MID-AMERICA
TRANSPORTATION CENTER

Report # MATC-MU: 184

Final Report
25-1121-003-184

University of Nebraska Lincoln
K-STATE Kansas State University
KU The University of Kansas
Missouri S&T University of Science & Technology
Lincoln University
University of Missouri
Iowa State University
The University of Iowa

Analysis of Driver Merging Behavior at Lane Drops on Freeways

Praveen Edara, Ph.D., P.E., PTOE
Associate Professor
Department of Civil Engineering
University of Missouri-Columbia

Yi Hou, M.S.
Graduate Research Assistant
Carlos Sun, Ph.D., P.E., JD
Associate Professor

Mizzou
University of Missouri

2013

A Coopertative Research Project sponsored by
U.S. Department of Tranportation-Research, Innovation and Technology Innovation Administration

MATC

Analysis of Driver Merging Behavior at Lane Drops on Freeways
Final Report

Praveen Edara, Ph.D., P.E., PTOE
Associate Professor
Department of Civil Engineering
University of Missouri-Columbia

Carlos Sun, Ph.D., P.E., JD
Associate Professor
Department of Civil Engineering
University of Missouri-Columbia

Yi Hou, M.S.
Graduate Research Assistant
Department of Civil Engineering
University of Missouri-Columbia

A Report on Research Sponsored by

Mid-America Transportation Center

University of Nebraska–Lincoln

December 2013

Technical Report Documentation Page

1. Report No. 25-1121-003-184	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Analysis of Driver Merging Behavior at Lane Drops on Freeways		5. Report Date December 2013
		6. Performing Organization Code
7. Author(s) Praveen Edara, Carlos Sun, and Yi Hou		8. Performing Organization Report No. 25-1121-003-184
9. Performing Organization Name and Address Mid-America Transportation Center 2200 Vine Street PO Box 830851 Lincoln, NE 68583-0851		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address Research and Innovative Technology Administration 1200 New Jersey Ave., SE Washington, D.C. 20590		13. Type of Report and Period Covered September 2012 – December 2013
		14. Sponsoring Agency Code MATC TRB RiP No. 1250770
15. Supplementary Notes		

16. Abstract

Lane changing assistance systems advise drivers on safe gaps for making mandatory lane changes at lane drops. In this study, such a system was developed using a Bayes classifier and a decision tree to model lane changes. Detailed vehicle trajectory data from the Next Generation Simulation (NGSIM) dataset were used for model development (US Highway 101) and testing (Interstate 80). The model predicted driver decisions regarding whether or not to merge as a function of certain input variables. The best results were obtained when both the Bayes and decision tree classifiers were combined into a single classifier using a majority voting principle. Predictive accuracy was 94.3% for non-merge events and 79.3% for merge events. In a lane change assistance system, the accuracy of non-merge events is more critical than accuracy for merge events. Misclassifying a non-merge event as a merge event could result in a crash, while misclassifying a merge event as a non-merge event would only result in a lost opportunity to merge. Sensitivity analysis performed by assigning a higher misclassification cost for non-merge events resulted in even higher accuracy for non-merge events, but lower accuracy for merge events.

17. Key Words Lane changing models, NGSIM, Safe gap acceptance, Mandatory lane changes	18. Distribution Statement

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 32	22. Price

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Federal Highway Administration (FHWA)
k Nearest Neighbor (kNN)
Mid-America Transportation Center (MATC)
Next Generation Simulation (NGSIM)

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation’s University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Abstract

Lane changing assistance systems advise drivers on safe gaps for making mandatory lane changes at lane drops. In this study, such a system was developed using a Bayes classifier and a decision tree to model lane changes. Detailed vehicle trajectory data from the Next Generation Simulation (NGSIM) dataset were used for model development (US Highway 101) and testing (Interstate 80). The model predicted driver decisions regarding whether or not to merge as a function of certain input variables. The best results were obtained when both the Bayes and decision tree classifiers were combined into a single classifier using a majority voting principle. Predictive accuracy was 94.3% for non-merge events and 79.3% for merge events. In a lane change assistance system, the accuracy of non-merge events is more critical than accuracy for merge events. Misclassifying a non-merge event as a merge event could result in a crash, while misclassifying a merge event as a non-merge event would only result in a lost opportunity to merge. Sensitivity analysis performed by assigning a higher misclassification cost for non-merge events resulted in even higher accuracy for non-merge events, but lower accuracy for merge events.

