PREDICTION OF MANDATORY LANE CHANGING BEHAVIOR USING ARTIFICIAL NEURAL NETWORK MODEL

PREDICTION OF MANDATORY LANE CHANGING BEHAVIOR USING ARTIFICIAL NEURAL NETWORK MODEL

By

Yile Wang, B.Eng.

A Thesis
Submitted to the School of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree of
Master of Applied Science

McMaster University

Master of Applied Science (2017)	McMaster University
(Mechanical Engineering)	Hamilton, Ontario, Canada

TITLE:	Prediction of Mandatory Lane Changing Behavior Using Artificial Neural Network Model
AUTHOR:	Yile Wang
SUPERVISOR:	Dr. Fengjun Yan
NUMBER OF PAGES:	ix, 77

Abstract

Recently, the applications of some driver assistance systems on vehicles have reduced vehicle accidents. However, studies have shown that the number of vehicle accidents caused by improper lane-changing behavior remains at a high level. Therefore, research has been focusing on developing a lane-changing assistance system to increase the safety level of driving in traffic. Many researchers have attempted to predict lane-changing behavior, and a general trend in the study of predicting driving behavior is the greater application of computational artificial intelligence.

Artificial Neural Network (ANN) is one of the artificial intelligence methods, and it is well-known for its high reliability in a variety of applications. An ANN model can mimic human thinking and behavior due to its ability to capture the complex relationship among different variables in an environment of uncertainty. In this thesis, a BP (back-propagation) Neural Network model established by two methods was developed to predict a driver's mandatory lane-changing decisions (merge or non-merge) at an early stage by considering driving environment features as the input vectors. Vehicle trajectory data from the Next Generation Simulation (NGSIM) dataset was used for training and testing the model. The results of the proposed model indicated that the prediction accuracies in advance of a driver's actual driving behavior with a lead time of 1s, 1.5s, and 1.8s were at 89.6%, 84.9%, 78.8% for merge events, and for non-merge events were at 92.2%, 87.5%, 81.1% respectively.

Acknowledgements

I would like to give special thanks to my supervisor Dr. Yan whose encouragement, guidance, and support sustained me through the Master research at McMaster University. He carefully read the whole draft of the thesis and offered painstaking and precious criticism.

Many thanks to the Department of Mechanical Engineering and its fellows, as well as McMaster University, for two years of beautiful memories. I also wish to sincerely thank one of my group members Yangliu Dou, whose brilliant ideas and perceptive observations were immensely constructive.

Finally, I wish to dedicate this thesis to my beloved family who has given me life, love, and support.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Research Motivation

Road crashes are increasingly costly in the global economy and are a growing societal concern. According to annual global road crash statistics, nearly 1.3 million people die in road crashes each year around the world—on average 3,287 deaths occur every day and an additional 20–50 million are injured or disabled [1]. Road traffic crashes rank as the 9th leading cause of death and account for 2.2% of all deaths globally [1]. Since the number of vehicle crashes remains at a high level and the cost is rapidly increasing, road safety is pushed to the forefront of the debate related to transportation development priorities [1].

For safe driving, it is necessary that drivers perceive the relevant objects of a situation, comprehend the meaning of these objects to form a holistic understanding of the current situation, and predict the future development of the situation. At present, due to the development of computer science and the popularity of driver-assisted vehicles, driver assistance systems are being widely researched. Several examples are Adaptive Cruise Control (ACC) [2], Rear-end Collision Avoidance systems [3], Emergency Driver Assistant, Intelligent Speed Adaptation or Intelligent Speed Advice (ISA) [4], and Lane-Changing Assistance systems [5].

Researchers have also estimated that lane-changing crashes account for 4% to 10% of all vehicle crashes in USA [6]. Although, this value is not very high, the delay time it causes accounts for 10% of the total time caused by all traffic accidents.

Lane change is defined as a driving execution that moves a vehicle from one lane to another where both lanes have the same direction of travel [7]. It is a driving behavior that can depend on the kinetics of multiple vehicles. Successful completion of a lane change requires attention to vehicles in the both the original lane and the adjacent lane. The fact that drivers could only look in one direction at a time, the presence of blind spot and human perception limitations lead to anomalous behavior.

Lane Change Maneuvers (LCMs) are usually classified as either mandatory or discretionary (forced or unforced). When a lane change is required due to, for example, a lane drop or yielding to traffic near a ramp, it is called a Mandatory Lane Change (MLC) [7]. A lane change that is intended to improve the perceived driving conditions not urgently (e.g. passing a slow lead vehicle) is called a Discretionary Lane Change (DLC). Statistical analysis has shown that there are obvious distinctions between these two kinds of maneuvers for vehicles due to the difference in driving situations (forced or self-initiated) [7].

In the main lanes of a freeway, the traffic flows efficiently and the road curvature changes gently. This means that it is relatively safe for drivers to do DLCs in the main lanes. In contrast, the traffic volume in the merging area of a freeway on-ramp is large, and the traffic conditions are complex. MLCs from the on-ramp often bring about the confusion of mainline traffic flow, the reduction in speed, the traffic delay and accidents. Statistical analysis showed that MLCs are more aggressive in gap acceptance and lane-changing execution than DLCs, and MLCs are approximately 4.5 times more likely to have a critical gap condition than DLCs [7]. According to the Federal Motor Carrier Safety Administration (2010) [7], vehicle crashes caused by MLCs are consistently the leading cause of occupational lane-changing crashes, accounting for 65% of all lane-changing crashes. Therefore, increasing the safety level of MLCs is significant in achieving congestion-free and accident-free traffic situations.

Recently, several lane-changing assistance systems have been developed in an attempt to ensure driver safety during the lane-changing process. Such systems operate by monitoring the related conflict vehicles with millimetre-wave radar or high-accuracy cameras. Once turn signals are recognized, the lane-changing assistance system may assume that a lane change will be executed at some time in the near future, and the system enters its working mode. When a conflict object is detected by the system within a given distance, warning signals are sent to remind the drivers of the potential danger [8]. However, in practice, prediction of lane-changing behavior on the basis of turn signals is unreliable. According to an experiment conducted under realistic on-road conditions, the operation rate of drivers' turn signals is below 5% by the initiation of the lane-changing behavior, which certainly affects the warning accuracy rate and reliability of a lane-changing assistance system which is triggered by turn signals [8].

In recent years, researchers have pursued ways to improve the performance of lane-changing assistance systems by predicting drivers' lane-changing behavior. Specifically, researchers have tried to predict drivers' lane-changing behavior via drivers' visual search behavior and vehicles' motion characteristics [9]. Based on gaze data of interest zones, Lethaus and Rataj [10] predicted drivers' lane-changing behavior via compound model, and they have proved that drivers may pay more attention to side mirrors than to inside mirrors when executing leftward lane changes. Doshi and Morris [11] proposed that in addition to eye movements, head movements could be used to detect drivers' lane-changing behavior. They developed a real-time on-road prediction system to detect drivers' lane-changing behavior.

If drivers' lane-changing behavior could be precisely predicted from their driving behavior, rather than relying on turn signals, the performance of lane-changing assistance systems could be greatly improved. However, challenges have arisen in the practical application of lane-changing prediction technologies [12]. Although an assistance system could detect a driver's lane-changing behavior, this does not mean that a lane change will definitely occur at some time in the future. Even if a driver intends to execute a lane change, he/she may change he/her mind for various reasons, such as a vehicle rapidly approaching from behind in the target lane. This is called the " intent revocation phenomenon" [12], and it cannot be easily predicted using the existing prediction technologies.

Researchers have developed other methods to predict a driver's lane-changing behavior. They have focused on using driving environment features to predict a driver's lane-changing decisions (merge or non-merge), and they have demonstrated that a driver's lane-changing decisions are essentially triggered by his/her surrounding driving environment features [13]. For example, a driver decides to change lanes only if there is an acceptable merging gap appearing in the target lane. In this case, the merging gap belongs to one of the surrounding driving environment features. Therefore, surrounding driving environment features can be used to predict a driver's lane-changing decisions.

Recently, a general trend in the study of predicting lane-changing behavior is the greater application of computational artificial intelligence. Since a driver's mental and physical behavior is non-deterministic and highly non-linear, it is difficult for traditional methods to embody this kind of uncertain relationship. The Artificial Neural Networks, fuzzy logic theory, and dynamic Bayesian networks, which include well-known hidden Markov models, have attracted many researchers to do the related research [14].

Some lane-changing prediction models have been proposed in previous studies [6]. However, few studies focused on the lane-changing prediction in merging areas of freeway on-ramp. Also, several experiences showed that the existing prediction models yield unsatisfactory performances [15]. Moreover, some of the existing models were only able to accurately predict a driver's lane-changing behavior after it actually happens [16]. Thus, the primary motivation of this thesis was to develop an accurate model for predicting a driver's mandatory lane-changing decisions in advance of the driver's actual driving behavior in the merging area of a freeway on-ramp.

1.2 Literature Review

An extensive literature review (in chronological order) of lane-changing models is presented in this section of the thesis. Gap acceptance was the principal focus of research in modeling driver lane-changing behavior. These models were based on the assumption that a driver makes a lane change when both the lead and the lag gaps in the target lane are acceptable.

1.2.1 Lane-Changing Models

Given the assumption on the distribution of critical lead and lag gap lengths, various gap acceptance models were developed in the 1960s and 1970s [17]. Herman and Weiss [18] assumed an exponential distribution for the critical gap, Drew [19] assumed a lognormal distribution, and Miller [20] assumed a normal distribution.

In 1981, Daganzo [21] modeled drivers' lane-changing from the minor leg of a stop controlled T-intersection to the major leg using a probit model for estimation of the gap acceptance model parameters. The function form of the critical gap for driver n at time t is assumed as:

$$G_n^{cr}(t) = G_n + \varepsilon_n^{cr}(t) \tag{1.1}$$

where G_n is the component of critical gap attributable to driver n and $\varepsilon_n^{cr}(t)$ is the random term that varies across different gaps for a given driver as well as across different drivers. All the factors in the function form are assumed to be mutually independent and normally distributed. However, this model was affected by the problem that non-negative estimated critical gap lengths were not guaranteed.

Mahmassani and Sheffi [22] solved Daganzo's problem by neglecting panel data formulation. The critical gap was assumed to have a normal distribution. The mean of the critical gap was formulated as a function of explanatory variables that affected driver gap acceptance behavior.

Gipps [23] designed a lane-changing decision model that has been widely implemented in a microscopic traffic simulator. The model was developed to model a driver's lane-changing behavior at a variety of traffic conditions such as traffic signals, obstructions, and the presence of heavy vehicles. Necessity, desirability and safety were included as three major factors in the lane-changing decision-making process. Drivers may face conflicting goals in some driving conditions. The model deterministically prioritized different goals. However, the model did not consider the inconsistency (a driver may behave differently under an identical condition at a different time) and heterogeneity (different drivers may behave differently under an identical condition) in driver behavior.

Kita [24] modeled drivers' lane-changing behavior at freeway on-ramps using a logit model to estimate the gap acceptance model. The random utility was formulated as a function of explanatory variables that have an impact on a driver's lane-changing behavior. In addition to the gap length, the relative speed of the merging vehicle to the mainline vehicles and the remaining distance to the end of the merging lane were found to affect a driver's lane-changing behavior.

Yang and Koutsopoulos [25] established a rule-based lane-changing model. The model only applied to driver lane-changing behavior in freeways. In their model, lane changes were classified as either mandatory lane changes (MLCs) or discretionary lane changes (DLCs). They employed a probabilistic framework instead of prioritizing deterministically when drivers face conflicting goals. A gap acceptance model was developed for both MLCs and DLCs. However, they did not propose a formal parameter estimation framework.

Ahmed et al. [26] developed a general lane-changing model that captured drivers' lane-changing behavior under both MLCs and DLCs situations. A driver's decision-making process for lane-changing was described as a sequence of four steps: deciding to make a lane change, choosing a target lane, finding an acceptable gap, and performing the lane-changing behavior. They employed a discrete choice model to model the decision elements. Due to the difficulty of obtaining the indicator to differentiate the first step and the fourth step, the utilities capturing these two steps cannot be uniquely identified. The parameters of the model were estimated only for lane-changing at freeway on-ramps by using the data collected from a site at Interstate 95 northbound near Baltimore-Washington Parkway in 1983 [26]. Since in this case drivers have already made the decision to change lanes, the decision-making process involved only two steps: finding an acceptable gap and performing the lane-changing behavior. The model followed the same rule as stated in Yang and Koutsopoulos's model.

Kita [27] developed a game-theoretic lane-changing model. In the model, the behavior of a pair of merging and through vehicles was described at the same time. Both the merging and through vehicles attempted to make decisions that were best for themselves by predicting each other's action. This model has the advantage of building a simpler model by separating the direct and indirect impacts.

With the development of advanced intelligent methods, Hidas [28] used intelligent-agent-based techniques to model driver lane-changing behavior. The model was built using data with microscopic details of lane-changing and weaving maneuvers under congested traffic conditions. He also proposed to classify lane-changing maneuvers into mandatory, free, and cooperative lane changes. The model was implemented in the ARTEMiS traffic simulator, and the results demonstrated its capability to reproduce the observed behavior of individual vehicles in both freeways and urban arterial networks.

Toledo et al. [29] developed an integrated driving behavior model that captured both lane-changing and acceleration behavior. The model framework proposed concepts of short-term goal and short-term plan. The short-term goals were defined by the target lanes, and the short-term plans were defined by various gaps in traffic in the target lane. Drivers were assumed to accomplish short-term goals by conceiving and performing short-term plans by adapting their acceleration behavior to select target lane gap.

