DRIVER BEHAVIOUR PREDICTION MODELS USING ARTIFICIAL INTELLIGENCE ALGORITHMS AND STATISTICAL MODELING

DRIVER BEHAVIOUR PREDICTION MODELS USING ARTIFICIAL INTELLIGENCE ALGORITHMS AND STATISTICAL MODELING

BY
YANGLIU DOU, B.E., M.S.

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES IN
PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE DOCTOR OF PHILOSOPHY

McMaster University Doctor of Philosophy (2019) Hamilton, Ontario (Mechanical Engineering)

TITLE:	DRIVER BEHAVIOUR PREDICTION MODELS USING ARTIFICIAL INTELLIGENCE ALGORITHMS AND STATISTICAL MODELING
AUTHOR:	Yangliu Dou M.S., (Mechanical Engineering) Northwestern Polytechnical University B.E., (Electronic and Information Engineering) Xi'an University of Post & Telecommunication
SUPERVISOR:	Associate Professor Fengjun Yan
NUMBER OF PAGES:	xxiv, 132

To my parents, for their unconditional love and support.

Lay Abstract

Lane changing and car following are the two most frequently encountered driving behaviours for intelligent vehicles. Substantial research has been carried out and several prototypes have been developed by universities as well as companies. However, the low accuracy and high computational cost prevent the existing lane changing models from providing safer and more reliable decisions for intelligent vehicles. In the existing car-following models, there are also few models that consider the effects of cut-ins from adjacent lanes which may result in their poor accuracy and efficiency. To address these obstacles, advanced artificial intelligence algorithms combined with sufficient driving environmental factors are proposed due to their promise of providing accurate, efficient, and robust lane changing and car-following models. The main part of this thesis is composed of three journal papers. Paper 1 proposed a gated branch neural network for a mandatory lane changing suggestion system at the on-ramps of highways; paper 2 developed a recurrent neural network time-series algorithm to predict the surrounding vehicles' discretionary lane changing intention in advance; paper 3 researched the strategic car-following gap model considering the effect of cut-ins from adjacent lanes.

Abstract

To improve the safety and comfort of intelligent vehicles, advanced driver models offer promising solutions which can successfully capture and reflect the driver behaviours, accurately alert the driver of potential risks, and fully assist the driver to avoid obstacles. However, several shortcomings of these models, including lack of comprehensive driving environmental factors, low accuracy and high computational cost, prevent them from being widely applied in reality. This is especially true for lane changing and car following, which are two of the most frequently encountered driving behaviours for intelligent vehicles. To address these shortcomings, advanced artificial intelligence algorithms in conjunction with the sufficient driving environmental factors are proposed based on real-life driving data. More specifically, three typical problems will be addressed in this thesis: Mandatory Lane Changing (MLC) suggestion at the highway entrance; Discretionary Lane Changing (DLC) intention prediction; Car-Following gap model considering the effect of cuts-in from the adjacent lanes.

For the MLC suggestion system, in which the main challenges are efficient decision making and high prediction accuracy of both non-merge and merge events, an additional gated branch neural network (GBNN) is proposed. The proposed GBNN algorithm employs a compact gated branch in addition to a feedforward neural network, and uses the scaled exponential linear units (SeLU) activation function and Adam optimizer. It not only achieves the highest accuracy among conventional binary classifiers in terms of great performance on the non-merge accuracy, the merge accuracy, and receiver operating characteristic score but also takes less time.

For the DLC, we propose a recurrent neural network (RNN)-based time series classifier with a gated recurrent units (GRU) architecture to predict the surrounding vehicles' intention. It can predict the surrounding vehicles' lane changing maneuver 0.8 s in advance at a recall and precision of 99.5% and 98.7%, respectively, which outperforms conventional algorithms such as the Hidden Markov Model (HMM).

Finally, drivers are typically faced with two competing challenges when following a preceding vehicle: they need to leave sufficient space in front to ensure safety, while doing so the probability of cut-ins by other vehicles increases as the car-following gap becomes large. A method is proposed to address the problem through an overall objective function of car-following gap and velocity considering the safety hazard and the probability of cut-ins by other vehicles. Based on this, seeking the strategic car-following gap translates to finding the optimal solution that minimizes the overall objective function. With the support of field data, the method along with concrete models are instantiated and the application of the method is elaborated.

Acknowledgements

First and foremost, I would like to express my great appreciation to my supervisor, Dr. Fengjun Yan, for giving me the opportunity to work with him as a Ph.D. student. I was encouraged and motivated by his great enthusiasm, insight and criticism for academic research. I was also deeply impressed by his attitude, humor and wisdom within his personal life.

Great appreciations are also expressed to Dr. Hongjin Sun for his invaluable guidance and help throughout my research. He encouraged me, supported me, and guided me with what I wanted to achieve. I am grateful for working with him. I would also like to thank Dr. Stephen Veldhuis, with his great academic enthusiasm and dedication, he has always provided helpful suggestions and comments on my studies. He has set a perfect model of a great scientist for me. Special thanks also go to Dr. Allan D. Spence, Dr. Jianqiang Wang. They gave me lots of precious suggestions and instruction on my experiments.

Besides that, my sincere thanks also go to my fellow groupmates in the Powertrain Control Lab, Qi Zhang, Song Chen, Jinbiao Ning, Yi Huo, Chuan Hu, Jiangtao He, Liang Wang, Yicong Liu, Tongrui Li, Yile Wang, Xiaowei Zhang, Mingbo Jia, and Zixiong Liu for their friendship and assistance.

Last but not least, I would like to thank my parents for their unconditional love and support throughout my life. I would also like to extend special thanks to my sister, Yangqing Dou, who has taught me how to bravely pursue true happiness and love.

List of Abbreviations and Symbols

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BF	Bayesian Filter
CAN	Controller Area Network
CDF	Cumulative Distribution Function
CFG	Car-Following Gap
CIP	Cut-In Probability
CNN	Convolutional Neural Network
DLC	Discretionary Lane Changing
ECG	Electrocardiogram
EV	Ego Vehicle
FV	Feature Vector
GBNN	Gated Branch Neural Network
GD	Gradient Descent
GHR	Gazis-Herman-Rohery model
GPS	Global Positioning System
GRU	Gated Recurrent Unit
GSR	Galvanic Skin Response
HI	Hazard Index
HMM	Hidden Markov Model
HOV	High-Occupancy Vehicle

HPV	Host-lane Preceding Vehicle
IDM	Intelligent Driver Model
IMU	Inertial Measurement Unit
IPC	Industrial Personal Computer
LKAS	Lane Keeping Assist System
LPV	Left-lane Preceding Vehicle
LSTM	Long Short-Term Memory
MLC	Mandatory Lane Changing
MSE	Mean Square Error
NGSIM	Next Generation Simulation
NHTSA	National Highway Transportation Safety Administration
OBE	On-board Equipment
PCA	Principle Component Analysis
PDF	Probability Density Function
RBF	Radial Basis Function
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
RPV	Right-lane Preceding Vehicle
SeLU	Scaled exponential linear units
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
THW	Subject Vehicle Time Headway
Δx_ego	Lateral distance of the subject vehicle with respect to the centre line of the lane
y_ego	Longitudinal coordinate of the subject vehicle
v_ego	Velocity of the subject vehicle

a_ego	Acceleration of the subject vehicle
d_lead	Distance between the front-center of a vehicle to the front-center of the preceding vehicle
Δv_lead	Speed difference between the lead vehicle in the current lane and the subject vehicle
Δa_lead	Acceleration difference between the lead vehicle in the current lane and the subject vehicle
d_lag	Longitudinal coordinate Y difference between the lag vehicle in the current lane and the subject vehicle
Δv_lag	Speed difference between the lag vehicle in the current lane and the subject vehicle
Δa_lag	Acceleration difference between the lag vehicle in the current lane and the subject vehicle
d_lead_l	Longitudinal coordinate Y difference between the lead vehicle in the left lane and the subject vehicle
Δv_lead_l	Speed difference between the lead vehicle in the left lane and the subject vehicle
Δa_lead_l	Acceleration difference between the lead vehicle in the left lane and the subject vehicle.
d_lag_l	Longitudinal coordinate Y difference between the lag vehicle in the left lane and the subject vehicle
Δv_lag_l	Speed difference between the lag vehicle in the left lane and the subject vehicle.
Δa_lag_l	Acceleration difference between the lag vehicle in the left lane and the subject vehicle
d_lead_r	Longitudinal coordinate Y difference between the lead vehicle in the right lane and the subject vehicle
Δv_lead_r	Speed difference between the lead vehicle in the right lane and the subject vehicle
Δa_lead_r	Acceleration difference between the lead vehicle in the right lane and the subject vehicle.
d_lag_r	Longitudinal coordinate Y difference between the lag vehicle in the right lane and the subject vehicle

Δv_lag_r	Speed difference between the lag vehicle in the right lane and the subject vehicle.
Δa_lag_r	Acceleration difference between the lag vehicle in the right lane and the subject vehicle
d_lead_t	Longitudinal gap between the front vehicle in target lane and the merging vehicle
Δv_lead_t	Velocity difference between the front vehicle in target lane and the merging vehicle
Δa_lead_t	Acceleration difference between the front vehicle in target lane and the merging vehicle
d_lag_t	Longitudinal gap between the rear vehicle in target lane and the merging vehicle
Δv_lag_t	Velocity difference between the rear vehicle in target lane and the merging vehicle
Δa_lag_t	Acceleration difference between the rear vehicle in target lane and the merging vehicle
$\ddot{x}_i$	Longitudinal acceleration of vehicle i
$\ddot{x}_j$	Longitudinal acceleration of vehicle j
$\dot{x}_i$	Velocity of vehicle i
$\dot{x}_j$	Velocity of vehicle j
d_{ij}	Car-following gap between vehicles i and j
a	Scale parameter in Weibull distribution
b	Shape parameter in Weibull distribution
a_{NCI}	Scale parameter of the model without cut-ins
a_{CI}	Scale parameter of the model with cut-ins
b_{NCI}	Shape parameter of the model without cut-ins
b_{CI}	Shape parameter of the model with cut-ins
w	Weight of the HI function

Declaration of Academic Achievement

1. **For modeling mandatory lane changing (MLC) behaviour**:
   a) A novel gated branched neural network is developed, which enables more efficient modeling for an MLC behaviour at the on-ramps of highways with high accuracy of both non-merge and merge events;
   b) In the algorithm architecture, an additional gated branch based on the correlation analysis is proposed, which provides effective feature learning and explicitly capture the relationship between the surrounding driving environment and the lane changing decision;
   c) The presented method is lightweight, compared with existing deep learning algorithms, in computation and can be applied in advanced driver assistance systems (ADAS) for an efficient MLC suggestion.
2. **For modeling discretionary lane changing (DLC) behaviour**:
   a) A recurrent neural network (RNN)-based time series classifier with a gated recurrent units (GRU) architecture for modeling DLC maneuvers is proposed;
   b) An early and reliable prediction of the surrounding vehicles' DLC behaviour is provided, which helps improve the driving safety and efficiency of ADAS;
   c) The proposed method is lightweight in computation and can be applied in ADAS or autonomous vehicles in real-time applications.
3. **For modeling car-following behaviour**:
   a) A strategic car-following gap is devised by incorporating the probability of cut-ins into the formulation to reveal the practical effect of lateral motions;
   b) An overall objective function consisting of the car-following gap and velocity is formulated by considering the safety hazard and the probability of cut-ins by

other vehicles. Based on this, seeking the strategic car-following gap translates to finding the optimal solution that minimizes the overall objective function.

Publications

Journal Papers

[J1] **Yangliu Dou**, Yihao Fang, Chuan Hu, Rong Zheng, Fengjun Yan, “A Gated Branch Neural Network for Mandatory Lane Changing Suggestion at the On-ramps of Highway.” *IET Intelligent Transport Systems*.2019, 13(1):48-54.

[J2] **Yangliu Dou**, Yihao Fang, Chuan Hu, Rong Zheng, Fengjun Yan, “Prediction of the Surrounding Vehicles’ Discretionary Lane Changing Intention at freeway: A Gated Recurrent Units Approach.” *IET Intelligent Transport Systems.* (Submitted)

[J3] **Yangliu Dou**, Daiheng Ni, Zhao Wang, Jianqiang Wang, Fengjun Yan, “Strategy Car-Following Gap Model Considering the Effect of Cut-ins from Adjacent Lanes.” *IET Intelligent Transport Systems.* 2016, 10(10): 658-665.

Peer-Reviewed Conference Papers

[C1] **Yangliu Dou**, Fengjun Yan, Daiwei Feng. Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers. *2016 IEEE International Conference on Advanced Intelligent Mechatronics. AIM,* 2016, Banff, Canada, pp. 901-906.

[C2] **Yangliu Dou**, Fengjun Yan, Kun Bu. Reversing design methodology of ceramic core for hollow turbine blade based on measured data*. 2014 International Mechanical Engineering Congress and Exposition*. ASME, 2014, Montreal, Canada, pp. V02AT02A051.

Statement of Co-Authorship

Paper 1: **Yangliu Dou**, Yihao Fang, Chuan Hu, Rong Zheng, Fengjun Yan, "A Gated Branch Neural Network for Mandatory Lane Changing Suggestion at the On-ramps of Highway." *IET Intelligent Transport Systems.*2019, 13(1):48-54.

Yangliu Dou researched the mandatory lane changing scenario, proposed the algorithm, wrote the code, and composed the manuscript; Yihao Fang and Chuan Hu optimized the Python and TensorFlow code; Rong Zheng and Fengjun Yan are the supervisor of Yihao Fang and Yangliu Dou, respectively. The main contributor to this paper is Yangliu Dou (contributes more than 80%).

Paper 2: **Yangliu Dou**, Yihao Fang, Chuan Hu, Rong Zheng, Fengjun Yan, "Prediction of the Surrounding Vehicles' Discretionary Lane Changing Intention at freeway: A Gated Recurrent Units Approach." *IET Intelligent Transport Systems.* (Submitted)

Yangliu Dou researched the discretionary lane changing scenario, proposed the algorithm, wrote the code, and composed the manuscript; Yihao Fang and Chuan Hu optimized the Python and TensorFlow code; Rong Zheng and Fengjun Yan are the supervisor of Yihao Fang and Yangliu Dou, respectively. The main contributor to this paper is Yangliu Dou (contributes more than 70%).

Paper 3: **Yangliu Dou**, Daiheng Ni, Zhao Wang, Jianqiang Wang, Fengjun Yan, "Strategy Car-Following Gap Model Considering the Effect of Cut-ins from Adjacent Lanes." *IET Intelligent Transport Systems.* 2016, 10(10): 658-665.

Yangliu Dou researched the car-following scenario, proposed the algorithm, wrote the code, and composed the manuscript. The main contributor to this paper is Yangliu Dou (contributes more than 80%).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

With the rapid increase in the numbers of fatalities caused by traffic accidents, the factors which contribute to the vast casualties have been extensively investigated and discussed in greater detail both in academia and industry. According to an American study [1], a large majority of all traffic accidents (96.2%) are due to human errors. In recent years, intelligent vehicles have already raised a considerable amount of concerns due to their higher security, better road utilization, and greatly lower mobility costs. Several prototypes have been developed by universities as well as companies [2-5] and parallel studies and opinions on the impact of these vehicles on society started to be released [6, 7].

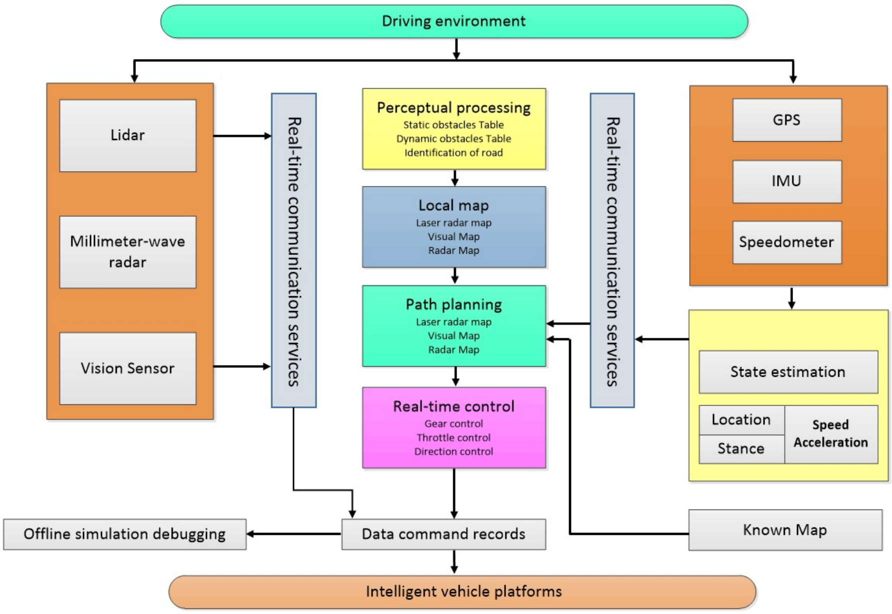


Figure 1.1 Intelligent vehicles' system architecture.

With the rapid growth in the deployment of sensor technology in automobiles [3-5, 8, 9], advanced driver assistance systems (ADAS), one of the fundamental technologies for intelligent vehicles, has become a reality to reduce road fatalities by minimizing human errors in recent years. Among ADAS, such as adaptive cruise control, collision avoidance systems, blind-spot warning systems, lane departure warning systems, and lane keeping assist system, they are all well developed to provide sufficient warning information and help drivers with safe decisions about driving maneuvers. The system architecture of an intelligent vehicle is demonstrated in Figure 1.1.

In order to ultimately reach the maturity level required by mass production [6, 7, 10], intelligent vehicles must be capable of real-time detection and have the ability to adapt to multiple driving conditions, such as all kinds of weather conditions, various types of roads, different sizes of surrounding vehicles, pedestrians, and traffic lights and road signs. In all of the corresponding traffic situations, intelligent vehicles should have the ability of making real-time decisions for improving safety and driving comforts, while maintaining an optimal route towards their destination. In order to meet this need, constructing the advanced driver models using real-time decision-making algorithms based on multi-sensor information fusion, drivers' perception, and recognition of the surrounding driving environment, have become an emerging research topic aiming at offering safer and more reliable solutions for intelligent vehicles.

From a safety standpoint, according to the research on the driver's action model from accidents, the cognition and judgment errors accounted for 84.1% as shown in Figure 1.2, which entails a large part in the total amount of casualties [11]. This also motivates the study on the cognitive mechanism of the driver's perception, high-level decision-making model based on human-vehicle-road, which in return, lays a solid foundation for developing safe and reliable driver models for intelligent vehicles.

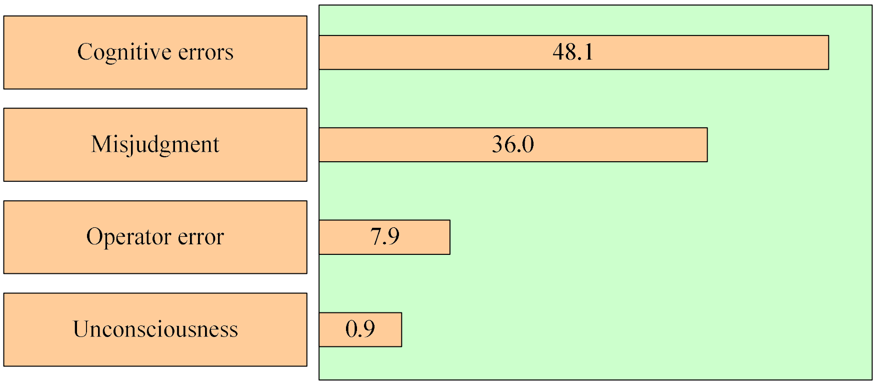


Figure 1.2 The proportion of accidents led by four errors ways.

According to the National Highway Transportation Safety Administration (NHTSA), 500,000 crashes per year in the United States [12, 13] are due to lane changing mistakes of the drivers. Of particular concern, even more crashes per mile happen during the merge maneuvers at highway ramps than at other segments [14]. It has also been noticed that 25% of all police-reported collisions in the US [15] and over 13% of all casualties from road accidents in Europe [16] are related to rear-end collisions. Additionally, as stated in the statistics from the NHTSA in the United States, unsuccessful car-following accidents accounts for 9.2% of the accidents [17]. Thanks to the potential of a lane keeping assist system (LKAS) [18], one of the ADAS, the passenger injury and fatality rate have decreased by about 9% and 15%, respectively. From a safety perspective, it is very crucial to construct safe and reliable driver models for the intelligent vehicles.

Many previous research achievements have been made on the driver models. However, several obstacles, including the unsatisfied accuracy and heavy computational load, still prevent the driver models from being widely adopted in practice. The aforementioned challenges and the ever-increasing demand for the driver behaviour models, such as lane changing and car-following models, necessitate the need for the development of

AI models that offer the possibility of better performance for safety, efficiency, and driving reliability.

As lane changing and car following are the two most frequently encountered driving behaviours for intelligent vehicles, this thesis will mainly focus on the following driver models: lane changing and car-following models.

1.2 Literature review of driver behaviour model

1.2.1 Review of lane changing model

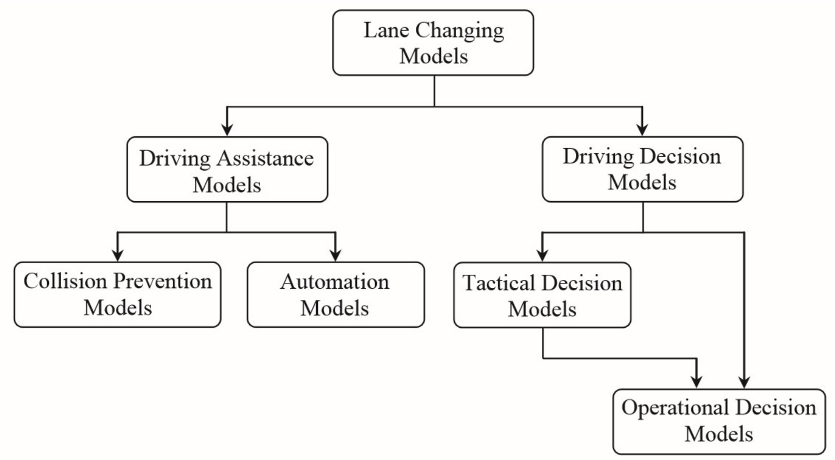


Figure 1.3 Classification of available approaches in lane changing studies [20].

Lane changing models have been developed to capture and model drivers' lane changing behaviours under various traffic conditions, which can be used in macroscopic and microscopic traffic flow research [19-22]. Recognizing drivers' lane changing behaviour and developing models to predict their decisions, both play a crucial role in the development of traffic management strategies. In recent years, numerous studies have been conducted on the lane changing models to enhance road capacity and safety [19, 20]. Different approaches have been categorized, as shown in Figure 1.3.

In exploring the definition range of the lane changing models, it can be classified as driving assistance models and driving decision models. Driving assistance models consist of collision prevention models and automation models, which consider the steering angle and lateral motions of the vehicles in the lane changing behaviours. This kind of model is mainly developed to reduce the danger of the lane changing maneuvers [23-25]. The other kind of lane changing models are driving decision models, which evaluates the intentions, desires, and needs of the driver to make and predict safe decisions in lane changing under different traffic conditions, different situational and environmental characteristics. With regard to the surrounding environment, driving decisions models can be categorized as either tactical or operational [26]. This classification is based on the required time for executing the decisions. The tactical decision models require 5 to 30 seconds while the operational decision models need less than 5 seconds, such as whether or not to accept a gap [27]. An example of operational decision model of lane changing is whether or not to merge into the highway from the on-ramp of the highway entrance, as shown in Figure 1.4.

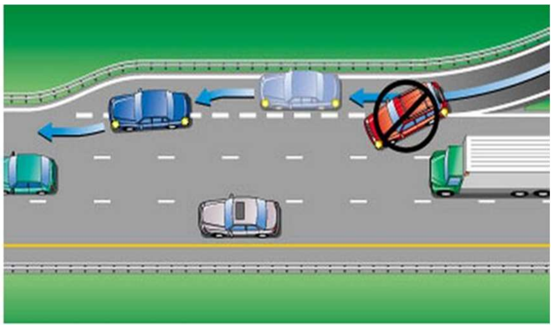


Figure 1.4 Schematic of an example of operational decision lane changing model.

Many studies have focused on the scope for lane changing and associated the lane changing decision of drivers to the surrounding traffic characteristics. They have generated models which can fall into one of two broad categories: rigid mechanistic models and artificial intelligence models [20]. Figure 1.5 demonstrates different models according to these two categories.

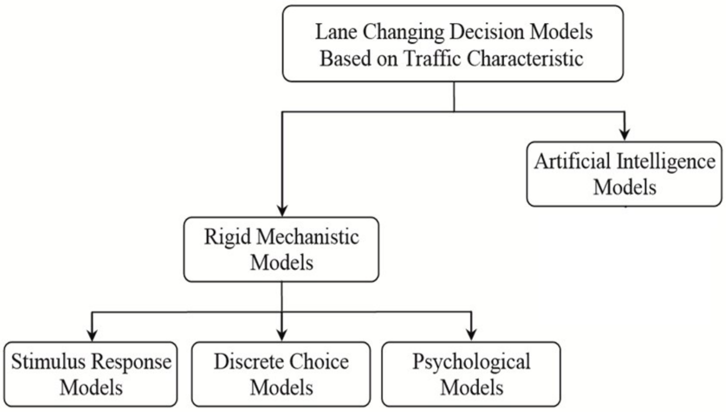

Lane Changing Decision Models Based on Traffic Characteristic
Artificial Intelligence Models
Rigid Mechanistic Models
Stimulus Response Models
Discrete Choice Models
Psychological Models

Figure 1.5 Classification of lane changing decision models based on traffic characteristics [20].

The rigid mechanistic models are composed of stimulus response models [28], discrete choice models [29], and psychological models [30, 31]. Rigid mechanistic models have the advantage of generating a clear relationship between the explanatory independent variables and the target dependant variable. Therefore, the magnitude of these models depends on the exact values of the independent variables. However, they do not usually incorporate the uncertainties and inconsistencies related to drivers' perceptions and decisions [32]. It prevents them from being widely applied in practice since drivers make their decisions based on their imprecise perceptions of the surrounding traffic [33, 34].

In recent years, as AI rolls out, AI based approaches on lane changing, as shown in Table 1.1, have emerged as a new direction and have been widely applied in this field [35- 41].

Table 1.1 AI based lane changing models

Category	Research	Model or Method	Test Platform	Objectives
Lane changing intent analysis	McCall [35]	Sparse Bayesian learning	A modular intelligent vehicle test bed	A driver intent inference system
Lane change intent prediction	Hwan [36]	ANN and SVM	Instrumented vehicle; Driving simulator	Driver lane changing intention model by SVM and ANN
Predictive lane change and merge	Sivaraman [37]	Dynamic program over probabilistic driving map	Instrumented automotive testbed	A general predictive system for lane changing and merge
Lane changing prediction	Dou [38]	SVM and ANN	NGSIM dataset	A lane changing intent classifiers
Lane changing prediction	Gao [39]	Deep neural network Physiological signals	Instrumented vehicle	A lane changing intent prediction model
Mandatory Lane Changing	Hou [40]	Bayes Classifier and Decision Trees	NGSIM dataset	Merge prediction model at highway entrance

A sparse Bayesian classifier for lane changing intent analysis was proposed in [35] based on lane positional information, vehicle parameters, and driver head motions. A novel algorithm was developed using an artificial neural network (ANN) model to augment the basic measurements, and the augmented information was fed to a support vector machine (SVM) algorithm to detect the driver's intention with an accuracy of 92% [36]. A general predictive system for lane changing on the highways and urban areas was presented using dynamic programming over the probabilistic driving map [37]. A combined model of SVM and ANN was introduced for a mandatory lane changing (MLC) prediction [38]. A group-wise convolutional neural network framework was proposed to predict the lane changing [39]. The feature data came from electrocardiogram (ECG), galvanic skin response (GSR), and respiration rate (RR) to best reflect a driver's response to the driving environment. A model was developed for MLC using Bayes classifier and decision trees with an accuracy of 94.3% [40].

In summary, AI based lane changing models have emerged as the new direction of the lane changing model in literature for the safety improvement, most of which have proved to be effective to some degree. However, MLC suggestion at the highway entrance, discretionary lane changing (DLC) intention prediction at the highway are still the main challenges for accurate lane changing models owing to the insufficient understanding of the driving context and situation, incomplete evaluation of drivers' intentions, desires, need for preemptive actions, and high computational cost for complex algorithms.

1.2.2 Review of car-following model

For over half a century, the modeling of longitudinal car-following behaviour of drivers has been the center of focus of many researchers in literature due to its crucial effects on safety. Several car-following models for reflecting a correlation between the ego vehicle (EV) and the preceding vehicle (PV), were proposed to address the challenges, as summarized in Table 1.2.

Table 1.2 Car-following models

Category	Research	Model or Method	Test Platform	Objectives
Pioneering Car-following model	Crooks [41]	Using perceptual field	Simulation	A car perceptual field
Car following	Gipps	Mathematical	Simulation	The effect of driver

Model and traffic flow	[42] Bando [43]	Models		Model to traffic flow
Mathematical model	Yousif [44] Li [45]	Mathematical models verified by experiment	Inductive loop detectors on these motorways	A mathematical model verified by experiment
Mathematical model	Schober [46]	Calibration model by operation test data	Measurements on German freeways and arterials.	A microscopic Car-Following Models
Collision warning	Lee [47]	Identification Gipps model by experiment	Experiment data; Traffic simulator	Driving model for collision warning and avoidance
Fuzzy inference system	Ma [48]	Genetic algorithm and neural-fuzzy system	Advanced instrumented vehicle	A neural-fuzzy framework for car-following model
AI based Car-following model	Wang [49]	Deep neural network	Experiment data	Consider effect of human drivers
AI-fuzzy logic based model	Hao [50]	Fuzzy logic artificial intelligence model	NGSIM dataset	Considering the psychological effects
Fuzzy logic based model	Vicente [51]	A fuzzy logic-based controller	Instrumented prototype vehicle	Consider gap error and control signal

Since the pioneering research work on driver behaviour in [41], many researchers have started to study car-following behaviour. Nevertheless, those early models mainly adopted pure mathematical approaches to express the time-variation of motion parameters for car-following behaviour [42 - 47]. There exist several drawbacks in the aforementioned models. First of all, most of these models are based on kinematic variables such as distance and speed, ignoring other important factors like traffic conditions and thus they cannot fully capture the car-following driving behaviours. The second drawback lies in the fact that too many assumptions and simplifications were made in those studies to model this complex behaviour. Therefore, more efforts are required to improve car-following models for better performance like precise model construction, while avoiding the aforementioned drawbacks.