Chapter 1 Introduction

With an increase in the deployment of sensor technology in automobiles, driver assistance systems such as adaptive cruise control, collision avoidance, and lane departure warning systems have become a reality in recent years. In terms of lane changing assistance, the current technology focuses primarily on blind spot identification and warning. Limited research exists on other forms of lane changing assistance systems. This report describes a lane changing assistance system that advises drivers of safe and unsafe gaps for making mandatory lane changes.

Lane changing models describe driver lane changing behaviors under various traffic conditions. These models are an essential component of microscopic traffic simulation, and have been extensively studied in the literature. Much of the literature on lane change models is based on gap acceptance. A driver makes a lane change when both the lead and the lag gaps in the target lane are acceptable. In the 1960s and 1970s, various gap acceptance models were developed based on assumed distributions of critical lead and lag gap lengths. Herman and Weiss (1) assumed an exponential distribution for critical gaps; Drew et al. (2) assumed a lognormal distribution; and Miller (3) assumed a normal distribution. Daganzo (4) modeled driver merging from the minor leg of a stop controlled T-intersection to the major leg using a probit model. Gipps (5) designed a hierarchical lane changing structure that was implemented in a microscopic traffic simulator. Kita (6) modeled driver merging behavior from a freeway on-ramp using a logit model for gap acceptance. Yang and Koutsopoulos (7) established a rule-based lane changing model that was incorporated into the microscopic simulator MITSIM. Ahmed et al. (8) developed a generic lane changing model that captured lane changing behavior under both mandatory and discretionary lane changes. Kita (9) also developed a two-person, non-zero, noncooperative game to model the interactions between drivers in the target lane and the merging lane. Hidas (10) used intelligent-agent-based techniques to model driver lane changing behavior, implementing the model in the ARTEMiS traffic simulator. Toledo et al. (11) proposed an integrated driving behavior model that captured both lane changing and acceleration behaviors. Recently, Meng and Weng (12) used statistical methods, such as the classification and regression tree (CART), to predict merging behavior near work zone tapers. In a recent study, Hou et al. (13) developed a genetic fuzzy model to predict the merging behavior of drivers at lane drops.

In summary, several types of lane changing models have been proposed in the literature, with the main goal of developing accurate traffic simulation models. However, none of these models were intended for use in a real-time lane changing assistance system that advises drivers on when it is safe or unsafe to merge. One main difference between simulation and lane change assistance system applications is the difference between merge and non-merge decisions in terms of the relative importance of misclassification. In a simulation model, the effect of a non-merge event misclassified as a merge event affects only mobility; however, the same misclassification in a lane change assistance system could impact traffic safety significantly. In other words, misclassifying a merge event as a non-merge event would result in a lost opportunity to merge, but would not have a negative impact on safety. Thus, any model of lane change targeted for use in vehicles as part of an assistance system must assign greater importance to not misclassifying non-merge events as merge events. Many of the models proposed in the literature are not focused on this new application.

In the current report, Bayes classifier and decision tree methods were applied to develop models for mandatory lane changes at lane drops. Both methods have been applied extensively in machine learning systems built for decision making in many disciplines. They have several advantages for modeling lane changing. Both relax the assumptions of the mathematical forms and variable distributions of traditional lane-changing models. Therefore, they can mimic the complex nonlinear nature of driver lane changing behavior more realistically. One additional advantage of the Bayes classifier is its ability to take into account the cost of misclassification. In a Bayes classifier, it is possible to assign a higher cost of misclassification to non-merge events.

Bayes classifier and decision tree models were developed using identical training and validation data. Then, both classifiers were combined into a single hybrid classifier. When tested on a new dataset from a different highway segment, the combined classifier outperformed the individual classifiers in terms of the accuracy of non-merge events. In this report, mandatory lane changes at lane drops refer only to those executed by traffic entering from a ramp. The lane changes made by vehicles exiting the mainline, although also mandatory, were outside the scope of this study. Discretionary lane changes, performed when drivers perceive driving conditions in the target lanes to be better, were also beyond the scope of this study.