A summary of findings from the proposed lane-changing models introduced above is presented as follows. First, the primary attention of the research has been on modeling driver gap acceptance behavior. Second, the majority of driver lane-changing models were developed based on discrete choice models (i.e. probit and logit models) which needed prior assumptions about the distribution of critical gaps. The estimation for model parameters was difficult. Third, it was hard to capture a driver's lane-changing decision-making process due to the difficulty to distinguish the decision of making a lane change and performing a lane-changing maneuver. Fourth, most of the proposed models mainly concerned about how to complete a lane-changing behavior. However, a lane-changing maneuver follows a hierarchy of processes, with each process being a combination of several performances at different levels. For example, a driver plays a major role in a lane-changing process. He/she must first decide whether to execute a lane-changing behavior and predict the behavior of other vehicles. This is also the need of the next generation of advanced lane-changing assistance systems. Fifth, few lane-changing models took into the variable of the distance from the merging vehicle to the end of the merging lane in a mandatory lane-changing situation, while it is intuitive that a driver's lane-changing behavior is more aggressive when the driver approaches the end of the merging lane. Last, most of the lane-changing models were not developed using congested traffic data, thus they cannot be applied to lane-changing situations in a congested traffic.

To satisfy the requirements of developing a more intelligent lane-changing model, a general trend in the study of predicting lane-changing behavior has been the greater application of computational artificial intelligence. Since a driver's mental and physical behavior is non-deterministic and highly non-linear, it has been difficult for traditional methods to embody this kind of uncertain relationship. The Artificial Neural Networks, fuzzy logic theory, and dynamic Bayesian networks, which include well-known hidden Markov models, have been widely used. For example, Tezuka et al. [30] developed a method to infer driver behavior with a driving simulator to evaluate continuous time-series steering angle data at the time of lane-changing. The proposed method used a static type conditional Gaussian model on Bayesian networks.

Moreover, Kuge et al. [16] proposed hidden Markov models (HMMs) using observations of vehicle parameters and lane positions to model trajectory of vehicles. Sathyanarayana et al. [31] proposed a method to model driver behavior signals using hidden Markov models. The hierarchical framework and initial results can encourage more investigations into driver behavior signal analysis and related safety systems employing a partitioned submodule strategy.

Pentland and Andrew [32] proposed that human behavior can be accurately described as a set of dynamic models sequenced together by a Markov chain. They considered the human as a device with a large number of internal mental states and used the dynamic Markov models to predict human behavior a few seconds into the future.

Macadam and Johnson [33] demonstrated the use of elementary neural networks (a two-layer back-propagation) to represent a driver's steering behavior in both lane-changing maneuvers and S-curve maneuvers. Due to the limited data source for neural networks, they concluded that the adaptive nature of neural networks should be used for modeling driver steering behavior under a variety of operation scenarios.

Tomar et al. [34] proposed a method to give the future lane-changing trajectory accurately for discrete patches using a multilayer perceptron (MLP). The proposed multilayer perception network was a simple single input, single output network based on a single hidden layer and was used for training, testing, and prediction of the vehicle trajectories. In addition to using neural networks, Wen Yao [35] used hidden Markov models to predict lane-changing trajectory based on the human driving data.

Kumar [36] proposed a solution to lane-changing behavior prediction based on a combination of a multiclass SVM classifier and Bayesian filter. This approach was able to predict a driver's lane-changing decisions on average 1.3s in advance. Also, Meng and Weng [37] adopted the statistical methods to predict a driver's lane-changing decisions near work zone tapers.

In summary, although there were many lane-changing models proposed by researchers, few models concerned about the lane-changing prediction in merging areas of freeway on-ramps. To solve this problem, Y. Hou [14] has done some excellent work in this field with Bayes and Decision Trees algorithms. He proposed a fuzzy logic-based lane-changing model, which was developed for mandatory lane-changing behavior in merging areas of freeway on-ramps. After training the model, the prediction accuracy of non-merge events could be as high as 86.3%, while the accuracy of merge events was 87.5%. In fact, if a merge event was predicted as a non-merge event, the driver may lose an opportunity to merge, but it would not cause an accident. Whereas if a non-merge event was predicted as a merge event, which would cause a potential danger to the driver. For this reason, Y. Hou carried out further research in this field with the help of Bayes classifier and Decision Trees. As for the results, the accuracy of non-merge events could be enhanced to 95.4%, while the accuracy of merge events was 73.6%.

1.2.2 Experimental Analysis of Merging Behavior Based on Literature Review

This section examines the experimental analysis of merging behavior based on the literature review. Fig. 1.1 illustrates the various variables characterizing the merging process. The merging vehicle and the vehicles surrounding it are focused, and the mainline drivers' lane choice modification is left out.

The merging behavior cannot be correctly observed through point-located measurement devices such as electromagnetic loops. Therefore, some researchers used trajectory measurement devices to present phenomenological observations of a merging behavior [38]. Two trajectory measurement methods exist: either measuring the trajectory of the merging vehicle with an equipped vehicle (using GPS) or measuring the trajectory outside of the merging vehicle from video camera recordings.

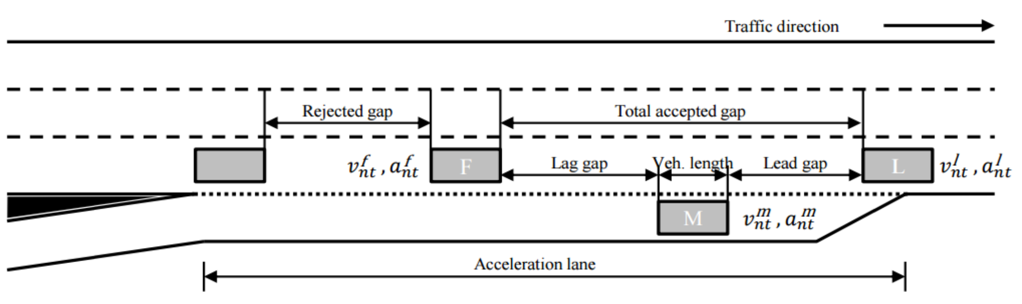

Figure. 1.1 Description of the variables of a merging process [38].

A few studies done by researchers focused on experimental observation of drivers' lane-changing trajectories at merging locations. Some of the studies used instrumented-vehicles (Kondyli [39]; Sarvi [40]), and the trajectory of the subject vehicle was estimated from GPS data. The GPS device was accompanied by a set of devices allowing to capture the positions of surrounding vehicles. This experimental device enabled the researchers to focus on the value of acceptable gaps rather than the rejected gaps which could be accessed with more difficulties.

Sarvi and Kuwahara [41] focused on the acceleration and deceleration phase of a merging vehicle. They combined two types of information (instrumented-vehicle data collection and video data collection) to calibrate their model of the acceleration and deceleration of the merging vehicle during the merging process. Wu et al. [42] studied the impact of ramp metering on drivers' behavior, both on the passing traffic and on the merging vehicle. They concluded that ramp metering had no effect on the passing traffic, but it had some effects on the merging characteristics: the presence of ramp metering increased the acceptable gap size and reduced the merging vehicle speed. Moreover, none of the studies analyzed the data in observing the rejected gaps [38]: the gaps a driver could have chosen (because the rejected gaps were present when the driver drove along the acceleration lane, but the driver preferred driving ahead and merged into an another gap, see Figure. 1.1).

1.2.3 Conclusions of Literature Review

Lane-changing models are one of the basic driving behavior interactions in the microscopic traffic simulations for traffic, safety, and transportation system analysis. Given the importance of lane-changing in traffic situations, there have been many research studies over the past 30 years. When predicting lane-changing behavior, several techniques can be distinguished. Most of the models were based on gap acceptance theory, but also models based on discrete choice modeling. Recently, many researchers have focused on using artificial intelligence methods to predict driver lane-changing behavior.

The literature review of the existing lane-changing models is summarized as follows. First, most of the lane-changing prediction models were only able to accurately predict a driver's lane-changing behavior (merge or non-merge) after it actually happens. Second, few models were intended to predict a driver's mandatory lane-changing behavior in the merging area of a freeway on-ramp. Third, most of the models considered drivers' visual search behavior and vehicles' motion features as the input vectors to predict drivers' lane-changing behavior. However, studies have demonstrated that a driver's lane-changing behavior is essentially triggered by his/her surrounding driving environment. Fourth, some of the models analyzed lane-changing behavior by offline processing of data rather than real-recording data. However, using the offline processing of data could make the prediction relatively unreliable and impractical. Fifth, few models focused on lane-changing prediction for instrumented-vehicles due to the limitations of poor sensor measurements and incomplete scene information. Even though an instrumented-vehicle has an extensive knowledge of its own dynamics and fairly reliable knowledge of its immediate surrounding information, the instrumented-vehicle cannot determine the information of vehicles beyond the immediate surrounding vehicles, which could affect the driving behavior of the instrumented-vehicle.

1.3 Research Objective

The objective of this thesis is to develop a model for predicting a driver's mandatory lane-changing decisions (merge or non-merge) in advance of the driver's actual driving behavior with a lead time of 1s, 1.5s, and 1.8s in the merging area of a freeway on-ramp. With the continuous improvements in the model accuracy, the model can be applied to lane-changing assistance systems.

The prediction model developed in this thesis was established based on BPNN (back-propagation neural network). BP [43] is a common method of training ANNs used in conjunction with an optimization method such as gradient descent, which calculates the gradient of a loss function on all the weights in a network. Typically, two advantages contribute to the popularity of BPNN. One is that BPNN can handle noisy data and approximate any degree of complexity in nonlinear systems. The other one is that BPNN does not require any simplifying assumptions or prior knowledge of problem-solving, compared with statistical models. Imprecision and uncertainty are often associated with a lane-changing behavior, and BPNN is capable of making accurate predictions under an environment of uncertainty. Moreover, BPNN offers a potential alternative to other methods, such as HMMs based techniques for prediction of lane-changing within road traffic scenarios [6].

In this thesis, two different methods were used to establish the lane-changing prediction model based on BPNN. One was that using one frame of variables as the input vectors for the BPNN model. The other was that using six frames of variables derived from a continuous time-series driving environment features as the input vectors for the BPNN model. The outputs of the model were the class labels of merge or non-merge. Detailed vehicle trajectory data provided by the Federal Highway Administration's (FHWA) Next Generation Simulation (NGSIM) project [44] was used for training and testing the model.

In this thesis, mandatory lane changes refer only to those executed by drivers entering from the on-ramp of a freeway. The lane changes made by drivers exiting the mainline of a freeway, although also mandatory, were outside the scope of this thesis. Discretionary lane changes, performed when drivers perceive driving conditions in the target lanes to be better, were also beyond the scope of this thesis.

1.4 Thesis Organization

This thesis consists of 5 chapters.

In the early part of Chapter 1, the motivation and objective of this thesis are presented. Then, an extensive literature review of existing lane-changing models is given in the remainder part of Chapter 1. In the literature review, both the contributions and disadvantages of the existing models are presented.

Chapter 2 describes the methodology used in this thesis. In Chapter 2, the BPNN method is demonstrated its reliability and practicality in predicting a driver's mandatory lane-changing decisions. Moreover, the additional momentum method which could improve the BP algorithm is introduced in Chapter 2.

Chapter 3 introduces the data used for training and testing the BPNN model. In Chapter 3, details of the datasets, data processing, and data filtering are described.

Chapter 4 presents the model design process and prediction results. In Chapter 4, identification of the input variables and required parameters for the BPNN model, model training and testing, and prediction results for both merge events and non-merge events are presented.

Chapter 5 summarizes this thesis and gives the contributions and future work of this thesis.

Chapter 2: Methodology

2.1 AI Methods

Recently, more and more researchers adopt AI (Artificial Intelligence) methods to solve traffic problems. AI attempts to understand and build intelligent entities that think and act rationally like humans for solving problems or making decisions [17]. AI methods include two major categories which are computational intelligence and symbolic AI. Computational intelligence consists of methods such as fuzzy system, neural network, and evolutionary computing. Symbolic AI concentrates on the development of knowledge-based systems that are capable of making decisions in a particular domain utilizing knowledge from a human expert [17]. The difference of computational intelligence from symbolic AI is that the output is generated without using knowledge bases such as rules, frames, or cases.

There are numerous advantages of AI methods. Among the most important of advantages are the following. First, AI is especially suitable for capturing the complex relationship among different variables in an environment of uncertainty. In many cases where uncertainty exists, direct mathematical relationships cannot be established. AI methods are able to deal with uncertainty by encapsulating the existing knowledge with uncertainty and probability inference theorems. Second, AI methods have the advantage of permanency. The knowledge incorporated in an AI framework is valid as long as the problems are relevant or decision circumstances are not changed. Third, AI methods are able to mimic human behavior and thinking, and they provide rational predictions or decisions with higher accuracy than traditional function fitting methods [17]. Fourth, AI methods provide fast solutions to complex problems by gathering and processing data.

The fact that the transportation systems are complex systems including interactions and conflicts among a lot of diverse individuals adds more difficulties for transportation professionals to solve the transportation problems. Instead of using traditional and classical methods, researchers have tried to seek the possibility of utilizing AI methods to address transportation problems [45]. Human behavior and interactions in transportation systems are hard to understand and model with traditional methods, but AI methods are suitable to simulate human behavior and interactions based on knowledge of a human or observed data. AI methods do not require assumptions about the mathematical forms of the relationships, and they can capture the non-linear nature of transportation problems. Traditional methods cannot effectively deal with uncertainties, but AI methods have been shown to make accurate predictions in an environment of uncertainty. Researchers have demonstrated that AI is a good choice when dealing with qualitative data like the knowledge provided by a human [46].

2.1.1 Various AI Classification Methods

There are several AI classification methods, such as Supporting Vector Machines (SVMs) [47], Decision Trees [48], Naive Bayes [49], Logistic Regression [49], Artificial Neural Networks (ANNs), and so on. AI methods are suitable for predicting driver lane-changing behavior because they can deal with the mixed data and capture non-linear nature of lane-changing behavior [50]. Several AI methods are introduced below.