A general framework expressed by a neural-fuzzy system was introduced to model the driver behaviour from real car-following data by regarding complex driver behaviour as non-linear problems [48]. In [49], a deep neural network-based car-following model was presented with higher accuracy by capturing human drivers' behaviours into the model in a natural and efficient way. A car-following model, composed of a classic stimulus-response framework, an extensive five-layer structure, a fuzzy logic-based inference mechanism, and perception anticipation inference strategy action, was developed to accurately imitate a human driver in [50]. A fuzzy logic-based car-following controller was proposed in [51], which has been tuned to minimize a cost function in order to get a trade-off between a proper car-following gap error and the smoothness of the control signal.

It is worth noting that, regardless of all kinds of model structures and expression forms, the existing car-following models have only taken the driving states of host-lane preceding vehicles into consideration. Nevertheless, the driving states of vehicles in adjacent-lanes also play a critical role in successfully modeling car following. Centering on this concept, preliminary studies have verified that a driver's perception of risk is affected by multiple PVs in his sight on multilane roads [52]. The headways of multi-lanes were researched and the results showed that there existed small headways in multi-lanes compared to bigger headways with single lanes [53]. The research implied the car following will be affected by vehicles from adjacent lanes. With the help of the Grey

System Theory, a method on how to extend the intelligent driver model (IDM) was presented to support multiple lanes and model driver imperfections [54]. However, the model was only based on a behaviour map with a distance of 400 meters, which can be too short for practical applications. In [55], a method was established to describe the car-following behaviour in multi-lanes and the result was not entirely consistent with that of single-lane analysis, which indicates that it is necessary to explore the drivers' behaviour in multi-lanes.

In summary, the shortcomings of existing car-following models in the aforementioned studies would fall into one of two drawbacks: insufficient parameters calibration based on single-lane behaviour and incomprehensive microscopic analytical formulation to fully reflect the relationship between multiple PVs and the subject vehicle.

1.2.3 Problems statement

The research is divided into two categories: lane changing, which is further subcategorized by Mandatory Lane Changing (MLC) and Discretionary Lane Changing (DLC), and Car Following.

For lane changing behaviours, as shown in Figure 1.6, the goal of the research is to provide accurate and efficient models for assisting the driver to eliminate unsafe lane changing maneuvers which account for the vast majority of driving accidents.

For car-following behaviours, the purpose of the research is to develop an accurate and robust car-following model that targets to fill the research gap, by taking the effects of vehicles in adjacent lanes into consideration.

Figure 1.6 Mandatory and Discretionary lane changing [56, 57].

More specifically, the problems addressed in the thesis for lane changing and car-following behaviours are as follows:

Problem 1: Mandatory lane changing suggestion (discussed in Chapter 3)

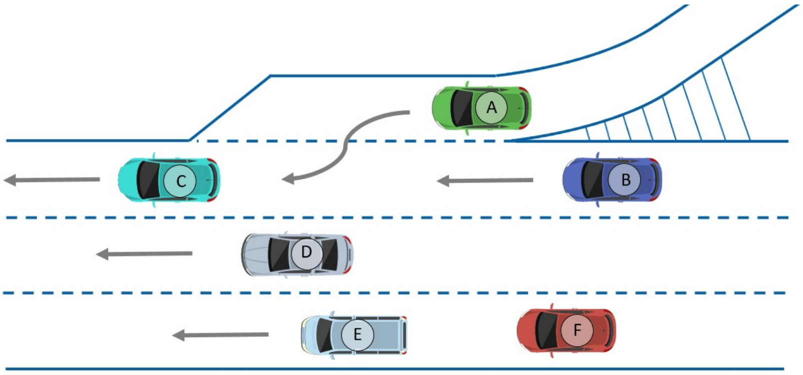


Figure 1.7 Mandatory lane changing scenario.

Vehicle "A" entering the highway from the on-ramp is a typical mandatory lane changing scenario. As shown in Figure 1.7, vehicle "A" will merge from the on-ramp into the highway mainline. The objective of this part is to construct an accurate and efficient mandatory lane changing model for assisting the driver with safe decisions of whether to merge or not.

Problem 2: Discretionary lane changing intention prediction (discussed in Chapter 4)

The discretionary lane changing scenario is shown in Figure 1.8. For vehicle “C”, the intentions, desires, and needs of lane changing of vehicle “A” can help it to avoid dangerous situations. It is obvious that the earlier vehicle “C” can predict the lane changing intent of vehicle “A”, the much safer vehicle “C” can be. The purpose of this part is to develop a discretionary lane changing model that aims at high accuracy and efficiency improvements, by adopting effective feature learning from the lane changing behaviour in conjunction with advanced AI algorithms, which are presented in detail in Chapter 4.

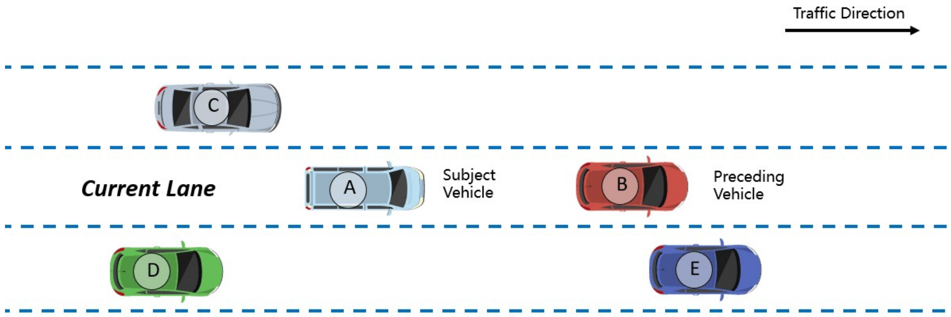


Figure 1.8 The typical multi-car scenario.

Problem 3: Car-following model considering the effect of the vehicle in the adjacent lane (discussed in Chapter 5)

After lane changing execution, the strategies of how vehicle “A” follows the preceding vehicle “B” is worth investigating by considering both effects from the longitudinal direction and the lateral direction, as shown in Figure 1.9. As a matter of fact, the existence of cut-in vehicle “C” has a great influence on the car-following strategy of vehicle “A”, which has been mentioned in part 1.2.2.

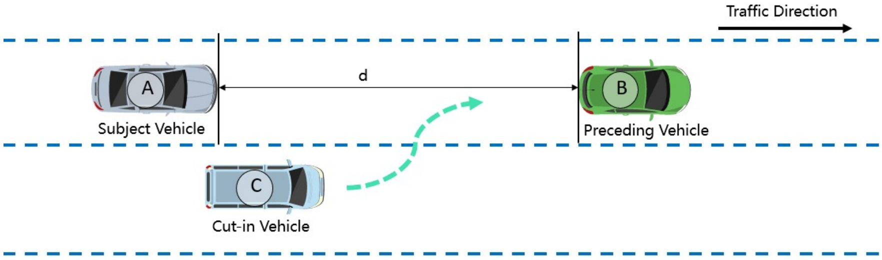


Figure 1.9 Car following with cut-in from the adjacent lane.

This thesis is mainly devoted to seeking solutions to these problems. The proposed models, with both safety and efficiency gains, are anticipated to be a promising solution for providing better prediction performance, thus generating greater road security and a better driving experience in a complex driving environment.

1.3 Contributions and research objectives

1.3.1 Mandatory Lane Changing Suggestion at the On-ramps of Highway

1) An additional gated branch neural network (GBNN) algorithm based on the correlation analysis is proposed. The gated branch provides effective feature learning and explicitly captures the relationship between the surrounding driving environment and the lane changing decision.

2) The proposed GBNN algorithm is used to model the MLC behaviour at the on-ramps of highways with high accuracy of both non-merge and merge events. The accuracy can be as high as 97.7% for non-merge and 96.3% for merge behaviours, respectively. The results are much more accurate than that of other AI algorithms, like CNN (AlexNet), SVM, ANN and so on. Furthermore, the proposed method is

lightweight in computation, compared with the existing deep learning algorithms, and can be applied in ADAS for an efficient MLC suggestion system.

1.3.2 Prediction of Surrounding Vehicles' Discretionary Lane Changing Intention at freeway

1) A recurrent neural network (RNN)-based time series classifier with a gated recurrent unit (GRU) is developed to predict and classify the surrounding vehicles' discretionary lane changing intention in advance and thus it can provide an early notification to the ego-vehicle for driving assistance.

2) The proposed algorithm can predict the surrounding vehicles' lane changing maneuver 0.8 s in advance at a recall and precision of 99.5% and 98.7%, respectively. The model can also predict the lane changing intention 1.6 s in advance with a recall of 92.2%. Furthermore, the proposed method is computationally inexpensive and can be applied in ADAS or autonomous vehicles in real-time applications.

1.3.3 Car-Following Gap Model Considering the Effect of Cut-ins from the Adjacent Lanes

1) A strategic car-following gap is devised by incorporating the probability of cut-ins into the formulation to reveal the practical effect of lateral motions. This model can successfully seek a driver model that operates the vehicle in a way to achieve both safety and efficiency.
2) An overall objective function consisting of the car-following gap and velocity is formulated by considering the safety hazard and the probability of cut-ins by other vehicles. Based on this, seeking the strategic car-following gap translates to finding

the optimal solution that minimizes the overall objective function. With the support of field data, most collected data points are bounded between the 5-th and 95-th percentile curves of the proposed model, which demonstrates the effectiveness of the constructed model. Thus, this model can be a vitally important input to advance the design of driver assistance systems, promoting smooth traffic flow with increased safety.

1.4 Outline

In Chapter 2, data processing is conducted and illustrated. Two kinds of datasets are processed and prepared for further research. What's more, the feature vector to represent the driving environment is constructed for the training and validation of AI algorithms.

In Chapter 3, based on the datasets of Highway 101 and I-80 from NGSIM, the proposed GBNN algorithm has the advantage over other artificial intelligence algorithms (AlexNet, SVM, ANN) in terms of high accuracy and efficiency, to model the MLC behaviours at the on-ramps of highways. The effective feature learning is adopted to explicitly capture the relationship between the surrounding driving environment and the lane changing decision. Furthermore, the proposed method is lightweight in computation, and can be applied in ADAS for an efficient MLC suggestion.

In Chapter 4, a recurrent neural network (RNN)-based time series classifier with a gated recurrent unit (GRU) is proposed to predict and classify surrounding vehicles' discretionary lane changing intention in advance and thus it can provide an early notification to the ego-vehicle for driving assistance. Results reveal that the proposed algorithm can predict the surrounding vehicles' lane changing intention in advance with high accuracy. The method presented in this chapter can be used for ADAS and autonomous driving that promises high traffic safety and effective traffic management.

In Chapter 5, in order to consider both effects from the longitudinal and lateral direction, we propose a strategic car-following gap model to fill this research gap by taking the safety hazard and the probability of cut-ins as a function of the car-following gap and velocity, respectively. Subsequently, an objective function is constructed as the weighted sum of both functions and then the optimal gap is obtained by finding the solution to that overall objective function. The model can give the car-following gap as a function of velocity of the subject vehicle.

In Chapter 6, we summarize the concluding remarks of this thesis and provide a brief introduction on the prospective research directions as future work.

References

[1] J. Engström, E. Johansson, and T. Victor, "Real-Time Driver Drowsiness Detection and Countermeasures: State-of-Art Review II". Gothenburg, Sweden: Volvo Technology Corp., 2001.

[2] T. Folsom, "Energy and Autonomous Urban Land Vehicles". IEEE Technology & Society Magazine, 2012, 31(2): 28-38.

[3] T. Luettel, M. Himmelsbach , H. J. Wuensche, "Autonomous Ground Vehicles Concepts and a Path to the Future", Proceedings of the IEEE, 2012, 100 (Special Centennial Issue): 1831-1839.

[4] N. Suganuma and T. Uozumi, "Development of an autonomous vehicle-System overview of test ride vehicle in the Tokyo Motor Show 2011", Proceedings of the SICE Annual Conf., Japan, Aug. 2012: 215-218.

[5] J. Leonard, J. How, S. Teller, "A perception-driven autonomous urban vehicle", Journal of Field Robotics, Special Issue: Special Issue, 2007, DARPA Urban Challenge, Part III, 2008, 25(10): 727-774.

[6] S. A. Beiker, "Legal aspects of autonomous driving", Santa Clara Law Review 1145, 2012, 52(4):1145-1156.

[7] KPMG llp & CAR (Center for Automotive Research): "Self-driving cars: The next revolution", online report. Available: https://staff.washington.edu/jbs/itrans /self_driving_cars[1].pdf.

[8] R. Dominguez, E. Onieva et al., "LIDAR based perception solution for autonomous vehicles", Proceedings of 11th International Conference on Intelligent Systems Design & Applications., 2011:790-795.

[9] C. Fernandez, M. Gagvilan, “Free space and speed humps detection using Lidar and vision for urban autonomous navigation”, Proceedings of Intelligent Vehicles Symposium, 2012:698-703.

[10] D. Bigman, “Driverless Cars Coming To Showrooms By 2020”, Says Nissan CEO Carlos Ghosn, Forbes On-line.

[11] R. He, G.G. Liu, “Study of Accident Prevention Measures Based on Cognitive Characteristics of Drivers”, Value Engineering, 2011, 30(7):182-183.

[12] Traffic Safety Facts, Department of Transportation, National Highway Traffic Safety Administration, Washington, DC, USA, 2011.

[13] General Estimates System, National Automotive Sampling System, National Highway Traffic Safety Administration, Washington, DC, USA, 2007.

[14] A. T. McCartt, V. S. Northrup, and R. A. Retting, “Types and characteristics of ramp-related motor vehicle crashes on urban interstate roadways in Northern Virginia”, Journal of Safety Research, 2004, 35(1):107-114.

[15] “Traffic safety factor 2011” Available: http://www.nrd.nhtsa.dot.gov/Pubs/8117 54AR.pdf.

[16] K. Malone, I. Wilmink, G. Noecker, K. Robrucker, R. Galbas, and T. Alkim, “Final Report and Integration of Results and Perspectives for market introduction of IVSS,” 2008.

[17] Martin, Peter G., Burgett, August, “Rear-end Collision Events: Characterization of Impeding Crashes”, Proceeding of 1st Human Centered Transportation Simulation Conference, November 2001.

[18] A .Varhelyi, “Dynamic speed adaptation based on information technology - a theoretical background”, Information Technology, 1996.

[19] M. Rahman, M. Chowdhury, Y. Xie, et al, “Review of Microscopic Lane-Changing Models and Future Research Opportunities”. IEEE Transactions on Intelligent Transportation Systems, 2013, 14(4):1942-1956.

[20] S. Moridpour , M. Sarvi , G .Rose, “Lane changing models: a critical review”, Transportation Letters: The International Journal of Transportation Research, 2010, 2(3):157-173.

[21] A. Adelakun and C. Cherry, “Exploring Truck Driver Perceptions and Preferences Congestion and Conflict, Managed Lanes, and Tolls”, Proceeding of the 88th Transportation Research Board Annual Meeting, Washington, DC, CD-ROM, 2009.

[22] C. H. Yang, and A. C. Regan, “Prioritization of Potential Alternative Truck Management Strategies using the Analytical Hierarchy Process”, Proceeding of the 88th Transportation Research Board Annual Meeting. Washington, DC, Transportation Research Board, 2009.

[23] A. Doshi, and M. Trivedi, “A Comparative Exploration of Eye Gaze and Head Motion Cues for Lane Change Intent Prediction”, IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands: 49-54, 2008.

[24] C. Oh, J. Choi, S. Park, “In-depth understanding of lane changing interactions for in-vehicle driving assistance systems”, International Journal of Automotive Technology, 2017, 18(2):357-363.

[25] G. Cesari , G. Schildbach, A. Carvalho, et al, “Scenario Model Predictive Control for Lane Change Assistance and Autonomous Driving on Highways”, IEEE Intelligent Transportation Systems Magazine, 2017, 9(3):23-35.

[26] R. Sukthankar, S. Baluja, and J. Hancock, “Evolving an Intelligent Vehicle for Tactical Reasoning in Traffic”, Proceedings of the International Conference on Robotics and Automation, New Mexico, 519-524, 1997.

[27] V. Alexiadis and R. Hranac, “Next Generation Simulation (NGSIM): Identification and Prioritization of Core Algorithm Categories”, Department of Transportation Federal HighWay Administration (FHWA), 2004.

[28] P. G. Gipps, “A model for the structure of lane-changing decisions”, Transportation Research, Part B (Methodological), 1986, 20(5):403-414.

[29] K. I. Ahmed, “Modeling Drivers’ Acceleration and Lane Changing Behaviour”, Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Master of Science, 1999.

[30] A. Champion, S. Éspié, and J. M. Auberlet, “Behavioural Road Traffic Simulation with ARCHISIM”, Summer Computer Simulation Conference, Orlando, USA, 2001.

[31] A. Champion, M. Y. Zhang, J. M. Auberlet, et al, “Behavioural Simulation: Towards High-Density Network Traffic Studies”, International Conference on Traffic and Transportation Studies”, Guilin, China, 2002.

[32] M. McDonald, J. Wu, and M. Brackstone, “Development of a Fuzzy Logic Based Microscopic Motorway Simulation Mode”, Proceedings of the IEEE Conference on Intelligent Transportation Systems, Boston, U.S.A, 1997.

[33] E. Balal, R. L. Cheu , T.S. Gyan, “A binary decision model for discretionary lane changing move based on fuzzy inference system”, Transportation Research Part C Emerging Technologies, 2016, 67:47-61.

[34] Y. Li, Z. Li, S. Peeta, et al, “Non-lane-discipline-based car-following model considering the effects of two-sided lateral gaps”, Nonlinear Dynamics, 2015, 80(1-2):227-238.

[35] J. C. Mccall, D. P. Wipf , M. M. Trivedi, et al., “Lane Change Intent Analysis Using Robust Operators and Sparse Bayesian Learning”. IEEE Transactions on Intelligent Transportation Systems, 2007, 8(3):431-440.

[36] K. Il-Hwan , B. Jae-Hwan , P. Jooyoung, et al. “Prediction of Driver’s Intention of Lane Change by Augmenting Sensor Information Using Machine Learning Techniques”, Sensors, 2017, 17(6):1350.

[37] S. Sivaraman, and M. M. Trivedi, “Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance”, IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5): 2063-2073.

[38] Dou, Y., Yan, F., Feng, D., “Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers”, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM, pp. 901-906, 2016.

[39] J. Gao , Y. L. Murphey, H. Zhu, “Multivariate time series prediction of lane changing behaviour using deep neural network”, Applied Intelligence, 2018, 48(10):3523-3537.

[40] Y. Hou, P. Edara , C. Sun, "Modeling Mandatory Lane Changing Using Bayes Classifier and Decision Trees", IEEE Transactions on Intelligent Transportation Systems, 2014, 15(2):647-655.

[41] L. E. Crooks, "A theoretical field-analysis of automobile driving", The American Journal of Psychology, 1938, 51(3):453-471.

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

[43] M. Bando, K. Hasebe, K. Nakanishi, et.al., "Phenomenological Study of Dynamical Model of Traffic Flow", Journal de Physique I, 1995, 5(11):1389-1399.

[44] S. Yousif, A. Obaedi, "Close following behaviour: Testing visual angle car following models using various sets of data", Transportation Research Part F: Traffic Psychology & Behaviour, 2011, 14(2):0-110.

[45] L. Li, X. Chen, Z. Li, "Asymmetric stochastic Tau Theory in car-following", Transportation Research Part F: Traffic Psychology and Behaviour, 2013, 18(5):21-33.

[46] M. Schober, P. Wagner, R. Kühne, et al., "Calibration of Microscopic Car-Following Models on the Basis of the Intelligent Cruise Control Field Operation Test Data", ITS Safety and Security Conference. 2004, 191-196.

[47] K. Lee, H. Peng, "Identification and verification of a longitudinal human driving model for collision warning and avoidance systems", International Journal of Vehicle Autonomous Systems, 2004, 2(1/2):3-17.

[48] X. Ma, "A neural-fuzzy framework for modeling car-following behaviour", IEEE International Conference on Systems, Man and Cybernetics, 2006, 2:1178-1183.

[49] X. Wang, R. Jiang, L. Li, et al., "Capturing Car-Following Behaviours by Deep Learning". IEEE Transactions on Intelligent Transportation Systems, 2017:1-11.

[50] H. Hao, W. Ma, H. Xu, "A fuzzy logic-based multi-agent car-following model", Transportation Research Part C: Emerging Technologies, 2015: S0968090X150 03472

[51] M. Vicente, M. Marouf, J. P. Rastelli, et al., "Low-Speed Cooperative Car-Following Fuzzy Controller for Cybernetic Transport Systems", IEEE International Conference on Intelligent Transportation Systems, 2014.

[52] J. Weng, Q. Meng, "Effects of environment, vehicle and driver characteristics on risky driving behaviour at work zones", Safety Science, 2012, 50(4):1034-1042.

[53] S. Y. Fang, F. Li, "Research on Abnormal Car-Following Behaviour of Multi-Lane". Advanced Materials Research", 2012, 468-471:2989-2992.

[54] H. X. Ge, X. P. Meng, H. B. Zhu, et al., "Feedback control for car following model based on two-lane traffic flow", Physica A: Statistical Mechanics and its Applications, 2014, 408(15):28-39.

[55] Y. Chenfei, W. Jianqiang, "Drivers' car-following correlative behaviour with preceding vehicles in multilane driving", IEEE Intelligent Vehicles Symposium (IV) June 8-11, 2014, Dearborn, Michigan, USA.

[56] http://www.mogi.bme.hu/TAMOP/jarmurendszerek_iranyitasa_angol/book.html

[57] https://driversed.com/driving-information/driving-techniques/merging.aspx.

Chapter 2

Data Processing and Feature Extraction

2.1 Data source

2.1.1 NGSIM I-80 and U.S. 101 Datasets

To evaluate the model for highway lane changing prediction, the dataset from the Federal Highway Administration's Next Generation Simulation (NGSIM) program was employed [1]. Specifically, the Interstate 80 Freeway and U.S. Highway 101 datasets were used in this thesis, which were collected on eastbound I-80 in the San Francisco Bay area in Emeryville, CA, on April 13th, 2005 and southbound US 101, also known as the Hollywood Freeway, in Los Angeles, CA, on June 15th, 2005, respectively. Both datasets are freely accessible on the NGSIM website [1].

Figure 2.1 displays a schematic illustration of the vehicles' available trajectory on the Interstate 80 Freeway. The data represents travel on the northbound direction of Interstate 80 in Emeryville, California. The data was collected using video cameras mounted on a 30-story building, Pacific Park Plaza, which is located in 6363 Christie Avenue and is adjacent to the interstate freeway I-80. The study area is approximately 500 meters (1,640 feet) in length, with an on-ramp at Powell Street, as marked by the rhombus solid line in Figure 2.1. The on-ramp at Ashby Avenue is just downstream of the study area. Lane numbering from 1 to 6 is incremented from the left-most (the high-occupancy vehicle, HOV) lane to the right lane. The Interstate 80 dataset consists of six freeway lanes. This vehicle trajectory dataset provides the precise location of each vehicle on different lanes within the study area every one-tenth of a second. The data was collected within the peak traffic period, in order to represent the buildup of congestion, or the transition between uncongested and congested conditions, and full congestion during the peak period. The dataset contains a total of 45 minutes that is segmented into three 15-minute periods: 4:00 p.m. to 4:15 p.m.; 5:00 p.m. to 5:15 p.m.; and 5:15 p.m. to 5:30 p.m. on April 13th, 2005.

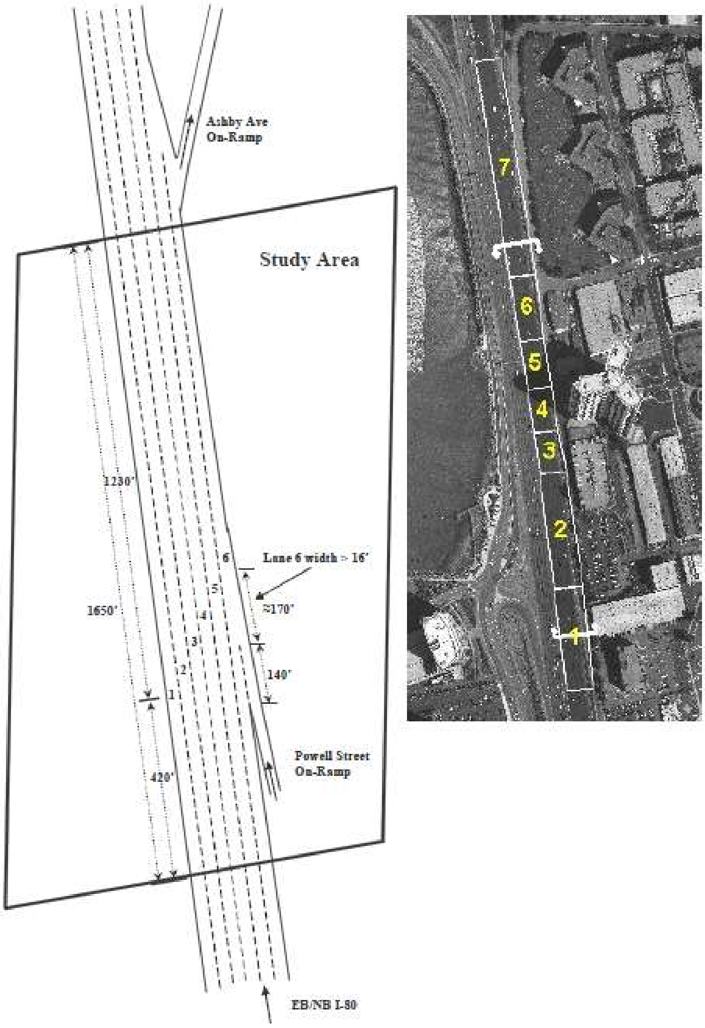


Figure 2.1 Study area schematic and camera coverage of NGSIM I-80 [1].

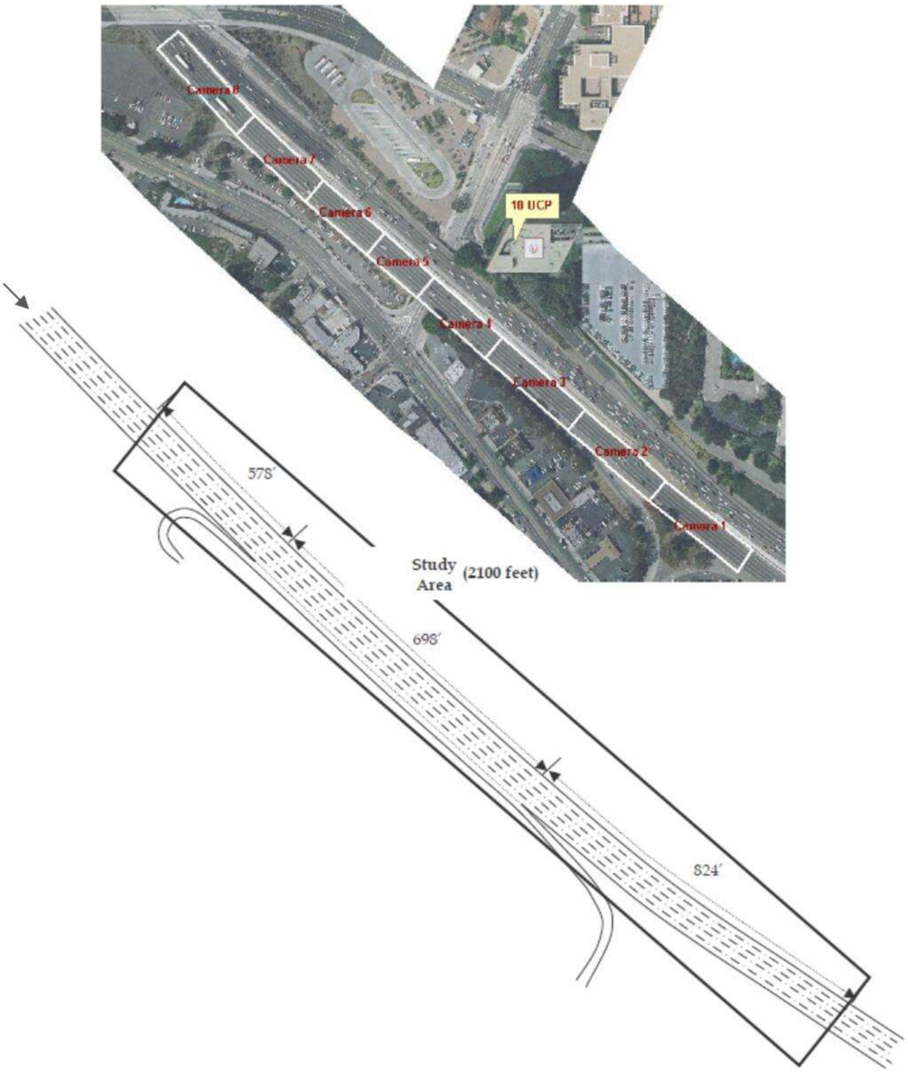


Figure 2.2 Study area schematic and camera coverage of U.S. 101 [1].

Figure 2.2 shows the schematic and the optical image of the study area of Highway 101, which is approximately 640 meters (2100 feet) in length with five mainline lanes. An auxiliary lane is located through a portion of the corridor between the on-ramp at Ventura Boulevard and the off-ramp at Cahuenga Boulevard. Lane numbering is incremented from the left-most lane in the driving direction. The video dataset was collected using eight video cameras with camera 1 recording the southernmost and camera 8 recording the northernmost section of the study area and the others recording the rest of the areas as marked in Figure 2.2. A total of 45 minutes of data is segmented into three 15 minute periods: 7:50 a.m. to 8:05 a.m.; 8:05 a.m. to 8:20 a.m.; and 8:20 a.m. to 8:35 a.m. on June 15, 2005 for better representation of the buildup of congestion, or the transition between uncongested and congested conditions, and full congestion during the peak period.

The data structure of the NGSIM dataset is shown in Table 2.1. The vehicle information and trajectory data were obtained with a camera recording 10 frames every second, which include vehicle ID, lateral coordinate, longitudinal coordinate, vehicle length, vehicle velocity, vehicle acceleration, spacing, and so on. Detailed information of this data structure can be found from NGSIM [1]. Through analysis of the vehicles' trajectory, we are able to identify the behaviours such as merge or not, lane changing, and car following.