Chapter 2 Data

2.1. Data Reduction

In this study, traffic data provided by the Federal Highway Administration's (FHWA) Next Generation Simulation (NGSIM) project (14) were used to build the lane changing models. NGSIM is an open source dataset that has been used in previous research on simulation model development and testing (15, 16). The NGSIM data included vehicle trajectories on a segment of southbound US Highway 101 (Hollywood Freeway) in Los Angeles, California and a segment of Interstate 80 in San Francisco, California. US Highway 101 data were collected for 45 minutes, from 7:50 a.m. to 8:35 a.m., on June 15, 2005. Interstate 80 data were also collected for 45 minutes, from 4:00 p.m. to 4:15 p.m. and from 5:00 p.m. to 5:30 p.m. on April 13, 2005. Both datasets represented two traffic states—conditions when congestion was building up (the period of the first 15 minutes), denoted as the transition period, and congested conditions (the period of the remaining 30 minutes). Table 2.1 shows the aggregate speed and volume statistics of the NGSIM dataset for every 15 minutes. During the congested period, flows and speeds both decreased. As depicted in figure 2.1, the study segment of US Highway 101 was located between an on-ramp and off-ramp, and was 2,100 feet long, with five freeway lanes and an auxiliary lane. The study segment of Interstate 80 was 1,650 feet in length, and also had five freeway lanes and an auxiliary lane, and one on-ramp.

Previous research (18, 19, 20, 21) has shown that NGSIM speed measurements exhibit noises (random errors). Data smoothing techniques such as moving average (21), Kalman filtering (22), and Kalman smoothing (23) have been used to improve speed data quality. In this report, the moving average method was adopted to smooth the speed measurements.

The longitudinal and lateral coordinates, speed, acceleration, and headway for each vehicle were obtained from trajectory data at a resolution of 10 frames per second. Given the focus of this study on mandatory lane changes, only trajectory data for vehicles in the auxiliary lane and the adjacent lane were used for model development. Hereafter, the auxiliary lane is referred to as the merge lane, and the adjacent lane as the target lane. The speed and position of each vehicle were identified in one-second intervals. The one-second intervals produced data with comparable sample sizes for both lane changing and non-lane changing events. Other researchers (12) have also used one-second intervals to analyze driver lane changing behavior. Since it is impossible to determine the intent of a driver using vehicle trajectory data alone, the observed behavior of drivers was modeled. During every one-second interval, a driver's behavior was identified as either merge or no-merge. Merge events occurred when a vehicle's lateral coordinate began to shift toward the adjacent target lane direction without oscillations. Otherwise, these were deemed non-merge events. A single driver could participate in several non-merge events, but only one merge event.

A total of 686 observations were obtained from US Highway 101, 373 being non-merge and 313 being merge events. As discussed in Hastie et al. (24), there is no general rule on how many observations should be assigned to training and validation. In order to obtain a high degree of accuracy, a large training dataset is required. Other studies have used 80% of the dataset for training and 20% for model validation (25, 26). Based on these studies, the current dataset was divided into two groups—80% of observations were used for training, and 20% were used for validation. The model was tested using the Interstate 80 dataset consisting of 667 observations, 459 being non-merge and 208 being merge events.