The first one is the Decision Trees (DTs) method [48], it uses the knowledge extracted from data in a recursive hierarchical structure composed of nodes and branches. Each internal node represents an attribute and is associated with a test relevant for data classification. Leaf nodes of the tree correspond to classes. Moreover, branches represent each of the possible results of the applied tests. A new example can be classified following the nodes and branches accordingly until a leaf node is reached. The DTs induction process aims to maximize the correct classification of all training data. To avoid the problem of overfitting, a pruning phase is usually applied to the trained tree. It prunes ramifications with low expressive power according to some criteria, like the expected error rate [51]. One advantage of DTs is the comprehensibility of the classification structures generated.

The second one is the k-Nearest Neighbor (kNN) method [52], it is the simplest representative of instance-based ML techniques. It stores all training data and classifies a new data point according to the class of the majority of its k nearest neighbors in the given dataset [53]. To obtain the nearest neighbors for each data, kNN uses a measure to compute the distance between pairs of data items. In general, the measure employed is the Euclidean distance. kNN is able to build local approximations of the objective function. This characteristic may be advantageous when the objective function is complex, but may be described by several local approximations of low complexity.

Another advantage of kNN is its simplicity. Nevertheless, prediction time is usually costly, since all training data must be revisited.

The third one is the Naive Bayes (NB) method [49] and it is a probabilistic classifier based on the Bayes theorem for conditional probabilities. It builds a function to be optimized using a narrow assumption that all attributes in a dataset are independent. Therefore, it assumes that the presence/absence of a characteristic describing a certain class is unrelated to the presence/absence of any other characteristic, which is not true for the majority of classification tasks [54]. NB training is usually performed through the use of maximum likelihood algorithms. Despite its simplicity, NB has been successful in complex practical applications, especially in text mining. It also shows low training and prediction time.

The fourth one is the Support Vector Machines (SVMs) method [47], which is based on concepts from Statistical Learning Theory. SVMs seek for a hyperplane $w \cdot \Phi(x) + b = 0$ which is able to separate the data with minimum error. The hyperplane also maximizes the margin of separation between the classes. In the equation of the hyperplane, Φ represents a mapping function that maps the data to a space of higher dimension, such that the classes become linearly separable. In SVMs training and predictions, the mapping function appears as dot products in the form $\Phi\ (xi) \cdot \Phi\ (xj)$, which can be efficiently computed by Kernel functions. Some of the most used Kernel functions are the Gaussian or RBF (Radial-Basis Function) functions [55]. SVMs have good generalization ability and also stand out for their robustness to high dimensional data. Their main deficiency is related to the difficulty of interpreting the generated model and their sensibility to a proper parameter tuning.

The last one is the Logistic Regression (LR) [49] classifiers method. LR classifiers are statistical models in which a logistic curve is fitted to the dataset, and they can model the probability of the occurrence of a class. LR classifiers are also known as a logistic model, logit model, and maximum-entropy classifiers. The first step in LR consists of building a logit variable, containing the natural log of the odds of the class occurring or not [56]. A maximum likelihood estimation algorithm is then applied to estimate the probabilities. LR models are widely employed in statistics and have demonstrated success in solving several real-world problems.

2.2 BP Neural Network

2.2.1 Feasibility Analysis of BP Neural Network

This thesis adopted the Back-Propagation Neural Network (BPNN) method as one of the AI methods to predict driver mandatory lane-changing behavior. The feasibility and reliability of using the BPNN method in lane-changing prediction are discussed in this section of the thesis.

BPNN algorithm belongs to the ANNs, which is a computational system based on the structure, processing methods, and learning ability of the human brain [57]. This system is composed of simple processing units that simulate biological neurons. These artificial neurons, also called nodes, are disposed in one or more layers to model some functions of the human nervous system in order to take advantage of its computational strength [58]. Each node is connected to one or more nodes through weighted connections that simulate biological synapses. BPNN is a multilayered feed-forward network trained according to an error back-propagation algorithm, which is one of the most widely applied neural networks [59]. The schematic of a typical multilayer feed-forward network is shown in Figure. 2.1. A BPNN can be used to learn and store a great deal of the mapping relationships of an input-output model, but no need exists to disclose in advance the mathematical equation that describes these mapping relationships [60].

BPNN is a commonly used nonlinear function approximation algorithm, and studies have shown its great advantages in predicting, pattern identification, optimization, and data processing for its flexible, non-linear, and valid self-organization properties [60]. Several problem areas are modeled using BPNN, and in many cases, BPNN has shown superior results compared to other modeling methods. BPNN is always used in cases in which the rules and criteria for searching for an answer are not clear, which is why BPNN models often are called black box [61], since they can be used to solve a problem, but cannot always explain how the problem was solved.

Compared to other statistical models, there is a great advantage of BPNN is that it does not require simplifying assumptions or prior knowledge of problem-solving. For example, with regression models, it is necessary to specify the underlying relationship (linear, polynomial, exponential, rational, etc.) between the independent and dependent variables before the model estimation [62]. However, such specifications are not necessary for the inputs and outputs of a BPNN model. Among the advantages of BPNN is its robustness to noisy data and its ability to represent the linear and non-linear functions of various forms and complexities [60].

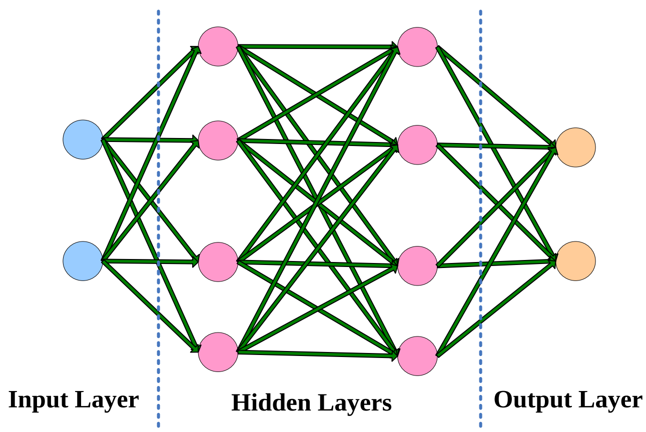

Figure. 2.1 Schematic of a multilayer feed-forward neural network.

The problem of predicting drivers' lane-changing decisions involves imprecision and uncertainty because a driver's mental and physical behavior are highly non-linear and difficult to capture. BPNN is suitable for predicting drivers' lane-changing decisions because it is capable of making accurate predictions under an environment of uncertainty.

Lane-changing behavior at on-ramps is a crucial component of microscopic traffic simulation that involves a high level of interaction between vehicles. This interaction involves a complex decision-making process, and a driver needs to decide whether to merge or not based on the relevant driving environment factors. So this is essentially a binary classification problem. The output of the problem is the driver's lane-changing decisions (merge or non-merge).

The biggest issue for solving the problem of lane-changing prediction is that the relevant factors do not linearly affect a driver's lane-changing decisions. To handle the uncertainty and nonlinearity of this problem, it is necessary to find the primary factors that directly lead to a driver's lane-changing decisions, then to approximate and merge the secondary factors. It is possible to generate a primary function to take the place of the non-linear function and create a universal approximation system to produce the output. Researchers have demonstrated that a BPNN model can be realized to be a universal approximation system to solve non-linear problems. Thus, it is feasible and reliable to predict a driver's lane-changing decisions using a BPNN model.

2.2.2 Back-Propagation Algorithm

BP algorithm is a method to monitor learning in a network [63]. It uses the approach of gradient descent to achieve the modification of connection weights in a network. The modification of the connection weights value aims to reveal the minimum error sum of squares in the network [63]. BP learning process can be described as the following two procedures:

(1) Forward-propagation of operating signal: the input signal is propagated from the input layer, via the hidden layer, to the output layer. During the forward-propagation of transferring signal, the weight value and threshold value of the network are maintained, and the status of each layer of nodes only

exerts an effect on the next layer of nodes. In case that if the expected output could not be achieved in the output layer, it would be switched into the back-propagation of error signal.

(2) Back-propagation of error signal: the difference between the computed output and expected output of a network is defined as the error signal, and a mean squared error is commonly used to evaluate the performance of the model. In the back-propagation of error signal, the error signal is propagated from the output end to the input layer in a layer-by-layer manner [64]. During the back-propagation of error signal, the weight value of the network is adjusted by the error feedback. The continuous modification of weight value and threshold (offset) value is applied to make the computed output of the network closer to the expected output.

The ideology guiding the learning rules of the BP algorithm is: the modification of the weight values and threshold values should be done along the negative gradient direction reflecting the fastest declining of function [64]. The learning process of the BP algorithm is described below in mathematics, and a BPNN is typically represented by the diagram in Figure. 2.2.

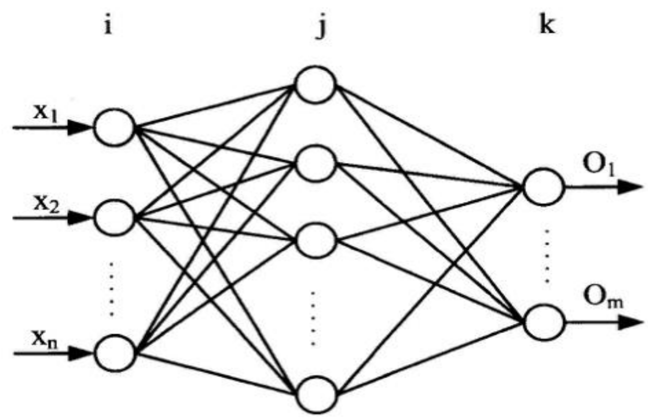

Figure. 2.2 Schematic of a BPNN with one hidden layer.

The input vectors of the network are:

$$X = (x_1, x_2, ..x_n)^T \tag{2.1}$$

The output values in the hidden layer of the network are:

$$O_k = (o_{k1,} o_{k2}, ...o_{km})^T \tag{2.2}$$

The output values in the output layer of the network are:

$$O_j = (o_{j1,} o_{j2}, ...o_{jl})^T \tag{2.3}$$

The connection weights connecting the input nodes and the hidden nodes are:

$$w_{ij} (i = 1,2,...,n;\ j = 1,2,...,l) \tag{2.4}$$

The connection weights connecting the hidden nodes and output nodes are:

$$w_{jk} (j = 1,2,...,l; k = 1,2,...,m) \tag{2.5}$$

During the process of signal forward-propagation, the output value of each node in the hidden layer is formulated as:

$$o_j = f(\sum_{i=1}^{n} w_{ij} x_i + \theta_j), (j = 1,2,...,l) \quad (2.6)$$

The output value of each node in the output layer is formulated as:

$$o_k = f(\sum_{j=1}^{l} w_{jk} o_j - \theta_k), (k = 1,2,...,m) \quad (2.7)$$

The tan-sigmoid activation function of the network is given by:

$$f(x) = 2/(1+e^{-2x}) - 1 \quad (2.8)$$

The two general approaches for neural network training are usually called batch training and online training [65]. The two approaches are similar, but they can produce very different results. The general consensus among neural network researchers is that when using the back-propagation training algorithm, the batch approach is better than the online approach. With online training, weight and threshold values are adjusted for every training item based on the difference between the computed outputs and the expected outputs [65]. However, when using this approach, the convergence rate of the network is usually slow during the training process. With batch training, the adjustment delta values are accumulated over all training items to give an aggregate set of deltas, and then the aggregated deltas are applied to each weight and threshold [65]. The convergence rate of the network is relatively high when using the batch training approach. This thesis used the batch training approach to train the BPNN. The global mean squared error taken as an evaluation of the performance of the BPNN during the training process is given by:

$$E(X)=\sum_{x=1}^{n}E_x=\frac{1}{2}\sum_{x=1}^{n}\sum_{k=1}^{m}(t_x(X)-o_k(X))^2 \tag{2.9}$$

where $t_k(X)$ is the expected output value for the sample X.

Based on the gradient descent method, the modification of each weight is formulated as:

$$\Delta w_{jk}=-\eta\frac{\partial E}{\partial w_{jk}} \tag{2.10}$$

After integrating the equation (2.7) and (2.9), the modification of each weight can be expressed as:

$$\Delta w_{jk}=-\eta\frac{\partial E}{\partial w_{jk}}=\eta\left(-\frac{\partial E}{\partial net_k}\right)\frac{\partial net_k}{\partial w_{jk}}=\eta(t_k-o_k)o_k(1-o_k)o_j \tag{2.11}$$

δ_j and δ_k are defined as:

$$\delta_j=-\frac{\partial E}{\partial o_j}o_j(1-o_j) \tag{2.12}$$

$$\delta_k=\left(-\frac{\partial E}{\partial net_k}\right)=(t_k-o_k)o_k(1-o_k) \tag{2.13}$$

Since $\frac{\partial E}{\partial o_j}$ cannot be calculated as a result, it is expressed as:

$$-\frac{\partial E}{\partial o_j} = -\sum_{k=1}^{m}\frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial o_j} = \sum_{k=1}^{m}(-\frac{\partial E}{\partial net_k})(\frac{\partial(\sum w_{jk}o_j)}{\partial o_j}) \tag{2.14}$$

$$= \sum_{k=1}^{m}(-\frac{\partial E}{\partial net_k})w_{jk} = \sum_{k=1}^{m}\delta_k w_{jk}$$

After integrating the equations above, the modification of each weight can be expressed as:

$$\Delta w_{jk} = \eta\ (t_k - o_k)o_k(1-o_k)o_j \tag{2.15}$$

$$\Delta w_{ij} = \eta\delta_j o_i \tag{2.16}$$

where η is the learning rate of network. δ_j can also be expressed as:

$$\delta_j = o_j(1-o_j)\sum_{k=1}^{m}\delta_k w_{jk} \tag{2.17}$$

In general, training is the process of modifying the weights and thresholds using a suitable method. BPNN uses a learning mode, in which the inputs and expected outputs are presented to BPNN and the weights and thresholds are adjusted so that BPNN attempts to produce the expected outputs. After training, weights and thresholds contain meaningful information, however, before training, they are random and have no meaning.