Table 2.1 Data structure of NGSIM dataset

1	**2**	**3**	**4**	**5**	**6**
Vehicle ID	Frame ID	Total Frames	Global Time	Local X	Local Y
7	**8**	**9**	**10**	**11**	**12**
Global X	Global Y	Vehicle Length	Vehicle Width	Vehicle Class	Vehicle Velocity
13	**14**	**15**	**16**	**17**	**18**
Vehicle Acceleration	Lane Identification	Preceding Vehicle	Following Vehicle	Space Headway	Time Headway

Figure 2.3 shows the vehicle detection and tracking process of the NGSIM dataset [1]. Vehicle trajectory data was transcribed from the video data using a customized software developed by NGSIM. The program can automatically detect and track most vehicles on the lanes with the assist of video recording and transcribe the vehicles' trajectory into a database.

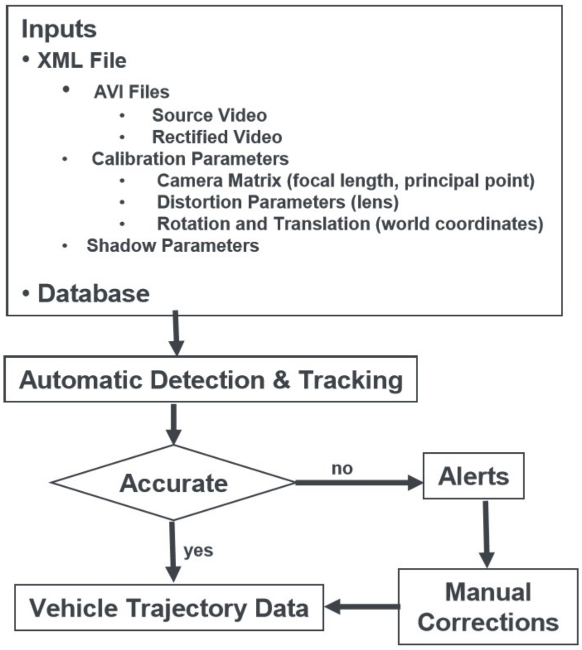


Figure 2.3 Vehicle detection and tracking process [1]. The software identified vehicles in a user-defined detection zone that was tracked by eight cameras.

2.1.2 Experiment data

Another data source comes from Tsinghua University. The experimental vehicle platform is shown in Figure 2.4, which is equipped with a driver recorder (DR), industrial personal computer (IPC), frontal view camera and vehicle millimeter wave radar. The sampling frequencies of the DR, camera, and wave radar are 10 Hz, 30 Hz, and 15 Hz, respectively. Given that the sampling frequency of the DR, cameras, and millimeter-wave radar are not the same, we developed a synchronization program to synchronize collective information into 10 Hz. Besides the equipment shown in Figure 2.4, the test vehicle was also equipped with radars and sensors to collect information about vehicle speed, acceleration, accelerator pedal depression, and brake pressure. These signals were recorded at a frequency of 10 Hz.

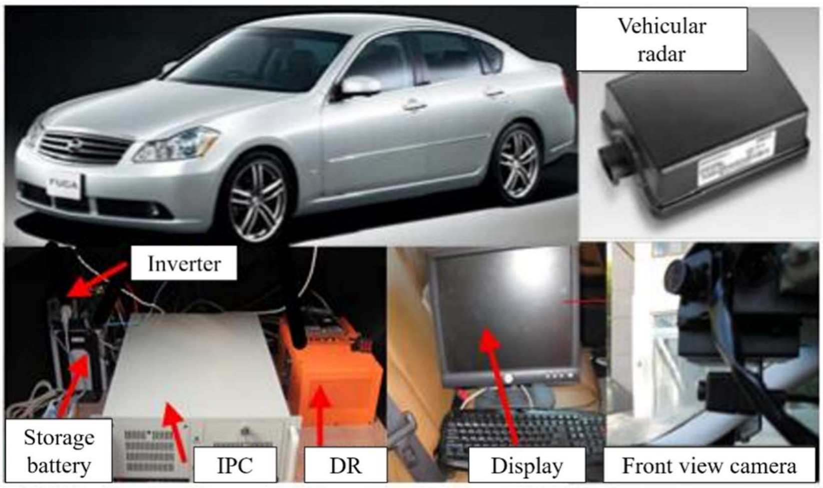


Figure 2.4 Experimental platform and devices.

The Fourth Ring traffic lane of Beijing, China was selected as the experimental section as shown in Figure 2.5. It is a highway that contains a minimum of 4 lanes in each driving direction. The vehicle started from Tsinghua University (point 0) and entered Fourth Ring Road (section 1) from Wanquanhe (point 1) to the Xiaocun (point 2), then went back along section 2 and returned to Zhongguancun (point 3). Finally, the vehicle arrived at Tsinghua University (point 0). The tested distance is approximately 130 km. The driving speed limit is 80km/h for the test road.

The experiments were performed to avoid peak traffic congestion time and no trucks were involved during our experiments. Experimental subjects are 12 non-professional drivers including 10 men and 2 women. Their average age is 37 years old with the standard deviation of 13 years. Their average driving experience is 15 years with a standard deviation of 11 years to promise the effectiveness of our experimental database.

.

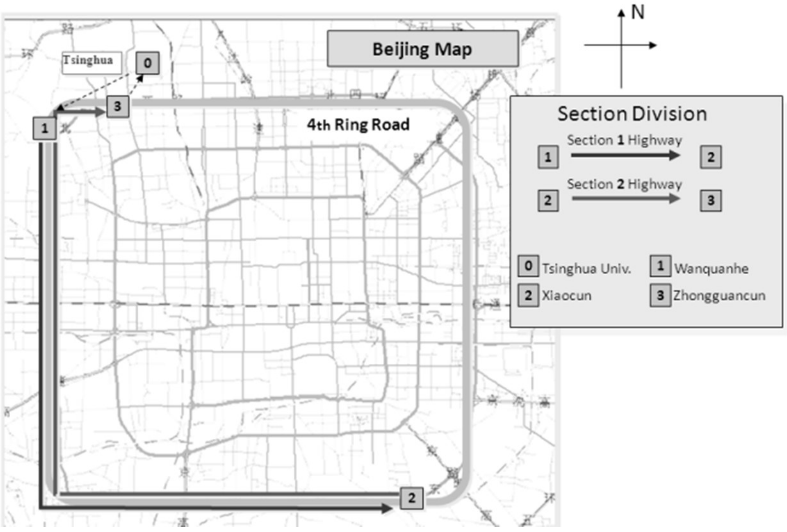


Figure 2.5 The experiment route. The arrow and close number from 1 to 3 indicate the moving direction and the route of the tested vehicles.

2.2 Data analysis

2.2.1 Data filtering

Earlier research has shown that the trajectory data from NGSIM is unfiltered and has some noise [2-4]. A moving average filter was used to smooth the data. In this paper, the speed and acceleration rate are smoothed by the moving average filter. We designed and applied a moving average filter to all trajectories before any further data analysis. The moving average filter as shown in equation (2.1) plays a vital role as a low pass filter. Figure 2.6 and Figure 2.7 show the unfiltered and filtered velocity and acceleration from the I-80 database, respectively. It can be found that after filtering the data becomes smoother. All the data was filtered before being used for our studies.

$$x[i] = \frac{1}{M}\sum_{j=-(\mathrm{M}-1)/2}^{(M-1)/2} y[i+j] \qquad (2.1)$$

Where M represents the window size of the filter and is an odd number. y represents the variable that needs to be filtered, x represents the filtered value; i represents the center time frame when the vehicle was pictured, j represents the surrounding time frames around the center time frame i in the window size of M.

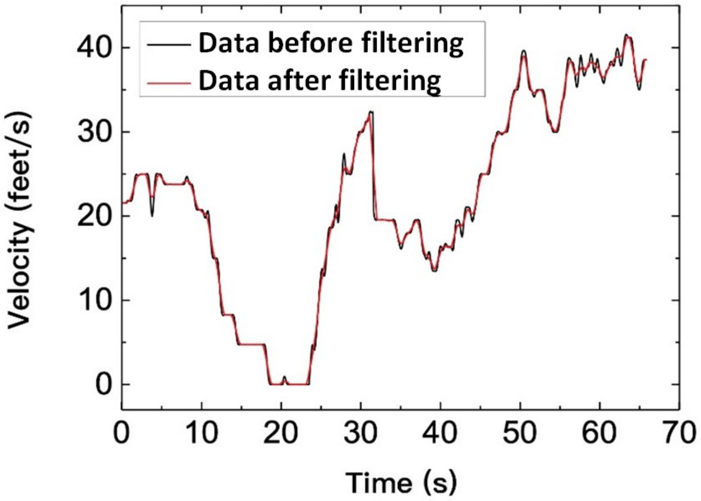


Figure 2.6 Comparison of unfiltered and filtered velocity data. The black curve is the data before filtering, while the red curve is the data after filtering.

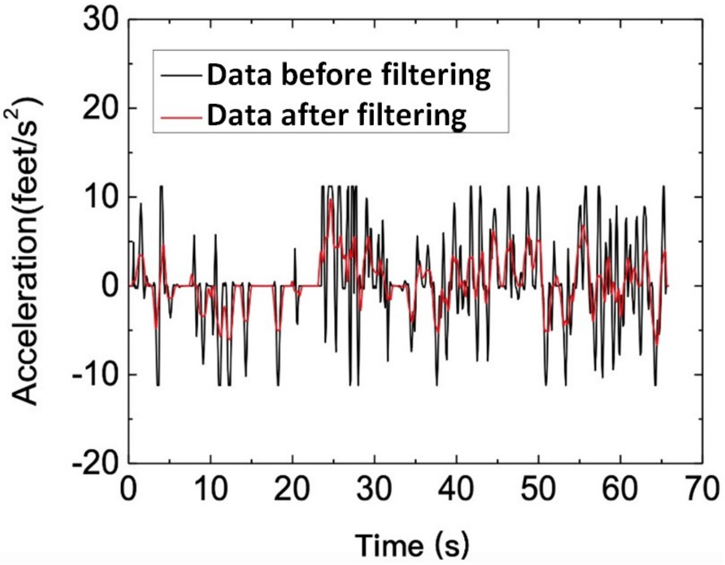
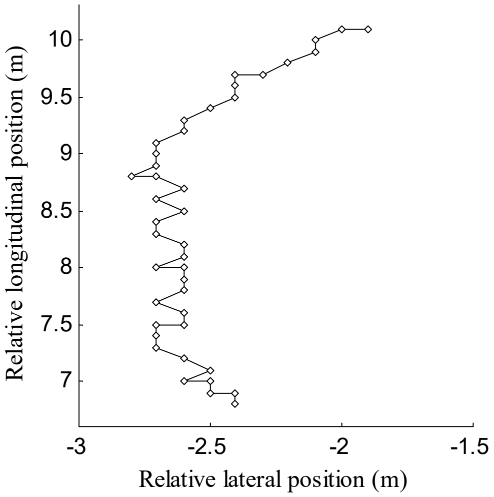


Figure 2.7 Comparison of unfiltered and filtered acceleration data.

2.2.2 Data pre-treatment

In each radar signal cycle, the information of relative lateral positions is the basis to judge which lane the target is in. Among the vehicles ahead of the ego vehicle (EV), the preceding vehicle (PV) is the one with the smallest longitudinal distance captured by the vehicular radar. The relative lateral position of the left-lane preceding vehicle (LPV) should satisfy: $d_{x-l} \in [-p_l - w_r, -p_l]$; the host-lane preceding vehicle (HPV) should satisfy: $d_{x-h} \in [-p_l, p_r]$; and the right-lane preceding vehicle (RPV) should satisfy: $d_{x-r} \in [p_r, p_r + w_r]$; w_r is the width of the lane, p_l and p_r are the left and right lane line positions.

Although the distance can be accurately measured using standard vehicular radar, the measurements produce signal errors due to noise and digital effects. For the cut-in vehicle, estimating its lateral velocity is the key information to judge its intention to cut in or not. Figure 2.8 shows that LPV switches to the lane of EV, and the measured value of the lateral distance fluctuates. This motion exhibited by the LPV in the lateral direction may be confused with the minor lateral changes due to noise.



0

Figure 2.8 Position of original radar target information for a cut-in vehicle from the left lane.

Considering the difficulties in tracking the vehicles in adjacent lanes, two requirements should be satisfied to make data applicable and accurate for our studies:

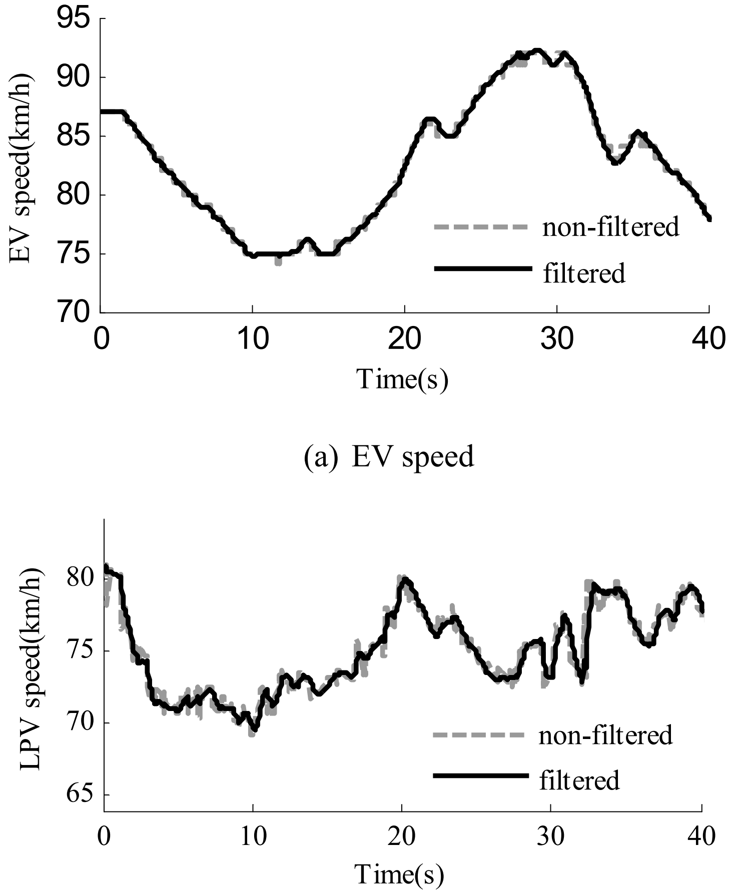
(1) Accumulated errors should not be generated. The estimated result of the current moment should not be affected by the estimated result of the previous moment but only affected by previous observations.

(2) Both accuracy and effectiveness can be obtained. To estimate the target position and identify the driving risk in real-time, the target tracking algorithm should rely on a small amount of real-time data rather than a large amount of data.

To solve these problems, an adjacent lane vehicle tracking method based on the idea of total least squares is implemented. The detailed algorithm can be found in Ref.[5]. The parameters used in the car-following model is shown in Table 2.2.

Table 2.2 Collected Parameters

Symbol	Name	Unit	Meaning
T	Time	s	Time
k	Number of sampling points		Number of sampling points
B	Brake signal		0=non-braking, 1=braking
v_{ego}	EV speed	m/s	Speed of EV
$v_{rel\text{-}l}$, $v_{rel\text{-}h}$, $v_{rel\text{-}r}$	Relative speed of PV	m/s	Relative speed of LPV/HPV/RPV, positive value means approaching
$d_{y}\text{-}_{l}$, $d_{y}\text{-}_{h}$, $d_{y}\text{-}_{r}$	Relative distance of PV	m	LPV/HPV/RPV
$d_{x}\text{-}_{l}$, $d_{x}\text{-}_{h}$, $d_{x}\text{-}_{r}$	Lateral position of PV	m	LPV/HPV/RPV, positive value in the right side of EV
p_l, p_r	Lane line position	m	Distance of EV longitudinal centerline from the left and right lane lines
N_d	Number of the tested driver		1-12



(a) EV speed

(b) LPV speed

Figure 2.9 Estimated speed after Kalman Filtering: (a) EV speed; (b) LPV speed.

Noise and glitches are inevitable in the speed information from a Controller Area Network (CAN), thus the velocity should be filtered in order to ensure the reliability of the data analysis results.

In order to exclude the impact of noise in the speed of EV, LPV, HPV, and RPV, we used the Kalman filter to process the data collected from EV and the three PVs. The Kalman filter is widely used in vehicular radars and the machine vision for object state estimation. It can quantitatively estimate the position and motion of the target via a sequence of sensor readings. The Kalman filter is capable of tracking targets in real-time, and the basic idea is as follows. Based on the estimated value of the previous moment and the observed value of the current moment, it updates the state variables of the current moment based on the state space model of the signal and noise [6]. Figure 2.9(a) is an example of filtering EV speed in a series. Figure 2.9(b) is an example of filtering LPV speed in a series. It is clearly shown that after filtering, data jitter and glitches in the relative speed from the signal have been removed and the signal quality is improved.

2.3 Driving environment and feature vector construction

To construct driver models, the feature vector, which can represent different driving conditions, must be obtained firstly. According to the information provided by the NGSIM dataset, a feature vector was extracted to represent the driving environment, as shown in Table 2.3. We take the left lane changing behaviours shown in Figure 2.10 as an example. The right lane changing behaviour has the same feature vector as shown in Table 2.3. Since *THW* is a dependent variable and can be expressed by *d_lead* and *v_ego*, in order to avoid the redundancy of the input vector, it is not suitable to be used as a separate feature variable. Therefore, the total number of the variables in the feature vector is 23.

Table 2.3 Feature vector represents the driving environment

Number	**1**	**2**	**3**	**4**	**5**	**6**
Subject vehicle	*time*	*y_ego*	*Δx_ego*	*v_ego*	*a_ego*	*THW*
Number	**7**	**8**	**9**	**10**	**11**	**12**

Subject vehicle	*d_lead*	*Δv_lead*	*Δa_lead*	*d_lag*	*Δv_lag*	*Δa_lag*
Number	**13**	**14**	**15**	**16**	**17**	**18**
Left lane vehicle	*d_lead_l*	*Δv_lead_l*	*Δa_lead_l*	*d_lag_l*	*Δv_lag_l*	*Δa_lag_l*
Number	**19**	**20**	**21**	**22**	**23**	**24**
Right lane vehicle	*d_lead_r*	*Δv_lead_r*	*Δa_lead_r*	*d_lag_r*	*Δv_lag_r*	*Δa_lag_r*

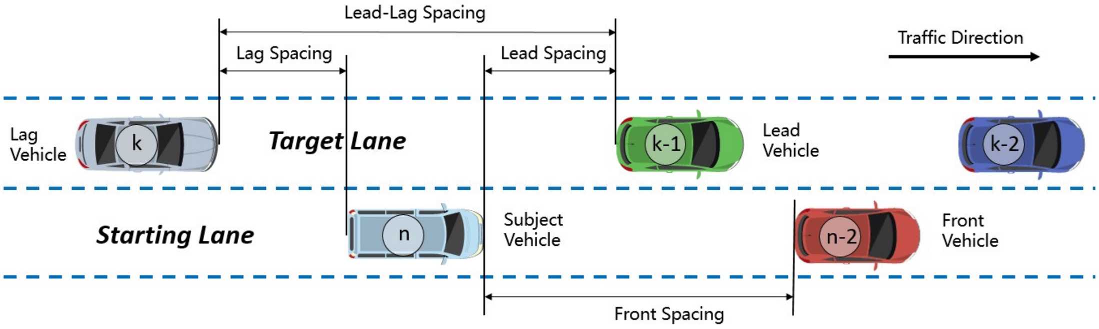


Figure 2.10 The feature variables represent the driving environment.

The meanings of the feature variables are as follows:

1. *time*: the duration of the frame ascending from the start point in units of 100 ms.
2. *y_ego*: the longitudinal coordinate Y of the subject vehicle.
3. *Δx_ego*: the lateral distance of the subject vehicle with respect to the centre line of the lane, shown in Figure 2.11.
4. *v_ego*: velocity of the subject vehicle.
5. *a_ego*: acceleration of the subject vehicle.
6. Time Headway (*THW*): headway provides the time to travel from the front-center

of a vehicle (at the speed of the vehicle) to the front-center of the preceding vehicle.

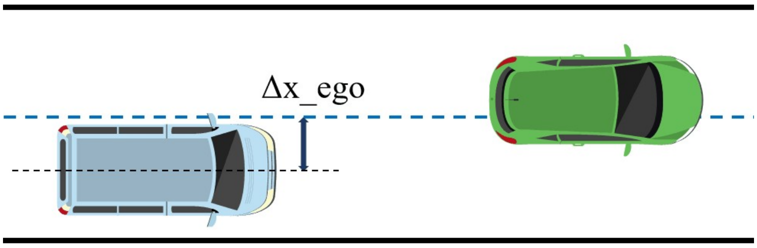


Figure 2.11 Lateral distance (*Δx_ego).*

7. *d_lead*: spacing provides the distance between the front-center of a vehicle to the front-center of the preceding vehicle.
8. *Δv_lead*: speed difference between the lead vehicle in the current lane and the subject vehicle.
9. *Δa_lead*: acceleration difference between the lead vehicle in the current lane and the subject vehicle.
10. *d_lag*: longitudinal coordinate Y difference between the lag vehicle in the current lane and the subject vehicle.
11. *Δv_lag*: speed difference between the lag vehicle in the current lane and the subject vehicle.
12. *Δa_lag*: acceleration difference between the lag vehicle in the current lane and the subject vehicle.
13. *d_lead_l*: longitudinal coordinate Y difference between the lead vehicle in the left lane and the subject vehicle.
14. *Δv_lead_l*: speed difference between the lead vehicle in the left lane and the subject vehicle.

15. *Δa_lead_l*: acceleration difference between the lead vehicle in the left lane and the subject vehicle.
16. *d_lag_l*: longitudinal coordinate Y difference between the lag vehicle in the left lane and the subject vehicle.
17. *Δv_lag_l*: speed difference between the lag vehicle in the left lane and the subject vehicle.
18. *Δa_lag_l*: acceleration difference between the lag vehicle in the left lane and the subject vehicle.
19. *d_lead_r*, 20.*Δv_lead_r*, 21.*Δa_lead_r*, 22.*d_lag_r*, 23.*Δv_lag_r*, 24.*Δa_lag_r* have nearly the same meaning as variables 13-18, but is relative to the right lane.

The feature vector (*FV*) can be represented by these variables in equation (2.2).There are 23 variables in the feature vector to characterize the driving environment. It must be noted that during different driver model construction, the size of the feature vector will change according to the specific problems that tend to be solved. Some variables may contain duplicate information, which may be omitted during different research. What's more, with more sensors, the size of the feature vector can be expanded.

$$\begin{gathered} FV = \{\, time, y\_ego, \Delta x\_ego, v\_ego, a\_ego, d\_lead, \\ \Delta v\_lead, \Delta a\_lead, d\_lag, \Delta v\_lag, \Delta a\_lag, \\ d\_lead\_l, \Delta v\_lead\_l, \Delta a\_lead\_l, d\_lag\_l, \Delta v\_lag\_l, \Delta a\_lag\_l, \\ d\_lead\_r, \Delta v\_lead\_r, \Delta a\_lead\_r, d\_lag\_r, \Delta v\_lag\_r, \Delta a\_lag\_r\} \end{gathered} \tag{2.2}$$

According to the lane changing label method, the distribution of some variables under different behaviours (lane changing or lane keeping) will be illustrated in the following.

Figure 2.12 and Figure 2.13 show some typical variables used in the feature vector. From the figures, it can be found that the *y_ego*, *Δx_ego*, *a_ego*, *Δv_lead*, *d_lead, d_lead_l*, and *Δv_lead_l* of lane changing behaviours are different from those of lane keeping behaviours.

For example, Figure 2.12(b) shows the distribution of *Δx_ego*. The left figure represents the cases of lane keeping behaviours while the right figure represents the lane changing behaviours. It is obvious that the *Δx_ego* of lane keeping behaviour and that of lane changing behaviour are much different from each other. For this reason, the variable can be used as the feature variable.

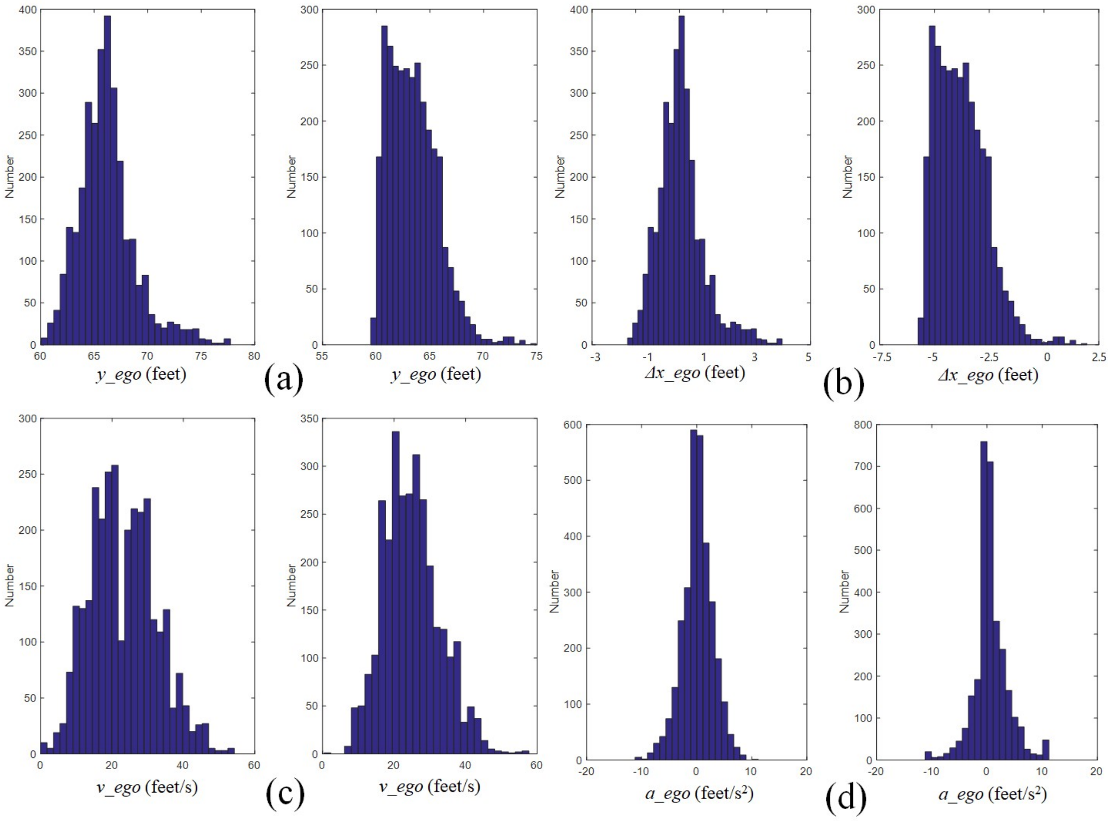


Figure 2.12 The distribution of *y_ego* (a), *Δx_ego* (b), *v_ego* (c), *a_ego* (d)

for lane keeping (left) and lane changing (right).

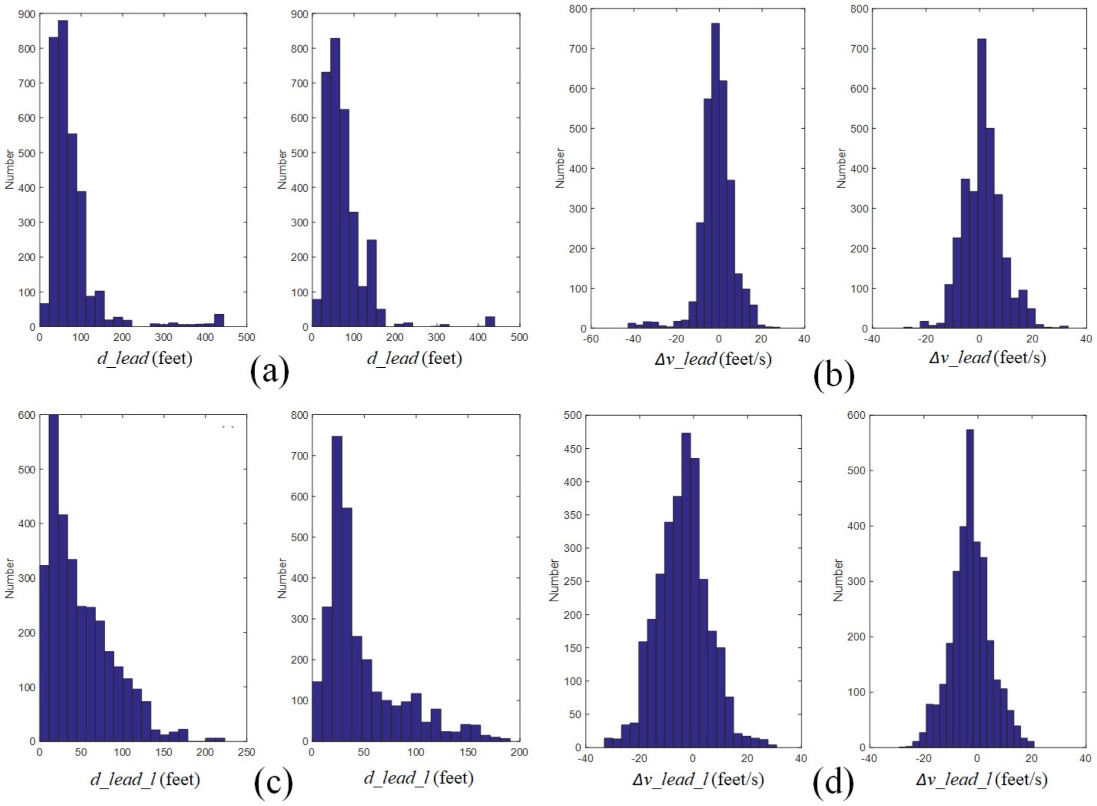


Figure 2.13 The distribution of *d_lead* (a), *Δv_lead* (b), *d_lead_l* (c), *Δv_lead_l* (d) for lane keeping (left) and lane changing (right).

It is noteworthy that the variable of LK and LC with different distributions may indicate that this variable can be used for distinguishing two behaviours, but if the variable of LK and LC has nearly the similar distribution, like the *a_ego* as shown in Figure 2.12, it may also be an important variable in the feature vector. The reason is that the distribution of the feature variable can not reflect the time information. In other words, a similar distribution can be obtained by the same value at different times.

2.4 Conclusions

In this chapter, the data sources are discussed in detail. The data in the thesis comes from two datasets: NGSIM, the Highway 101 and I-80 data, and experiment data from Tsinghua University. The data was filtered by the moving average filter, Kalman filter, and the total least squares method. Furthermore, the distributions of the variables are given in detail for lane keeping and lane changing behaviours, which will be used for the research in the following chapters. After processing the data, the feature vector constructed by 23 variables will be extracted to characterize the driving environment.