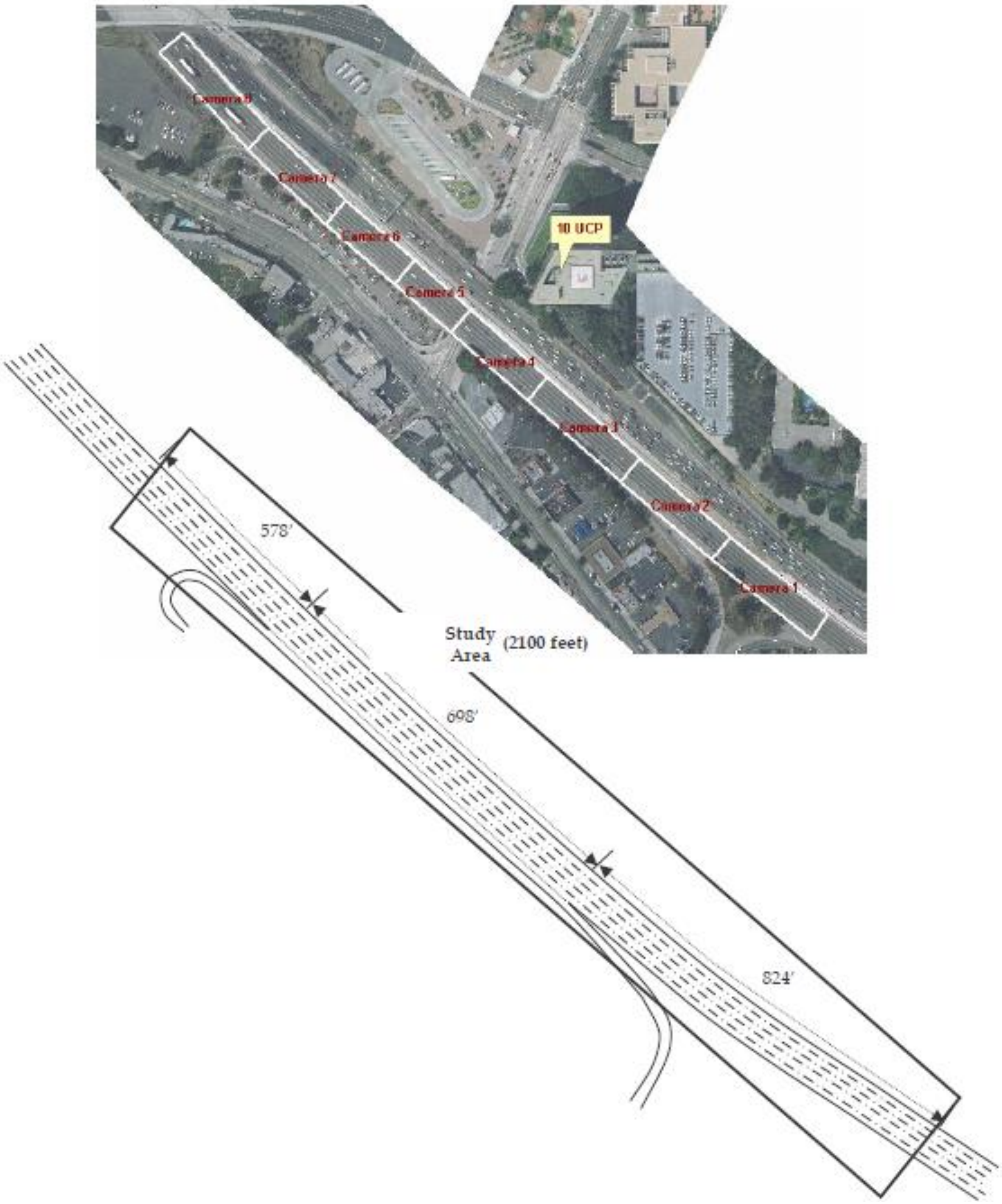
Table 2.1 Summary statistics

A． Summary statistics of US Highway 101 dataset

Traffic condition	**Time period**	**Flow (vph)**	**Time mean speed**	
			m/s	**km/h**
Transition	7:50 a.m. – 8:05a.m.	8612	12.55	45.16
Congested	8:05 a.m. – 8:20a.m.	8016	11.10	39.96
	8:20 a.m. – 8:35a.m.	7604	9.74	35.05

