neural network analysis on heart rate variability data to assess driver fatigue. In: *Expert Systems with Applications* 38 (2011), no. 6, pp. 7235 – 7242

[QLH12] Qin, H. ; Liu, J. ; Hong, T.: An Eye State Identification Method Based on the Embedded Hidden Markov Model. In: *IEEE International Conference on Vehicular Electronics and Safety*, 2012, pp. 255 – 260

[Rab89] Rabiner, L. R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. In: *Proceedings of the IEEE* 77 (1989), pp. 257 – 286

[Ran94] Ranney, T. A.: Models of Driving Behavior: a Review of their Evolution. In: *Accident Analysis and Prevention* 26 (1994), no. 6, pp. 733 – 750

[Ras83] Rasmussen, J.: Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. In: *IEEE Transactions on systems, man, and cybernetics* 13 (1983), no. 3, pp. 257 – 266

[RJ86] Rabiner, L. R. ; Juang, B. H.: An Introduction to Hidden Markov Models. In: *IEEE Acoustics, Speech, Signal Processing Magazine* 3 (1986), pp. 4 – 16

[RS16] Rothe, S. ; Söffker, D.: Comparison of Different Information Fusion Methods Using Ensemble Selection Considering Benchmark Data. In: *IEEE International Conference on Information Fusion* (2016), pp. 73 – 78

[SBH08] Sathyanarayana, A. ; Boyraz, P. ; Hansen, J. H. L.: Driver Behavior Analysis and Route Recognition by Hidden Markov Models. In: *IEEE International Conference on Vehicular Electronics and Safety* (2008), pp. 276 – 281

[SLZ+11] Shiwu, Li ; Linhong, Wang ; Zhifa, Yang ; Bingkui, Ji ; Feiyan, Qiao ; Zhongkai, Yang: An active driver fatigue identification technique using multiple physiological features. In: *IEEE International Conference on Mechatronic Science, Electric Engineering and Computer*, 2011, pp. 733 – 737

[SS02] Schölkopf, B. ; Smola, A.J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. The MIT Press, London, 2002. – 15 – 20 S

[SSPE15] Sagberg, F. ; Selpi, S. ; Piccinini, G. F. B. ; Engstrom, J.: A Review of Research on Driving Styles and Road Safety. In: *Human Factors* 57 (2015), no. 7, pp. 1248 – 1275

[TJBB06] Teh, Y. W. ; Jordan, M. I. ; Beal, M. J. ; Blei, D. M.: Hierarchical Dirichlet processes. In: *Journal of the American Statistical Association*, 2006, pp. 1566 – 1581

[TNH+15] Taniguchi, T. ; Nagasaka, S. ; Hitomi, K. ; Takenaka, K. ; Bando, T.: Unsupervised Hierarchical Modeling of Driving Behavior and Prediction of Contextual Changing Points. In: *IEEE Transactions on Intelligent Transportation Systems* 16 (2015), no. 4, pp. 1746 – 1760

[TNH+16] Taniguchi, T. ; Nagasaka, S. ; Hitomi, K. ; Chandrasiri, N. P. ; Bando, T. ; Takenaka, K.: Sequence Prediction of Driving Behavior Using Double Articulation Analyzer. In: *IEEE Transactions on*

Systems, Man, and Cybernetics: Systems 46 (2016), no. 9, pp. 1300 – 1313

[TSL14] Tadesse, E. ; Sheng, W. ; Liu, M.: Driver Drowsiness Detection through HMM based Dynamic Modeling. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 4003 – 4008

[TSLL15] Tran, D. ; Sheng, W. ; Liu, L. ; Liu, M.: A Hidden Markov Model Based Driver Intention Prediction System. In: *The 5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 2015, pp. 115 – 120

[Vap95] Vapnik, V.: *The nature of statistical learning theory.* Springer-Verlag, 1995

[VB15] Varghese, J.Z. ; Boone, R.G.: Overview of Autonomous Vehicle Sensors and Systems. In: *International Conference on Operations Excellence and Service Engineering*, 2015, pp. 178 – 191

[WCY14] Wu, B. F. ; Chen, Y. H. ; Yeh, C. H.: Driving behaviour-based event data recorder. In: *IET Intelligent Transport Systems* 8 (2014), no. 4, pp. 361 – 367

[Win16] Winner, H.: Fundamentals of Collision Protection Systems. In: *in Chapter 6 of Handbook of Driver Assistance Systems* (2016), pp. 1149 – 1176

[WM97] Wolpert, D. H. ; Macready, W. G.: No free lunch theorems for optimization. In: *IEEE transactions on evolutionary computation* 1 (1997), no. 1, pp. 67 – 82

[Wor15] World Health Organization: Global Status Report on Road Safety 2015. In: *WHO Library Cataloguing-in-Publication Data*, 2015

[WWW+14] Wang, F. ; Wang, S. ; Wang, X. ; Peng, Y. ; Yang, Y.: Design of Driving Fatigue Detection System Based on Hybrid Measures Using Wavelet-packets Transform. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 4037 – 4042

[WXC14] Wang, W. ; Xi, J. ; Chen, H.: Modeling and recognizing driver behavior based on driving data: A survey. In: *Hindawi Publishing Corporation - Mathematical Problems in Engineering* (2014), pp. 1 – 20

[WXH19] Wang, W. ; Xi, J. ; Hedrick, J. K.: A Learning-Based Personalized Driver Model Using Bounded Generalized Gaussian Mixture Models.