In biologically inspired neural networks, the activation function is usually an abstraction representing the rate of the action potential firing in the cell. In a BPNN, the activation function of a node defines the output of that node given an input or set of inputs. Three kinds of activation functions are commonly used in BPNN.

The first one is the log-sigmoid [66] activation function illustrated in Figure. 2.3. The function log-sigmoid generates outputs between values of 0 and 1 as the network input of nodes goes from negative to positive infinity [66].

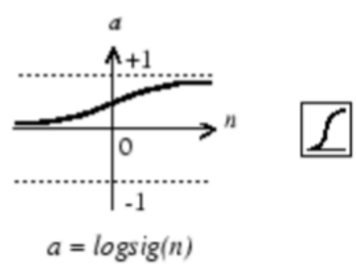

Figure. 2.3 Log-sigmoid activation function [66].

The second one is the tan-sigmoid [66] activation function illustrated in Figure. 2.4. The function tan-sigmoid generates outputs between values of -1 and 1.

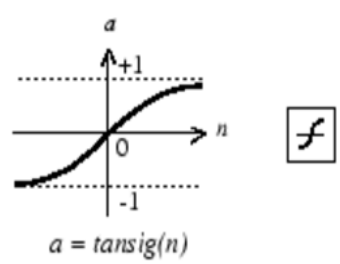

Figure. 2.4 Tan-sigmoid activation function [66].

The third one is the linear activation function [66] illustrated in Figure. 2.5. The linear activation function is commonly used for solving function fitting problems.

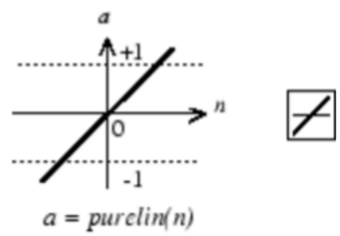

Figure. 2.5 Linear activation function [66].

Sigmoid activation functions are often used for solving pattern recognition problems. This thesis adopted the tan-sigmoid activation function to produce outputs in the BPNN model. For the problem of mandatory lane-changing prediction, the inputs of the BPNN model are the factors that could affect a driver's lane-changing decisions, and the outputs of the BPNN model are the driver's lane-changing decisions (merge or non-merge). In this thesis, the input factors were derived from a driver's surrounding driving environment, and the outputs were the class labels of merge or non-merge.

2.2.3 Improvement of BP Algorithm

In the applications of artificial neural networks, the BPNN and its varied patterns are adopted in most of the neural network models. However, this does not mean that BPNN is perfect and there are still inevitable deficiencies in its algorithm, for example, falling into the minimum point of a local part, the convergence rate being rather slow, the network tending to have more redundancy and the samples newly added may affect the samples learned [67]. Researchers have put forward many improved methods to solve these deficiencies. The improved methods can be classified into three categories: the first one is to improve the accuracy of BPNN training; the second one is to improve the speed of BPNN training; the third one is to avoid dropping into the minimum point of a local part. Figure. 2.6 shows a part of the error surface of BPNN.

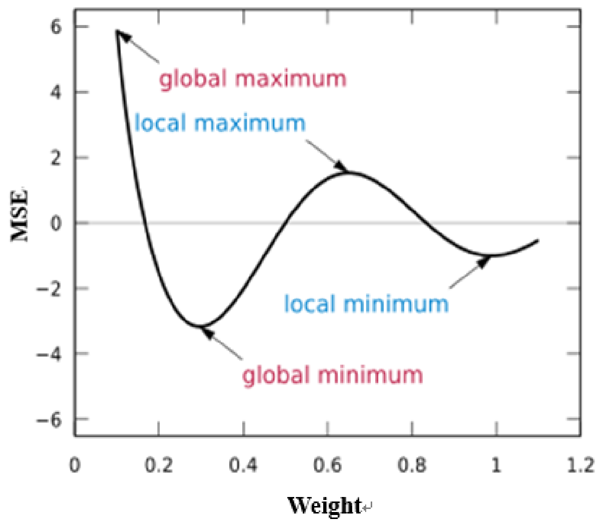

Figure. 2.6 Error surface of BPNN [68].

$E(w)$ is assumed as the error function of BPNN. Usually, a local minimum point could appear when the partial derivative $\partial E / \partial w$ becomes zero or very small. However, the error function $E(w)$ can still remain at a very high error level. Based on the gradient descent method, the BP searching algorithm with the formulation shown as:

$$\Delta w = -\eta \frac{\partial E}{\partial w} \approx 0 \tag{2.18}$$

would be stopped, even though $E(w)$ still remains at a high error level in a certain weight space.

A method that avoids dropping into the minimum point of a local part is called additional momentum method [67]. In this method, a momentum coefficient α is introduced into the gradient descent algorithm. The formulation that illustrates the modification of the weight values and includes the additional momentum coefficient is expressed as:

$$\Delta w\,(t+1) = \alpha \Delta w(t) + \eta(1-\alpha)\frac{\partial E}{\partial w} \tag{2.19}$$

where $\Delta w\,(t+1)$ and $\Delta w\,(t)$ represent the weight modifications after the iterations of $(t+1)^{th}$ and t^{th}. The value of the momentum coefficient α must be determined between values of 0 and 1 [67]. $\frac{\partial E}{\partial w}$ represents the negative gradient of error sum of squares to weight in the BP algorithm.

After introducing the momentum coefficient into the BP algorithm, the correction result of last time is used to affect the correction of this time. When the correction of last time is oversized, the symbol of the second item in the formulation will be contrary to that of correction of last time, in order to reduce the correction of this time and lower the oscillation. When the correction of last time is undersized, the symbol of the second item in the formulation will be the same with that of correction of last time, in order to amplify the correction of this time and speed up the correction.

It is clear that the application of additional momentum method always tries to increase the corrections which are in the same direction of gradient. This method is able to accelerate the convergence rate and reduce the probability of falling into the minimum point of a local part. This method was adopted in this thesis to improve the performance of BPNN training, and 0.9 was determined as the value of the momentum coefficient.

Researchers have also demonstrated that the starting weights of BPNN can exert an enormous effect on the final result of BPNN training. Starting weights are the important factors that affect the possibility of BPNN to achieve certainly acceptable accuracy [67]. Generally, the starting weights of a BPNN are generated at random in a certain interval. The BPNN training starts with a starting point and reaches gradually to a minimum of error along the slope of error function. Hence, once the starting weight values are defined, the convergence direction of BPNN is determined. In case that the starting weights are not determined well, the convergence direction of BPNN can be towards the direction of divergence and this causes the non-convergence of concussion in BPNN. In addition, the convergence rate of BPNN training is also related to the determination of starting weights. Thus, it is significant to determine appropriate starting weights, so as to accelerate the convergence rate of BPNN training and avoid the concussion during the training process.

In this thesis, the approach used to determine the starting weights followed a rule: at the first time of the weighted combination of inputs in each node during the training process, the output value of each node should be close to zero [69]. To satisfy this requirement, the starting weight values of the BPNN were randomly determined between values of -1 and +1.

Chapter 3: Data

3.1 Data Source

To develop the BPNN model for driver lane-changing prediction in merging areas of freeway on-ramps, the data from the Federal Highway Administration's Next Generation Simulation (NGSIM) program [44] was used in this thesis. NGSIM is an openly available data source that has been used in many previous research studies for traffic simulation models. Two datasets used in the this thesis were the Interstate 80 Freeway Dataset and the U.S. Highway 101 Dataset, which were collected on the eastbound I-80 in the San Francisco Bay area in Emeryville, CA on April 13, 2005; and on the southbound US 101, also known as the Hollywood Freeway, in Los Angeles, CA on June 15th, 2005; respectively.

For the Interstate 80 dataset, the study area (see Figure. 3.1) was approximately 500 meters (1,640 feet) in length and consisted of six freeway lanes, which included a high-occupancy vehicle (HOV) lane. This dataset was collected using video cameras mounted on a 30-story building called Pacific Park Plaza. Video data was collected using seven video cameras—cameras 1 through 7—with camera 1 recording the southernmost and camera 7 recording the northernmost sections of the study area. This vehicle trajectory data provided detailed lane positions and the precise location of each vehicle within the study area every one-tenth of a second. A total of 45 minutes of data was available in the full dataset, which was 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.

For the U.S. Highway 101 dataset, the study area (see Figure. 3.2) was approximately 640 meters (2,100 feet) in length and consisted of five freeway lanes. An auxiliary lane was present through a portion of the corridor between the on-ramp at Ventura Boulevard and the off-ramp at Cahuenga Boulevard. This dataset was collected using video cameras mounted on a 36-story building called 10 Universal City Plaza. Video data was collected using eight video cameras—cameras 1 through 8—with camera 1 recording the southernmost and camera 8 recording the northernmost sections of the study area. A total of 45 minutes of data was available in the full dataset, which was 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.

Both of the two datasets represented two traffic states: conditions when congestion was building up (period of the first 15 minutes) denoted as the transition period; and congested conditions (period of the remaining 30 minutes). According to the aggregate speed and volume statistics of NGSIM dataset, both the flows of traffic and speeds of vehicles decreased during the congested period [70].

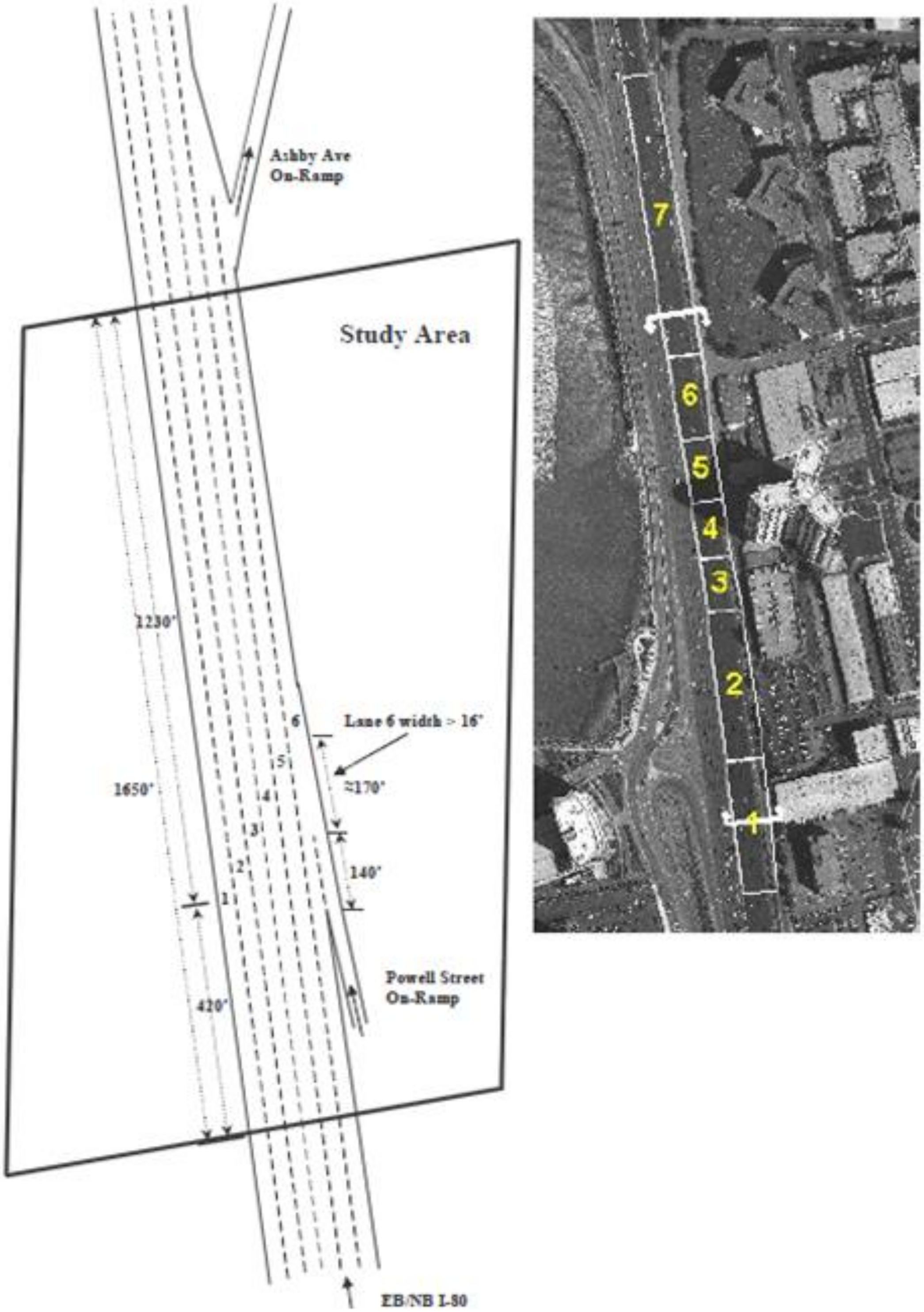

Figure. 3.1 Study area schematic and camera coverage of NGSIM I-80 dataset [44].