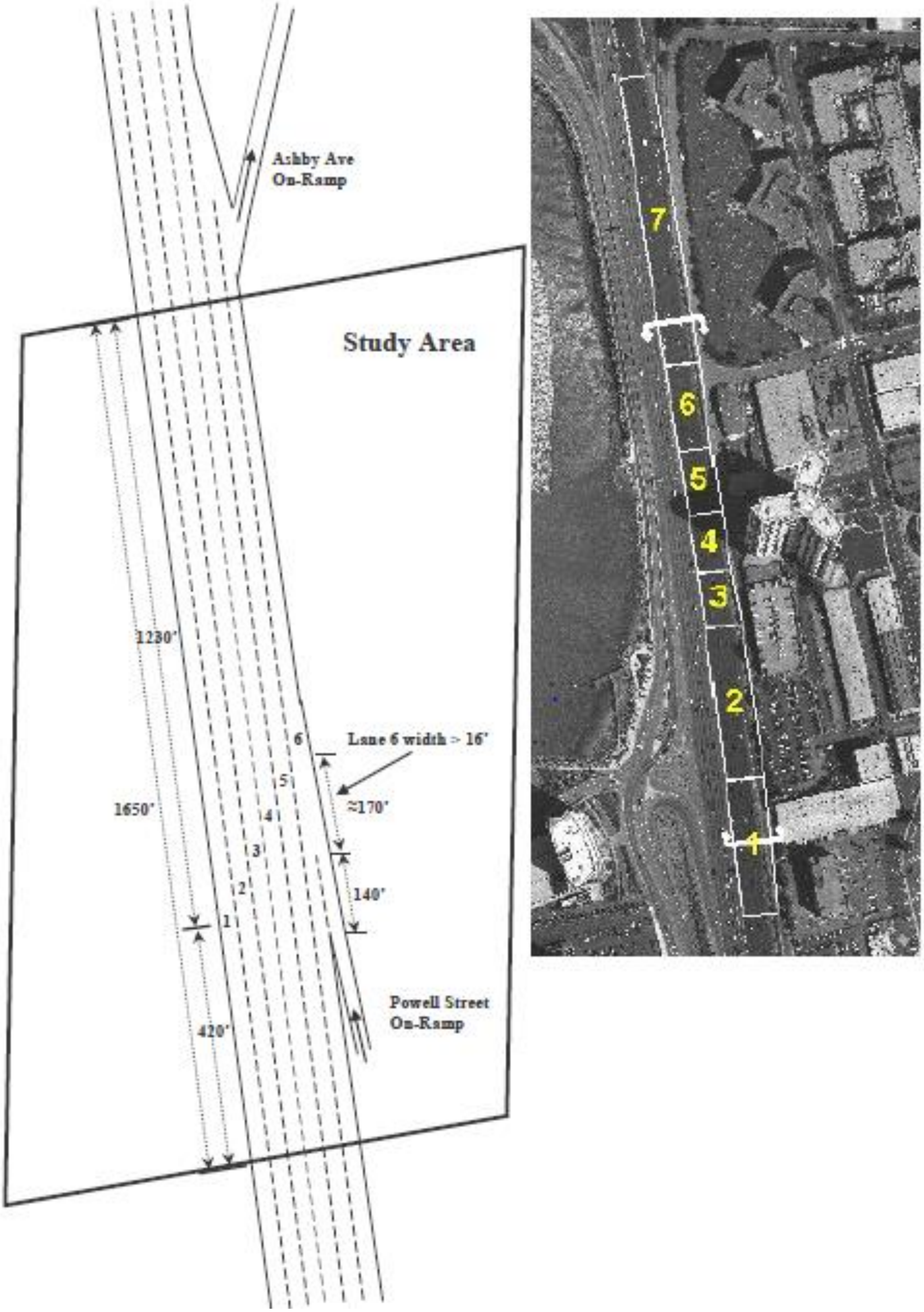
B. Summary statistics of Interstate 80 dataset

Traffic condition	**Time period**	**Flow (vph)**	**Time mean speed**	
			m/s	**km/h**
Transition	4:00 p.m. – 4:15p.m.	8144	9.92	35.71
Congested	5:00 p.m. – 5:15p.m.	7288	8.34	30.13
	5:15 a.m. – 5:30a.m.	7048	7.78	28.00



(a)

Figure 2.1 US Highway 101 (a) and Interstate 80 (b) study corridor from NGSIM (14)



(b)

Figure 2.1 US Highway 101 (a) and Interstate 80 (b) study corridor from NGSIM (14) (cont'd)

2.2. Input Variables

At any given instant, a driver traveling in the merge lane assesses traffic conditions in both the target lane and the merge lane in order to decide whether to merge. Several factors may affect a driver's lane changing decision. In this study, five factors, or, dimensions that were found to affect driver merging decisions in previous studies (8, 10) were considered as input variables for the models. These factors are shown in figure 2.2, and are defined below.

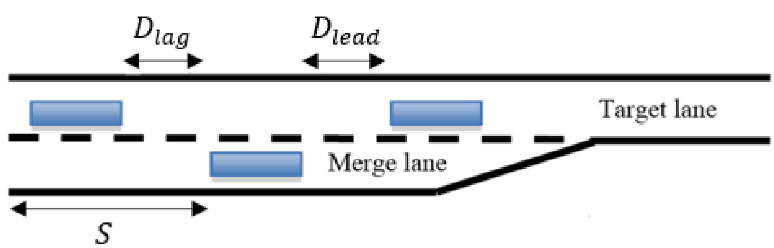


Figure 2.2 Schematic illustrating input variables

- ΔV_{lead}(m/s): The speed difference between the lead vehicle in the target lane and the merging vehicle, in feet per second. ΔV_{lead} can be expressed as

$$\Delta V_{lead} = V_{lead} - V_{merge},$$

where,

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

- ΔV_{lag}(m/s): The speed difference between the lag vehicle in the target lane and the merging vehicle, in feet per second, ΔV_{Lag}, can be expressed as:

$$\Delta V_{lag} = V_{lag} - V_{merge},$$

where,

V_{Lag} is the speed of the lag vehicle.