In: *IEEE Transactions on Vehicular Technology* 68 (2019), no. 12, pp. 11679 – 11690

[WXZ18a] Wang, W. ; Xi, J. ; Zhao, D.: Driving style analysis using primitive driving patterns with bayesian nonparametric approaches. In: *IEEE Transactions on Intelligent Transportation Systems* 20 (2018), no. 8, pp. 2986 – 2998

[WXZ18b] Wang, W. ; Xi, J. ; Zhao, D.: Learning and Inferring a Driver's Braking Action in Car-Following Scenarios. In: *IEEE Transactions on Vehicular Technology* 67 (2018), no. 5, pp. 3887 – 3899

[WY10] Wu, Q. ; Yu, W.: A Driver Abnormality Recondition Model based on Dynamic Bayesian Network for Ubiquitous Computing. In: *IEEE International Conference on Advanced Computer Theory and Engineering* 1 (2010), pp. 320 – 324

[WZHX18] Wang, W. ; Zhao, D. ; Han, W. ; Xi, J.: A learning-based approach for lane departure warning systems with a personalized driver model. In: *IEEE Transactions on Vehicular Technology* 67 (2018), no. 10, pp. 9145 – 9157

[XCL17] Xiong, X. ; Chen, L. ; Liang, J.: A New Framework of Vehicle Collision Prediction by Combining SVM and HMM. In: *IEEE Transacions on Intelligent Transportation Systems* 19 (2017), pp. 699 – 710

[XLW+14] Xiong, G. ; Li, Y. ; Wang, S. ; Li, X. ; Liu, P.: HMM and HSS Based Social Behavior of Intelligent Vehicles for Freeway Entrance Ramp. In: *International Journal of Control and Automation* 7 (2014), no. 10, pp. 79 – 90

[YZBZ13] Yao, W. ; Zhao, H. ; Bonnifait, P. ; Zha, H.: Lane Change Trajectory Prediction by using Recorded Human Driving Data. In: *IEEE Intelligent Vehicles Symposium* (2013), pp. 430 – 436

[ZB02] Zhao, J. ; Bose, B.K.: Evaluation of membership functions for fuzzy logic controlled induction motor drive. In: *IEEE Industrial Electronics Society* 1 (2002), pp. 229 – 234

[ZGL+17] Zhao, C. ; Gong, J. ; Lu, C. ; Xiong, G. ; Mei, W.: Speed and Steering Angle Prediction for Intelligent Vehicles Based on Deep Belief Network. In: *IEEE International Conference on Intelligent Transportation Systems* (2017), pp. 301 – 306

[ZKZ+15] Zhu, T. ; Khajepour, A. ; Zong, C. ; Mai, L. ; Deng, H.: Driving condition identification based on HHMM model Application to the simulation of heavy duty vehicle. In: *The dynamics of vehicles on roads and tracks: proceedings of the 24th Symposium of the International Association for Vehicle System Dynamics (IAVSD)*, 2015, pp. 663 – 668

[ZSF14] Zheng, J. ; Suzuki, K. ; Fujita, M.: Predicting drivers lane-changing decisions using a neural network model. In: *Simulation Modelling Practice and Theory*, 2014, pp. 73 – 83

[ZVL14] Zylius, G. ; Vaitkus, V. ; Lengvenis, P.: Driving Style Analysis using Spectral Features of Accelerometer Signals. In: *Proceedings of 9th International Conference ITELMS*, 2014, pp. 267 – 273

[ZWL15] Zhang, L. ; Wu, X. ; Luo, D.: Human Activity Recognition with HMM-DNN Model. In: *IEEE Cognitive Informatics and Cognitive Computing*, 2015, pp. 192 – 197

[ZWYY09] Zong, C. ; Wang, C. ; Yang, D. ; Yang, H.: Driving Intention Identification and Maneuvering Behavior Prediction of Drivers on Cornering. In: *IEEE International Conference on Mechatronics and Automation*, 2009, pp. 4055 – 4060

[ZZWX20] Zhang, C. ; Zhu, J. ; Wang, W. ; Xi, Junqiang: Spatiotemporal Learning of Multivehicle Interaction Patterns in Lane-Change Scenarios. In: *arXiv preprint arXiv* (2020), pp. 1 – 19

This thesis is based on the results and development steps presented in the following previous publications:

Journal articles

[DSW+19] Deng, Q.; Wang, J.; Hillebrand, K.; Benjamin, C.R.; Söffker, D.: Prediction performance of lane changing behaviors: a study of combining environmental and eye-tracking data in a driving simulator. IEEE Transactions on Intelligent Transportation Systems (T-ITS), vol. 21, no. 8, 2019, pp. 3561-3570.