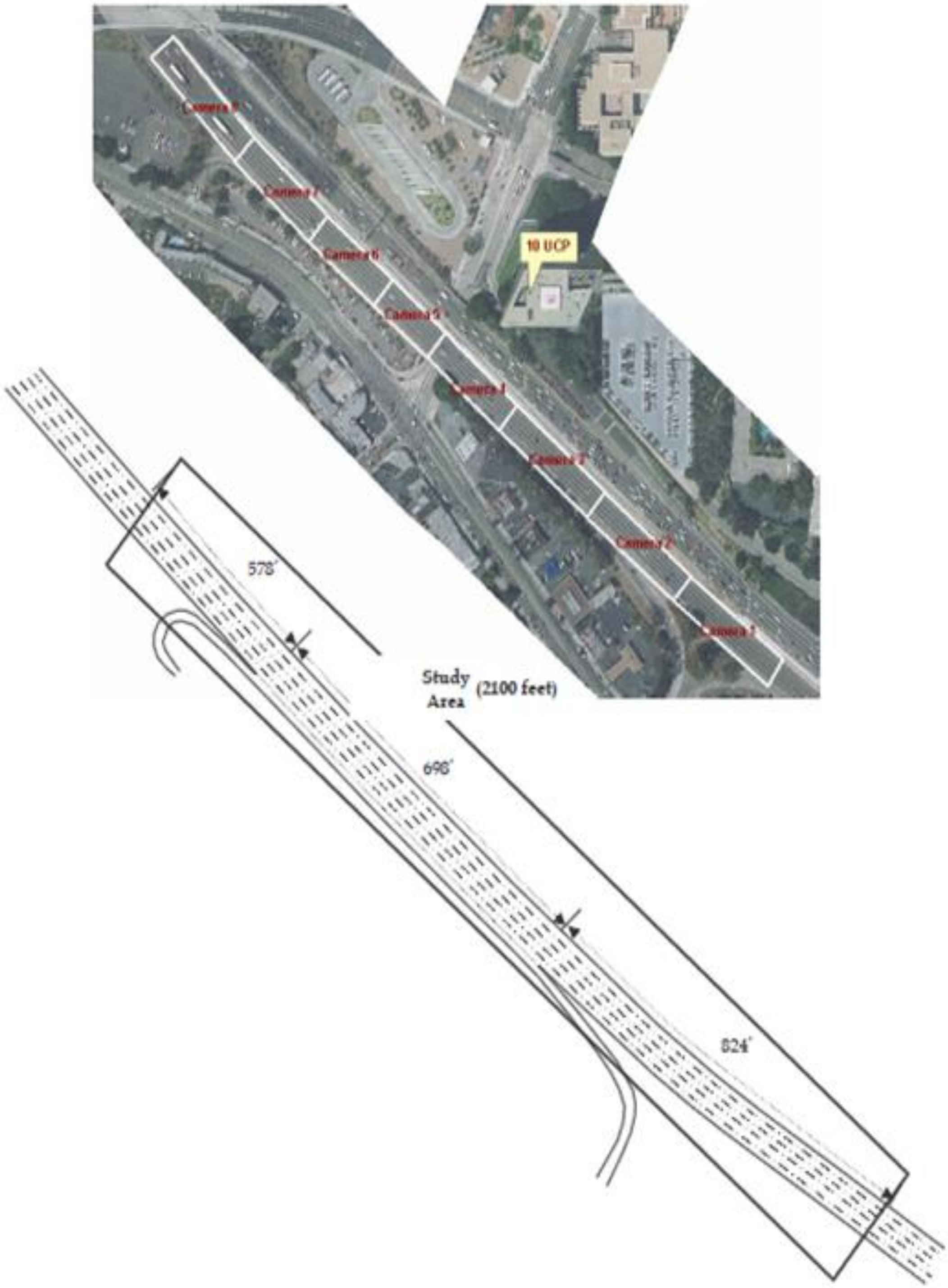

Figure. 3.2 Study area schematic and camera coverage of NGSIM U.S.101 dataset [44].

The vehicle trajectory data was obtained at a resolution of 10 frames per second, which included vehicle ID, lateral coordinate, longitudinal coordinate, vehicle length, vehicle width, vehicle class, vehicle velocity, vehicle acceleration, land identification, and so on. Further information about the trajectory data structure can be found from the NGSIM [44].

With the help of the trajectory data, the behavior of vehicles could be identified, for example, lane-changing, lane-keeping, vehicle following, and even the natural lane-changing path. Given the focus of this thesis on mandatory lane-changing behavior, only the trajectory data of vehicles in the auxiliary lane (on-ramp) and the adjacent lane was used for model development. The auxiliary lane was referred to as the merging lane, and the adjacent lane was referred to as the target lane. The velocity and position of each vehicle were identified in 0.1-second intervals. The 0.1-second interval produced data with comparable sample sizes for both merge events and non-merge events.

To establish the model, a driver's behavior must be identified by the vehicle trajectory data. In fact, the NGSIM dataset did not present this information directly, and the approach adopted in this thesis was to try to understand a driver's driving behavior. The standard used to identify a driver's behavior was as follows: when the lateral coordinate of a vehicle began to shift toward the adjacent target lane direction without oscillations, the behavior of the driver was recognized as a merge event. Otherwise, the behavior of the driver was deemed as a non-merge event. A single driver could participate in several non-merge events but only one merge event.

After identifying the non-merge events and merge events based on the trajectory data, a total of 1,000 observations was obtained from the U.S. Highway 101 dataset and Interstate 80 dataset, 500 of them being non-merge events and 500 of them being merge events. There is no general rule on how many observations should be assigned to training and testing. In order to obtain high prediction accuracy, a large training data size was required. Previous studies have used 80% of the dataset for training and 20% for testing the model. Based on these studies, the dataset was divided into two groups: 80% of observations were used for training, and 20% of observations were used for testing.

3.2 Data Filtering

Previous studies have shown that the vehicle acceleration data from NGSIM exhibits noises (random errors) [71]. Data smoothing techniques such as moving average filter [72], Kalman filtering [73], and Kalman smoothing [73] have been used to improve the acceleration data quality.

In this thesis, the moving average filter was adopted to smooth the acceleration data. A moving average filter was designed and applied to all trajectories before any further data analysis. The moving average filter is the most common filter because it is easy to understand. In spite of its simplicity, this kind of filter is optimal for dealing with common tasks: it is good at reducing random noise while still retaining a sharp step response, which makes it the premier filter for time domain encoded signals [72]. The formulation that shows the moving average filter is expressed as:

$$y(i) = \frac{1}{M}\sum_{j=0}^{M-1} x(i+j) \tag{3.1}$$

where *x* is the input signal, *y* is the output signal. Figure. 3.3 shows the unfiltered and filtered acceleration data from NGSIM dataset. It can be found that the acceleration data needed to be filtered before using for model development.

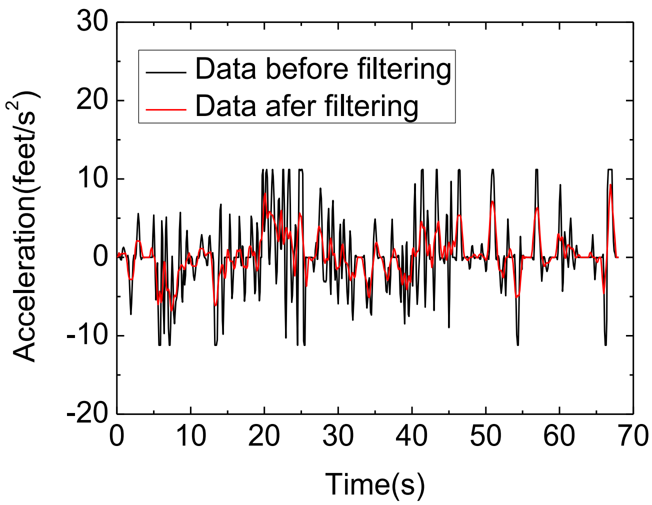

Figure. 3.3 Comparison of unfiltered and filtered acceleration data.

Chapter 4: Model Design and Results

4.1 Input Variables

To establish the BPNN model for predicting a driver's mandatory lane-changing decisions, the input variables for the model must be defined.

Several features can be considered as the input variables to predict a driver's lane-changing decisions. Generally, the features can be divided into two categories. The first category includes the drivers' visual search behavior and vehicles' motion features [8]. The second category includes the surrounding driving environment features which could affect a driver's lane-changing decisions [14]. Researchers have demonstrated that a driver's lane-changing decisions are essentially triggered by his/her surrounding driving environment features [13].

In any given instance, a driver traveling in the merging lane assess driving environment in both the target lane and merging lane in order to decide whether to merge or not. Whether the driving environment meets the requirements for executing a lane change depends mainly on two conditions, namely, the merging gap acceptance and the time to collision (TTC) [45], which is obtained by dividing the distance by the relative speed between the conflict vehicles. Both the merging gap acceptance and the TTC must be sufficiently large to ensure the safety of the lane-changing behavior [45].

Several factors may affect a driver's lane-changing decisions. In this thesis, five factors or dimensions that were found to affect a driver's lane-changing decisions in previous studies [13] were considered as the input variables for the BPNN model. These factors are shown in Figure. 4.1, and they are defined below.

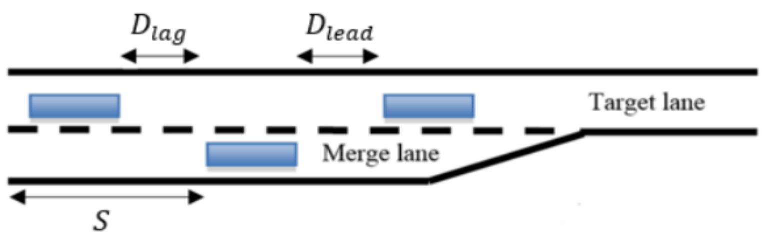

Figure. 4.1 Schematic illustrating input variables [13].

- $\Delta V_{lead}(feet/s)$: The speed difference between the lead vehicle in the target lane and the merging vehicle. It is defined as:

$$\Delta V_{lead} = V_{lead} - V_{merge} \tag{4.1}$$

where, V_{lead} is the speed of the lead vehicle, and V_{merge} is the speed of the merging vehicle.

- $\Delta V_{lag}(feet/s)$: The speed difference between the lag vehicle in the target lane and the merging vehicle. It is defined as:

$$\Delta V_{lag} = V_{lag} - V_{merge} \tag{4.2}$$

where, V_{lag} is the speed of the lag vehicle.

- $D_{lead}(feet)$: The gap distance between the lead vehicle in the target lane and the merging vehicle.

- $D_{lag}(feet)$: The gap distance between the lag vehicle in the target lane and the merging vehicle.

- $S(feet)$: The distance from the beginning of the merging lane to the merging vehicle.

The five input variables used for the BPNN model development can be explained as follows: the gap variables have an effect on the merging gap acceptance. A large merging gap makes merging relatively easier and hence increases the probability of merging; the speed variables have an effect on the potential risk of lane-changing behavior. When the lead vehicle is slower than the merging vehicle, the merging vehicle is more likely to slow down to match its speed with the speed of the lead vehicle first so as to focus exclusively on the interaction with the lag vehicle. When the lag vehicle is faster than the merging vehicle, the merging vehicle is more likely to speed up before attempting to establish the right of way; the remaining distance from the beginning of the merging lane to the merging vehicle has an effect on a driver's thinking. As the remaining distance increases, a driver becomes more concerned about merging and hence more aggressive.

4.2 Driver Mandatory Lane-Changing Prediction

4.2.1 Model Establishment and Training

Some of the existing models were only able to predict a driver's lane-changing behavior after it actually happens [71]. The BPNN model developed in this thesis was aimed at predicting a driver's lane-changing decisions in advance of the driver's actual driving behavior with a lead time of 1s, 1.5s, and 1.8s.

For the dataset used for the model development, a total of 1,000 observations (500 merge events and 500 non-merge events) was used for training and testing the model. As explained previously in this thesis, 80% of observations were used for training, and 20% of observations were used for testing the performance of the model. Figure. 4.2 shows a part of the processed data for training the model. As shown in Figure. 4.2, values from the first five columns represented the input variables for samples in 0.1-second intervals, and values from the last column represented the corresponding outputs which were the class labels of merge events or non-merge events. Value 1 represented the class labels of merge events, and value -1 represented the class labels of non-merge events.

2.5093000e+001	4.3880000e+000	3.8577000e+002	-1.9500000e+000	-4.4533334e+000	-1.0000000e+000
2.6084000e+001	8.8000000e-001	3.9425700e+002	-6.9444440e-001	-2.6377778e+000	-1.0000000e+000
1.6800000e+000	1.7380000e+001	4.0319900e+002	-3.0488889e+000	-4.6000000e+000	-1.0000000e+000
4.5770000e+000	1.4727000e+001	4.1176300e+002	-2.2644445e+000	-1.8766667e+000	-1.0000000e+000
6.5950000e+000	1.3633000e+001	4.1875600e+002	6.0000000e-002	2.0855556e+000	-1.0000000e+000
5.7520000e+000	1.6697000e+001	4.2291300e+002	-1.1400000e+000	-6.7777800e-002	-1.0000000e+000
9.3510000e+000	1.5524000e+001	4.2998500e+002	-4.6788889e+000	-2.4488889e+000	-1.0000000e+000
1.3913000e+001	1.1283000e+001	4.3957600e+002	-4.7466667e+000	-4.9088889e+000	-1.0000000e+000

Figure. 4.2 A part of processed data.

To improve the BPNN training efficiency, data normalization must be carried out. If the original data of the input vectors possessed a high degree of dispersion, larger parameter values would occupy the learning process of the BPNN. This was likely to affect the prediction of the BPNN model [36]. In this thesis, BPNN training was conducted using the MATLAB software [74].

The hidden layers are the most important parts in a BPNN. It is clear that without hidden layers, simple perceptron with linear output nodes is equivalent to linear statistical prediction models. At the very beginning, researchers hypothesized that a multiple-layers BPNN could perform well when solving non-linear problems [75]. Recently, researchers have also demonstrated that a multiple-layers BPNN could make the network very sophisticated and increase the possibility of non-convergence [76]. According to the Kolmogorov theory [77], a single hidden layer is sufficient for BPNN to approximate any complex non-linear function with any desired accuracy. Thus, this thesis used a single hidden layer BPNN to develop the model.

For the gradient descent algorithm, a step size, which is called the learning rate in BPNN, must be specified. The learning rate is crucial for back-propagation learning algorithm since it determines the magnitude of the weight modifications. It is well known that the steepest descent suffers the problem of slow convergence rate. Furthermore, it can be very sensitive to the determination of the learning rate. Smaller learning rates tend to slow the learning process while larger learning rates may cause the problem of network oscillation in the weight space. To overcome this problem, a small learning rate is commonly recommended [69]. In this thesis, 0.05 was determined as the learning rate.