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

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

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

Chapter 3 Methodology

3.1. Bayes Classifier

3.1.1 Bayes Decision Theory

Let y_1, y_2 denote the merge and non-merge classes. According to the Bayesian classification rule (27),

$$P(y_i|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y_i)P(y_i)}{p(\boldsymbol{x})}, \ i = 1,2 \tag{3.1}$$

where,

$\boldsymbol{x}$ is the input vector, $P(.)$is the probability, and $p(.)$ is the probability density function.

The Bayes classification rule (24) is stated as follows:

- If $P(y_1|\boldsymbol{x}) > P(y_2|\boldsymbol{x})$, $\boldsymbol{x}$ is classified to y_1.
- If $P(y_1|\boldsymbol{x}) < P(y_2|\boldsymbol{x})$, $\boldsymbol{x}$ is classified to y_2.
- If $P(y_1|\boldsymbol{x}) = P(y_2|\boldsymbol{x})$, $\boldsymbol{x}$ can be assigned to either y_1 or y_2.

Using equation 3.1, the classification decision is equivalently based on the inequalities,

$$p(\boldsymbol{x}|y_1)P(y_1) > (<) p(\boldsymbol{x}|y_2)P(y_2) \tag{3.2}$$

3.1.2 Risk of Misclassification

Risk considers both the likelihood of misclassification and the cost of the misclassification. A penalty term λ_{ki} denotes the cost of misclassifying $\boldsymbol{x}$ to a wrong class y_i while belonging to class y_k (27). In order to minimize the average risk, the classification decision inequalities (3.2) become,

$$(\lambda_{12} - \lambda_{11}) p(\boldsymbol{x}|y_1) P(y_1) > (<) (\lambda_{21} - \lambda_{22}) p(\boldsymbol{x}|y_2) P(y_2) \tag{3.3}$$

Adopting the assumption that $\lambda_{ij} > \lambda_{ii}$ and $\lambda_{ii} = 0$, the Bayes classification rule becomes,

$$\boldsymbol{x} \text{ belongs to } y_1(y_2) \text{ if } l_{12} = \frac{p(x|y_1)}{p(x|y_2)} > (<) \frac{P(y_2)\lambda_{21}}{P(y_1)\lambda_{12}} \tag{3.4}$$

where,

l_{12} is likelihood ratio.

3.1.3 k Nearest Neighbor Density Estimation

A driver's merging behavior can be predicted using the class-conditional probability density function, $p(\boldsymbol{x}|y_i)$. In this study, the k nearest neighbor (kNN) density estimation method (28) was used to estimate the class-conditional probability density functions. The kNN estimation method was chosen because, similar to kernel estimation, it is a non-parametric method; thus, there is no need to assume a distributional form, unlike maximum likelihood. Using this method, the class-conditional probability density functions is estimated as,

$$p(\boldsymbol{x}|y_i) = \frac{k}{N_i V_i}, i = 1,2 \tag{3.5}$$

where,

N_i is the total number of training samples in class y_i, and V_i is the volume of the five-dimensional hypersphere (i.e., input data space) centered at $\boldsymbol{x}$ that contains k points from class y_i.

$P(y_i)$ is easily estimated from observations as follows:

$$P(y_i) = \frac{N_i}{N},\ i = 1,2 \tag{3.6}$$

where,

N_i is the total number of training samples in class y_i, and N is the total number of training samples.

By substituting equations 3.5 and 3.6 into equation 3.4, the Bayes classification rule is equivalent to,

$$\boldsymbol{x} \text{ belongs to } y_1\ (y_2) \text{ if } l_{12} = \frac{V_2}{V_1} > (<) \frac{\lambda_{21}}{\lambda_{12}} \tag{3.7}$$

Let r_i denote the radius of the hypersphere centered at $\boldsymbol{x}$ that contains k points from class y_i. Since the hypersphere dimension in this study is five (the total number of input variables), the likelihood ratio can be computed as,

$$l_{12} = \frac{V_2}{V_1} = \left(\frac{r_2}{r_1}\right)^5 \text{(29)} \tag{3.8}$$

3.1.4 Distance Measurement

The hypersphere radius r_i can be easily obtained by searching for the kth nearest distance from all the training vectors of class y_i. The weighted distance measure was used to calculate the hypersphere radius. Let $\boldsymbol{x_j}$ and $\boldsymbol{x_k}$ denote two vectors of l features. The weighted distance is:

$$D(\boldsymbol{x_j}, \boldsymbol{x_k}) = \sqrt{\sum_{i=1}^{l} w_i (x_{ji} - x_{ki})^2} \quad (3.9)$$

where,

w_i is the weights associated with features.