References

[1] Next Generation Simulation Fact Sheet, https://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm.

[2] A. Khodayari, A. Ghaffari, R. Kazemi, and R. Braunstingl, “A Modified Car-Following Model Based on a Neural Network Model of the Human Driver Effects”, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 2012, 42(6):1440-1449.

[3] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, “Estimation of vehicle trajectories from observed discrete positions and next-generation simulation program (NGSIM) data,” Annual Meeting, Transportation Research Board, Washington, DC, USA, 2009.

[4] C. Thiemann, M. Treiber, and A. Kesting, “Estimating acceleration and lane-changing dynamics based on NGSIM trajectory data”, Transportation Research Record Journal of the Transportation Research Board, 2008, 2088(2088):90-101.

[5] Wang, Jianqiang , et al. “A Forward Collision Warning Algorithm With Adaptation to Driver Behaviors.” IEEE Transactions on Intelligent Transportation Systems 2016, 17.4:1157-1167.

[6] Alexander, T. . “Book Review - Introduction to random signal analysis and kalman filtering.” IEEE Communications Magazine 2003, 22.10:34-34.

Chapter 3

A Gated Branch Neural Network for Mandatory Lane Changing Suggestion at the On-ramps of Highway

3.1 Citation and Main Contributor

Yangliu Dou, Yihao Fang, Chuan Hu, Rong Zheng, Fengjun Yan, "A Gated Branch Neural Network for Mandatory Lane Changing Suggestion at the On-ramps of Highway." *IET Intelligent Transport Systems*.2019, 13(1):48-54.

The main contributor to this paper is the first author - Yangliu Dou (contributes more than 80%).

3.2 Copyright

Published with permission from *IET Intelligent Transport Systems.*

3.3 Abstract

A gated branch neural network (GBNN) is proposed for modeling mandatory lane changing (MLC) behaviour at the on-ramps of highways. It provides a core algorithm for an MLC suggestion system for advanced driver assistance systems (ADAS), in which the main challenges are efficient decision and high prediction accuracy of both non-merge and merge events. The GBNN algorithm employs a gated branch based on correlation analysis, scaled exponential linear units (SeLU) activation function, and Adam optimizer. The algorithm has been evaluated using the real-world datasets of U.S.

Highway 101 and Interstate 80 from the Federal Highway Administration's Next Generation Simulation (NGSIM). Input features are extracted from NGSIM and pre-processed by standardization and principal component analysis. TensorFlow framework and Python are used as the development platform. Results show that the proposed GBNN algorithm with the Pearson correlation method has values of 97.7%, 96.3%, and 0.990 for non-merge accuracy, merge accuracy, and receiver operating characteristic score, respectively. It outperforms conventional binary classifiers for MLC applications in accuracy and is more efficient computationally than a convolutional neural network (AlexNet) of deep learning algorithm. Owing to its compact architecture, the GBNN provides high accuracy and efficiency, demonstrating promising usage as an MLC suggestion system in ADAS.

3.4 Introduction

Advanced driver assistance systems (ADAS), such as adaptive cruise control [1], collision avoidance [2], and lane changing prediction systems [3], have attracted much attention in recent years owing to their improvements for driving safety and efficiency [4, 5]. Among these, lane changing is an important driver behaviour, which is prone to accidents. According to Ref. [6], most traffic accidents are caused by drivers' mistakes and nearly 1.6% of fatal crashes happen during lane changing maneuvers.

Lane changing maneuvers can be classified as mandatory lane changing (MLC) and discretionary lane changing (DLC) [7]. An MLC occurs when a driver is forced to leave the current lane, for instance merging to the main lane of the highway from an on-ramp or taking an exit to an off-ramp. DLC is not mandatory and happens mostly when the driver is not satisfied with the situation of the current lane and wishes to change to an adjacent lane.

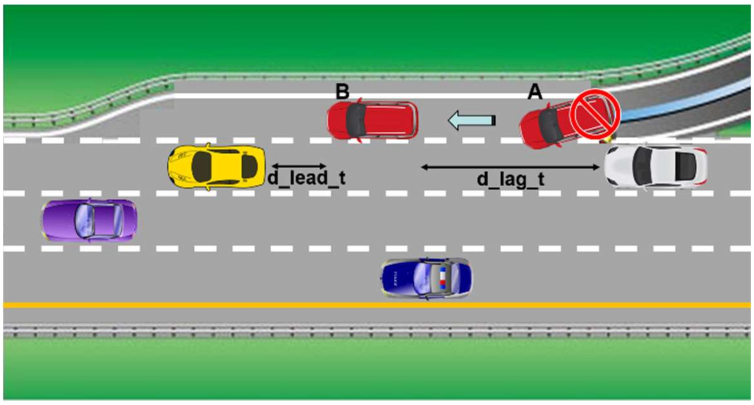


Figure 3.1 Schematic illustrating mandatory lane changing at an on-ramp of a highway. Red car is the subject vehicle; A and B denote different positions and moments for the same vehicle.

Figure 3.1 illustrates a scenario of an MLC at an on-ramp of a highway. The red car is the subject vehicle, and the forbidden symbol at position A indicates that it is not safe to merge lanes at that position and moment because it is too close to the white car behind it. The red car should move forward to position B and assure that the distance between itself to the front car (*d_lead_t*) and to the back car (*d_lag_t*) in the target lane are sufficiently large. An MLC is more vital than a DLC due to its mandatory nature and the potential of leading to traffic jams. On a highway, on-ramps are often the most congested places, as this is where drivers need to merge to the main lane in a limited distance and time. It's urgent, but safety is the most important concern, and drivers always need to slow down to ascertain that the MLC behaviour is safe.

In this work, we propose an MLC suggestion system, which can be integrated in an ADAS, to help drivers to complete MLC maneuvers in a more efficient and safer way. The MLC suggestion system works as follows: 1) it obtains the information of surrounding vehicles using Radar, Lidar, or camera sensors; 2) it analyses the data and gives a suggestion of "merge" or "non-merge"; 3) drivers execute an MLC maneuver according to the "merge" suggestion, or keep in the same lane according to the "non-merge" suggestion. The suggestion system can eliminate the time of hesitation in manual MLC maneuvers and therefore reduce traffic congestion. The core of this system is an algorithm that takes sensor inputs and makes suggestions efficiently. We focused on the development of such an algorithm suitable for the application of an MLC suggestion system.

Research on lane changing mainly focuses on constructing a microscopic behaviour model [8, 9], and many machine learning algorithms are utilized. Kumar *et al.* [10] proposed a model for lane changing intention prediction based on a combination of a multiclass support vector machine (SVM) classifier and a Bayesian filter (BF). Li *et al.* [11] applied a hidden Markov model (HMM) to recognize lane changing intention. Tang *et al.* [12] reported an adaptive fuzzy neural network to predict the intention of lane changing. However, these work aim to predict the surrounding vehicles' lane changing behaviour to assure the safety of the subject vehicle. Hou *et al.* [13] focused on an MLC, and proposed a lane changing model combining a Bayes classifier with a decision tree for an MLC at lane drops. However, for a decision tree algorithm, there are no consistent standards to choose the threshold values. They combined two algorithms to achieve good prediction accuracy of non-merge (no lane changing) events at the cost of prediction accuracy for merge (lane changing) events. Our previous work for an MLC combined SVM and artificial neural network (ANN) achieves good accuracy of non-merge events but also at the cost of accuracy of merge events [14]. The inaccuracy of non-merge predictions, i.e., misclassifying a non-merge event as a merge event, will result in an accident; meanwhile, the inaccuracy of merge predictions, i.e., misclassifying a merge event as a non-merge event, will delay the merge operation from an on-ramp to the main lane, potentially increasing the occurrence of traffic jams. Therefore, an algorithm that can achieve high accuracy of both non-merge and merge events is desired for an MLC at on-ramps of highways. Recently, deep learning algorithms such as convolutional neural networks (CNNs) [15] and long short-term memory (LSTM) networks [16], a type of recurrent neural network (RNN), have been investigated for their applications in intelligent transport systems. The results are promising, but deep learning algorithms generally require much computational resource to obtain highly accurate results.

To achieve high accuracy of both non-merge and merge events with a computationally inexpensive algorithm, we proposed a compact architecture composed of a feedforward neural network and an additional gated branch for performance improvement, called *Gated Branch Neural Network* (GBNN).

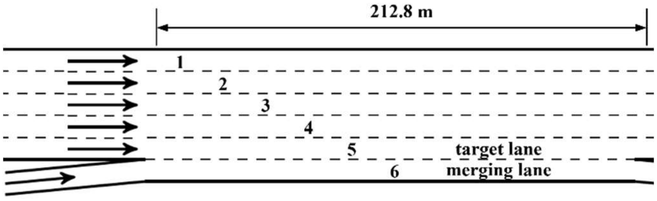
The main contributions are three-fold: (1) a novel GBNN algorithm is proposed, which enables more efficient modeling for an MLC behaviour at the on-ramps of highways with high accuracy of both non-merge and merge events; (2) an additional gated branch based on the correlation analysis is proposed, which provides effective feature learning and explicitly capture the relationship between the surrounding driving environment and the lane changing decision; (3) the proposed method is lightweight, compared with existing deep learning algorithms, in computation and can be applied in ADAS for efficient MLC suggestions.

The rest of the paper is organized as follows. Section 3.5 provides the data source and the detailed methodology. Section 3.6 shows the results and discussion. Section 3.7 concludes the paper and discusses the limitation of the proposed approach and future work.

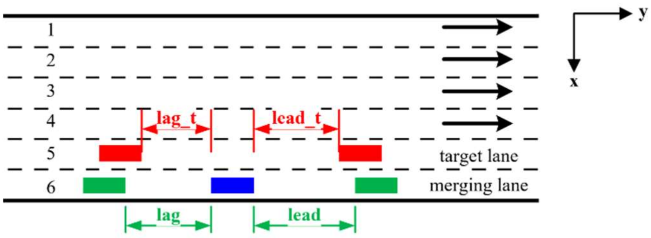
3.5 Methodology

The real-world datasets of U.S. Highway 101 (US 101) and Interstate 80 (I-80) are utilized from the Federal Highway Administration's Next Generation Simulation

(NGSIM) program [17]. The NGSIM datasets are open-source and have been widely used in academic and industry research [18-20]. Therefore, they are used to model the MLC behaviour in this work. A total of 45 minutes of vehicles' trajectory data are collected either on US 101 in Los Angeles or on I-80 in the San Francisco Bay area, using synchronized digital video cameras and customized software. The US 101 dataset is segmented into three 15-minute periods: 7:50 a.m. to 8:05 a.m.; 8:05 a.m. to 8:20 a.m.; and 8:20 a.m. to 8:35 a.m. on June 15, 2005. The I-80 dataset is also segmented into three 15-minute periods: 4:00 p.m. to 4:15 p.m.; 5:00 p.m. to 5:15 p.m.; and 5:15 p.m. to 5:30 p.m. on April 13, 2005. The total dataset is around 1230.6 MB, and the vehicle trajectory data provide the precise location of each vehicle within the study area at every one-tenth of a second, resulting in detailed lane positions and locations relative to other vehicles. The data is of two-dimensional structure, with rows organized by the vehicles' identification (ID) number and columns for information of each vehicle (such as locations, velocity, acceleration, lane number, size and type of vehicle, and ID numbers of preceding and following vehicles). In order to obtain the information of the surrounding vehicles, including the information of vehicles in the adjacent lanes, a customized program is developed to extract suitable features for modeling MLC behaviours.



(a)



(b)

Figure 3.2(a) Study area for modeling of an MLC (from lane 6 to lane 5) from NGSIM US-101 and I-80 Dataset. (b) Definition of an MLC scenario, centre blue - subject (merging) vehicle; right green (lead) - front vehicle in merging lane; left green (lag) - rear vehicle in merging lane; right red (lead_t) - front vehicle in target lane; left red (lag_t) - rear vehicle in target lane.

The schematic of the study area for modeling MLC behaviours is shown in Figure 3.2(a), where vehicles at the on-ramp are forced to merge from lane 6 to lane 5. Lane 6 and lane 5 are defined as the merging lane and the target lane, respectively. Figure 3.2(b) shows the detail of the MLC surrounding environment. Generally, when a subject vehicle (the centre blue vehicle) tries to change lanes from the merging lane to the target lane, it needs to assess its surrounding driving environment, such as the status of front and rear vehicles both in the merging lane and target lane as well as its own information. Four symbols, *lead*, *lag*, *lead_t*, and *lag_t*, are introduced to represent four categories of surrounding vehicles (the front vehicle in the merging lane, the rear vehicle in the merging lane, the front vehicle in target lane, and the rear vehicle in the target lane). Every category of vehicle is characterized by distance, velocity, and acceleration relative to the subject vehicle, respectively.

The GBNN algorithm is proposed to model the MLC behaviour and suggest drivers to perform an MLC maneuver at the on-ramps of highways. The architecture of GBNN is shown in Figure 3.3.

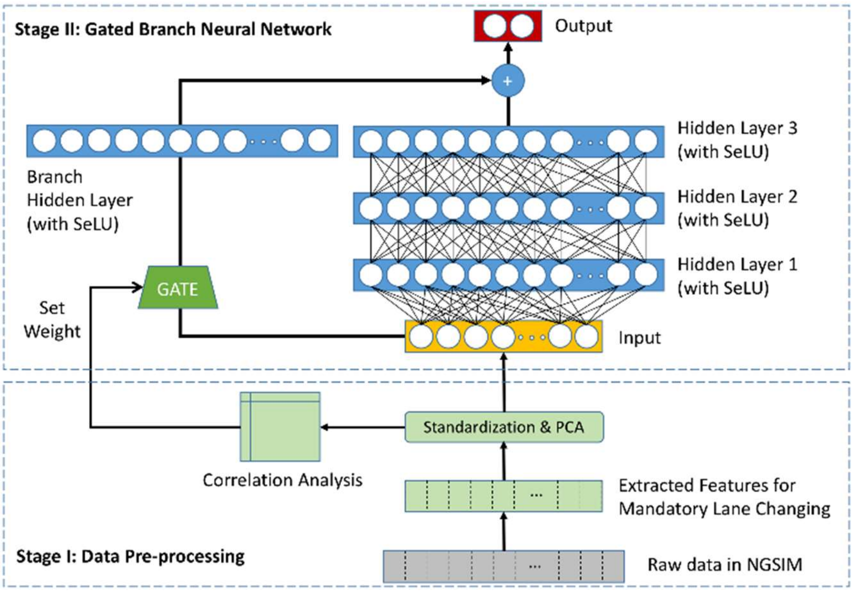


Figure 3.3 Data flow and architecture of the proposed GBNN.

It is composed of two stages. Stage I is for data pre-processing and stage II includes a compact neural network. The trajectory data provided by NGSIM includes total information for all the vehicles that appeared in lane 1 through lane 6, but only data relevant to an MLC from lane 6 to lane 5 is needed. Therefore, in the first stage, 16 features are extracted as inputs to our algorithm as listed in Table 3.1. The input data have 20002 instances (rows) and each instance has 16 features and 1 label (columns). Label with value of 0 and 1 represents non-merging and merging events, respectively. Of the same number of non-merge and merge instances in all the data, 80% is taken randomly as the training dataset, and the remaining 20% of instances as the test dataset. In the training dataset, the 10-fold cross validation method is used to train and optimize the model. Once the model parameters are optimized, the test dataset is used to predict the non-merge events and the merge events. Before feeding data into the neural network, data pre-processing, including standardization and principle component analysis (PCA), has to be carried out.

Standardization is an essential data pre-processing step and is always used before PCA. In standardization, the input data is normalized to be of unit variance and zero mean [21]. PCA is able to derive low-dimensional features from a large set of features, while keeping as much variation as possible [22]. After standardization and PCA, the number of features is reduced to 13. After conducting standardization and PCA for the training dataset, the mean, standard deviation, and eigen vectors are derived. These derived values are then used for processing the standardization and PCA for the test dataset [23]. Afterwards, data flow is divided into two streams: one goes to the input of the main neural network, and the other is used to calculate the correlation coefficients between the label and each feature. A typical method (Pearson) to calculate the correlation coefficients between two random variables X and Y is:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \tag{3.1}$$

Where cov is the covariance operator, and σ the standard deviation. The calculated correlation coefficients are used to set the weight of the gate in the branch shown in Figure 3.3. Generally, in machine learning, the upper bound of the weight is often chosen to be 1 and the lower bound is a small value greater than 0, so we scaled and shifted the correlation coefficients by equation (3.2) and their values are in the range [0.1, 1]:

$$f(x) = \frac{|x|}{|x|_{\max}} \times 0.9 + 0.1 \tag{3.2}$$

Where $|x|$ is the absolute value of the calculated correlation coefficient between each feature and the label, and $|x|_{\max}$ is the maximum one. In statistics, there are three common methods to calculate correlation coefficients: Pearson, Kendall Tau, and Spearman. We evaluated all these three methods and compared their effects on the prediction accuracy.

Table 3.1 Features extracted from NGSIM for modeling an MLC at on-ramps of highways

No	Features	Unit	Meaning
1	*Δx_ego*	feet	deviation of the lateral coordinate, *x* in Figure 3.2(b), of the merging vehicle with respect to the centre line of the merging lane;
2	*y_ego*	feet	longitudinal coordinate, *y* in Figure 3.2(b), of the merging vehicle with respect to the left-most entry edge, where the vehicle enters into the study area;
3	*v_ego*	feet/s	velocity of the merging vehicle;
4	*a_ego*	feet/s^2	acceleration of the merging vehicle;
5	*d_lead*	feet	longitudinal gap between the front vehicle in merging lane and the merging vehicle;
6	*Δv_lead*	feet/s	velocity difference between the front vehicle in the merging lane and the merging vehicle;
7	*Δa_lead*	feet/s^2	acceleration difference between the front vehicle the in merging lane and the merging vehicle;
8	*d_lag*	feet	longitudinal gap between the rear vehicle in the merging lane and the merging vehicle;
9	*Δv_lag*	feet/s	velocity difference between the rear vehicle in the merging lane and the merging vehicle;
10	*Δa_lag*	feet/s^2	acceleration difference between the rear vehicle in the merging lane and the merging vehicle;
11	*d_lead_t*	feet	longitudinal gap between the front vehicle in the target lane and the merging vehicle;
12	*Δv_lead_t*	feet/s	velocity difference between the front vehicle in the target lane and the merging vehicle;

13	*Δa_lead_t*	feet/s^2	acceleration difference between the front vehicle in the target lane and the merging vehicle;
14	*d_lag_t*	feet	longitudinal gap between the rear vehicle in the target lane and the merging vehicle;
15	*Δv_lag_t*	feet/s	velocity difference between the rear vehicle in the target lane and the merging vehicle;
16	*Δa_lag_t*	feet/s^2	acceleration difference between the rear vehicle in the target lane and the merging vehicle.

Stage II is composed of a main neural network with three hidden layers and an additional branch. The branch is gated, and the weight of the gate is set by the aforementioned values derived from the correlation coefficients. The main neural network consists of three hidden layers, and each hidden layer has the same number of neurons. The branch consists of a weighted gate and one hidden layer, whose number of neurons is set to the same as the main neural network. The main neural network and the branch are used together to generate a two-bit binary label, with label [1 0] representing a non-merge event and label [0 1] representing a merge event. The gate provides a soft approach to incorporate correlation between label and input features into the branch network.

The activation function used in each hidden layer is scaled exponential linear units (SeLU) [24]. Compared with the traditional Sigmoid and the popular rectified linear unit (ReLU), SeLU automatically converges towards zero mean and unit variance. Thus it has the same effect as batch normalization and can avoid exploding and vanishing gradients. The SeLU activation function is given by equation (3.3):

$$\mathrm{selu}(x) = \lambda \begin{cases} x & \text{if } x > 0 \\ \alpha \cdot e^{x} - \alpha & \text{if } x \leq 0 \end{cases} \tag{3.3}$$

Where λ and α are two parameters with typical values of around 1.0507 and around 1.6733, respectively.

Softmax with the cross-entropy function is used to compute the training loss as equation (3.4) that quantifies the agreement between the predicted scores and the ground truth labels. The smaller the loss, the more consistent the predicted class scores with the ground truth labels are in the training data [25]:

$$L = -\frac{1}{N}\sum_{i=0}^{N-1} \log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right) + \frac{1}{2}\lambda' \sum_k \sum_l W_{k,l}^2 \tag{3.4}$$

Where N is the number of instances during training, s the predicted score function that maps the input data to class scores, s_j for j-th class, s_{y_i} for the class of ground truth label corresponding to the i-th input instance, λ' the regularization penalty, and W the weight matrix. $\sum_k \sum_l W_{k,l}^2$ denotes the summation of all the squared elements of W. The first term at the right-hand side of the equation represents the data loss, and the second term represents the regularization loss.

Adaptive moment estimation (Adam) [26] is used as the optimizer to minimize the training loss, and its effect on this algorithm is investigated and compared with the traditional Mini-batch gradient descent (GD) optimizer. For the Adam optimizer, the same stochastic batch fetch method for Mini-batch GD is used.

The algorithm is implemented with Python and TensorFlow framework. Pandas, Numpy, and Matplotlib are used for file and data manipulation, array operation, and visualization.

It is worth noting that the accuracy of non-merge events has higher priority than merge events, because misclassifying a non-merge event as a merge event could result in a traffic crash. However, for an MLC at on-ramps of highways, the accuracy of merge events is also vital because inaccurate merge prediction could potentially lead to the increase of traffic jams. Therefore, the optimization strategy in our research is to weight more on the accuracy of non-merge event prediction at limited cost of the accuracy of predicting merge events.

Table 3.2 Confusion matrix for an MLC

	Predicted non-merge (N)	Predicted merge (P)
Actual non-merge (N)	TN	FP
Actual merge (P)	FN	TP

It is common to evaluate the performance of a binary classifier using the confusion matrix and plot the receiver operating characteristic (ROC) curve [21]. For MLC applications, the confusion matrix is given in Table 3.2. Each row in the confusion matrix represents an actual class, and each column represents a predicted class. *TN* (true negatives) and *FP* (false positives) are the numbers of actual non-merge events correctly and wrongly classified as non-merge events and merge events, respectively. *FN* (false negatives) and *TP* (true positives) are the numbers of actual merge events wrongly and correctly classified as non-merge events and merge events, respectively. The accuracy of non-merge and merge are also called specificity and recall (sensitivity), respectively. They can be calculated by equation (3.5) and (3.6).

$$\text{Accuracy of non-merge (specificity)} = \frac{TN}{TN+F} \tag{3.5}$$

$$\text{Accuracy of merge (recall)} = \frac{TP}{TP+F} \tag{3.6}$$

ROC curve plots the true positive rate (recall) against the false positive rate ($1 -$ specificity). The area under the curve, also called ROC score, is used to compare the performance of different algorithms. A perfect binary classifier will have a ROC score equal to 1. In this work, some traditional binary classifiers - SVM, random forest, stochastic gradient descent (SGD) - are compared against the GBNN algorithm. The ROC curve for SVM, random forest, and SGD are implemented by the Scikit-Learn package, and for GBNN the method in Ref. [27] is used.

3.6 Results and Discussion

Input features contribute differently to the output prediction. They are controlled by the neural network's weights updated by backpropagation. Backpropagation in general does not guarantee to reach the global optimum. In our research, it is observed that through the proposed gated branch, correlation statistics can help neural networks converge to a better optimum. The gate coefficients are the normalized correlation statistics. Input features after data pre-processing are multiplied by gate coefficients so that they are weighted explicitly by the correlation statistics. Correlation statistics with the gated branch influence the network's output through the fusion of the gated branch and the main neural network at the output.

After data standardization and PCA, three commonly used correlation analysis methods, Pearson, Kendall Tau, and Spearman, are compared. Figure 3.4 shows the Pearson correlation coefficient between each principal component (pc_i, $i = 1, 2, \ldots, 13$) and the label. It can be seen that some components contribute more than others, and those with higher correlation coefficients have higher weights in the gate of branch.

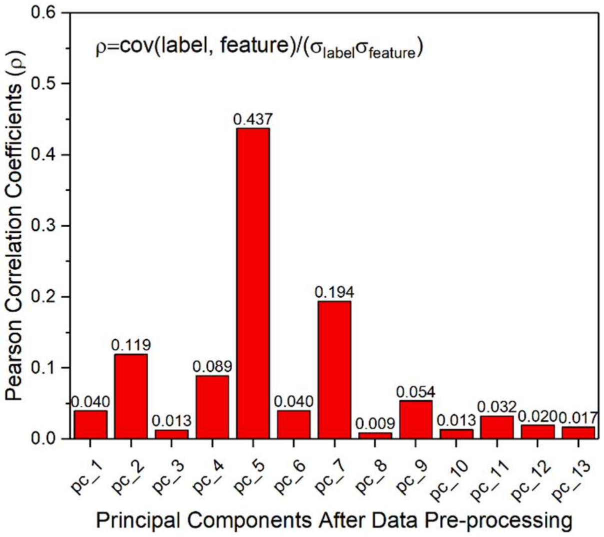


Figure 3.4 Pearson correlation coefficients between label and each principal component of training data (after standardization and PCA).

Table 3.3 shows the correlation coefficients, which is calculated by the pandas. DataFrame.corr( ) function provided in the Pandas package, as well as the calculated weights for the gate in GBNN. Pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. The pandas.DataFrame.corr( ) function computes the pairwise correlation of columns, excluding null values. The weights are calculated by equation (3.2), which limits the values between 0.1 and 1. It eliminates the difference of correlation coefficients between the Kendall Tau and Spearman methods leading to outputs with the same weights. In the following discussion, we will reduce these two methods to one and refer to as Kendall/Spearman.

Table 3.3 Correlation coefficients (corr.) between input features (feat.) and label; calculated weights (wei.) for gates.

input	Pearson		Kendall Tau		Spearman	
feat.	corr.	wei.	corr.	wei.	corr.	wei.
1	0.040	0.182	0.024	0.158	0.029	0.158
2	0.119	0.346	0.113	0.373	0.139	0.373
3	0.013	0.126	0.018	0.144	0.022	0.144
4	0.089	0.284	0.077	0.285	0.094	0.285
5	0.437	1	0.373	1	0.457	1
6	0.040	0.183	0.040	0.196	0.049	0.196
7	0.194	0.499	0.162	0.490	0.198	0.490
8	0.009	0.118	0.012	0.128	0.014	0.128
9	0.054	0.210	0.059	0.243	0.073	0.243
10	0.013	0.127	0.026	0.162	0.032	0.162
11	0.032	0.167	0.029	0.169	0.035	0.169
12	0.020	0.140	0.019	0.146	0.023	0.146
13	0.017	0.135	0.017	0.140	0.02	0.140

Figure 3.5 shows the effects of different correlation methods on the non-merge accuracy and merge accuracy. We set the weight of the gate by the Pearson and Kendall/Spearman correlation method as well as the random method. Since we aim to maximize the accuracy of non-merge events, the Pearson method is the preferred one with the highest non-merge accuracy and good merge accuracy. In this comparison, the

number of neurons in each hidden layer is fixed at 384, and 500 epochs are run to train the model.

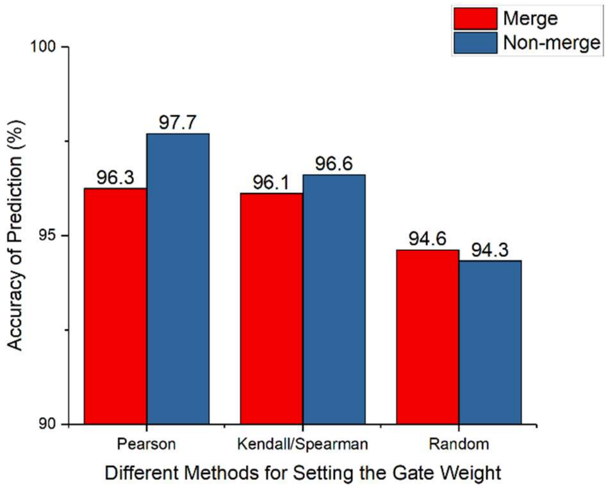


Figure 3.5 Comparison of different correlation methods (Pearson, Kendall/Spearman, and random), which are used for setting the weights of gate, on the effects of the prediction accuracy of non-merge and merge events.

Adam computes adaptive learning rates for each parameter in algorithms. In addition to storing an exponentially decaying average of past squared gradients, it also keeps an exponentially decaying average of past gradients, which makes it more efficient and stable than the traditional GD method and its variances [26]. Figure 3.6 investigates the training loss vs. the epochs running for Adam and Mini-batch GD. Mini-batch is the most efficient method in GD variances, and the common batch size ranges between 50 and 256. In Figure 3.6, we use a batch size of 200 for both Mini-batch GD and Adam. It can be seen that Adam achieves minimum loss around 100 epochs, while Mini-batch GD requires around 2000 epochs. Adam also has smaller variation at the stable stage. For model training, the computational time is proportional to the number of epochs. We found through experiments that beyond 500 epochs, the training loss of Adam does not change much. Therefore, 500 epochs are sufficient for training. This value is also used for the subsequent experiments.

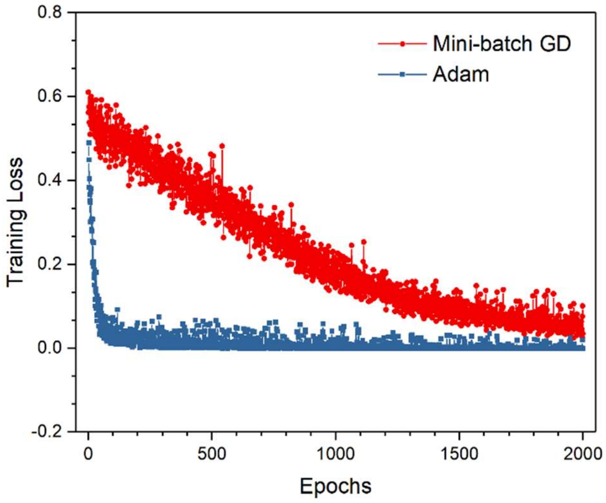


Figure 3.6 Training loss vs. the number of epochs for different optimizers.