An expected global error must be determined in BPNN training process. There is no criterion that determines when to stop training a BPNN. Generally, a BPNN would stop training when the expected error is achieved. However, if the expected error cannot be achieved, the training of a BPNN would be stopped once the training iterations (epoch) reach the maximum number. In this thesis, 0.01 was determined as the expected global error, and 10000 was determined as the maximum number of the training iterations.

The hidden nodes play very important roles for many successful applications of BPNN. It is the hidden nodes in the hidden layer that allow BPNN to detect the feature, to capture the pattern in the data, and to perform the complicated non-linear mapping between input and output variables [78]. The determination of optimal number of hidden nodes is crucial in BPNN. In general, a BPNN with fewer hidden nodes is preferable as it usually has better generalization ability and less overfitting problem. However, a BPNN with too few hidden nodes may not have enough power to model and learn the data [78].

There is no theoretical basis for determining this parameter although a few systematic approaches were proposed. The most common way in determining the number of hidden nodes is via experiments or by the trial-and-error approach. Several rules have also been proposed, for example, the number of hidden nodes depends on the number of input patterns. To avoid the overfitting problem, some researchers have provided empirical rules to restrict the number of hidden nodes [79].

This thesis used the trial-and-error approach to determine the optimal number of hidden nodes. In this approach, a range of the optional numbers was determined based on the empirical formulas, then different number of hidden nodes chosen from the range was used in the BPNN to observe the training performance. The empirical formulas [69] are given by:

$$l = \sqrt{mn} \tag{4.3}$$

$$l = \sqrt{m+n} \tag{4.4}$$

$$l = 2n+1 \tag{4.5}$$

where l is the number of hidden nodes; n is the number of input nodes; and m is the number of output nodes. For the model developed in this thesis, n was 5 and m was 1, so l was supposed to be 2.3, 2.4, or 11 based on the empirical formulas.

Figure. 4.4 shows the MMSE (minimized mean squared error) value of the BPNN when using different number of hidden nodes from the range of [2, 16]. This was the case of BPNN training for predicting a driver's lane-changing decisions in advance with a lead time of 1s.

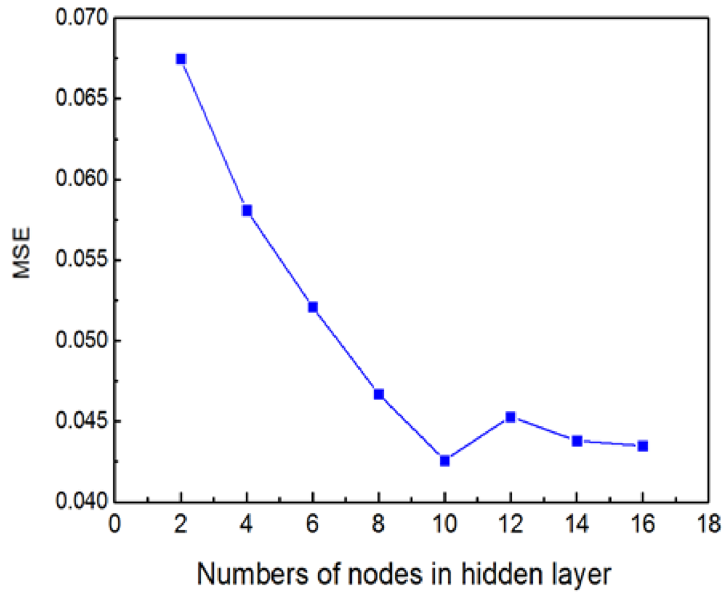

Figure. 4.3 MMSE of the BPNN with different number of hidden nodes.

As Figure. 4.3 shows, with the increase in the number of hidden nodes used in the BPNN, the MMSE value first decreased to a certain point, increased slightly, and finally became relatively constant. It was clear that the best performance of the BPNN appeared when 10 hidden nodes were used in the BPNN. Therefore, the optimal number of hidden nodes was 10. Although using more than 10 hidden nodes may decrease the MMSE value, using too many hidden nodes could weaken the generalization ability of the BPNN. Thus, in this case, a total of 10 hidden nodes was used in the BPNN. Similarly, for predicting a driver's lane-changing decisions in advance with a lead time of 1.5s and 1.8s, a total of 10 hidden nodes was also used in the BPNN.

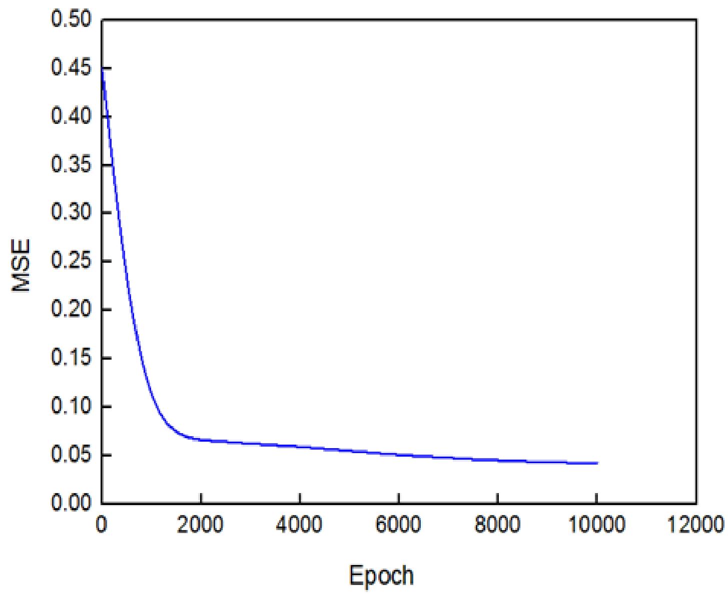

Figure. 4.4 Training error curve of the BPNN.

Training error curve of the BPNN is shown in Figure. 4.4. As indicated in Figure. 4.4, by increasing the number of iterations, the training performance of the BPNN improved. The MSE (mean squared error) value converged to 0.0426 after training the BPNN for 1,0000 iterations. It was also worth noting that, at the early stage of the training process, the MSE value decreased suddenly. This indicated that the convergence rate of the BPNN was high at the early stage of the training process. However, when the number of iterations was large enough, an increase in the number of iterations would no longer reduce the MSE value. Table. 4.1 shows the required parameters for the BPNN training.

Parameter	Value
Number of nodes in input/hidden/output layer	5/10/1
Number of total layers	3
Learning rate	0.05
Iterations	1,0000
Expected error	0.01
Momentum coefficient	0.9

Table 4.1 Parameters of BPNN training.

Figure. 4.5 plots the comparison of the MMSE value of the BPNN with different prediction lead time (1s, 1.5s, and 1.8s). As shown in Figure. 4.5, by increasing the prediction lead time, the MMSE value of the BPNN also increased. This indicated that by increasing the prediction lead time, the BPNN became more uncertain and imprecise, and the BPNN was less likely to be well-convergent.

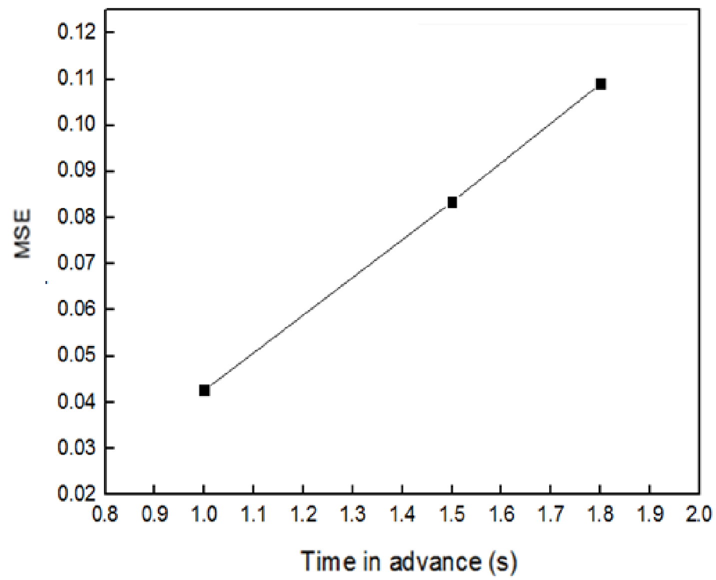

Figure. 4.5 MMSE of the BPNN with different prediction lead time.

4.2.2 Model Testing and Results

After training the model and assessing the training performance of the BPNN with respect to the training observations, the remaining observations (20% of all observations) could be used in the BPNN model to predict a driver's lane-changing decisions. Given the BPNN produced an output value between -1 and +1, a threshold value must be determined to identify the result properties. In this thesis, according to the input and output characteristics of all the observations, 0 was determined as the threshold value. If the BPNN output was larger than 0, a driver was predicted to execute a merging behavior, whereas if the BPNN output was less than 0, a driver was predicted to execute a lane-keeping behavior. By comparing the prediction results for each observation with its actual attribute, the prediction accuracy of the BPNN model could be assessed. The prediction results for non-merge events and merge events are presented in Figure. 4.6, Figure. 4.7.

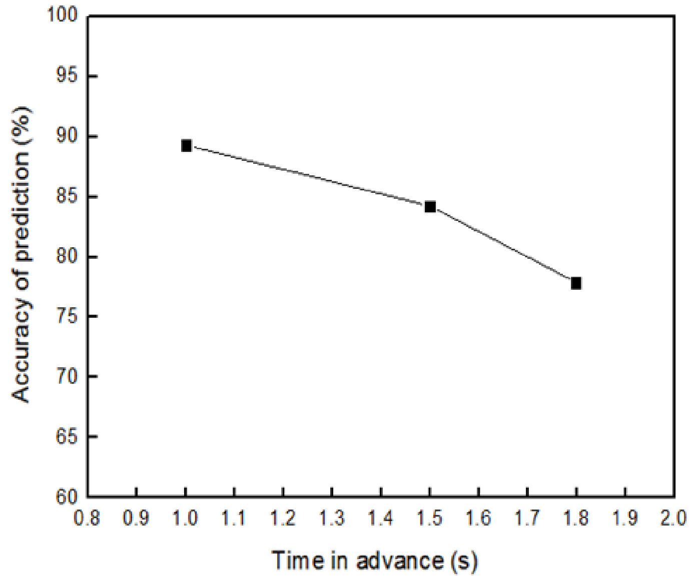

Figure. 4.6 Prediction results for non-merge events.

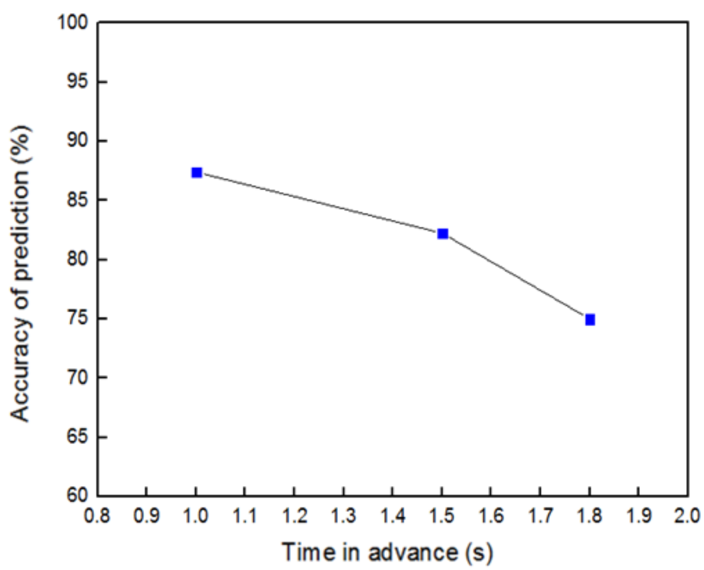

Figure. 4.7 Prediction results for merge events.

The prediction accuracy with respect to time was considered in the evaluation of the performance of the BPNN model. Figure. 4.6 plots the comparison of prediction accuracy for non-merge events with different prediction lead time, and Figure. 4.7 plots the comparison of prediction accuracy for merge events with different prediction lead time. As shown above, both the non-merge events and merge events' prediction accuracies decreased by increasing the prediction lead time. For merge events predictions, the prediction accuracy was approximately 75.6% at 1.8s in advance of a merging behavior. The accuracy increased to 82.2% at 1.5s in advance of a merging behavior and then reached 87.3% at 1s in advance of a merging behavior. As a result, the prediction performance of the BPNN model was better when predictions were made closer to a merging behavior. A summary of the prediction results for both non-merge events and merge events is shown in Table. 4.2.

Time in advance	Accuracy of non-merge events	Accuracy of merge events
1	89.1%	87.3%
1.5	84.4%	82.2%
1.8s	77.9%	75.6%

Table 4.2 Summary of prediction results.

4.2.3 New Method of Model Establishment and Training

To improve the prediction accuracy of the BPNN model, a new method is described in this section of the thesis. Figure. 4.9 shows a scenario of typical lane-changing in the merging area of a freeway on-ramp.

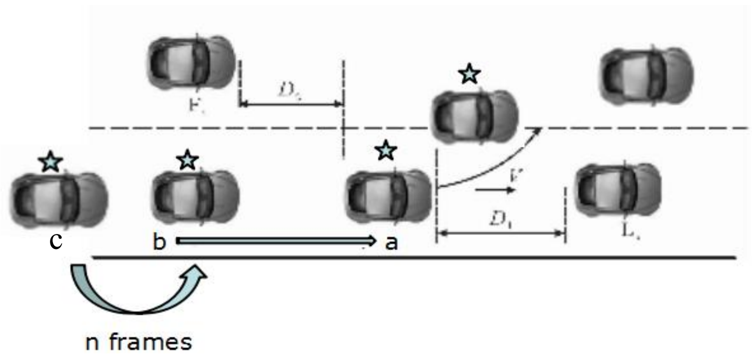

Figure. 4.8 Typical lane-changing scenario.