The maximum margin decision boundary established by support vector machines (SVM) is used to determine the weights (30). Let $\boldsymbol{q}$ be the query point whose class label is to be predicted. The SVM classifier gives the decision hyperplane $g(\boldsymbol{x})$. Let $\boldsymbol{p}$ be the point with the closest Euclidean distance to $\boldsymbol{q}$ on decision hyperplane $g(\boldsymbol{x})$**.** $R(\boldsymbol{q})_j$ is defined as,

$$R(\boldsymbol{q})_j = |\boldsymbol{e}_j^T \nabla g(\boldsymbol{p})| \quad (3.10)$$

where,

$\boldsymbol{e}_j$ denotes the canonical unit vector along input feature j. The weights are given by,

$$w(\boldsymbol{q})_j = \frac{(R(\boldsymbol{q})_j)^t}{\sum_{i=1}^{l} (R(\boldsymbol{q})_i)^t} \quad (3.11)$$

where,

t is a positive integer.

In this study, t values ranging from 1 to 4 were applied, and $t = 2$ produced the best model performance. In this case, the SVMs decision hyperplane was in linear form $g(\boldsymbol{x}) = \boldsymbol{b}^T \boldsymbol{x} + b_0 = 0$, Thus, $R(\boldsymbol{q})_j \equiv b_j$.

3.2 Decision Tree Model

A decision tree achieves a classification decision by performing a sequence of tests on feature vectors along a path of nodes (31). Each internal node in the tree provides a question, “Is feature $x_i \geq a$?”, where a is a threshold value. The binary answer to the question corresponds to a descendant node. At the end, each terminal node returns a class label. The size of a decision tree is the key factor in developing the decision tree model. If the size of a tree is too small, the tree results in high misclassification rates. On the other hand, if a tree grows too large, it could overfit the training data and perform poorly on testing data. Therefore, the suggested approach is to grow a tree with a large enough size, then prune the branches according to a set of pruning rules.

3.2.1 Node Splitting

In order to construct a decision tree, the set of questions at tree nodes are to be determined. Each node t is associated with a subset of training set $\boldsymbol{X}_t$. The root node is assigned with the entire training set. The goal of the binary split at each node is to produce subsets that are more homogeneous or purer than the parent subset. In this model, Shannon’s information theory (32) was adopted to measure the impurity of subset $\boldsymbol{X}_t$, also known as node impurity.

Let y_1, $y_2$denote the two classes: merge and non-merge. Let $P(y_i|t)$denote the probability that a sample in subset $\boldsymbol{X}_t$ belongs to class y_i, $i = 1,2$. Node impurity is then defined as,

$$I(t) = -\sum_{i=1}^{2} P(y_i|t) \log_2 P(y_i|t) \qquad (3.12)$$

$P(y_i|t)$ can be easily estimated by N_t^i/N_t, where N_t^i is the number of vectors in subset $\boldsymbol{X}_t$ that belong to class y_i, and N_t is the total number of vectors in subset $\boldsymbol{X}_t$. After performing a binary split at node t, a subset $\boldsymbol{X}_{tY}$ with an answer "Yes" is assigned to node t_Y, and a subset $\boldsymbol{X}_{tN}$ with answer "No" is assigned to node t_N. The decrease in node impurity $\Delta I(t)$ is given by,

$$\Delta I(t) = I(t) - \frac{N_{tY}}{N_t} I(t_Y) - \frac{N_{tN}}{N_t} I(t_N) \tag{3.13}$$

where,

N_{tY} and N_{tN} are the numbers of vectors in subsets $\boldsymbol{X}_{tY}$ and $\boldsymbol{X}_{tN}$.

By exhaustively searching for all candidate questions, the one that leads to the maximum impurity decrease is selected.

3.2.2 Stop-Splitting Criteria and Class Assignment

A threshold probability value P_0 is necessary to stop the node splitting process at any node. Splitting stops when more than $P_0 \times 100\%$ of vectors in the subset belong to any one single class, i.e., $\max_{i} P(y_i|t) > P_0$. In this model, 0.9 is selected as the threshold value, as this value will also ensure the tree grows large enough for pruning. Once a terminal node is determined, the class label is given by y_j where,

$$j = arg \max_{i} P(y_i|t) \tag{3.14}$$

3.2.3 Tree Pruning

Minimal cost-complexity pruning (33) was employed as the pruning rule in this report. Due to its computational efficiency, minimal cost-complexity pruning is one of the most common methods of pruning a decision tree. The sequence of subtrees generated by this pruning process is nested, meaning that the nodes that were previously cut off will not reappear in subsequent subtrees. The cost-complexity measure $R_\alpha(T)$ of decision tree T is defined as,

$$R_\alpha(T) = R(T) + \alpha|\tilde{T}| \tag{3.15}$$

where,

$R(T)$ is the substitution estimate for the overall misclassification rate of tree T, $\alpha \geq 0$ is the complexity parameter, and $|\tilde{T}|$ is the total number of terminal nodes in tree T.