The numbers of hidden layers and neurons of each hidden layer are two basic parameters of neural networks. In theory, one hidden layer with enough neurons can handle all binary classification problems, but two or three hidden layers can help to reduce the number of required neurons drastically. If the total number of neurons is not large enough, the model is prone to underfit, which means the model is not able to obtain a sufficiently low loss value on the training set, but if the neurons are too many, the model may be overfitting, namely, the model can achieve an extremely low error for the training set but gives a large error when used to predict the test set. The main cause of overfitting is the unsuitably large capacity of the model [28]. The neurons number in each hidden layer plays an important role for its performance. Traditionally, these numbers are decreased from input layer to output layer. Recently, especially for deep learning, researchers have found that keeping the same number of neurons in each hidden layer is a simple and efficient way to improve the performance [21]. In the proposed method, three hidden layers were selected in the main neural network and one in the gated branch, as shown in Figure 3.3. We vary the same number of neurons at each hidden layer among 64, 96, 128, 192, 256, 384, and 512. Both the accuracy of predictions for merge events and non-merge events are investigated and the results are shown in Figure 3.7. The Pearson method and Adam optimizer are used, and the epochs are chosen to be 500 as stated in the aforementioned discussion.

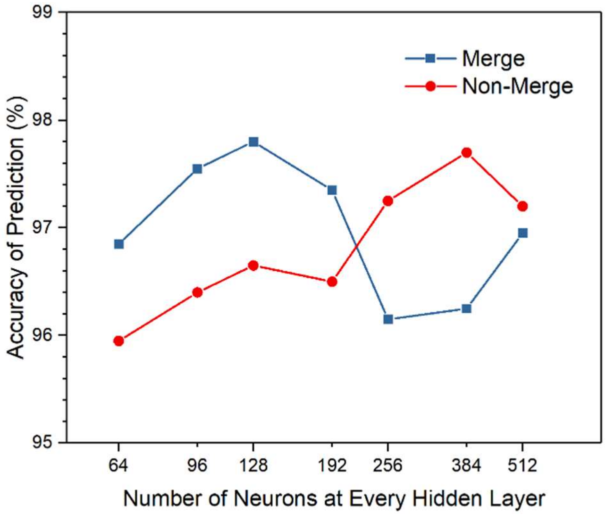


Figure 3.7 Accuracy of prediction for merge events and non-merge events with different number of neurons in each hidden layer (x-axis in log2 scale).

With a small number of neurons, the accuracy of merge events turns out to be better than the accuracy of non-merge events. As the number of neurons increases, the accuracy of non-merge events increases and achieves the highest value of 97.7% at 384 neurons. The accuracy of merge events is not the highest at that point, but still has a good value of 96.3%. Since the accuracy of non-merge events has priority over merge events due to its relation to driving safety, 384 neurons are chosen.

Training the GBNN model takes around 230 s, but the prediction (of merge or non-merge) only needs around 30 ms. The running time information is obtained from a regular laptop, Macbook Pro mid-2014 (2.5 GHz Intel i7, 16 GB 1600 MHz DDR3), and TensorFlow is the CPU-only 1.4 version. It is expected that hardware used for assisted driving will be more powerful. For instance, Tesla's Autopilot uses Nvidia's Drive PX2 platform with 2 SoCs and 2 GPUs, which brings the capacity closer to the power needed to enable level 5 full autonomy [29]. Therefore, GBNN can be executed in real-time on such platforms.

To see the computation efficiency, we compare the training and inference time of GBNN with a baseline CNN (AlexNet [30]) algorithm. AlexNet is implemented in the same computational platform. Hyperparameters, such as receptive field, stride, and zero-padding, are adapted and tuned for one-dimensional inputs. Results show that AlexNet will need 8798 s to train and 482 ms to predict, with a prediction accuracy of 96.6% and 97.5% for non-merge and merge events, respectively. In other words, GBNN is more than 38 times faster in training time and 16 times faster in inference times when compared to CNN (AlexNet). Other popular deep learning models, such as GoogLeNet [31] and VGGNet [32] are more complicated than AlexNet in architecture and thus likely require even more time to train and predict.

An MLC is a typical binary classification application. Therefore, many traditional binary classifiers can be adopted. We implement the stochastic gradient descent (SGD), random forest, SVM with radial basis function (RBF) kernel (*gamma* and *c* were set to be 0.2 and 5), and SVM with linear kernel (*c* was set to 5) by Scikit-Learn package. The decision tree, Bayes classifier, and combined algorithm from Hou *et al.* [13] are compared. Our previous work of combining SVM and ANN for an MLC, implemented by MATLAB, is also used for comparison [14].

The results are shown in Table 3.4. The proposed GBNN algorithm and AlexNet outperform all other algorithms in accuracy of both non-merge and merge events.

However, as previously described, the accuracy of the non-merge event is more important than that of the merge. GBNN not only achieves the highest accuracy among all classifiers for non-merge events but also uses much shorter time. The combined methods, which combined two binary classifiers as a new classifier, achieve good accuracy of non-merge events at the cost of the accuracy of merge events.

Table 3.4 Comparison of different binary classifiers for MLC applications

Algorithms	Accuracy of non-merge	Accuracy of merge
SGD	64.7%	73.2%
Random Forest	89.4%	94.6%
SVM with RBF Kernel	93.9%	95.9%
SVM with Linear Kernel	75.6%	71.3%
Decision Tree [13]	84.3%	80.8%
Bayes Classifier [13]	79.5%	92.3%
Combined Decision Tree and Bayes Classifier [13]	94.3%	79.3%
Combined SVM and ANN [14]	94%	78%
Proposed GBNN	**97.7%**	**96.3%**
CNN (AlexNet)	96.6%	97.5%

Although the accuracy of the non-merge is the more important one, the low accuracy of the merge will delay merging from on-ramps to the main lane, leading to an increase of potential traffic jams at the on-ramps of highways. Therefore, achieving high accuracy for both non-merge and merge events is important for on-ramps MLC applications. The proposed GBNN algorithm achieves this goal due to the introduction of an additional gated branch to the main neural network, the SeLU activation function, and the Adam optimizer.

Figure 3.8 shows the comparison of ROC curves among GBNN, SVM with RBF kernel, random forest, and SGD. As mentioned before, ROC curve plots the true positive rate (recall) against the false positive rate $(1 - \text{specificity})$. The area under the curve, also called ROC score, is used to compare the performance of different classifiers. The diagonal dash line represents a ROC curve of a pure random classifier, and a good classifier stays as far away from this line as possible (the closer to the top left, the better). ROC scores for GBNN, SVM with RBF kernel, random forest, and SGD are 0.990, 0.980, 0.972, and 0.754, respectively.

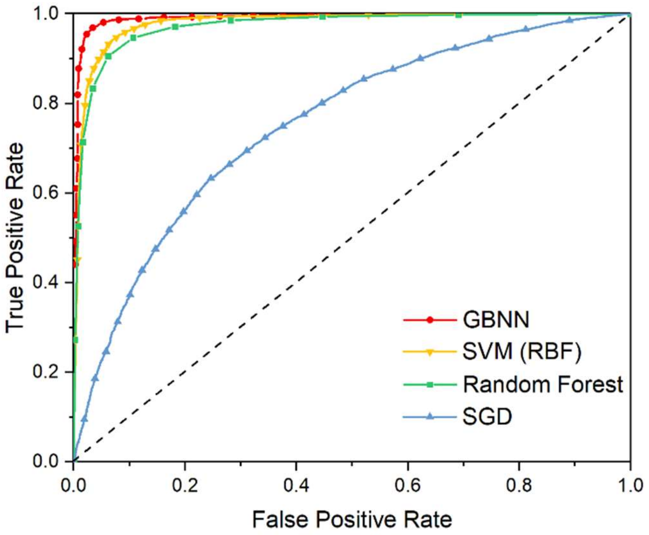


Figure 3.8 Comparison of ROC curves among GBNN, SVM (with RBF kernel), Random Forest, and SGD. The black dash line represents a ROC curve of a pure random classifier.

This clearly demonstrates the advantages of the proposed GBNN algorithm over traditional binary classifiers. The main contributor is the additional gated branch based on correlation analysis, which explicitly captures the relationship between the surrounding driving environment and the lane changing decision.

3.7 Conclusions

In this paper, we proposed GBNN, a lightweight feedforward neural network and an additional gated branch to predict vehicles' lane changing behaviours. The proposed GBNN algorithm achieved a high accuracy in predicting both non-merge events (97.7%) and merge events (96.3%) as well as a satisfactory ROC score (0.990) using the NGSIM dataset. It outperformed state-of-the-art binary classifiers reported in MLC applications. With inference time at 30 ms with high accuracy on a regular laptop, GBNN is promising for real-time on-vehicle applications. We compared GBNN with AlexNet, a deep algorithm, to demonstrate its lightweight characteristics. The evaluation is limited as only one dataset is utilized. As future work, we planned to collect more data with a real vehicle equipped with radar, LIDAR, and camera sensors to evaluate the performance of GBNN and realize a real-time MLC suggestion system.

3.8 Acknowledgements

This research is in part supported by NSERC Discovery Grant, NSERC Strategic Partnership Grant, and China Scholarship Council.

References

[1] Dey, K.C., Yan, L., Wang, X., et al., "A Review of Communication, Driver Characteristics, and Controls Aspects of Cooperative Adaptive Cruise Control (CACC)", IEEE Transactions on Intelligent Transportation Systems, 2016, 17(2):491-509.

[2] Jeannin, J.B., Ghorbal, K., Kouskoulas, Y., et al., "A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system", International Journal on Software Tools for Technology Transfer, 2015, 230(1):1-25.

[3] Kim, I.-H., Bong, J.-H., Park, J., Park, S., "Prediction of Driver's Intention of Lane Change by Augmenting Sensor Information Using Machine Learning Techniques", Sensors, 2017, 17, (6): 1350.

[4] Flanigan, M., Majka, K., Blatt, A., Lee, K., "Utility of Advanced Driver Assistance Systems and Connected Vehicle Technologies for Emergency Medical Services: Emergency Event Use Cases in an Automated and Connected Transportation System", Nature Reviews Cancer, 2015, 5(3):161-71.

[5] Gaikwad, V., Lokhande, S., "Lane Departure Identification for Advanced Driver Assistance", IEEE Transactions on Intelligent Transportation Systems, 2015, 16, (2): 910-918.

[6] "FARS Encyclopedia - Vehicles Involved in Single- and Two-Vehicle Fatal Crashes by Vehicle Maneuve (2015)", https://www.fars.nhtas.dot.gov/Vehicles/VehiclesAll/Vehicle.aspx

[7] Talebpour, A., Mahmassani, H.S., Hamdar, S.H., “Modeling lane-changing behaviour in a connected environment: A game theory approach”, Transportation Research Part C, 2015, 7: 420-440.

[8] Morton, J., Wheeler, T.A., Kochenderfer, M.J., “Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behaviour”, IEEE Transactions on Intelligent Transportation Systems, 2017, 18(5):1289-1298.

[9] Bahram, M., Hubmann, C., Lawitzky, A., Aeberhard, M., Wollherr, D., “A Combined Model- and Learning-Based Framework for Interaction-Aware Maneuver Prediction”, IEEE Transactions on Intelligent Transportation Systems, 2016, 17(6):1538-1550.

[10] Kumar, P. , et al. “Learning-based approach for online lane change intention prediction.” Intelligent Vehicles Symposium IEEE, 2013.

[11] Li, K., Wang, X., Xu, Y., Wang, J., “Lane changing intention recognition based on speech recognition models”, Transportation Research Part C Emerging Technologies, 2016, 69:497-514.

[12] Tang, J., Liu, F., Zhang, W., Ke, R., Zou, Y.: “Lane-changes prediction based on adaptive fuzzy neural network”, Expert Systems with Applications, 2018, 91: 452-463.

[13] Hou, Y., Edara, P., Sun, C. , “Modeling mandatory lane changing using bayes classifier and decision”, IEEE Transactions on Intelligent Transportation Systems, 2014, 15(2):647-655.

[14] Dou, Y., Yan, F., Feng, D., “Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers”, IEEE

International Conference on Advanced Intelligent Mechatronics, 2016(86): 901-906.

[15] Kim, J., Kim, J., Jang, G.-J., Lee, M., “Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection”, Neural Networks, 2017, 87: 109-121.

[16] Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., Liu, J., “LSTM network: a deep learning approach for short-term traffic forecast”, IET Intelligent Transport Systems, 2017, 11(2):68-75.

[17] Next Generation Simulation Fact Sheet, https://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm.

[18] Monteil, J., Nantes, A., Billot, R., Sau, J., “Microscopic cooperative traffic flow: calibration and simulation based on a next generation simulation dataset”, IET Intelligent Transport Systems, 2014, 8(6):519-525.

[19] Henclewood, D., Suh, W., Guin, A., Guensler, R., Fujimoto, R.M., Hunter, M.P., “Real-time data-driven traffic simulation for performance measure estimation”, IET Intelligent Transport Systems, 2016, 10(8):562–571.

[20] Coifman, B., Li, L.,“A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset”, Transportation Research Part B: Methodological, 2017, 105362-377.

[21] Géron, A., “Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems”, O’Reilly Media, Sebastopol, 2017, 1st edition.

[22] James, G., Witten, D., Hastie, T., Tibshirani, R., “An introduction to statistical learning: with Applications in R”, Springer, 2017.

[23] Friedman, J., Hastie, T., Tibshirani, R., "The elements of statistical learning", Springer series in statistics New York, 2001.

[24] Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S., "Self-Normalizing Neural Networks", arXiv Prepr., 2017, (arXiv:1706.02515v5), 1-102.

[25] "Convolutional Neural Networks for Visual Recognition", http://cs231n.github.io/.

[26] Kingma, D.P., Ba, J., "Adam: A Method for Stochastic Optimization", arXiv Prepr., 2014, (arXiv:1412.6980v9), 1-15.

[27] Woods, K., Bowyer, K.W., "Generating ROC curves for artificial neural networks", IEEE Transactions on Medical Imaging, 1997, 16 (3): 329-337.

[28] Goodfellow, I., Bengio, Y., Courville, A.. "Deep learning" (MIT press, 2016).

[29] Lambert, F., "Tesla has a new Autopilot '2.5' hardware suite with more computing power for autonomous driving", https://electrek.co/2017/08/09/tesla-autopilot 2-5-hardware-computer-autonomous-driving/,March 2018.

[30] Krizhevsky, Alex , I. Sutskever , and G. Hinton . "ImageNet Classification with Deep Convolutional Neural Networks." Advances in neural information processing systems, 2012, 25(2).

[31] Szegedy, Christian , et al. "Going Deeper with Convolutions." , 2014.

[32] Simonyan, K., Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition", arXiv Prepr. arXiv, 2015, (1409):1556.

Chapter 4

Prediction of the Surrounding Vehicles' Discretionary Lane Changing Intention at Freeway: A Gated Recurrent Units Approach

4.1 Citation and Main Contributor

Yangliu Dou, Yihao Fang, Chuan Hu, Rong Zheng, Fengjun Yan, "Prediction of Surrounding Vehicle's Discretionary Lane Changing Intention at freeway: A Gated Recurrent Units Approach." *IET Intelligent Transport Systems.* (Submitted)

The main contributor to this paper is the first author - Yangliu Dou (contributes more than 70%).

4.2 Abstract

Predicting a surrounding vehicle's intention early and accurately can help avoid potential congestions and accidents and improve driving safety and efficiency in advanced driver assistance systems (ADAS) and autonomous driving. We propose a recurrent neural network (RNN)-based time series classifier with a gated recurrent unit (GRU) to predict and classify the surrounding vehicles' discretionary lane changing intention in advance and provide an early notification to the ego-vehicle for driving assistance. The algorithm has been evaluated using the real-world datasets of U.S. Highway 101 and Interstate 80 from the Federal Highway Administration's Next Generation Simulation (NGSIM). The proposed algorithm outperforms classical

algorithms such as the long short-term memory (LSTM) and hidden Markov model (HMM). Results reveal that the proposed algorithm can predict the surrounding vehicles' lane changing maneuver 0.8 s in advance at a recall and precision of 99.5% and 98.7%, respectively. Furthermore, the model can predict the lane changing intention 1.6 s in advance with a recall of 92.2%. The method presented in this study can be used for ADAS and autonomous driving that promises high traffic safety and effective traffic management.

4.3 Introduction

Prediction of the surrounding vehicles' intention is an essential building block for advanced driver assistance systems (ADAS) [1-3] and autonomous vehicles [4-6] due to its importance in driving safety. There are three fundamental maneuvers in traffic control [7]: Mandatory lane changing (MLC), Discretionary lane changing (DLC) and Car following. Car following is relatively simpler than lane changing since it involves only one-dimensional motion. To predict the surrounding vehicle's intention, DLC plays a more important role because MLC always happens at the on-ramp or exit-ramp of a freeway where vehicles have a high possibility of lane changing. Therefore, drivers near the ramp have more awareness to the MLC behaviour of vehicles in the surrounding lanes. In contrast, DLC is more random, unexpected, and prone to accidents. For example, if a preceding car in the adjacent lane cuts in suddenly, this unpredictable behaviour may cause the subject vehicle to adjust itself accordingly in an abrupt manner, resulting in an unexpected deceleration or acceleration that contributes to a high possibility of accidents. Therefore, it is valuable to predict the surrounding vehicle's lane changing behaviour in advance.

DLC is a binary classification problem, and its performance can be evaluated by a confusion matrix [8]. We define lane changing and lane keeping events as positive and negative events, respectively. *FN* (false negatives) and *TP* (true positives) are the numbers of actual lane changing events wrongly and correctly classified as lane keeping events and lane changing events, respectively. *TN* (true negatives) and *FP* (false positives) are the numbers of actual lane keeping events correctly and wrongly classified as lane keeping events and lane changing events, respectively. Recall, accuracy of lane changing, is defined as equation (4.1); specificity, accuracy of lane keeping, is defined as equation (4.2); precision is defined as equation (4.3):

$$\text{recall} = \frac{TP}{FN+} \tag{4.1}$$

$$\text{specificity} = \frac{TN}{TN+FP} \tag{4.2}$$

$$\text{precision} = \frac{TP}{FP+T} \tag{4.3}$$

Recall and precision are trade-offs, and a perfect binary classifier will achieve three performance metrics all with a value of 1. For lane changing intention, prediction time in advance at a specific recall threshold is another important performance metric.

Due to the rapid development of artificial intelligent (AI), especially machine learning, several recent studies for DLC modeling have been conducted using AI. Aoude *et al.* [8] used a support vector machine (SVM) and a Bayesian filter (BF) to classify agent intentions at road intersections, and achieved a recall of 100%, but only has a precision at 77%. Kumar *et al.* [9] also worked with an SVM and BF algorithm. They developed a modified model for lane changing intention prediction and achieved a 1.3 s in advance prediction for lane changing events at a recall and precision of 100% and 72%, respectively. Park *et al.* [10] constructed a joint probability distribution of DLC in speed difference and density difference domains and applied logistic regression to quantify the relations and modelled DLC behaviours under congested traffic. Balal *et al.* [7] proposed a binary DCL model based on fuzzy inferences and reported an 82.2% accuracy in predicting lane changes. Tang *et al.* [11] also presented an adaptive fuzzy neural network and predicted the intention of lane changing successfully. DLC behaviour is a time series problem in nature rather than an instantaneous event. So, it is better to use a pure time series algorithm to handle the DLC problem [9].

To handle time series, the hidden Markov model (HMM) and recurrent neural network (RNN), two typical machine learning algorithms, have been applied. Li *et al.* [10] combined HMM and BF to recognize lane changing intention, and achieved a high recognition recall of 93.5% and 90.3% for right and left lane changing, respectively [11]. Although HMMs can capture strong discriminative features, RNN has displayed a slightly better overall performance for making earlier decisions [12]. RNNs were increasingly being used due to its advantages for successfully handling time series data, but a basic RNN suffered from the vanishing/exploding gradient problem [13]. To overcome this problem, the long short-term memory (LSTM) variant has been developed and widely adopted. Zhao *et. al.*[14] used LSTM to forecast the short-term traffic flow successfully. Zyner *et. al.* [15] applied LSTM to predict a driver's intention as the vehicle entered an intersection with the position and velocity obtained from the sensor of the ego-vehicle. Olabiyi *et al.*[16] reported a deep RNN algorithm based on camera data and predicted the lane changing maneuvers, acceleration, braking, and other essential behaviours. All these works have achieved good results and demonstrated RNN's promising applications in driver's behaviour modeling.

Recently, a simplified but effective modification of LSTM, named gated recurrent units (GRU), has been proposed [17]. As it decreases the computational cost, GRU is more suitable for in-vehicle applications. To accurately, computationally, and efficiently model the DLC maneuver of surrounding vehicles in advance, we proposed a GRU-based time series classifier and investigated the performance of the optimization, aiming at an early prediction to assist driving maneuvers.

Our envisioned system uses camera and radar to obtain the information of surrounding vehicles in real time. It will provide in-advance warnings for predicting the dangerous behaviours of the surrounding vehicles. This in-advance warning is dependent on the value of recall that a driver or autonomous vehicle wants. The driver or autonomous vehicle specifies a recall value (threshold) for different scenarios, and a corresponding in-advance warning will be calculated and presented by the proposed algorithm. This algorithm is suitable to simultaneously meet different drivers' demands and autonomous driving's criteria.

The main contributions are three-fold: (1) a novel GRU-based time series classifier for modeling DLC maneuver is proposed; (2) an early and reliable prediction of surrounding vehicles' DLC behaviour is provided, which helps improve the driving safety and efficiency of ADAS; (3) the proposed method is lightweight in prediction and can be applied in ADAS or autonomous vehicles in real-time applications.

The remainder of this paper is organized as follows. Section 4.4 describes the datasets and the detailed methodology. Section 4.5 provides the results and discussion. Section 4.6 draws conclusions and discusses future applications.

4.4 Datasets and Methodology

The real-world datasets of U.S. Highway 101 (US 101) and Interstate 80 (I-80) are obtained from the Federal Highway Administration's Next Generation Simulation (NGSIM) program [18]. The NGSIM datasets are open-source and have been widely used in academic and industry research [19-21]. Therefore, they are used to model the

DLC behaviour in this work. A total of 45 minutes of trajectory data for vehicles is collected on US 101 in Los Angeles and on I-80 in the San Francisco Bay area, respectively, using synchronized digital video cameras at 10 Hz. The US 101 dataset is segmented into three 15-minute periods: 7:50 a.m. to 8:05 a.m.; 8:05 a.m. to 8:20 a.m.; and 8:20 a.m. to 8:35 a.m. on June 15th, 2005.The I-80 dataset is also segmented into three 15-minute periods: 4:00 p.m. to 4:15 p.m.; 5:00 p.m. to 5:15 p.m.; and 5:15 p.m. to 5:30 p.m. on April 13th, 2005.

The schematic of study areas and a scenario of DLC behaviours are shown in Figure 4.1(a). Generally, when a subject vehicle (the center blue vehicle) tries to change lanes from the original lane to the target lane (left or right lane), assessment of its surrounding driving environment, such as the status of the preceding and following vehicles both in the original lane and target lane as well as its own information, is a necessity.

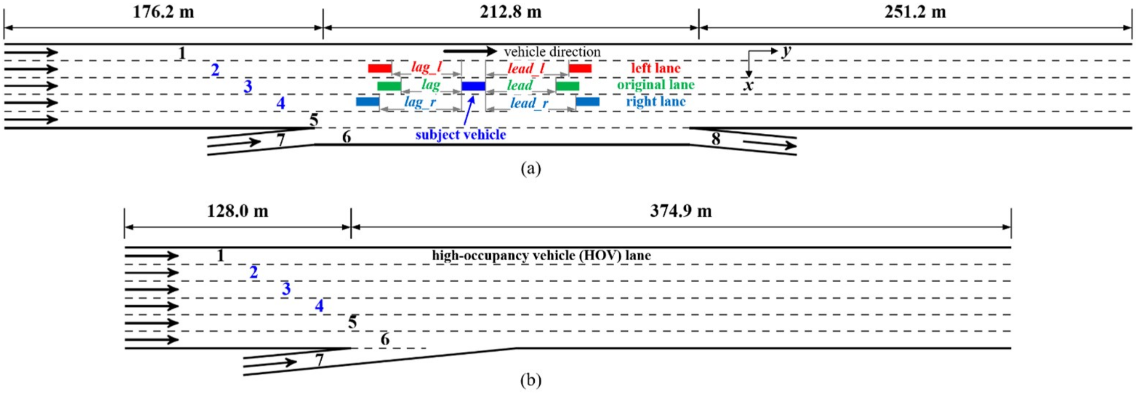


Figure 4.1 Schematic of study area for (a) US-101 and (b) I-80.

In Figure 4.1(a), a DLC scenario is defined: centre blue - subject vehicle; right green - preceding vehicle in original lane; left green - following vehicle in original lane; right red - preceding vehicle in left lane; left red - following vehicle in left lane; right dark blue - preceding vehicle in right lane; left dark blue - following vehicle in right lane.

Six symbols, *lead*, *lag*, *lead_l*, *lag_l*, *lead_r*, and *lag_r* are introduced to represent six categories for the surrounding vehicles (the preceding vehicle in the original lane, the following vehicle in the original lane, the preceding vehicle in the left lane, the following vehicle in the left lane, the preceding vehicle in the right lane, and the following vehicle in the right lane). Each category of vehicle is characterized by the gap, relative velocity, and relative acceleration from the surrounding vehicles with regards to the subject vehicle. The gap is defined as the distance of two successive vehicles.

The vehicles' trajectory data are processed based on the following rules [7, 22]. The statistical data from the US-101 0750 dataset (07:50 a.m. to 08:05 a.m.) are shown below as an example:

1) Only passenger cars are selected for study. The reason is two-fold: the numbers of motorcycles and trucks are much less than that of passenger cars; the lane changing characteristics of motorcycles and trucks are very different than that of passenger cars. There are 30 motorcycles, 53 trucks, and 2086 passenger cars, respectively.
2) Only the subject cars that originally travelled in lane 1, 2, 3, and 4 are selected. This is because the lane changing behaviours of cars in lane 5 and 6 have a large possibility of MLC. Therefore, the number of researched cars is reduced to 1553.
3) Those making multiple lane changes or those changing back to the original lane are dropped. To be consistent with traffic simulation models, a lane change is defined as the act of changing from one lane to the lane immediately next to it. The behaviour of multiple lane changing is different from a normal DLC. After this procedure, the number of remaining cars is 1461.
4) The cars are classified into two categories: lane changing and lane keeping. The corresponding numbers are 204 and 1257, respectively.
5) The data from the lateral and longitudinal position is smoothed with the moving average filter, and the velocity and acceleration data are recalculated by the

differential method. The original position data provided by NGSIM has noise due to the extracting algorithm from the camera video [21]. The moving average filter operates by averaging a number of points from the input signal to produce each point in the output signal, and is implemented by:

$$Y[i] = \frac{1}{M}\sum_{j=-(M-1)/2}^{(M-1)/2} X[i+j] \tag{4.4}$$

Where $X[\,]$ is the input signal, $Y[\,]$ the output signal, and *M* is required to be an odd number, representing the average number of points. We vary the *M* value to investigate the different smoothing effects.

6) The information of surrounding vehicles is collected, and the input features for the DLC algorithm are generated. The original trajectory data provided by NGSIM only includes the identity numbers of the preceding and following vehicles. Python with the Pandas data analysis library is used to collect the information of the surrounding vehicles (gap, velocity, and acceleration relative to the subject vehicle). The data required for the DLC algorithm includes one label (0 and 1 denotes lane keeping and lane changing, respectively), car identity number, and 23 input features, which are shown in Table 4.1.

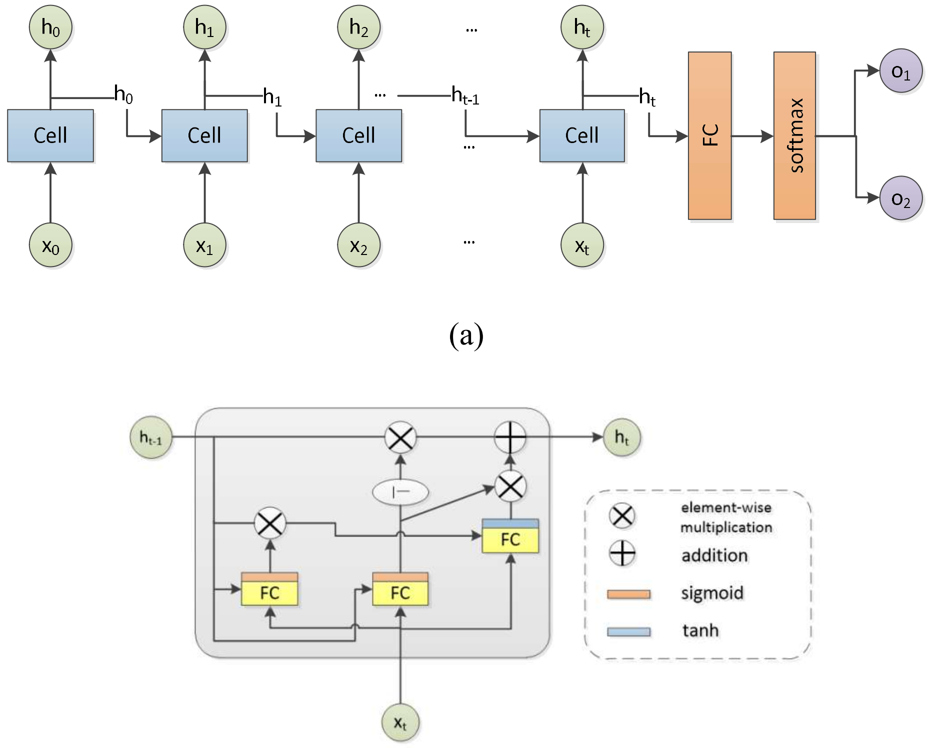
7) The trajectories are truncated to a uniform segment of six seconds ending at the time when the vehicle crosses the lane boundary (the data point at this time is defined as the ground truth point). This is because those six seconds are sufficient to include normal DLC behaviour according to our analysis in section 4.5. A total of 1443 effective vehicles are identified. Of the researched vehicles, 186 are classified as lane changing and 1257 as lane keeping. Table 4.2 lists the statistical information of all six datasets in this work.

Table 4.1 Input features extracted from NGSIM for modeling DLC behaviour

No.	Description
1	time in unit of 100 ms
2	Longitudinal position of subject vehicle
3	deviation of lateral position respect to the center line of the lane
4	velocity of subject vehicle
5	acceleration of subject vehicle
6-8	gap, velocity, and acceleration of preceding vehicle relative to the subject vehicle
9-11	gap, velocity, and acceleration of following vehicle relative to the subject vehicle
12-17	gap, velocity, and acceleration of preceding / following vehicle in left lane relative to the subject vehicle
18-23	gap, velocity, and acceleration of preceding / following vehicle in right lane relative to the subject vehicle

Table 4.2 Statistics of effective trajectories of passenger cars according to the processing rules for US-101 and I-80 datasets. keep - lane keeping (no lane change); left- left lane changing; right - right lane changing

	US-101			I-80			total
	0750	0805	0820	0400	0500	0515	
keep	1257	1245	1166	963	899	869	6399
left	109	94	103	93	84	67	550
right	77	66	39	31	15	18	246
total	1443	1405	1308	1087	998	954	7195



(a)

(b)

Figure 4.2 Time series classifier based on GRU cell: (a) structure of classifier; (b) structure of GRU cell.