In Figure. 4.8, the merging vehicle (marked with an asterisk) maintains a lane-keeping behavior until it reaches at the merging location “a.” The merging vehicle starts changing lanes from location “a.” In this thesis, the BPNN model was aimed at predicting the oncoming lane-changing behavior at an early stage (at location “b”). The duration time from location “b” to location “a” is specified here as 1s, 1.5s, and1.8s.

For the BPNN model developed previously in this thesis, only one frame of variables derived from the surrounding driving environment features of location “b” was used as the input vectors to predict a driver’s lane-changing decisions. In fact, a driver’s lane-changing decisions are usually affected by a continuous time-series driving environment features. In another word, a driver usually assesses surrounding driving environment for a period of time in order to decide whether to merge or not. To increase the practicality and reliability of the BPNN prediction model in this thesis, six frames of variables derived from a continuous time-series driving environment features (from location “c” to location “b”) were used as the input vectors for the BPNN model. As a result, a total of thirty input variables in each sample was used for the BPNN model. This was a new method to establish the BPNN model.

For the dataset used for the model development in the new method, 80% of observations were used for training, and 20% of observations were used for testing the performance of the model. The maximum number of training iterations and the expected global error were determined to 12000 and 0.01, respectively.

In the new method of BPNN model establishment, the trial-and-error approach was also adopted to determine the number of hidden nodes. A range of the optional numbers was determined based on the empirical formulas, then different number of hidden nodes chosen from the range was used in the BPNN to observe the training performance.

After determining the number of hidden nodes based on the empirical formulas, the optional numbers were 5.5, 5.6, or 61. Figure. 4.9 shows the MMSE value of the BPNN when using different number of hidden nodes from the range of [5, 65]. This was also the case of BPNN training for predicting a driver's lane-changing decisions in advance with a lead time of 1s.

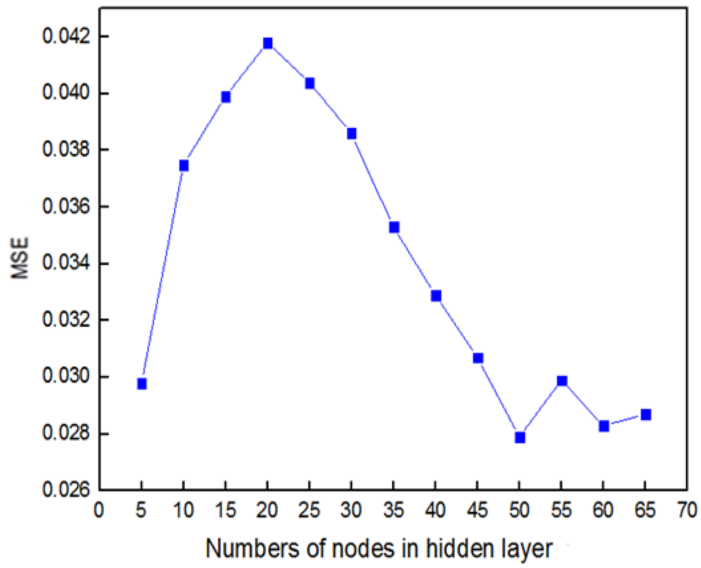

Figure. 4.9 MMSE of the BPNN with different number of hidden nodes.

As shown in Figure. 4.9, with the increasing number of hidden nodes, the MMSE value increased and the training performance of the BPNN decreased down to a certain point. The decreasing trend of the MMSE value was not significant when using more than 60 hidden nodes in the BPNN. Particularly, the BPNN with 50 hidden nodes was found to be performing well. This result also demonstrated the reliability of the empirical formulas in determining the number of hidden nodes. As a result, a total of 50 nodes was determined to be used in the BPNN.

Figure. 4.10 shows the comparison of the MMSE value with different prediction lead time. As shown in Figure. 4.10, with the increase in the prediction lead time, the MMSE value of the BPNN also increased. This indicated that by increasing the prediction lead time, the convergence rate of the BPNN decreased gradually.

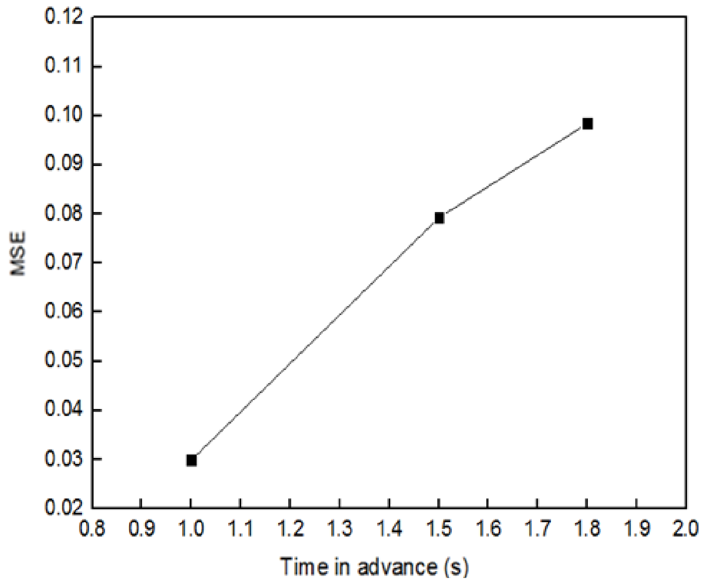

Figure. 4.10 MMSE of the BPNN with different prediction lead time.

4.2.4 Model Testing and Results of New Method

In this section of the thesis, the prediction results based on the new method are presented. Also, the comparison of the prediction results between the two methods proposed in this thesis is also shown in figures. The BPNN model was tested using the testing dataset which included 20% of all observations.

In the new method, 0 was also determined as the threshold value to identify the result properties. If the computed output of the BPNN was large than 0, a driver's driving behavior was predicted to be a merging behavior, whereas if the computed output was less than 0, a driver's driving behavior was predicted to be a lane-keeping behavior. The prediction performances of the two methods are compared in Figure. 4.11, Figure. 4.12. The performances were measured for both merge events and non-merge events.

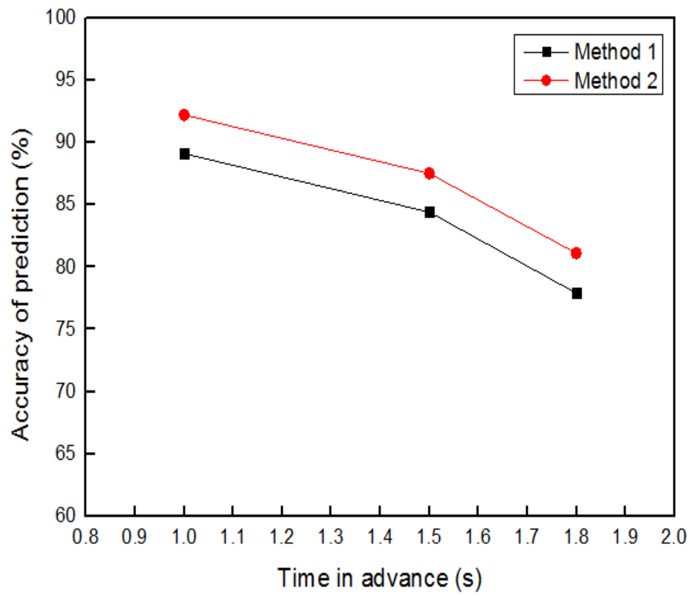

Figure. 4.11 Comparison of prediction results for non-merge events.

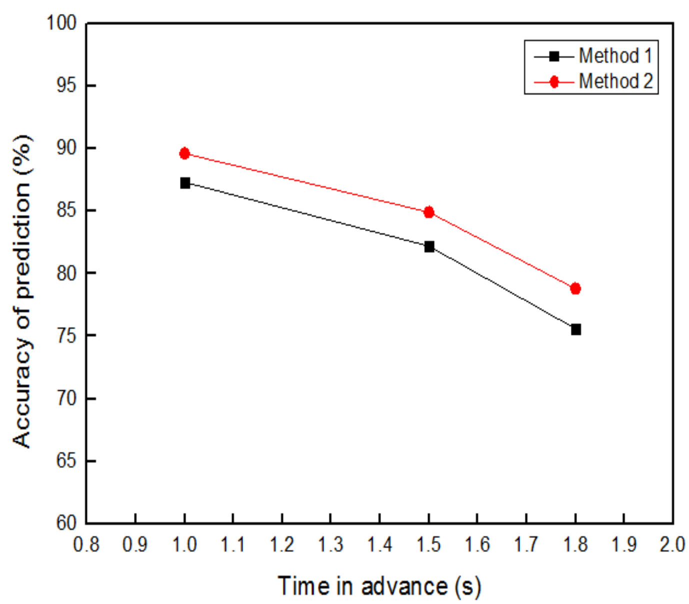

Figure. 4.12 Comparison of prediction results for merge events.

In method 1, the BPNN model was established using one frame of variables as the input vectors. Whereas in method 2 (new method), the BPNN model was established using six frames of variables as the input vectors. As the results shown in Figure. 4.11, Figure. 4.12, with the increase in the prediction lead time, the prediction accuracies for both of the two methods gradually decreased. The results revealed that the prediction performance of the BPNN model was better when predictions were made closer to a merging behavior. Predictions made earlier than 1.8s before a merging behavior were unreliable.

The results also indicated that the BPNN model established by method 1 performed poorly compared to the BPNN model established by method 2. Based on method 2, the prediction accuracies for merge events were improved to 89.6%, 84.9%, and 78.8% with a prediction lead time of 1s, 1.5s, and 1.8s, respectively. As expected, the method using six frames of variables as the input vectors for the BPNN model could improve the model prediction accuracy. The prediction results based on method 2 are also displayed in Table. 4.3.

Time in advance	Accuracy of non-merge events	Accuracy of merge events
1	92.2%	89.6%
1.5	87.5%	84.9%
1.8s	81.1%	78.8%

Table 4.3 Summary of prediction results based on new method.

Chapter 5: Conclusions

5.1 Research Conclusions

Lane-changing behavior at freeway on-ramps has a significant effect on driving safety and the stability of traffic flow. During the lane-changing process, the information processed by drivers is more complicated than that processed while remaining in a lane. If drivers fail to accurately judge the appropriate lane-changing time or the relative movement characteristics of related vehicles, vehicles accidents may occur. Thus, accurate prediction of lane-changing behavior is essential for a driving assistance system to ensure driver safety.

In this thesis, a BPNN model was developed to predict a driver's mandatory lane-changing decisions in advance of the driver's actual driving behavior with a lead time of 1s, 1.5s, and 1.8s. The main motivation for using a BPNN was its ability to learn and incorporate the uncertainties from real driving data. The publicly available NGSIM vehicle trajectory dataset that consists of transition traffic conditions and congested traffic conditions was used for the BPNN model training and testing. A total of 1,000 observations (500 merge events and 500 non-merge events) was obtained from the NGSIM dataset, 80% of observations were used for training the model and 20% of observations were used for testing the model. The model employed factors such as vehicle speeds relative to lead and lag vehicles in the target lane, lead and lag distances, and the distance from the beginning of the merging lane to the merging vehicle. Studies have shown that a driver's lane-changing decisions are essentially triggered by his/her surrounding driving environment.

The prediction results demonstrated that the method using six frames of variables as the input vectors for the BPNN model could improve the model prediction accuracy. Also, the number of nodes used in the hidden layer had a significant impact on the performance of the BPNN model. The results indicated that the best prediction accuracies in advance of a driver's actual driving behavior with a lead time of 1s, 1.5s, and 1.8s were at 89.6%, 84.9%, 78.8% for merge events, and for non-merge events were at 92.2%, 87.5%, 81.1% respectively.

5.2 Research Contributions

The main contributions of this thesis are presented below:

(1) The BPNN model established by two methods was able to accurately predict a driver's lane-changing decisions in advance of the driver's actual driving behavior with a lead time of 1s, 1.5s, and 1.8s. The model was shown to be more accurate than conventional logit models.

(2) The method using six frames of variables as the input vectors for the BPNN model improved the model prediction accuracy. Microscopic traffic simulation using this method will lead to more realistic lane-changing behavior prediction at freeway on-ramps.

(3) As a commonly used non-linear function approximation technique, the BPNN showed great advantages in predicting a driver's lane-changing decisions for its flexible and valid self-organization properties.

(4) The BPNN model proposed in this thesis was developed using traffic data collected under challenging traffic conditions: approaching congestion and congested conditions. Thus, the model enhanced the performance of microscopic simulation models at freeway on-ramps under congested traffic conditions.

(5) The driving environment features which were considered as the input variables for the BPNN model demonstrated very high reliability in predicting a driver's lane-changing decisions.

5.3 Research Future Work

A few related research topics can be explored in the future. First, it is possible to enhance the performance of the BPNN model by further improving the BP algorithm. In the future, the variable learning rate method can be used in the BP algorithm to realize the self-adaptive adjustment of learning rate. Second, the BPNN model can be used as a basic model or an initial point to create more complex lane-changing models. For future studies, perhaps the results of the prediction could be improved by using more input variables. Third, further studies can be performed considering various weather conditions to investigate the effect of weather conditions on the lane-changing behavior of drivers.

Also, further studies need to investigate the driving characteristics by considering the gender of a driver. Fourth, future studies can be intended to real-time predict lane-changing trajectory based on BPNN. Meanwhile, an on-road lane-changing detector can be developed to predict a driver's future driving state. This detector extracts signals from vehicle sensors and processes them into feature vectors, which are then used for offline training and online prediction. Finally, the simulation results could be used to assess the safety of lane-changing behavior and lay a necessary foundation for further development of the driver assistance systems. Last, other AI and advanced time-series methods [80], such as SVMs, Bayesian networks, and HMMs may be explored to predict a driver's lane-changing behavior.

REFERENCES

[1] "Annual Global Road Crash Statistics" [Online]. Available: http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics. [Accessed: 04-Sep-2014].

[2] M. Hoedemaeker, and K. A. Brookhuis, "Behavioural adaptation to driving with an adaptive cruise control (ACC)," *Transportation Research Part F: Traffic Psychology and Behaviour,* vol. 1, no. 2, pp. 95-106, 1998.