[DS20a] Deng, Q.; Söffker, D.: Novel HMM-based classifiers for human behavior prediction applied for assistance and supervision of human operators. IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2020, submitted.

[DS20b] Deng, Q.; Söffker, D.: A Review of the current HMM-based Approaches of Driving Behaviors Recognition and Prediction. IEEE Transactions on Intelligent Vehicles (T-IV), 2020, submitted.

[DS20c] Deng, Q.; Söffker, D.: Strategy for improved training of conventional algorithms for human driver behavior prediction. IEEE Transactions on Intelligent Transportation Systems (T-ITS), 2020, submitted.

Conference papers

[DWS18] Deng, Q.; Wang, J.; Söffker, D.: Prediction of human driver behaviors based on an improved HMM approach. 2018 IEEE Intelligent Vehicles Symposium, Changshu, Suzhou, China, 2018, pp. 2066-2071.

[DS18] Deng, Q.; Söffker, D.: Improved driving behaviors prediction based on Fuzzy Logic-Hidden Markov Model (FL-HMM). 2018 IEEE Intelligent Vehicles Symposium, Changshu, Suzhou, China, 2018, pp. 2003-2008.

[DS19a] Deng, Q.; Söffker, D.: Classifying Human Behaviors: Improving Training of Conventional Algorithms. IEEE Transactions on Intelligent Transportation Systems Conference (ITSC 2019), Auckland, New Zealand, 2019, pp. 1060-1065.

[DS19b] Deng, Q.; Söffker, D.: Modeling and Prediction of Human Behaviors based on Driving Data using Multi-Layer HMMs. IEEE Transactions on Intelligent Transportation Systems Conference (ITSC 2019), Auckland, New Zealand, 2019, pp. 2014-2019.

Workshop presentations

[DWS17] Deng, Q.; Wang, J.; Söffker, D.: Defining Feature Properties for Optimal HMM-based Situation Recognition for Human Drivers. 6. Interdisziplinärer Workshop Kognitive Systeme: Mensch, Teams, Systeme und Automaten, Neubiberg bei Mnchen, Germany, 2017.

[DS18b] Deng, Q.; Söffker, D.: Improved human driving behaviors prediction based on Fuzzy Logic-Hidden Markov Model. 7. Interdisziplinärer Kognitive Systeme: Mensch, Teams, Systeme und Automaten, Braunschweig, Germany, 2018.

[DS19c] Deng, Q.; Söffker, D.: Multi-Level HMMs-based Cognitive modeling for Human Driving Intentions Recognition. 8. Interdisziplinärer Kognitive Systeme: Mensch, Teams, Systeme und Automaten, Duisburg, Germany, 2019.

Other publications, which are not included in this thesis:

Journal articles

[MWD+15] Muthig, O.; Wang, J.; Deng, Q.; Söffker, D.: Integrating situated human interaction modeling and stochastic state automata for improved technical situation awareness. IFAC-PapersOnLine, vol. 48, no. 1, 2015, pp. 87-92.

[ADS20a] Ameyaw, D. A.; Deng, Q.; Söffker, D.: A new kind of visualization and interpretation of classifier performance. Artificial Intelligence, 2020, submitted.

[ADS20b] Ameyaw, D. A.; Deng, Q.; Söffker, D.: A new measure comparing machine learning-based classification approaches applied to dynamically changing situations. IEEE Intelligence Systems, 2020, submitted.

Conference papers

[ADS19] Ameyaw, D. A.; Deng, Q.; Söffker, D.: Probability of Detection (POD)-based metric for evaluation of Classifiers used in Driving Behavior Prediction. Proceedings of the Annual Conference of the PHM Society, 11(1) , Scottsdale, Arizona, USA, September 21-26, 2019.

[DST+20] Deng, Q.; Saleh, M.; Tanshi, F.; Söffker, D.: Online Intention Recognition Applied to Real Simulated Driving Maneuvers. IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA 2020), British Columbia, Canada, 2020, pp. 1-6.

In the context of research work at the Chair of Dynamics and Control, the following student theses have been supervised by Qi Deng and Univ.-Prof. Dr.-Ing. Dirk Söffker. Development steps and results of the research work and the student theses are integrated with each other and hence are also part of this thesis.

[Hil17] Hillebrand. K., Entwicklung eines Fahrerassistenzsystems zur Erhöhung der Fahrsicherheit und -effizienz, Master Thesis, December 2017

The following student theses have been supervised by Qi Deng and Univ.-Prof. Dr.-Ing. Dirk Söffker, which are not included in this thesis:

[Lee19] Lee, T. S., Exploring and analyzing effectiveness and usefulness of driving models based on different machine learning algorithms, Bachelor Thesis, January 2019

[Ong19] Ong, M. H., Toward the Realization of Autonomous Driving using CARLA Simulator, Bachelor Thesis, April 2019

[Sal19] Saleh, M., Data-driven-based Situation and Intention Recognition for real time driving evaluation, Master Thesis, March 2019

[Sad20] Sadiq, R., Development of a driver assistance system for decision support, Master Thesis, July 2020

DOI: 10.17185/duepublico/73923
URN: urn:nbn:de:hbz:464-20210126-160312-4