A GRU-based time series classifier is trained to predict the intention of the surrounding vehicles' maneuvers. The structure of the classifier is shown in Figure 4.2(a), where x_i (i=0, 1, 2, … t) is the input vector, and h_i (i=0, 1, 2, … t) is the hidden state of a cell, which is a function of the input at that time and its state at the previous time step: $h_t = f(h_{t-1}, x_t)$. The hidden state of the last time step is connected to a fully-connected (FC) layer, followed by a softmax layer. o_1 and o_2 are final output class labels, indicating lane keeping and lane changing behaviours, respectively. Subsequently, we merge left lane changing and right lane changing to a lane changing class, since their features are similar in our dataset [22]. GRU is a variant of LSTM [17]. It retains the LSTM's advantages, such as its resistance to the vanishing/exploding gradient problem, but its internal structure is simpler than LSTM. Therefore, GRU is faster to train and validate than LSTM. Researchers also found that the simplification of GRU's architecture provides better performance over LSTM [23]. A GRU cell contains an update gate (z_t) and a reset gate (r_t) as illustrated in Figure 4.2(b), and their rules are shown in equation (4.5) - (4.7):

$$z_t = \text{sigmoid}(W_z[h_{t-1}, x_t] + b_z) \tag{4.5}$$

$$r_t = \text{sigmoid}(W_r[h_{t-1}, x_t] + b_r) \tag{4.6}$$

$$h_t = (1 - z_t)\otimes h_{t-1} + z_t \otimes \tanh(W_h[h_{t-1}\otimes r_t, x_t] + b_h) \tag{4.7}$$

Where W_z, W_r, and W_h are the weight matrices for the update gate, reset gate and hidden state, respectively; b_z, b_r, and b_h are the bias vectors of the update gate, reset gate and hidden state, respectively; ⊗ is the element-wise multiplication operator; sigmoid and tanh are activation functions.

The main purpose of our proposed algorithm is to predict the surrounding vehicles' lane changing intention in advance. We merged all the lane changing cars in US-101 and I-80 to generate an entire lane changing dataset, and then chose the same number of cars from lane keeping category to form a complete lane keeping dataset. The datasets are set to have the same number for balance. Each car instance includes 60-time steps at an interval of 0.1 s, and each time step includes 23 features as shown in Table 4.1. 80% of the data is randomly chosen as the training set and the remaining 20% as the test set. The distribution of lane changing cars and lane keeping cars is kept at a 1:1 ratio for both the training set and test set. At the training stage, six-second trajectories are used. At the test stage, the characteristics of early prediction are investigated by the following procedures:

a) Within a six-second trajectory, a three-second segment is chosen for testing, and the corresponding prediction accuracy of the lane changing event is evaluated.

b) First, we choose the segment of the last three seconds as the input sequence for the proposed model and evaluated the recall (accuracy of lane changing). Because this segment includes the ground truth point, the recall can be equal to or close to 100%.

c) Then, we keep the three second duration and shift this segment backwards t_{back} seconds before the ground truth point and recalculate the recall. Different t_{back} have different recall values.

d) Repeat step c), and vary the t_{back} among (0.2, 0.4, … 2). Figures of recall vs. t_{back} are plotted. A threshold of the recall is specified, and then a prediction time in advance can be determined. For example, the first decreasing point where the recall drops below 100% can be regarded as the threshold point, and then the corresponding t_{back} is denoted as the prediction time in advance.

The algorithm is implemented with Python and the TensorFlow framework. Pandas, Numpy, and Matplotlib are used for file and data manipulation, array operation, and visualization.

4.5 Results and Discussion

The visualization and effect of the moving average filter are investigated. After applying the data processing rules mentioned in the previous section, there are a total of 7195 effective trajectories left for algorithm development. To visualize the details of these trajectories and investigate the effect of the moving average filter, results from the US-101 0750 dataset are shown in Figure 4.3. All the trajectories of the passenger cars for left lane changing and right lane changing are illustrated in Figure 4.3(a), and the numbers are 109 and 77, respectively. For clear visualization of the whole process, the observation window is extended from 6 s to 20 s, including two 10 s segments before and after the ground truth point. The positions of the crossed lane boundaries are shifted to 0 for all trajectories.

As shown in Figure 4.3(a), six seconds before the ground truth point is sufficient for capturing the DLC behaviour, and most DLC cars cross the line boundary within three seconds. As literature mentioned [7, 21, 22], the original trajectory data provided by the NGSIM have noise due to the difficulties of extracting accurate data from high-mounted video cameras. We analyzed the NGSIM data and found that an upper limit for the absolute value of acceleration was set at 11.2 feet/s^2 to ensure the practical acceleration of the vehicle is within the physical constraint. Acceleration is the differential of velocity which can be recalculated from the velocity data provided by the original NGSIM data.

The recovered acceleration for trajectory no. 64 is shown in Figure 4.3(b). There are four truncations between the 0 to 10 s segments. This acceleration difference indicates that the noise of position and velocity data in the original NGSIM remain even when the acceleration truncations are carried out. To eliminate this noise, a moving average filter is used to filter x and y position data until the derived acceleration data are within ±11.2 feet/s^2. Through varying the M parameter, the velocity and acceleration from y are calculated based on the filtered position data.

Figure 4.3(c) shows three typical trajectories before and after smoothing. The x-axis is the longitudinal position (y) and the y-axis is the lateral position (x), illustrating the real scenarios on US-101 highway. These three trajectories are obtained from vehicle no. 64 (left changing from lane 3 to lane 2), no. 125 (right changing from lane 3 to lane 4), and no. 5 (lane keeping and staying in lane 4). Figure 4.3(d) compares the lateral position of the no. 64 trajectory under different M values. The trajectory becomes smoother with the increase of the M value. Figure 4.3(e) and (f) show the velocity and acceleration curves according to different M values, respectively. The original velocity data provided by NGSIM has sharp changes, which leads to high peaks or valleys in acceleration. From Figure 4.3(f), we can see that the filter with the M value of 11 still exceeds the physical constraints of 11.2 feet/s^2 and the one with the M value of 19 is the best when only considering the smoothing effect, but it filters out too much information.

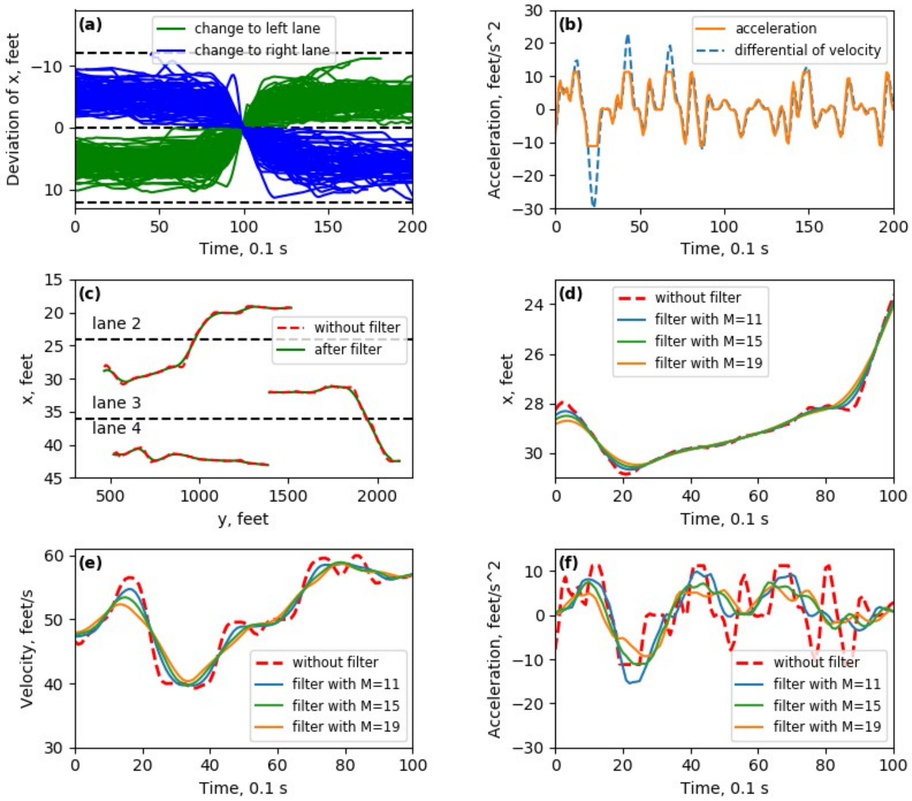


Figure 4.3 Typical trajectories of passenger cars in the NGSIM US-101 dataset (07:50 a.m. to 08:05 a.m.). (a) trajectories of left lane changing and right lane changing, and the positions of the crossed lane boundaries are shifted to 0 for all trajectories. (b) comparison of acceleration data provided in NGSIM, which are truncated at ±11.2 feet/s^2, and those calculated from the differential of velocity data provided in NGSIM (trajectory no. 64). (c) three trajectories before and after moving average filter for left lane changing (trajectory no. 64), right lane changing (trajectory no. 125), and lane keeping (trajectory

no. 5). (d) comparison of lateral positions (trajectory no. 64) for different values of filter parameter M. (e) comparison of velocity (trajectory no. 64) for different values of filter parameter M. (f) comparison of acceleration (trajectory no. 64) for different values of filter parameter M.

Therefore, the M parameter of the moving average filter is chosen at 15 to reach the optimal smoothness. Because no. 64 trajectory is a representative noise case in the NGSIM dataset, the M value of 15 is used for all the dataset in this work.

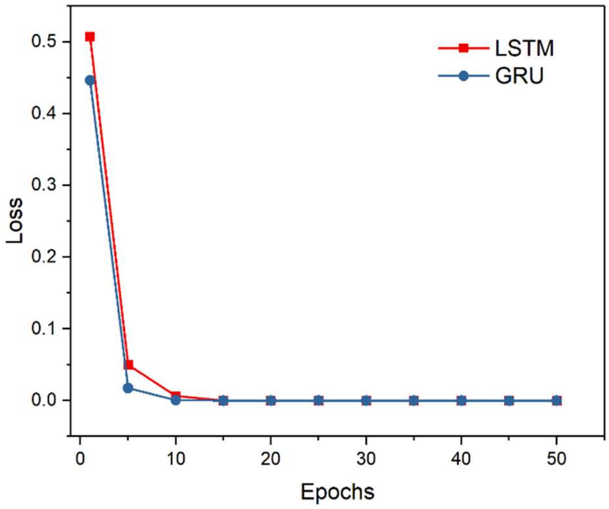


Figure 4.4 Comparison of training loss vs. the number of epochs between GRU and LSTM classifier. (The smaller loss, the better performance).

We train the GRU-based time series classifier with the training set, and then evaluate the prediction recall using the test set. During training, the training loss is calculated by the cross-entropy function and minimized by an optimizer named adaptive moment estimation (Adam) [24]. Both the GRU and LSTM algorithms are developed by Tensorflow. Figure 4.4 shows the comparison of training loss vs. the number of epochs between GRU and LSTM. The numbers of neurons are the same and are set at 128. The loss decreases quickly due to the advantage of Adam, and after 20 epochs, it becomes stable and stays near zero. As shown in Figure 4.4, both GRU and LSTM show good convergence, but GRU turns out to be slightly better than LSTM.

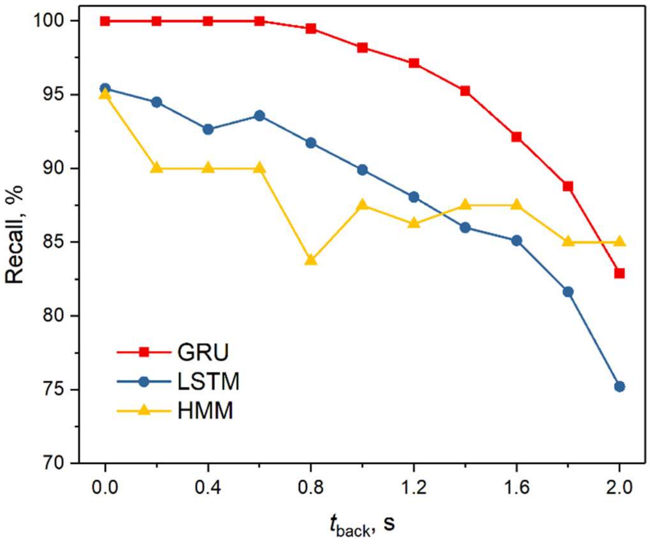


Figure 4.5 Comparison of recall vs. t_{back} among GRU, LSTM, and HMM algorithms.

We investigate the recall vs. t_{back}, and the results for GRU, LSTM, and HMM are compared in Figure 4.5. HMM is implemented with MATLAB. The number of neurons for both GRU and LSTM are set at 128. When the t_{back} equals zero, which denotes that the segment used for test includes the ground truth point, the prediction accuracy can reach 100% if the model captures the fact that the car actually crosses the lane boundary. For the GRU classifier, the recall of the first four points are 100%. Subsequently, the recall begins to decrease with the value of 99.5% and 98.2% for the 5th and 6th points, respectively. If we choose 99.5% as the threshold of recall, the proposed algorithm can predict the surrounding vehicles' lane changing behaviour 0.8 s in advance at the recall of 99.5%.

The LSTM and HMM classifiers present with lower recall at a maximum value of 95.4 % and 95.0 %, respectively. The GRU outperforms LSTM at all t_{back}, but HMM turns out to have a good performance at a large t_{back}. By taking both larger t_{back} and higher recall into consideration, GRU provides the best performance among these three algorithms.

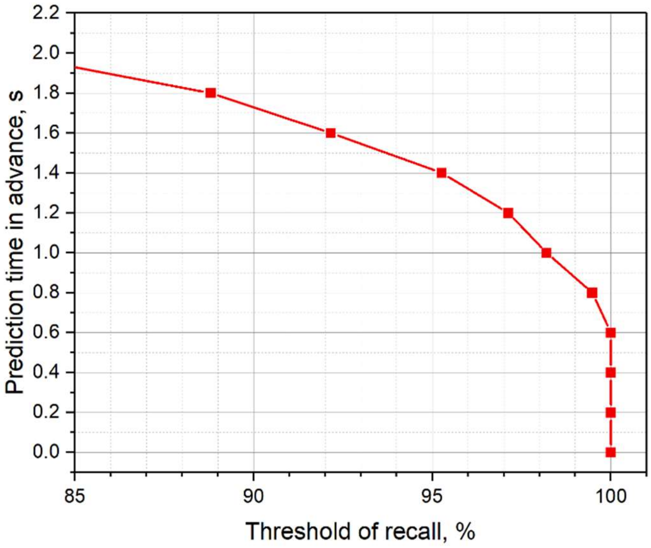


Figure 4.6 Determination of prediction time in advance by different threshold of recall.

Figure 4.6 provides a lookup graph for the determination of prediction time in advance from different threshold selections. For example, if we choose a threshold recall value of 92.2%, the proposed system can provide a prediction 1.6 s in advance of the surrounding vehicles' lane changing behaviour. In ADAS or autonomous vehicle applications, the corresponding prediction time in advance can be determined and adjusted by the specified threshold values of recall. Therefore, this model is able to suit the need of different drivers with their individual preferences or in various driving situations of autonomous vehicles and offers an early notification and assistance.

In order to optimize the performance of the GRU-based classifier, the neurons of each FC network within a GRU cell, refer to Figure 4.2, is adjusted among 32, 64, 128, 256, and 512 to investigate its effects. The prediction recall vs. t_{back} is shown in Figure 4.7.

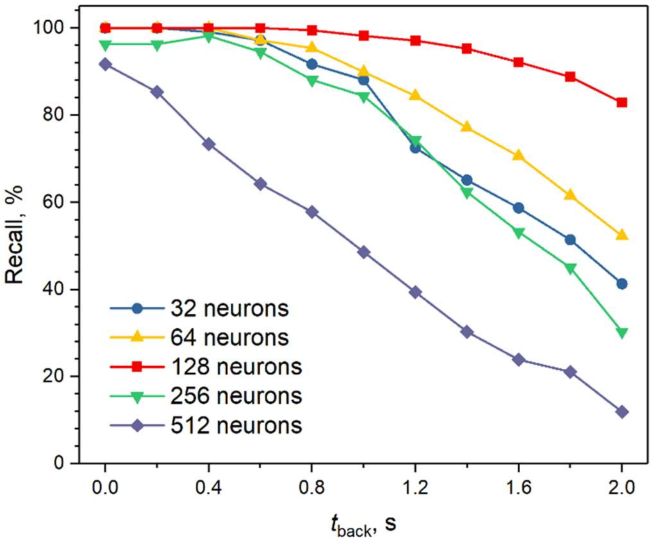


Figure 4.7 Prediction recall vs. t_{back}, varying the number of neurons as 32, 64, 128, 256 and 512.

The GRU classifier with 128 neurons demonstrates the best performance. As can be seen from the graph, both neurons with fewer and larger numbers have lower recall values. Having too few neurons cannot effectively represent the nonlinear relationship between input features and output labels; while too many neurons would easily lead to overfitting due to more model variables and a larger model size. Therefore, 128 is chosen as the optimal value of our proposed algorithm.

Besides recall, precision is another performance metric which always needs to be given for binary classifiers. Recall and precision are often combined into a single metric called F_1 score, which is the harmonic mean of recall and precision to provide a simple way to compare two classifiers. Table 4.3 compares the recall, precision, and F_1 score of GRU, LSTM, and HMM based classifiers.

Both the GRU and LSTM use 128 neurons in gates and the values are calculated at t_{back} of 0.8 s. According to the Table 4.3, the proposed GRU-based classifier offers the best performance.

Table 4.3 Recall, precision, and F1 score for GRU, LSTM, and HMM based classifiers

	Recall	Precision	F_1 Score
GRU	99.50%	98.70%	99.10%
LSTM	91.70%	98.00%	94.80%
HMM	83.80%	82.70%	83.20%

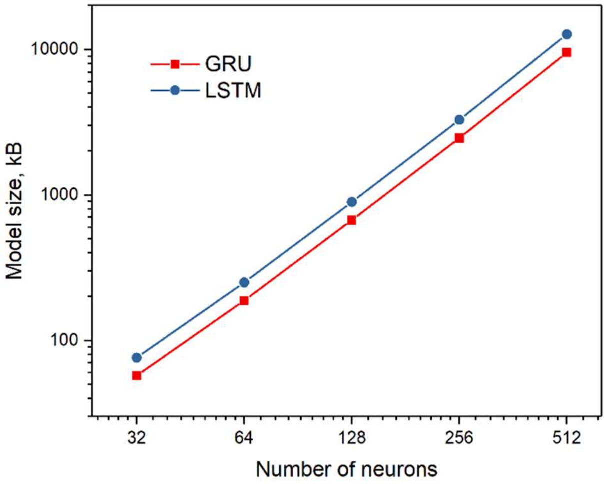


Figure 4.8 Model size vs. number of neurons.

Figure 4.8 compares the model size of GRU and LSTM for different numbers of neurons. As can be seen in the graph, the model size of the GRU is smaller than that of LSTM at all the different neuron numbers. To be specific, the numerical model sizes at the neurons of 32, 64, 128, 256, and 512 for GRU and LSTM are 57.2, 188.2, 671.3, 2463, and 9534 kB and 76.1, 250.4, 894, 3282, and 12708 kB, respectively. The former is about 3/4 of the latter. The reason GRU has smaller model sizes is that its structure is simpler than LSTM. GRU has three weight matrices as shown in Figure 4.2, while LSTM has four. The rate of the number of weight matrices is exactly the same as the rate of their model sizes. The rate of their model sizes is in proportion to the rate of the number of weight matrices. This means that the model size is mainly dominated by the number of weight matrices, whose size is affected by the number of neurons. Additionally, the curves in Figure 4.8 show an excellent linearity. We use a log2, and log10 scale for the x-axis and y-axis, respectively. Linear regression method is used to evaluate the linearity. Both GRU and LSTM turn out a R-square value of 0.99945.The intercepts of GRU and LSTM are -1.04649 and -0.92371, respectively; while the slope for GRU and LSTM are 0.55606 and 0.55629, respectively. The linearity means that the model size has exponential growth with the number of neurons. This relation can be described by $y = a \cdot x^b$, where ln$a$ is the intercept at when the x value is 1, and 0.3b is the slope of the curve. Model size is an important characteristic of algorithms, especially for embedded systems or mobile platforms. From this viewpoint, GRU is more suitable for in-vehicle applications than LSTM.

4.6 Conclusions

In this paper, a GRU-based time series classifier is proposed to predict the surrounding vehicles' behaviour and provides the prediction time in advance if lane changing is about to occur. The number of neurons is investigated for optimizing the algorithm. LSTM and HMM are also studied for comparison. Results indicate that the proposed algorithm can predict the subject vehicle's lane changing maneuver 0.8 s in advance with a prediction recall of 99.5%, or 1.6 s at a recall of 92.2%, before it actually crosses the lane boundary. The threshold of recall can be specified by the driver assistance systems or autonomous vehicles as a vital input parameter. This early notification of the surrounding vehicles' DLC behaviour is beneficial for drivers to take actions beforehand and will improve the safety and efficiency of driving on freeways. The envisioned driving assistant system based on the proposed algorithm shows promising applications for ADAS and autonomous vehicles.

4.7 Acknowledgements

This research is in part supported by NSERC Discovery Grant, NSERC Strategic Partnership Grant, and China Scholarship Council.

References

[1] Otaegui, O., “Embedding vision-based advanced driver assistance systems: a survey”, IET Intelligent Transport Systems, 2017, 11(3):103-112.

[2] Kim, I.-H., Bong, J.-H., Park, J., Park, S., “Prediction of Driver’s Intention of Lane Change by Augmenting Sensor Information Using Machine Learning Techniques”, Sensors, 2017, 17(6): 1350.

[3] Jeannin, J.B., Ghorbal, K., Kouskoulas, Y., et al., “A formally verified hybrid system for safe advisories in the next-generation airborne collision avoidance system”, International Journal on Software Tools for Technology Transfer, 2015, 230(1):1-25.

[4] Schwarting, W., Alonso-Mora, J., Rus, D., “Planning and Decision-Making for Autonomous Vehicles”, Annual Review of Control, Robotics, and Autonomous Systems, 2018(1):1-24.

[5] de La Fortelle, A., “Autonomous Intersection Management systems: criteria, implementation and evaluation”, IET Intelligent Transport Systems, 2017, 11(3):182-189.

[6] Li, W., “Trajectory control for autonomous electric vehicles with in-wheel motors based on a dynamics model approach”, IET Intelligent Transport Systems, 2016, 10(5):318-330.

[7] Balal, E., Cheu, R.L., Sarkodie-Gyan, T., “A binary decision model for discretionary lane changing move based on fuzzy inference system”, Transportation Research Part C Emerging Technologies, 2016, 67:47-61.

[8] Géron, A., “Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems”, O’Reilly Media, Sebastopol, 2017, 1st edition.

[9] Längkvist, M., Karlsson, L., Loutfi, A., “A review of unsupervised feature learning and deep learning for time-series modeling”, Pattern Recognition Letters, 2014, 42 (1):11-24.

[10] Li, K., Wang, X., Xu, Y., Wang, J., “Lane changing intention recognition based on speech recognition models”, Transportation Research Part C Emerging Technologies, 2016, 69: 497-514.

[11] Zheng, Z., “Recent developments and research needs in modeling lane changing”, Transportation Research Part B Methodological, 2014, 60(1):16-32.

[12] Panzner, M., Cimiano, P., “Comparing Hidden Markov Models and Long Short Term Memory Neural Networks for Learning Action Representations”, International Workshop on Machine Learning. Springer International Publishing, 2016: 94-105.

[13] Zaremba, W., Sutskever, I., Vinyals, O., “Recurrent Neural Network Regularization”, Eprint Arxiv, 2014, (1409.2329v5):1-8.

[14] Zhao, Z., Chen, W., Wu, X., Chen, P.C.Y., Liu, J., “LSTM network: a deep learning approach for short-term traffic forecast”, IET Intelligent Transport Systems, 2017, 11(2):68-75.

[15] Zyner, Alex, et al. “Long short term memory for driver intent prediction.” Intelligent Vehicles Symposium IEEE, 2017.

[16] Olabiyi, O., Martinson, E., Chintalapudi, V., Guo, R., "Driver Action Prediction Using Deep (Bidirectional) Recurrent Neural Network", arXiv:1706.02257, 2017.

[17] Cho, K., van Merrienboer, B., Gulcehre, C., et al., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", Computer Science, 2014(1406.1078):1-15.

[18] Next Generation Simulation Fact Sheet, https://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm.

[19] Monteil, J., Nantes, A., Billot, R., Sau, J., "Microscopic cooperative traffic flow: calibration and simulation based on a next generation simulation dataset", IET Intelligent Transport Systems, 2014, 8 (6):519-525.

[20] Henclewood, D., Suh, W., Guin, A., Guensler, R., Fujimoto, R.M., Hunter, M.P., "Real-time data-driven traffic simulation for performance measure estimation", IET Intelligent Transport Systems, 2016, 10(8):562-571.

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

[22] Wang, Q., Li, Z., Li, L., "Investigation of discretionary lane-change characteristics using next-generation simulation data sets", Journal of Intelligent Transportation Systems, 2014, 18(3):246-253.

[23] Zhou, G., Wu, J., Zhang, C., Zhou, Z., "Minimal Gated Unit for Recurrent Neural Networks", International Journal of Automation & Computing, 2016, 13(3):226-234.

[24] Kingma, Diederik P. , and J. Ba . "Adam: A Method for Stochastic Optimization." Computer Science, 2014.

Chapter 5

Strategic Car-Following Gap Model Considering the Effect of Cut-ins from Adjacent Lanes

5.1 Citation and Main Contributor

Yangliu Dou, Daiheng Ni, Zhao Wang, Jianqiang Wang, Fengjun Yan, "Strategic Car-Following Gap Model Considering the Effect of Cut-ins from Adjacent Lanes." *IET Intelligent Transport Systems.* 2016, 10(10): 658-665.

The main contributor to this paper is the first author - Yangliu Dou (contributes more than 70%).

5.2 Copyright

Published with permission from *IET Intelligent Transport Systems.*

5.3 Abstract

Drivers are typically faced with two competing challenges when following a preceding vehicle: they need to leave sufficient space in front to ensure safety, while doing so, the probability of cut-ins by other vehicles increases as the car-following gap becomes large. Therefore, a strategic car-following gap that addresses both challenges becomes critical. This paper proposes a method to address the problem through an overall objective function of the car-following gap and velocity considering the safety hazard and the probability of cut-ins by other vehicles. Based on this, seeking the strategic car-

following gap translates to finding the optimal solution that minimizes the overall objective function. With the support of field data, the method along with concrete models are instantiated and application of the method is elaborated. The method presented in this paper can be used to enhance traffic safety and improve traffic management in a connected vehicle environment that promises cooperative adaptive cruise control and cooperative crash avoidance systems.

5.4 Introduction

Recent developments of connected vehicle technologies such as vehicle-to-vehicle communication, in-vehicle computing, and on-board sensors, cooperative adaptive cruise control systems, cooperative collision avoidance systems, and lane-departure warning systems have led to extensive exploration of driver assistance systems. Adaptive cruise control (ACC), an important component of driver assistance systems, has been widely implemented in modern vehicles [1-3]. ACC adjusts vehicle speed to maintain a safe gap from vehicles ahead, and such control is imposed based on sensor information from on-board sensors only. Taking a step further, Cooperative Adaptive Cruise Control (CACC) extends the concept of ACC by means of cooperation between the vehicles in car following. As connected vehicle technology rolls out and vehicles are enabled to communicate with each other, CACC has emerged as a new direction [4-7]. In addition, the technology can further be used on automated vehicles in the longitudinal direction by using information gathered from infrastructure such as satellites, roadside equipment (RSE), and on-board equipment (OBE).

While ACC and CACC seem to have addressed safety gaps in car following, their smooth and reliable performance can frequently be interrupted by cut-in vehicles from adjacent lanes. Consequently, the subject vehicle has to adjust itself accordingly in an abrupt manner, resulting in unnecessary deceleration and acceleration that contribute to not only wasted fuel and emissions but also traffic waves [8-11] which further worsen the situation. As such, the control in the longitudinal direction has to be considered in conjunction with cut-ins from the lateral direction. Therefore, the objective of this research is to understand and model strategic car-following gaps that ensure safety distances in the longitudinal direction and, meanwhile, minimize the probability of cut-ins from the lateral direction. If successful, the research can be a vitally important input to advance the design of driver assistance systems, promoting smooth traffic flow with increased safety and reduced fuel consumption and emissions.

This paper is organized as follows. Section 5.5 reviews the state-of-the-art of car-following models, by which research gap is identified in its context. In Section 5.6, an overall objective function is set up which incorporates the safety hazard faced by a driver and the probability of cut-ins by adjacent vehicles. In this way, the formulation not only considers the effect of a preceding vehicle, but also incorporates the impact of potential cut-ins. Hence, optimizing the objective function translates to seeking a strategic gap that addresses both challenges in the longitudinal and the lateral directions. In Section 5.7, based on a set of empirical car-following data, the parameters in the car-following model are obtained though minimizing the overall objective function. Section 5.8 offers further discussion on model verification and application. In the last section, conclusions are drawn and contributions are clarified.

5.5 Review of Car-Following Models

Typically, underlying ACC and CACC systems are car-following models. Over more than half a century, there has been rich literature on this subject [12-18]. The goal of a car-following model is to use mathematical language to capture a driver's longitudinal control behaviour for a variety of purposes such as automatic control and traffic simulation. Modeling of this nature is difficult because driver's control strategies are not only complicated but also unobservable. Hence, many approaches have been explored to externalize driver's implicit strategies complemented by observable phenomena.