[3] M. Rahman, M. Chowdhury, Y. Xie, and Y. He, "Review of microscopic lane-changing models and future research opportunities," *IEEE transactions on intelligent transportation systems*, vol. 14, no. 4, pp. 1942-1956, 2013.

[4] O. M. Carsten, and F. Tate, "Intelligent speed adaptation: accident savings and cost–benefit analysis," *Accident Analysis & Prevention,* vol. 37, no. 3, pp. 407-416, 2005.

[5] J. D. Alonso, E. R. Vidal, A. Rotter, and M. Muhlenberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *IEEE Transactions on Vehicular Technology,* vol. 57, no. 5, pp. 2736-2746, 2008.

[6] C. Ding, W. Wang, X. Wang, and M. Baumann, "A neural network model for driver's lane-changing trajectory prediction in urban traffic flow," *Mathematical Problems in Engineering,* vol. 2013, 2013.

[7] D. Zhao, H. Peng, K. Nobukawa, S. Bao, D. J. LeBlanc, and C. S. Pan, "Analysis of mandatory and discretionary lane change behaviors for heavy trucks," *Ann*

Arbor, vol. 1001, pp. 48109-2133, 2014.

[8] J. Peng, Y. Guo, R. Fu, W. Yuan, and C. Wang, "Multi-parameter prediction of drivers' lane-changing behaviour with neural network model," *Applied ergonomics,* vol. 50, pp. 207-217, 2015.

[9] A. Doshi, and M. M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *IEEE Transactions on Intelligent Transportation Systems,* vol. 10, no. 3, pp. 453-462, 2009.

[10] F. Lethaus, and J. Rataj, "Do eye movements reflect driving manoeuvres?," *IET Intelligent Transport Systems,* vol. 1, no. 3, pp. 199, 2007.

[11] A. Doshi, B. Morris, and M. M. Trivedi, "On-road prediction of driver's intent with multimodal sensory cues," *IEEE Pervasive Computing,* vol. 10, no. 3, pp. 22-34, 2011.

[12] J. Lee, J. Choi, K. Yi, M. Shin, and B. Ko, "Lane-keeping assistance control algorithm using differential braking to prevent unintended lane departures," *Control Engineering Practice,* vol. 23, pp. 1-13, 2014.

[13] P. Edara, Y. Hou, and C. Sun, *Analysis of Driver Merging Behavior at Lane Drops on Freeways*, 2013.

[14] Y. Hou, P. Edara, and C. Sun, "A genetic fuzzy system for modeling mandatory lane changing." pp. 1044-1048.

[15] Y. Hou, P. Edara, and C. Sun, "A genetic fuzzy system for modeling mandatory lane changing." pp. 1044-1048.

[16] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, *A driver behavior recognition*

method based on a driver model framework, 0148-7191, SAE Technical Paper, 2000.

[17] Y. Hou, “Artificial intelligence techniques for modeling dynamic traffic behavior at bottlenecks,” UNIVERSITY OF MISSOURI-COLUMBIA, 2014.

[18] R. Herman, and G. Weiss, “Comments on the highway-crossing problem,” *Operations Research,* vol. 9, no. 6, pp. 828-840, 1961.

[19] D. R. Drew, L. R. LaMotte, J. A. Wattleworth, and J. H. Buhr, “Gap acceptance in the freeway merging process,” *Highway research record*, no. 208, 1967.

[20] A. J. Miller, “Nine estimators of gap-acceptance parameters,” *Publication of: Traffic Flow and Transportation*, 1900.

[21] C. F. Daganzo, “Estimation of gap acceptance parameters within and across the population from direct roadside observation,” *Transportation Research Part B: Methodological,* vol. 15, no. 1, pp. 1-15, 1981.

[22] H. Mahmassani, and Y. Sheffi, “Using gap sequences to estimate gap acceptance functions,” *Transportation Research Part B: Methodological,* vol. 15, no. 3, pp. 143-148, 1981.

[23] P. G. Gipps, “A model for the structure of lane-changing decisions,” *Transportation Research Part B: Methodological,* vol. 20, no. 5, pp. 403-414, 1986.

[24] H. KITA, “Effects of Merging Lane Length on the Merging Behavior at Expressway On-Ramps,” *Transportation and Traffic Theory,* 1993.

[25] Q. Yang, and H. N. Koutsopoulos, “A microscopic traffic simulator for evaluation

of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113-129, 1996.

[26] K. Ahmed, M. Ben-Akiva, H. Koutsopoulos, and R. Mishalani, "Models of freeway lane changing and gap acceptance behavior," *Transportation and traffic theory,* vol. 13, pp. 501-515, 1996.

[27] H. Kita, "A merging–giveway interaction model of cars in a merging section: a game theoretic analysis," *Transportation Research Part A: Policy and Practice,* vol. 33, no. 3, pp. 305-312, 1999.

[28] P. Hidas, "Modelling vehicle interactions in microscopic simulation of merging and weaving," *Transportation Research Part C: Emerging Technologies,* vol. 13, no. 1, pp. 37-62, 2005.

[29] T. Toledo, H. N. Koutsopoulos, and M. Ben-Akiva, "Integrated driving behavior modeling," *Transportation Research Part C: Emerging Technologies,* vol. 15, no. 2, pp. 96-112, 2007.

[30] S. Tezuka, H. Soma, and K. Tanifuji, "A study of driver behavior inference model at time of lane change using Bayesian networks." pp. 2308-2313.

[31] A. Sathyanarayana, P. Boyraz, and J. H. Hansen, "Driver behavior analysis and route recognition by hidden Markov models." pp. 276-281.

[32] A. Pentland, and A. Liu, "Modeling and prediction of human behavior," *Neural computation,* vol. 11, no. 1, pp. 229-242, 1999.

[33] C. C. MacAdam, and G. E. Johnson, "Application of elementary neural networks and preview sensors for representing driver steering control behaviour," *Vehicle*

System Dynamics, vol. 25, no. 1, pp. 3-30, 1996.

[34] R. S. Tomar, S. Verma, and G. S. Tomar, "Prediction of lane change trajectories through neural network." pp. 249-253.

[35] W. Yao, H. Zhao, P. Bonnifait, and H. Zha, "Lane change trajectory prediction by using recorded human driving data." pp. 430-436.

[36] P. Kumar, M. Perrollaz, S. Lefevre, and C. Laugier, "Learning-based approach for online lane change intention prediction." pp. 797-802.

[37] Q. Meng, and J. Weng, "Classification and regression tree approach for predicting drivers' merging behavior in short-term work zone merging areas," *Journal of Transportation Engineering*, vol. 138, no. 8, pp. 1062-1070, 2012.

[38] F. Marczak, W. Daamen, and C. Buisson, "Key variables of merging behaviour: empirical comparison between two sites and assessment of gap acceptance theory," *Procedia-Social and Behavioral Sciences*, vol. 80, pp. 678-697, 2013.

[39] A. Kondyli, and L. Elefteriadou, "Driver behavior at freeway-ramp merging areas based on instrumented vehicle observations," *Transportation Letters*, vol. 4, no. 3, pp. 129-142, 2012.

[40] M. Sarvi, A. Ceder, and M. Kuwahara, "Modeling of freeway ramp merging process observed during traffic congestion." pp. 483-502.

[41] M. Sarvi, and M. Kuwahara, "Microsimulation of freeway ramp merging processes under congested traffic conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 470-479, 2007.

[42] J. Wu, M. McDonald, and K. Chatterjee, "A detailed evaluation of ramp metering

impacts on driver behaviour," *Transportation Research Part F: Traffic Psychology and Behaviour,* vol. 10, no. 1, pp. 61-75, 2007.

[43] Y.-C. Chen, H.-C. Hsieh, and H.-C. Lin, "Improved precision recommendation scheme by BPNN algorithm in O2O commerce." pp. 324-328.

[44] "Next Generation Simulation Fact Sheet, Washington, D.C., USA. [Online]. Available: https://www.fhwa.dot.gov/publications/research/operations/its/06135/ [Accessed: 22-Dec-2006].

[45] Y. Hou, P. Edara, and C. Sun, "Modeling mandatory lane changing using Bayes classifier and decision trees," *IEEE Transactions on Intelligent Transportation Systems,* vol. 15, no. 2, pp. 647-655, 2014.

[46] S. Russell, P. Norvig, and A. Intelligence, "A modern approach," *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs,* vol. 25, pp. 27, 1995.

[47] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets," *Journal of Machine Learning Research,* vol. 6, no. Apr, pp. 363-392, 2005.

[48] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine Learning,* vol. 73, no. 2, pp. 185-214, 2008.

[49] Y. Lee, M. Sayyadian, A. Doan, and A. S. Rosenthal, "eTuner: tuning schema matching software using synthetic scenarios," *The VLDB Journal—The International Journal on Very Large Data Bases,* vol. 16, no. 1, pp. 97-122, 2007.

[50] V. Topuz, "Hourly Traffic Flow Prediction Using Different ANN Models," *Urban*

Transport and Hybrid Vehicles: InTech, 2010.

[51] T. Fawcett, “An introduction to ROC analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861-874, 2006.

[52] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification." pp. 3360-3367.

[53] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition." pp. 2126-2136.

[54] K. M. Leung, “Naive bayesian classifier,” *Polytechnic University Department of Computer Science/Finance and Risk Engineering*, 2007.

[55] V. Jakkula, “Tutorial on support vector machine (svm),” *School of EECS, Washington State University*, vol. 37, 2006.

[56] S. Dreiseitl, and L. Ohno-Machado, “Logistic regression and artificial neural network classification models: a methodology review,” *Journal of biomedical informatics*, vol. 35, no. 5, pp. 352-359, 2002.

[57] M. Khashei, and M. Bijari, “An artificial neural network (p, d, q) model for timeseries forecasting,” *Expert Systems with applications*, vol. 37, no. 1, pp. 479-489, 2010.

[58] T. Kohonen, “An introduction to neural computing,” *Neural networks*, vol. 1, no. 1, pp. 3-16, 1988.

[59] W. Jin, Z. J. Li, L. S. Wei, and H. Zhen, "The improvements of BP neural network learning algorithm." pp. 1647-1649.

[60] J. Mikulić, and D. Prebežac, “Accounting for dynamics in attribute-importance

and for competitor performance to enhance reliability of BPNN-based importance–performance analysis," *Expert Systems with Applications,* vol. 39, no. 5, pp. 5144-5153, 2012.

[61] D. Kanungo, M. Arora, S. Sarkar, and R. Gupta, "A comparative study of conventional, ANN black box, fuzzy and combined neural and fuzzy weighting procedures for landslide susceptibility zonation in Darjeeling Himalayas," *Engineering Geology,* vol. 85, no. 3, pp. 347-366, 2006.

[62] J. S. Long, and J. Freese, *Regression models for categorical dependent variables using Stata*: Stata press, 2006.

[63] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting,* vol. 14, no. 1, pp. 35-62, 1998.

[64] M. Şengül, S. Öztürk, H. B. Çetinkaya, and T. Erfidan, "New phenemenon on power transformers and fault identification using artificial neural networks." pp. 767-776.

[65] V. G. Gudise, and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks." pp. 110-117.

[66] " Introduction of Several Activation Functions" [Online]. Available: http://www-rohan.sdsu.edu/doc/matlab/toolbox/nnet/backpr52.html. [Accessed 06-November-2007].

[67] J. Li, J.-h. Cheng, J.-y. Shi, and F. Huang, "Brief introduction of back propagation

(BP) neural network algorithm and its improvement,” *Advances in Computer Science and Information Engineering*, pp. 553-558, 2012.

[68] "Maxima and Minima of Gradient Descent Method" [Online]. Available: https://en.wikipedia.org/wiki/Maxima_and_minima.

[69] W. J. Yan, “Research on Merging Behavior of Freeway On-ramp Merging Area,” *Chinese Academic Journals,* 2015.

[70] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "Estimation of vehicle trajectories from observed discrete positions and Next-Generation Simulation Program (NGSIM) data."

[71] C. Thiemann, M. Treiber, and A. Kesting, “Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 2088, pp. 90-101, 2008.

[72] J. Alvarez-Ramirez, E. Rodriguez, and J. C. Echeverría, “Detrending fluctuation analysis based on moving average filtering,” *Physica A: Statistical Mechanics and its Applications,* vol. 354, pp. 199-219, 2005.

[73] M. S. Grewal, *Kalman filtering*: Springer, 2011.

[74] Z. Guo, and W. Sha, “Modelling the correlation between processing parameters and properties of maraging steels using artificial neural network,” *Computational Materials Science,* vol. 29, no. 1, pp. 12-28, 2004.

[75] Gowrishankar, and P. Satyanarayana, “Neural network based traffic prediction for wireless data networks,” *International Journal of Computational Intelligence*

Systems, vol. 1, no. 4, pp. 379-389, 2008.

[76] T. Kaya, E. Aktas, İ. Topçu, and B. Ülengin, "Modeling toothpaste brand choice: an empirical comparison of artificial neural networks and multinomial probit model," *International Journal of Computational Intelligence Systems,* vol. 3, no. 5, pp. 674-687, 2010.

[77] V. Kůrková, "Kolmogorov's theorem and multilayer neural networks," *Neural networks,* vol. 5, no. 3, pp. 501-506, 1992.

[78] B. Aazhang, B.-P. Paris, and G. C. Orsak, "Neural networks for multiuser detection in code-division multiple-access communications," *IEEE Transactions on Communications,* vol. 40, no. 7, pp. 1212-1222, 1992.

[79] S. Ding, W. Jia, C. Su, X. Liu, and J. Chen, "An improved bp neural network algorithm based on factor analysis."

[80] L. Yingwei, N. Sundararajan, and P. Saratchandran, "Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm," *IEEE Transactions on neural networks,* vol. 9, no. 2, pp. 308-318, 1998.