Car-following models are usually described by differential equations considering the vehicles' dynamic variables, such as position, velocity and acceleration in response to inputs such as external stimuli and driver properties. The goal is to adapt the subject vehicle's speed and acceleration to maintain a safe distance behind the preceding vehicles [19]. For example, Reuschel [20] and Pipes [21] developed car-following models to describe a specific safety rule coded in the California Drivers' Manual. Subsequently, mathematical methods were gradually adopted all over the world for the development of driver models. Li proposed a car-following model based on the effect of the visual angle under the non-lane-discipline environment, which captures the impacts from the visual angle of the driver between the following vehicle and the preceding vehicle as well as its change rate on a road without lane discipline [22]. The systematic effort on car-following models was carried out at the General Motors Research Laboratory, known as the Gazis-Herman-Rohery (GHR) model [1, 23]. According to Brackstone and McDonald [20], the car-following model can be distinguished as follows: Gazis-Herman-Rothery (GHR) model, safety-distance or collision avoidance models, Linear (Helly) models, Psychophysical or action point models, and Fuzzy logic-based models. The GHR model is based on an intuitive hypothesis that a driver's acceleration is proportional to velocity difference. It is a stimulus-response type car-following model and intends to find a proper acceleration according to different driving characteristics between two vehicles. Different from the GHR model, the safety-distance model seeks to specify a safe following distance, which is the minimum distance such that a collision would be avoidable if the driver of the vehicle ahead were to act "unpredictably" [24]. The Linear model, proposed by Helly, includes additional terms for the adaptation of the acceleration according to whether or not the vehicles in front are braking [2]. The psycho-spacing models are based on theories of perceptual psychology and they describe the vehicles in a relative speed versus a relative distance plane. These thresholds delineate a relative speed-spacing plane in which the driver of a following vehicle does not respond to any change in dynamic conditions [5, 25, 26]. Fuzzy logic modeling represents the next logical step in attempting to accurately describe driver behaviour. Such models typically divide their inputs into a number of overlapping "fuzzy sets" and each one describes how adequate a variable is [27].

With the above understanding, research gaps were identified in two ways. First, the above models were proposed mainly for the purpose of traffic simulation. In traffic simulation, the primary goal is to mimic the behaviour of real world drivers who are typically not perfect. As such, the model somehow needs to reflect such imperfection to be realistic. Therefore, it is unfit to use in advanced driving assistance systems that seek a "perfect" driver model that operates the vehicle in an "ideal" way to achieve both safety and efficiency gains. This research fills this gap by proposing the strategic car-following gap model that is specifically formulated with application in advanced driving assistance systems in mind. Second, existing models mainly focus on keeping safety distance in the longitudinal direction, while in reality longitudinal operation is frequently affected by lateral motions such as lane changing and cut-ins [28, 29]. As a matter of fact, some studies have verified that a driver's perception of risk is affected by multiple vehicles in front [30-32], so it is significant to consider the effect of vehicles cutting-in from adjacent lanes. More specifically, the subject driver is typically faced with a dilemma when following a preceding vehicle on a highway: leaving a sufficient car-following gap poses an opportunity for other vehicles to cut in, while shortening the gap increases the likelihood of running into a crash. Therefore, it is critical for the same car-following gap, henceforth called the strategic car-following gap, to serve two competing goals: ensuring safety in the longitudinal direction and avoiding cut-ins from the lateral direction.

In order to consider both of the effects from the longitudinal direction and lateral direction, the proposed strategic car-following gap model fills this research gap as well by taking the safety hazard and the probability of cut-ins as a function of the car-following gap and velocity, respectively. Subsequently, an objective function is constructed as the weighted sum of both functions and the optimal gap is obtained by finding the solution to that overall objective function.

5.6 The Strategic Car-Following Gap Model

We shall now direct our attention to the formulation of the strategic car-following gap model. Such a model should be functional under different speeds and be sensitive to cut-ins by vehicles from adjacent lanes.

5.6.1 Problem Formulation

To help weigh the two competing goals (ensuring safety and avoiding cut-ins), the proposed methodology translates each goal into a function of the car-following gap. One function, named the hazard index (HI), represents the safety hazard faced by a subject driver, which is the probability of rear end collision without cut-ins. The other function, called the cut-in probability (CIP), reflects the likelihood of cut-ins by surrounding vehicles. Then, an overall objective function is constructed as the weighted sum of both functions. Therefore, seeking the strategic car-following gap is equivalent to finding the optimal solution of the overall objective function.

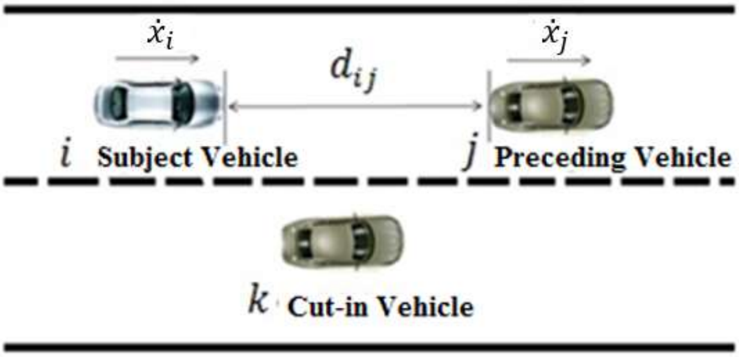


Figure 5.1 Steady-state car-following with potential cut-in vehicle.

To facilitate model formulation, Figure 5.1 illustrates a scene of car following where the subject vehicle i is following another vehicle j in the inner lane, while a third vehicle k in the adjacent lane is seeking opportunity to cut in. Based on this scene, all the variables used in the paper are defined in Table 5.1.

Table 5.1 Nomenclature

$\ddot{x}_i$(km/h^2)	longitudinal acceleration of vehicle i
$\ddot{x}_j$(km/h^2)	longitudinal acceleration of vehicle j
$\dot{x}_i$(km/h)	velocity of vehicle i
$\dot{x}_j$(km/h)	velocity of vehicle j
d_{ij}(m)	car-following gap between vehicles i and j
a(m)	scale parameter in Weibull distribution

b(m)	shape parameter in Weibull distribution
THW(s)	time headway between vehicles i and j
a_{NCI}(m)	scale parameter of the model without cut-ins
a_{CI}(m)	scale parameter of the model with cut-ins
b_{NCI}(m)	shape parameter of the model without cut-ins
b_{CI}(m)	shape parameter of the model with cut-ins
w(unit)	weight of the HI function

To avoid overly complicating the problem, the following analysis is made under steady-state car-following conditions:

(1) Vehicles in the target lane are at constant velocities, i.e.,

$$\dot{x}_i \,\&\, \dot{x}_j = C \text{ and } \ddot{x}_i = \ddot{x}_j = 0$$

Where $\dot{x}_i$ and $\dot{x}_j$ are the velocity of vehicles i and j respectively. C is a constant. $\ddot{x}_i$ and $\ddot{x}_j$ are longitudinal accelerations of vehicles i and j, respectively. The relative velocity between vehicles i and j is zero, i.e.

$$\dot{x}_i - \dot{x}_j = 0$$

(2) All vehicles are at substantial velocities to avoid stop-and-go and crawling conditions, i.e.,

$$\dot{x}_i \,\&\, \dot{x}_j \geq C_{min}$$

Where C_{min} is an arbitrary velocity that represents minimum velocity of vehicle in the highway.

Under this assumption, the car-following gap d_{ij} and the velocity of the subject vehicle $\dot{x}_i$ are the only two variables that affect the CIP and HI faced by driver i. Therefore, the HI and CIP can be represented by the functions of both d_{ij} and $\dot{x}_i$, respectively.

$$HI = h\left(d_{ij}, \dot{x}_i\right) \tag{5.1}$$

$$CIP = g(d_{ij}, \dot{x}_i) \tag{5.2}$$

An overall objective function describing the strategic car-following gap model is constructed as shown in equation (5.3):

$$\begin{cases} \min\ O\left(d_{ij}, \dot{x}_i\right) = (1 - w) \times g\left(d_{ij}, \dot{x}_i\right) + w \times h\left(d_{ij}, \dot{x}_i\right) \\ \qquad\qquad s.t.\ 0 \leq \mathrm{d_{ij}} \leq \infty,\ \ 0 \leq \dot{x}_i \leq \mathrm{V} \end{cases} \tag{5.3}$$

Where w is the weight of HI and a larger value of w means heavier weight is given to HI. The weight can be adjustable under different scenarios so that this model is able to suit the need of different drivers with their individual preferences. As such, seeking the strategic car-following gap translates to finding the optimal solution of the overall objective function subject to constraints of the car-following gap ($0 \leq \mathrm{d_{ij}} \leq \infty$) and vehicle velocity $0 \leq \dot{x}_i \leq \mathrm{V}$, where V is the maximum speed.

As illustrated in Figure 5.2, $\dot{x}_i{}^{*}$ and $\dot{x}_i{}^{**}$ are two different velocities of the subject vehicle. With a known $\dot{x}_i$, the optimal solution is to minimize the objective function. Common sense suggests that a long car-following gap translates to a large CIP, while a short car-following gap implies a large HI. Therefore, the weighted sum of the two terms, i.e., the objective function, is expected to reach its minimum at some point in the range of the car-following gap, i.e., at the strategic car-following gap.

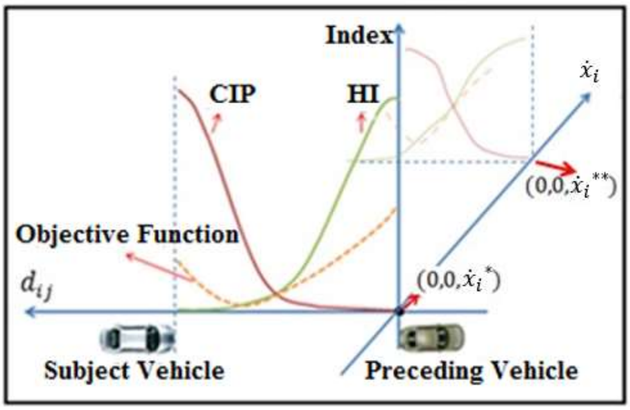


Figure 5.2 Illustration of the objective function varying with velocity and the car-following gap.

5.6.2 Model Instantiation

While the objective function in equation (5.3) and its component functions in equations (5.1) and (5.2) are given in generic terms, practical applications call for a strategic car-following model with specific form. As such, it is necessary to instantiate the above model with concrete functions that satisfy the velocity and car-following gap constraints. To formulate the specific model and identify its parameters, we base our development on examining field data. To avoid interrupting model formulation, the bulk of the field data analysis is deferred to the next section with only critical findings being brought forth here. In short, Weibull, Gamma, Normal, and Exponential distributions were adopted to represent statistical distributions fitted to the data. It turned out that Weibull distribution, which is fitted using the Kolmogorov-Smirnov test and the mean square error (MSE) methods, achieved the best result and thus was used in model formulation. With this, the empirical car-following data was divided into two groups. One group contained the car-following gap without cut-ins, while the other with successful cut-ins.

A probability density function (PDF) of the car-following gap was fitted for each group, from which a cumulative distribution function (CDF) was constructed. The HI faced by driver i was represented by the opposite probability of the CDF of the car-following gap without cut-ins. In addition, the CIP was represented by the CDF of the car-following gap with successful cut-ins. The empirical car-following data was two-dimensional (car-following gap d_{ij} and $\dot{x}_i$), and the parameters of the model were described as functions of $\dot{x}_i$.

The Weibull distribution is mathematically defined by its PDF equation, and its general expression has three parameters:

$$f(d_{ij}) = \frac{b}{a}(\frac{d_{ij}-\gamma}{a})^{b-1}e^{-\left(\frac{d_{ij}-\gamma}{a}\right)^{b}} \tag{5.4}$$

$$f(x) \geq 0;\ d_{ij} \geq \gamma;\ b \geq 0;\ a \geq 0;\ -\infty < \gamma < \infty$$

Where b is the shape parameter, and it is dimensionless. For specific values of b, Weibull distribution converts to other distributions. For example, Weibull distribution converts to the exponential distribution when b equates 1. Parameter a is the scale parameter, and has the same unit as d_{ij}. By holding b constant, the increase of a will stretch the PDF curve out. Parameter γ is the location parameter with the same unit as d_{ij}. Usually γ is set to zero. A non-zero γ shifts the PDF curve horizontally. In the car-following case, γ being zero is expected.

Therefore, the PDF can be simplified to a two-parameter form.

$$f(d_{ij}) = \frac{b}{a}(\frac{d_{ij}}{a})^{b-1}e^{-\left(\frac{d_{ij}}{a}\right)^{b}} \tag{5.5}$$

$$f(x) \geq 0;\ x \geq 0;\ b \geq 0;\ a \geq 0$$

The CDF is obtained by integrating the Weibull PDF as follows:

$$F(d_{ij}) = \int_0^{d_{ij}} f(d_{ij})\, d_{ij} = 1 - e^{-\left(\frac{d_{ij}}{a}\right)^b} \tag{5.6}$$

Based on our data fitting results, the shape parameter b is set to be constant, and the scale parameter a is a second order function of velocity which is shown in equation (5.7).

$$a = \delta \dot{x}_i^{\,2} + \vartheta \dot{x}_i + \varphi \tag{5.7}$$

Where δ, ϑ, and φ are constants. With these results, the CDF of the car-following gap without cut-ins, $F_{NC}\ (d_{ij})$ can be formulated as:

$$F_{NCI}(d_{ij}) = 1 - e^{-\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}}}$$

Based on which the specific form of HI is formulated as:

$$HI = h(d_{ij}) = 1 - F_{NCI}(d_{ij}) = e^{-\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}}} \tag{5.8}$$

In addition, the CDF of the car-following gap with successful cut-ins $F_{CI}(d_{ij})$ can take the form of:

$$F_{CI}(d_{ij}) = 1 - e^{-\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}}}$$

Hence, the specific form of CIP is specified as:

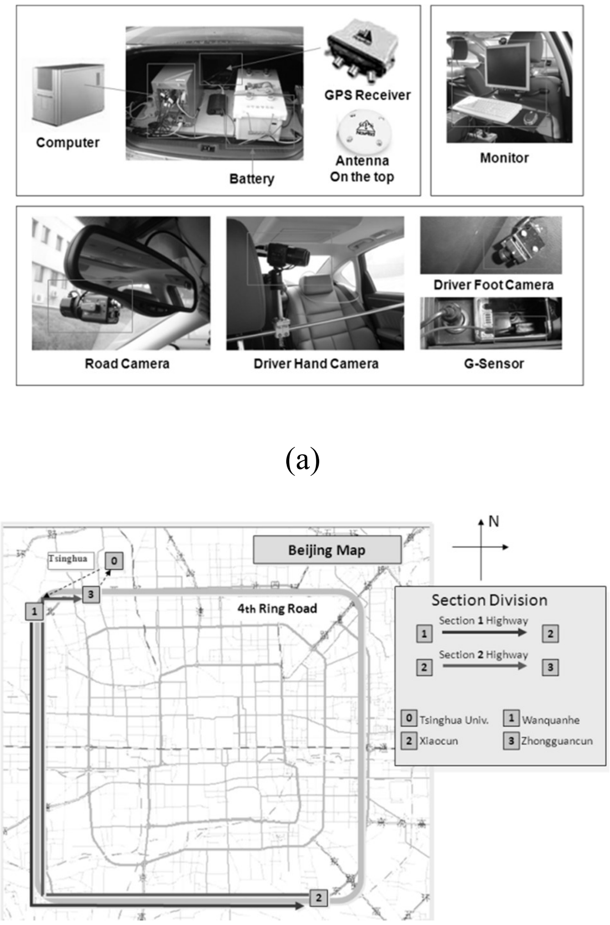
$$CIP = g(d_{ij}) = F_{CI}(d_{ij}) = 1 - e^{-\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}}} \tag{5.9}$$

Substituting the above formulation of HI and CIP into the objective function in equation (5.3), the specific form of the strategic car-following model is obtained:

$$\begin{cases}\min\ \ O\left(d_{ij},\dot{x}_i\right)=(1-w)\left[1-e^{-\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}}}\right]+w\left[e^{-\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}}}\right] \\ s.t.\ \ 0\le \mathrm{d_{ij}}\le\infty;0\le\dot{x}_i\le \mathrm{V}\end{cases} \tag{5.10}$$

5.7 Experiment and Parameters Identification

A set of car-following data collected in the field are used to support the model formulation above and this section presents the data analysis result.



(a)

(b)

Figure 5.3 (a) Installation of data collection system. (b) Map of experimental route.

5.7.1 Data Extraction and Analysis

This set of naturalistic driving data was recorded during field experiments. The data collection system and field experiment setup are shown in Figure 5.3(a). The test vehicle was equipped with radars and sensors to collect information including vehicle speed, acceleration, accelerator pedal depression, brake pressure. These signals were recorded at a frequency of 10 Hz. As shown in Figure 5.3(a), Fourth Ring Road in Beijing is selected as a valid experimental road. It is a highway that maintains a minimum of 4 lanes in each driving direction. The drivers drove from Tsinghua University (point 0) and entered Fourth Ring Road (section 1) from Wanquanhe (point 1) to the Xiaocun (point 2). Then went back along section 2 and returned to Zhongguancun (point 3). Subsequently they arrived at Tsinghua University (point 0).

This finished the driving route which provided approximately 130 km of valid road data. The speed limit of the test road was 80km/h. The experiments in this paper avoid peak traffic congestion time and no trucks are involved during our experiment. Experimental subjects are 12 non-professional drivers (10 men and 2 women). Their average age is 37 years old, and the standard deviation is 13 years. Their average driving experience is 15 years, with a standard deviation of 11 years. Additional data collection efforts were documented in our earlier work as reference in [33]. Then, the recorded experimental data were manually transcribed and grouped into two different cases: one containing car-following gaps without cut-ins, while the other with successful cut-ins.

Figure 5.4 and Figure 5.5 plot the subject vehicle's velocity versus car-following gap $\mathrm{d_{ij}}$ and $\mathrm{THW} = \mathrm{d_{ij}}/\dot{\mathrm{x}}_\mathrm{i}$, respectively according to the car-following process from the experiment data.

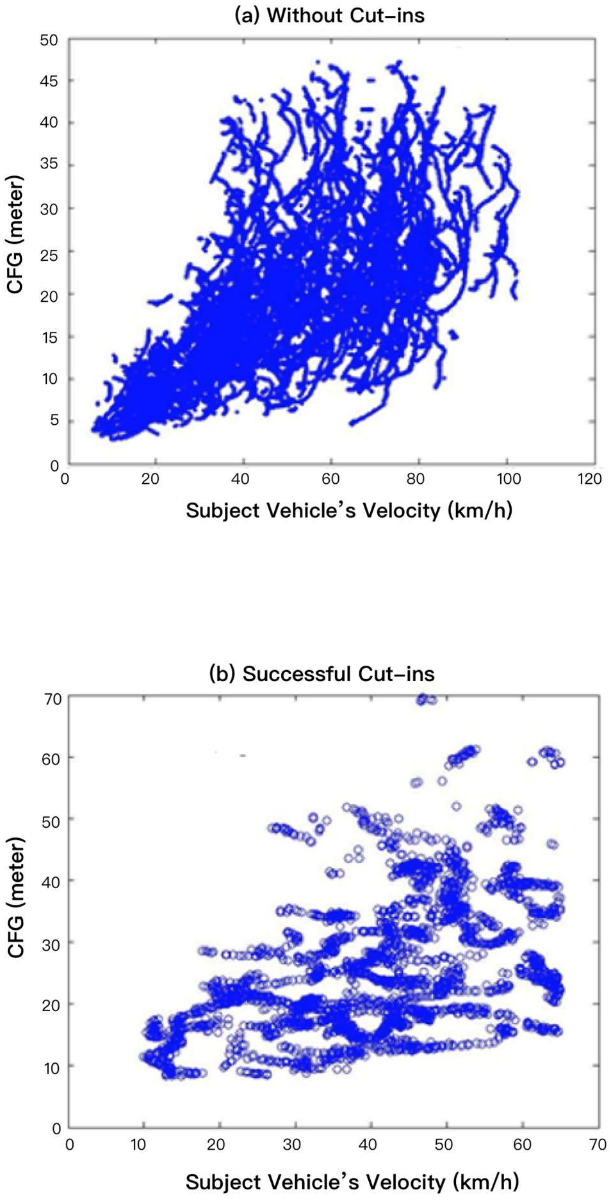


Figure 5.4 Plots of car-following gap versus subject vehicle's velocity:

(a) Without Cut-ins; (b) Successful Cut-ins.

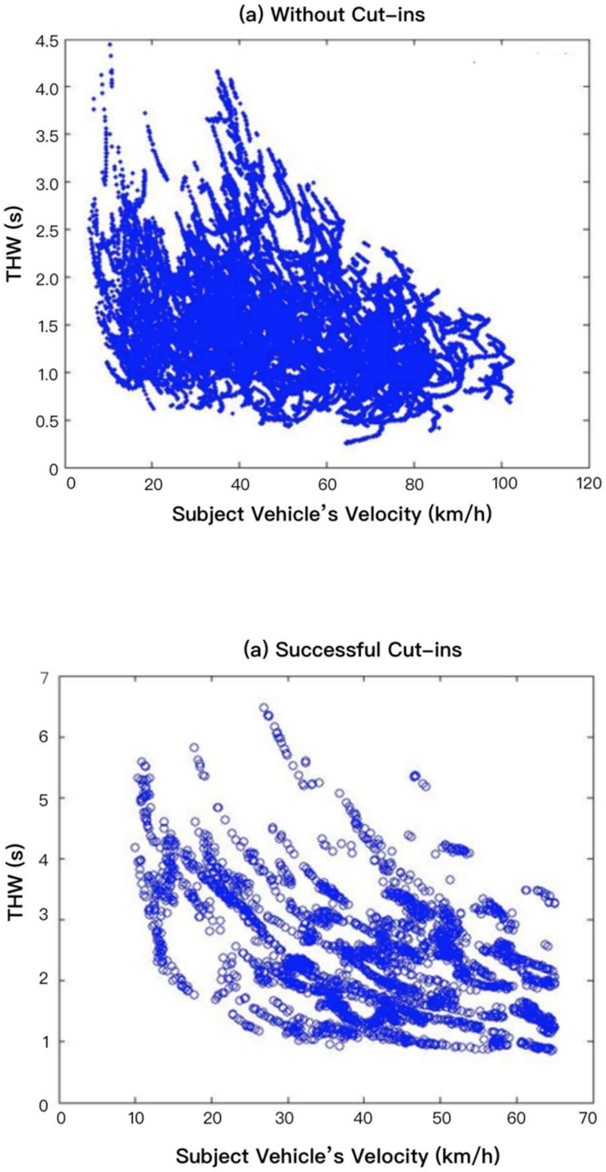


Figure 5.5 Plots of THW versus subject vehicle's velocity:

(a) Without Cut-ins; (b) Successful Cut-ins.

5.7.2 Data Fitting

The Weibull PDF of the car-following gap was fitted for each group in three steps. First, the data were divided into several segments according to $\dot{x}_i$. Figure 5.6 illustrates the segmentation. Second, the distribution model was fitted to the data in every segment to identify the scale and shape parameters. Finally, the scale parameters and the shape parameters were fitted by results from all the segments respectively.

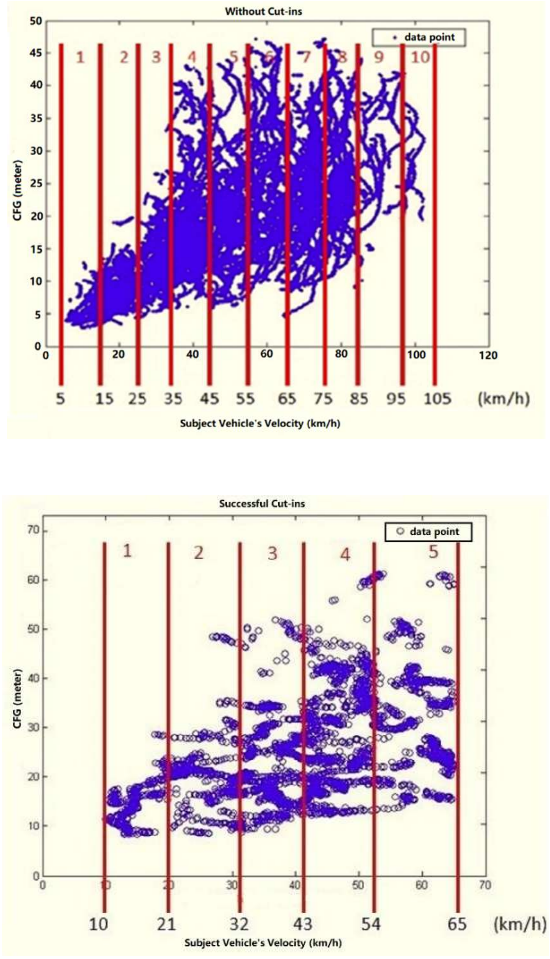
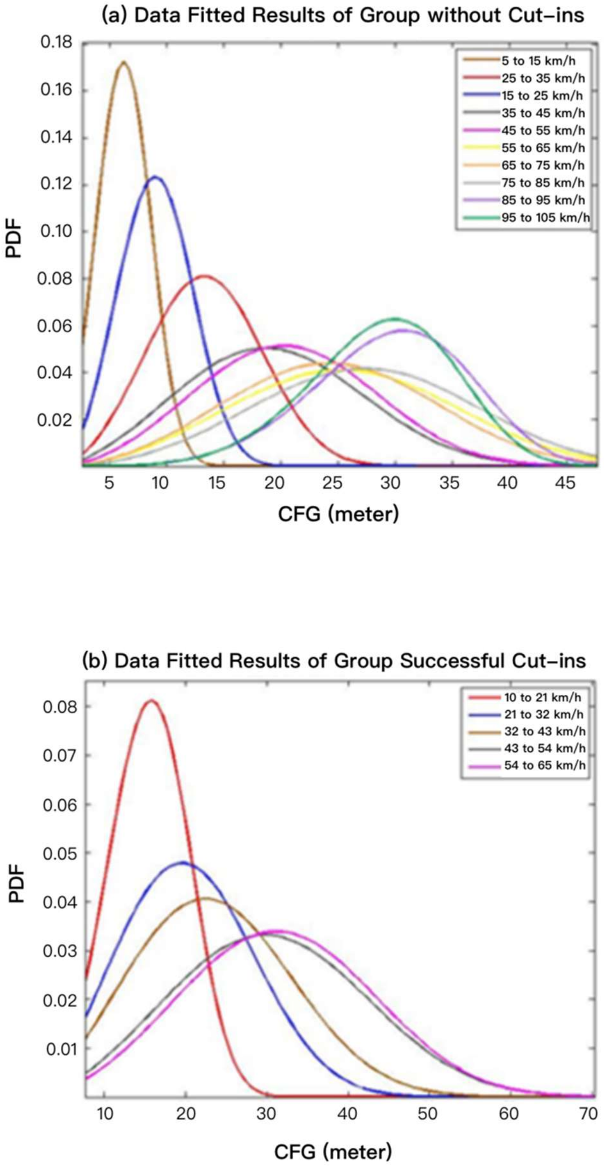


Figure 5.6 The data divided into several segments according to $\dot{\boldsymbol{x}}_{\boldsymbol{i}}$.

The data fitted results in every segment of both groups are shown in Figure 5.7. In both cases, it can be found that lower velocities in general lead to shorter car-following gaps for most segments. This matches the actual driving situation since vehicles don’t travel far at low velocities during the same reaction time.


(a) Data Fitted Results of Group without Cut−ins

(b) Data Fitted Results of Group Successful Cut−ins

Figure 5.7 Plots of PDF curves of fitted distribution model:
(a) Without Cut-ins; (b) Successful Cut-ins.

However, the difference in the two cases is that, at similar velocities, the car-following gap is larger in data with successful cut-ins than that without cut-ins. This result can be interpreted as, drivers tend to leave relatively large car-following gaps if there is a vehicle intending to cut-in, or vehicles successfully cut in because the opportunity to cut in exists due to the large gap. Bottom line, a cut-in is an important factor in formulating the strategic car-following gap model.

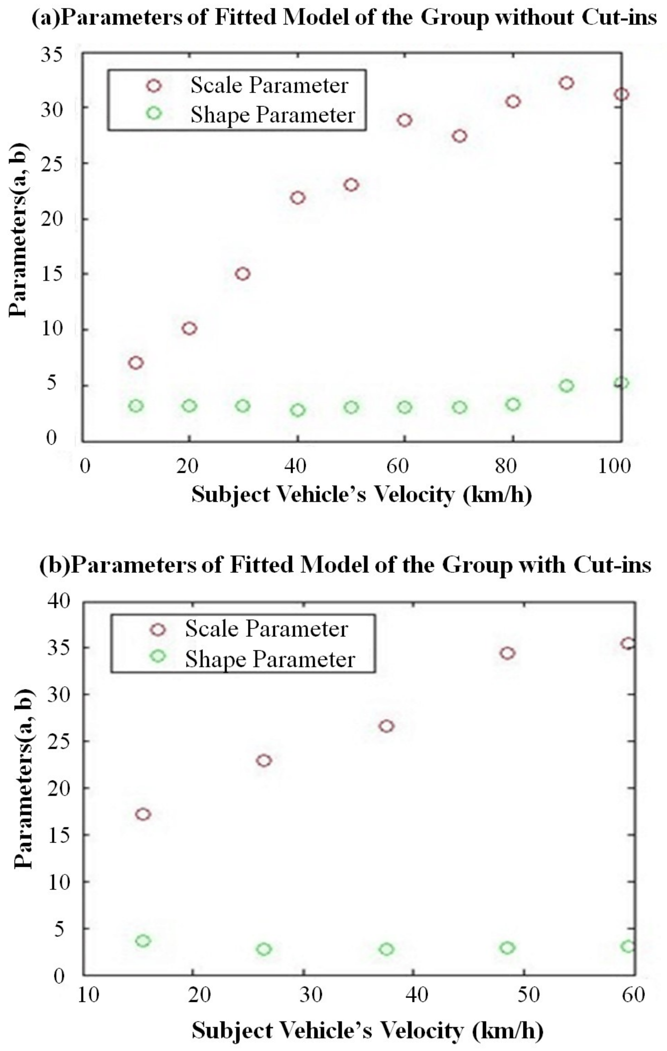


Figure 5.8 The plots of scale and shape parameters in different segments:
(a) Without Cut-ins; (b) Successful Cut-ins.

The plots of scale and shape parameters in different segments of both groups are shown in Figure 5.8. Every parameter was plotted as a function of its segments' median velocity. The variances of shape parameters in the two groups are both small, so they can be regarded as constants. The scale parameters of the fitted model are shown in equations (5.11) and (5.12).

Where, a_{NCI} is the scale parameter of the model without cut-ins, and a_{CI} is the scale parameter of the model with cut-ins. b_{NCI} and b_{CI} are shape parameters of the models without and with cut-ins, respectively, and their values are 3.44709 and 3.027836.

$$a_{NCI} = -0.0033\,\dot{x}_i{}^2 + 0.6515\,\dot{x}_i - 0.3184 \tag{5.11}$$

$$a_{CI} = -0.0031\,\dot{x}_i{}^2 + 0.6676\,\dot{x}_i + 7.4344 \tag{5.12}$$

5.8 Model Verification and Application

5.8.1 Model Verification

To verify our fitted parameters, Figure 5.9 shows how the car-following gap (CFG) varies with vehicle velocity. In particular, the figure shows two special CDF curves: the bottom one is the 5-th percentile car-following gap and the top is the 95-th percentile. It can be seen that, when holding velocity constant, the 95-th percentile car-following gap is greater than that of the 5-th percentile and both become larger as the velocity increases. In addition, since the CDF is a monotonically increasing function with the car-following gap, most data points are bounded between the 5-th and 95-th percentile curves, which demonstrates that the constructed mathematical model is able to represent the field data well. Additionally, Figure 5.10 provides the PDF curves of the model overlaid on top of the field data, allowing direct comparison of the fitted model against the field data. The comparison of PDF in Figure 5.11 reveals that it is of significance to construct a car-following gap model considering the effect of a vehicle cutting in from adjacent lanes since longer gaps are needed for cut-in vehicles to execute the intended maneuverer. Therefore, the strategic car-following gap model can be used as a tool to capture the effect of cut-ins and represent features of the field data.

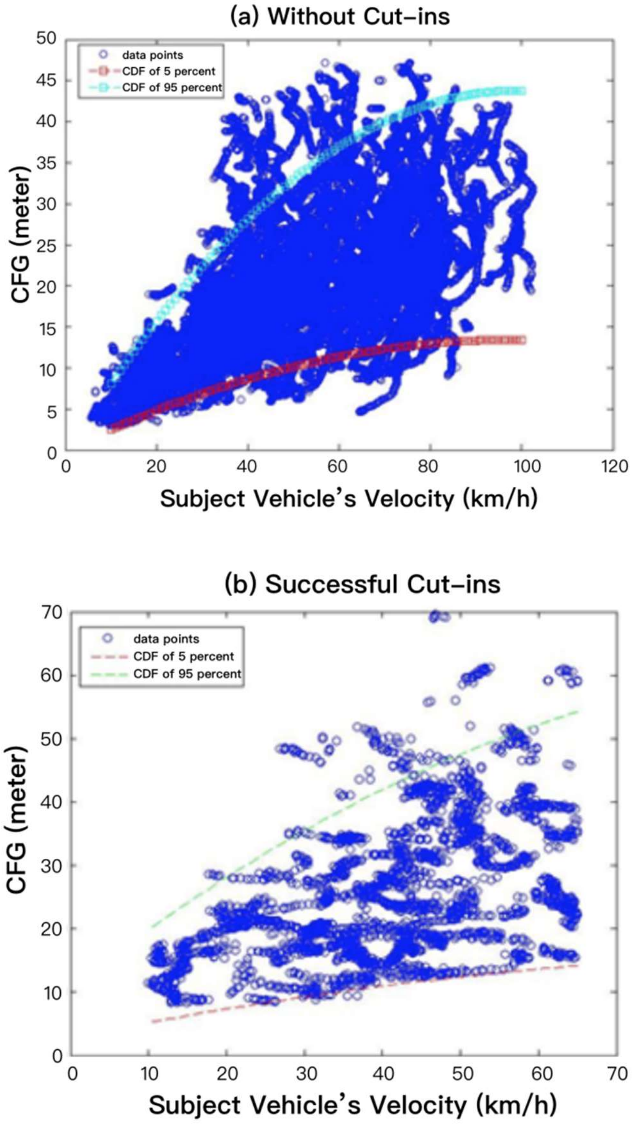


Figure 5.9 90% intervals of fitted distribution models and the empirical data: (a) Without Cut-ins; (b) Successful Cut-ins.

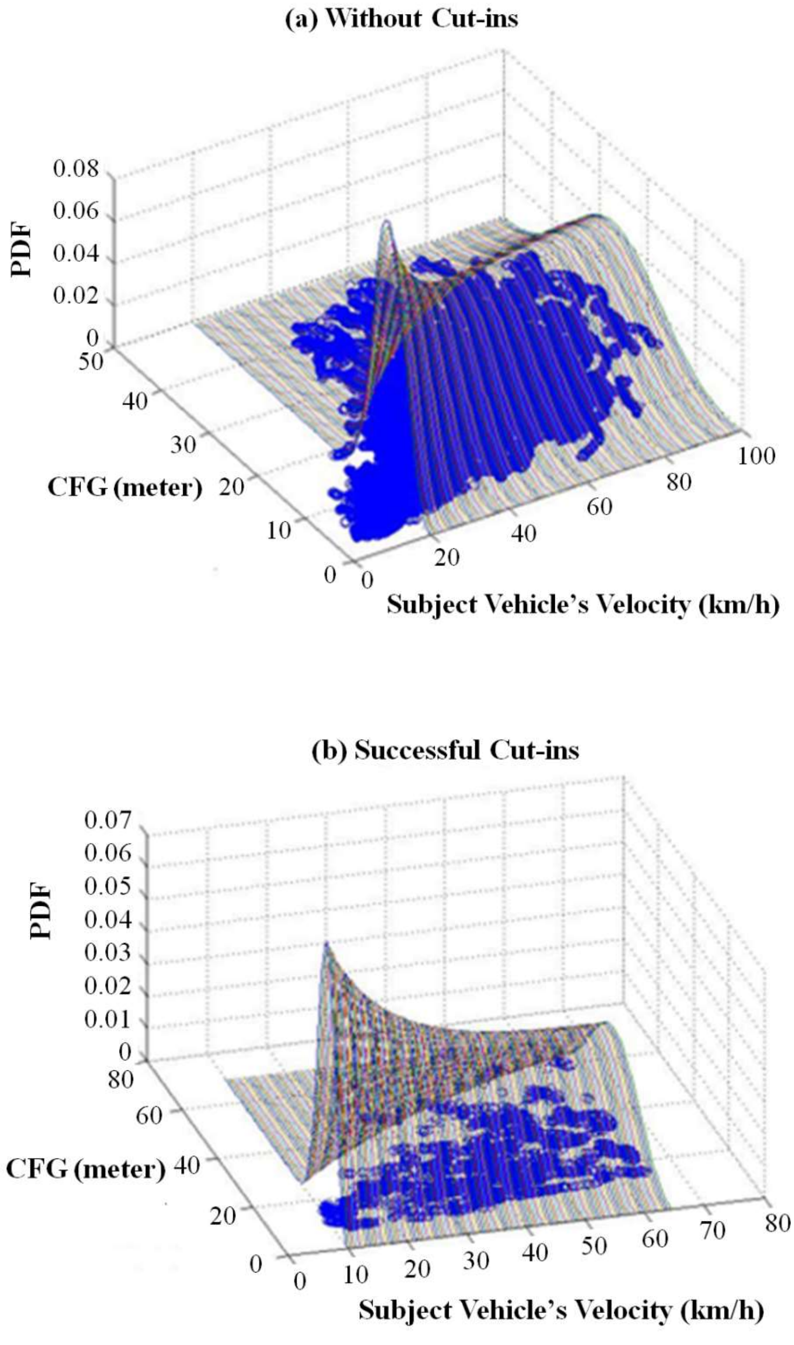


Figure 5.10 Plots of model PDF curves and the empirical data:

(a) Without Cut-ins; (b) Successful Cut-ins.

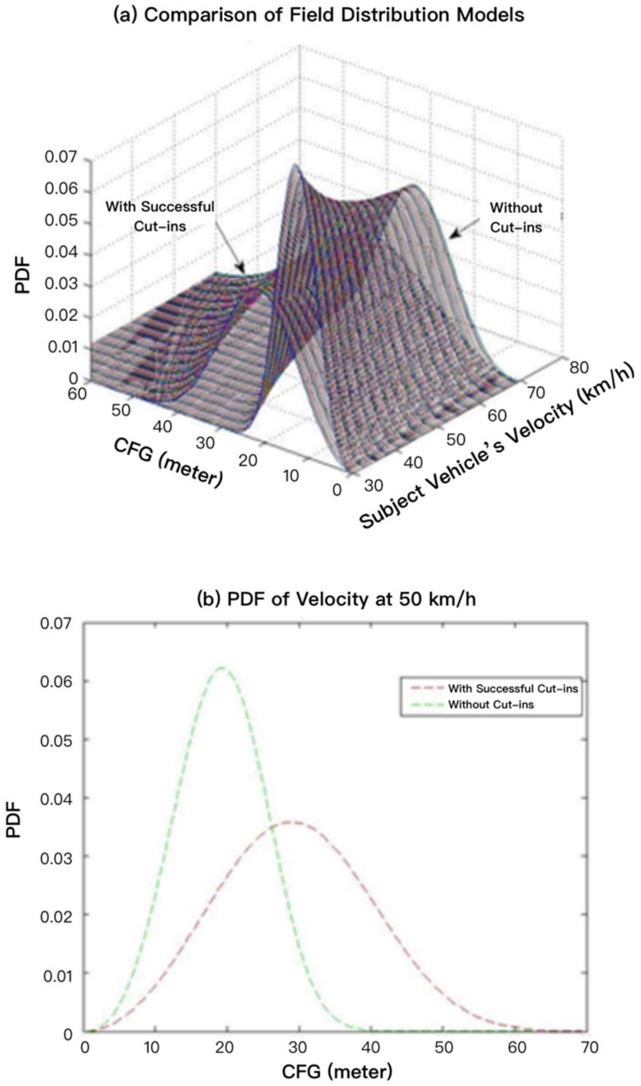


Figure 5.11 Comparison of the two fitted models:

(a) Comparison of field distribution models; (b) PDF of velocity at 50 km/h.

5.8.2 Model Application

As elaborated in Sections 5.6 and 5.7, the strategic car-following gap model formulated above can be used in advanced driver assistance systems to ensure safety in car following and minimize cut-ins from adjacent lanes, and thus smoothens traffic and reduces traffic waves which in turn saves fuel and lowers emissions.

The analysis below provides further details to facilitate the application of the proposed model that was formulated in general form in equation (5.3) and in specific form in equation (5.10). Searching for the strategic car-following gap translates to finding the optimal solution in equation (5.13), i.e. minimizing the overall objective function with a known $\dot{x}_i$.

$$\begin{cases} O\left({d_{ij}}^{*}\right) = \min O\left(d_{ij}\right) \\ s.t. 0 \leq d_{ij} \leq \infty,\ \dot{x}_i \text{ known} \end{cases} \tag{5.13}$$

Since the objective function is continuous within the constraint of the car-following gap, it can be differentiated with respect to d_{ij}:

$$\frac{\partial O}{\partial d_{ij}} = (1-w)\left(\frac{b_{CI}}{a_{CI}}\right)\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}-1} e^{-\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}}} - w\left(\frac{b_{NCI}}{a_{NCI}}\right)\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}-1} e^{-\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}}} \tag{5.14}$$

The strategic car-following gap is found when setting the first derivate of the objective function to zero, as in (5.15):

$$(1-w)\left(\frac{b_{CI}}{a_{CI}}\right)\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}-1} e^{-\left(\frac{d_{ij}}{a_{CI}}\right)^{b_{CI}}} - w\left(\frac{b_{NCI}}{a_{NCI}}\right)\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}-1} e^{-\left(\frac{d_{ij}}{a_{NCI}}\right)^{b_{NCI}}} = 0 \tag{5.15}$$

Figure 5.12 shows the strategic car-following gap at different velocities with different w, and it can be seen that the car-following gap is longer at a higher velocity and, if velocity is held constant, with a larger w value (i.e., more consideration is given to reducing the safety hazard).

Therefore, the strategic car-following gap calculated based on the subject vehicle's current velocity can be used to adjust the vehicle's position in the traffic to achieve safe car following while reducing the chance of cut-ins.

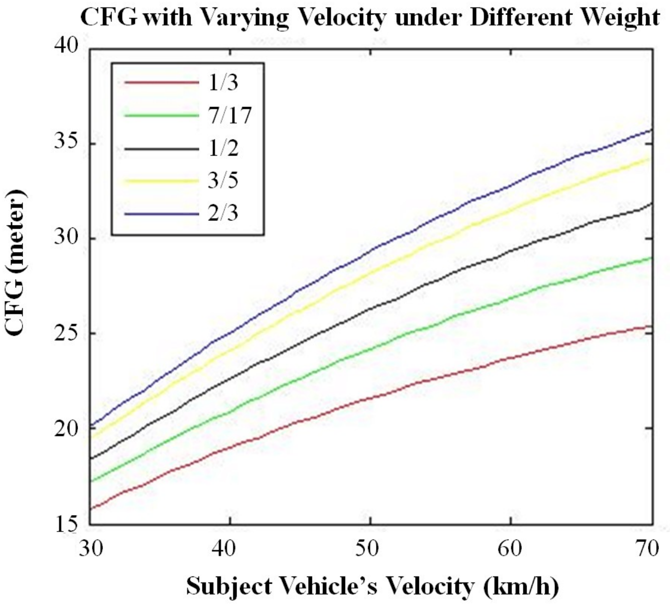


Figure 5.12 Plot of the strategic car-following gap at different velocities with different w, which is 1/3,7/17,1/2,3/5 and 2/3.

5.9 Conclusions and Discussions

In this paper, a strategic car-following gap model is proposed. It consists of the weighted effects of safety hazard and the probability of cut-ins by other vehicles. The weight can be adjusted under different scenarios so that this model is able to suit the needs of different drivers with their individual preferences and various driving situations. Field experiments were conducted, and the data was used to facilitate the specific formulation of the strategic car-following gap model along with its parameters. The results proved that the parameters of the car-following gaps without cut-ins are much different from those with successful cut-ins. More specifically, data with successful cut-ins are typically associated with longer gaps when compared with data without cut-ins, which is valid since drivers necessitate longer gaps to cut in. The good quality of the specific model is presented in comparison with the field data and plotted in cumulative form using the 5-th and 95-th percentile car-following gap curves as well as in PDF form overlaying on top of the field data. This verifies the feasibility and reliability of the proposed model.

This research contributes to the state-of-the-art of traffic flow modeling and advanced driving assistance systems in two ways. First, this research presents a strategic car-following gap model which can help seek a "perfect" driver model that operates the vehicle in an "ideal" way to achieve both safety and efficiency gains in advanced driving assistance systems. Second, the proposed model successfully incorporates the probability of cut-ins into the formulation of the car-following gaps to reveal the practical effect of lateral motions such as lane changing and cut-ins to the longitudinal operations. With the above advantages, this research can be a vitally important input to the design of advanced driver assistance systems, promoting smooth traffic flow with increased safety and reduced fuel consumption and emissions.

A future study will focus on making a more quantitative analysis and compare this with the existing models. Furthermore, we will collect more sufficient and representative data based on individual drivers' driving behaviours from actual driving conditions and further improve the performance by combining this current model with new theoretical approaches.

5.10 Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 51475254, China Scholarship Council and in part by USDOT University Transportation Center (UTC) program.

References

[1] Li Y., Zhang L., Peeta S., Pan H., Zheng T., Li Y. and He X., “Non-lane-discipline-based car-following model considering the effects of two-sided lateral gaps”, Nonlinear Dynamics, 2015, 80(1-2):227-238.

[2] Wu J., Geng S. F., Zhao Y., “A Complete Target Selection Method for ACC System Based on Statistics and Classification of Vehicle Trajectories”, Applied Mechanics & Materials, 2014, 533:316-320.

[3] Avi R., Zevi B., Claudia V. G, David J. L. and Omer T., “Learning Drivers’ Behaviour to Improve Adaptive Cruise Control”, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, 2014, 19(1):18-31.

[4] Javadi M. S., Habib S., Hannan M. A., “Survey on Inter-Vehicle Communication Applications: Current Trends and Challenges”, Information Technology Journal, 2013, 12(2):243-248.

[5] Somda F.H., Cormerais H., “Auto-adaptive and string stable strategy for intelligent cruise control”, IET Intelligent Transport Systems, 2011, 5(3):168-174

[6] Charles D. and Brahim C., “Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach”, IEEE Transactions on Intelligent Transportation Systems, 2011, 12(4):1248-1260.

[7] Yu S., Shi Z., “The effects of vehicular gap changes with memory on traffic flow in cooperative adaptive cruise control strategy”, Physica A: Statistical Mechanics and its Applications, 2015, 428(15):206-223.

[8] Li Y., Kang Y., Yang B., Peeta S., Zhang L., Zheng T. and Li Y., "A sliding mode controller for vehicular traffic flow", Physica A: Statistical Mechanics and its Applications, 2016, 462:38-47.

[9] Li Y., Zhang L., Zheng H., He X., Peeta S., Zheng T. and Li Y., "Evaluating the energy consumption of electric vehicles based on car-following model under non-lane discipline", Nonlinear Dynamics, 2015, 82(1-2): 629-641.

[10] Jin I.G. and Orosz G., "Dynamics of connected vehicle systems with delayed acceleration feedback", Transportation Research Part C: Emerging Technologies, 2014, 46:46-64.

[11] Gu H., Haiyan, Jin P. J., Wan X., Shah J., Ran B., "A Leading Vehicle Model for Comfortable Acceleration Among Cooperative Adaptive Cruise Control (CACC) Vehicle Platoons", Transportation Research Board 94th Annual Meeting, Washington DC, United States, 2015, 01558265.

[12] Alireza K., Ali G. , Reza K., Fatemeh A., Reinhard B., "Improved adaptive neuro fuzzy inference system car-following behaviour model based on the driver-vehicle delay", IET Intelligent Transport Systems, 2014, 8(4): 323-332.

[13] Biagio C., Vincenzo P., Marcello M., "Global sensitivity analysis techniques to simplify the calibration of traffic simulation models. Methodology and application to the IDM car-following model", IET Intelligent Transport Systems, 2014, 8(5): 479-489.

[14] Li Y., Zhang L., Peeta S., He X., Zheng T. and Li Y., "A car-following model considering the effect of electronic throttle opening angle under connected environment", Nonlinear Dynamics, 2016:1-11.

[15] Gennaro N. B. , Luigi P., Mark B., Michael M., “Driving behaviour models enabling the simulation of Advanced Driving Assistance Systems: revisiting the Action Point paradigm”, Transportation Research Part C: Emerging Technologies, 2013, (36):352-366.

[16] Tang T.Q., Shi W.F. Shang, H.Y. and Wang Y.P., “An extended car-following model with consideration of the reliability of inter-vehicle communication”, Measurement, 2014, (58):286-293.

[17] Li Y., Zhu H., Cen M., Li Y., Li R. and Sun D., “On the stability analysis of microscopic traffic car-following model: a case study”, Nonlinear Dynamics, 2013, 74(1-2):335-343.

[18] Tang T., Shi W., Shang H. and Wang Y., “A new car-following model with consideration of inter-vehicle communication”, Nonlinear dynamics, 2014, 76(4):2017-2023.

[19] Ma X., “Driver modeling based on computational intelligence approaches, PhD thesis”, Royal Institute of Technology, Stockholm, Sweden, 2006.

[20] Reusehel A., “Vehicle movements in a platoon with uniform acceleration or deceleration of the lead vehicle”, Zeitschrift des Oesterreichischen Ingenieur-und Architekten-Vereines., 1950, 95: 73-77.

[21] Pipes L.A., “An operational analysis of traffic dynamites”, Journal of Applied Physics, 1953, 24(3): 274-281.

[22] Li Y., Zhang L., Zhang B., Zheng T., Feng H. and Li Y., “Non-lane-discipline-based car-following model considering the effect of visual angle”, Nonlinear Dynamics, 2016, 85(3):1-12.

[23] Chandler, Herman & Montroll, "Traffic dynamics: studies in car following", research, 1958, 6(2):165-18.

[24] Kometani E., Sasaki T., "Dynamic behaviour of traffic with a nonlinear spacing-speed relationship. In Proceedings of the Symposium on Theory of Traffic Flow", Research Laboratories, General Motors, New York: Elsevier, 1959:105-119.

[25] Brackstone M., Sultan B., McDonald M., "Motorway driver behaviour: studies on car following", Transportation Research Part F: Traffic Psychology and Behaviour, 2002, 5(1):31-46.

[26] Mairi P., "Identification and modeling of behavioural changes in longitudinal driving through vehicle-based signals", Delft University of Technology, Master's Thesis.

[27] Kikuchi, C., Chakroborty, P., "Car following model based on a fuzzy inference system", Transportation Research Record, 1992:82-91.

[28] Peng J. , Guo Y., Fu R., Yuan W., Chang Wang, "Multi-parameter prediction of drivers' lane-changing behaviour with neural network model", Applied Ergonomics, 2015, 50: 207-217.

[29] Sivaraman, S.; Trivedi, M.M., "Dynamic probabilistic drivability maps for lane change and merge driver assistance", IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5):2063-2073.

[30] Yu, Chenfei, and J. Wang. "Drivers' car-following correlative behavior with preceding vehicles in multilane driving." 2014 IEEE Intelligent Vehicles Symposium (IV) IEEE, 2014.

[31] Ge H., Meng X., Zhu H., Li Z., “Feedback control for car following model based on two-lane traffic flow”, Physica A: Statistical Mechanics and its Applications, 2014, 408(32):28-39.

[32] Ngoduy D., “Linear stability of a generalized multi-anticipative car following model with time delays”, Communications in Nonlinear Science and Numerical Simulation, 2015, 22:420-426.

[33] Wang J.Q., Xiong C.F., Lu m., Li K.Q., “Longitudinal driving behaviour on different roadway categories: an instrumented-vehicle experiment, data collection and case study in China”, IET Intelligent Transport Systems, 2015, 9(5):555-563.

Chapter 6

Conclusions and Future Work

This chapter demonstrates the main conclusions and contributions derived from the research shown in the aforementioned chapters, followed by displaying our perspective on future work inspired by our findings.

6.1 Contributions

In this thesis, three challenging problems in intelligent vehicles (mandatory lane changing (MLC), discretionary lane changing (DLC), and car-following) are investigated and developed by AI algorithms and statistical modeling. A compact gated branch neural network for MLC, a deep learning recurrent neural network for DLC, and a statistical model for the car-following gap are established for these three scenarios, respectively. The contributions of research work on these three problems are summarized as follows:

1. Mandatory Lane Changing Suggestion at the On-ramps of Highways

1) Based on correlation analysis, an additional gated branch neural network (GBNN) algorithm is proposed. The gated branch offers effective feature learning and the ability to explicitly incorporate the interplay between the surrounding driving environment and the lane changing decision.

2) Featuring high accuracy in both events of merge and non-merge, the proposed GBNN algorithm is used to model the MLC behaviour at the on-ramps of highways. The accuracy could be as high as 97.7% for non-merge and 96.3% for merge behaviour. In comparison with other AI algorithms like CNN (AlexNet), SVM, and

ANN, the proposed algorithm yields much more accurate results. Furthermore, the proposed method is computationally cost-effective, compared with the existing deep learning algorithms, and thus can be applied in ADAS for an efficient MLC suggestion system.

2. Prediction of the Surrounding Vehicle's Discretionary Lane Changing Intention at Freeway

1) A recurrent neural network (RNN)-based time series classifier with a gated recurrent unit (GRU) is developed to classify and predict the surrounding vehicles' discretionary lane changing intention beforehand, and thus provides an early notification to the ego-vehicle for driving assistance.

2) The proposed algorithm is capable of predicting the surrounding vehicles' lane changing maneuver 0.8 s ahead of time at a recall and precision of 99.5% and 98.7%, respectively. The model can predict the lane changing intention 1.6 s in advance with a recall of 92.2%. Furthermore, the proposed method is lightweight in computation, being a suitable candidate to be applied in ADAS or autonomous vehicles in real-time applications.

3. Car-Following Gap Model Considering the Effect of Cut-ins from the Adjacent Lanes

1) A strategic car-following gap is devised by incorporating the probability of cut-ins into the formulation to reveal the practical effect of lateral motions. This model can successfully seek a driver model that operates the vehicle in a way to achieve both safety and efficiency.

2) An overall objective function consisting of the car-following gap and velocity is formulated by considering the safety hazard and the probability of cut-ins by other vehicles. Based on this, seeking the strategic car-following gap translates to finding

the optimal solution that minimizes the overall objective function. With the support of field data, most collected data points are bounded between the 5-th and 95-th percentile curves of the proposed model, which demonstrates the effectiveness of the constructed model. Thus, this model can be a vitally important input to advance the design of driver assistance systems, promoting smooth traffic flow with increased safety.

6.2 Conclusions

Lane changing and car following, the two most frequently encountered driving behaviours for intelligent vehicles, have become the focus in research because their errors are accountable for a large amount of traffic casualties. As security and driving comfort have been elevated as top priorities for the development of intelligent vehicles, this thesis aims for providing better solutions to these two challenging scenarios, and advances driver models by introducing novel algorithms for safer and more efficient driving assistance system development.

This thesis is organized by following the natural behaviour sequences of a driver entering from the on-ramp of the highway, staying in the mainline with awareness of the surrounding vehicles' lane changing intention, and keeping a reasonable gap to the preceding vehicle at the end. More specifically, the corresponding three typical scenarios have been addressed in the thesis, including Mandatory Lane Changing (MLC) suggestion at the highway entrance, Discretionary Lane Changing (DLC) intention prediction, and the car-following gap model considering the effects of cut-ins from adjacent lanes.

A mandatory lane changing suggestion model at highway entrance was introduced in Chapter 3. This mandatory lane changing suggestion model is based on a newly proposed lightweight GBNN algorithm and is validated using the real-world datasets of

U.S. Highway 101 and Interstate 80 from Federal Highway Administration's Next Generation Simulation (NGSIM). This model is capable of not only accurately modeling MLC behaviours at the on-ramps of highways but also temporally and spatially featuring the driving environment to correlate it with the lane changing decision. The non-merge accuracy, merge accuracy, and receiver operating characteristic score of the prediction model using the newly proposed GBNN algorithm are 97.7%, 96.3%, and 0.990, respectively. The results are more accurate than other artificial intelligence algorithms such as CNN (AlexNet), SVM, and ANN and faster than the AlexNet based on deep learning algorithms. The proposed GBNN algorithm achieves this goal due to the introduction of an additional gated branch to the main neural network and the adoption of SeLU activation function and the Adam optimizer. A mandatory lane changing suggestion model, Such as this, is anticipated to be applicable in ADAS allowing efficient MLC suggestions.

After entering the highway from the on-ramp, the driver will put more attention to the driving environment and be aware of the surrounding vehicles' lane changing intention. For Discretionary Lane Changing (DLC) intention prediction at the highway derived in Chapter 4, a recurrent neural network (RNN)-based time series classifier with a gated recurrent unit (GRU) is proposed and evaluated by using the real-world datasets of U.S. Highway 101 and Interstate 80 from Federal Highway Administration's Next Generation Simulation (NGSIM). The newly developed prediction model enables us to predict the surrounding vehicles' lane changing intention of 0.8 s in advance at a recall and precision of 99.5% and 98.7%, respectively, and as early as 1.6 s in advance with a recall of 92.2%, outperforming other classical rival algorithms such as LSTM and HMM. This early notification of the surrounding vehicles' DLC behaviour is beneficial for drivers to take actions beforehand and will improve the safety and efficiency of driving on freeways.

Once the driver finishes lane changing, car-following behaviour, the next scenario, is worth investigating by considering both effects from the longitudinal direction and lateral direction. In Chapter 5, a strategic car-following gap model is proposed. It consists of the weighted effects of safety hazard and the probability of cut-ins by other vehicles. The weight can be adjusted according to different scenarios so that this model is able to meet the needs of different drivers with their individual preferences and various driving situations. Field experiments have been conducted, and the data is used to facilitate the specific formulation of the strategic car-following gap model along with its parameters. The results prove that the parameters of the car-following gaps without cut-ins are much different from those with successful cut-ins. More specifically, the data with successful cut-ins is typically associated with longer gaps when compared with data without cut-ins, which is valid since drivers necessitate longer gaps to cut in. The good performance of the specific model is presented in comparison with the field data and plotted in cumulative form using the 5-th and 95-th percentile car-following gap curves as well as in probability density function (PDF) form overlaying on top of the field data. This verifies the feasibility and reliability of the proposed model.

6.3 Future Work

Based on the aforementioned work in the thesis, future work can be anticipated and carried out in the following topics.

6.3.1 Enhancing the applicability of AI-based prediction models to complex real-life conditions

To further put our models into real-life use in the future, the models based on our newly proposed algorithms should be applicable and accurate in different weather and various road conditions or scenes. Although our models of lane changing have been evaluated by the Federal NGSIM datasets from U.S. Highway 101 and Interstate 80, the data only comes from one sensor, a CCD camera. Limited by the datasets in this thesis, the robustness of the models needs to be fully validated in future work. Targeting more secure, intelligent and efficient vehicles to be suitable for very complicated driving conditions, more sensors such as LIDAR, millimeter wave radar and ultrasonic sensors have to be installed to collect comprehensive data of both the subject vehicle and the surrounding driving vehicles. In the near future, fully vehicle-to-vehicle communication may become accessible, allowing mutual sharing of driving conditions with each other to be integrated into our models. More data input may decrease the computational speed while increasing the accuracy of the proposed models. This means models optimized for parallel computation should be developed to suit complex conditions. Hence, consideration of novel algorithms in combination with our algorithms and other algorithms that can take into account more feature variables are greatly demanded and will be well-performed in the future.

6.3.2 Defining, choosing and enriching variables for accurate lane changing in complex conditions

Since there is a trade-off between accuracy and prediction time for lane changing, defining and choosing effective feature variables are of great importance for our prediction model. In the future, more databases will be integrated into our model to classify variables into groups to fit normally uniform-speed driving conditions and extreme speed-up conditions. In the future, more effective variables related to road conditions will also be included in our model to make it suitable for different highways and weather conditions.

6.3.3 Optimization of the strategic car-following model

As mentioned in Chapter 5, the strategic car-following gap model is effective to adjust the vehicle’s position in traffic flow to keep safe distance while reducing the chance of cut-ins. It is noteworthy that the key of the model is the PDF of the car-following gap under different driving scenes. In other words, the more data we have, the more accurate the PDF will be. Hence, in the future, more real-life experimental data will be used for optimization of our model. Also, whether the Weibull distribution is the most suitable distribution in our model will need to be verified by big data input.

Although the strategic car-following gap model can help vehicles adjust their positions in time to prevent cut-in behaviour, it will add complexity to the following cars and may induce a “butterfly” effect on traffic. Hence, developing an advanced algorithm to avoid chain effects of large-scale movements of vehicles and in some conditions that can allow deliberate cut-ins will be conceived in the future.

In summary, we will improve our algorithms by incorporating more real-life data and develop them as a subsystem of ADAS for intelligent vehicles. The efforts on intelligent vehicles are worthwhile due to the promising safety, intelligence, and comfortable driving experiences that they offer.