Each value of α is associated with a subtree $T(\alpha)$ that minimizes $R_\alpha(T)$. As α increases from 0 to a sufficiently large number, the size of $T(\alpha)$ decreases from its largest size to the smallest size (only for the root node). If a subtree minimizes $R_\alpha(T)$ for a given value of α, it will remain minimizing $R_\alpha(T)$ until α increases to a jump point. Let $\{\alpha_k\}$ be the increasing sequence of the jump points. For any $\alpha_k \leq \alpha \leq \alpha_{k+1}$, $T(\alpha) = T(\alpha_k) = T_k$. Finally, a sequence of minimal cost-complexity trees $\{T_k\}$ are generated. The correct size tree T^*can be selected by test sample estimates,

$$R^{ts}(T^*) = \min_k R^{ts}(T_k) \tag{3.16}$$

where,

$R^{ts}(.)$ denotes misclassification rate for test sample.

3.3. Combining Classifiers

The majority voting rule (27) was used as the combination rule to combine both the Bayes classifier and decision tree methods, owing to its robust performance. Let L denote the number of classifiers; the majority voting rule is stated as follows:

- If L is odd, the unknown pattern is classified to a class when at least $\frac{L+1}{2}$ of classifiers agree on the class label.
- If L is even, the unknown pattern is classified to a class when at least $\frac{L}{2} + 1$ of classifiers agree on the class label.

In this report, a vehicle will merge (class) only if both the Bayes classifier and decision tree agree on the decision to merge (same class label). Thus, the combined classifier is more conservative than either of the individual classifiers from which it was constructed. This is a valuable attribute for safety applications such as the lane change assistance system, where non-merge decisions are more critical and erring on the conservative side is safer.

Chapter 4 Results

4.1. Bayes Classifier

In developing the Bayes classifier, weights were first estimated using SVMs. The estimated weights shown in table 4.1 reveal that ΔV_{lead} had the largest weight, indicating ΔV_{lead} was the most relevant feature in classifying merge and non-merge events. Thus, a slight change in ΔV_{lead} may greatly change the merging distance. Speed differences ΔV_{lead} and ΔV_{lag} were more relevant than lead gap D_{lead} and lag gap D_{lag}. The distance from the beginning of the merge (auxiliary) lane, S, turned out to be the least relevant feature.

Table 4.1 Weight given by SVMs

Variables	Weights
ΔV_{lead}	0.7243
ΔV_{lag}	0.2449
D_{lead}	0.0292
D_{lag}	0.0012
S	0.0005

A Bayes classifier was developed from $k = 3$ and $\frac{\lambda_{21}}{\lambda_{12}} = 1$. The model's predictive accuracy in terms of validation data for merge and non-merge events is shown in table 4.2. The predictive accuracy for validation data was similar to the accuracy for training data (82% non-merge and 90% merge), indicating that the model did not overfit the training data. The performance in terms of test data is also shown in table 4.2. The accuracy of merge events was high (92.3%), but it was only 79.3% for the critical non-merge events.

Table 4.2 Accuracy of Bayes classifier for validation and test data

Decision	Validation data		Test data	
	Observations	**Accuracy**	**Observations**	**Accuracy**
Non-merge	73	82.2%	459	79.5%
Merge	56	91.1%	208	92.3%

4.2. Decision Tree Model

A decision tree with 62 terminal nodes was constructed using training data before pruning. After applying the pruning rules, a sequence of 16 minimal cost-complexity trees was generated. The total numbers of terminal nodes $|\widetilde{T_k}|$ are shown in table 4.3.

Table 4.3 Number of terminal nodes in minimal cost-complexity trees

Tree	$\boldsymbol{|\widetilde{T_k}|}$
T_1	62
T_2	58
T_3	53
T_4	46
T_5	29
T_6	27
T_7	22
T_8	18
T_9	12
T_{10}	10
T_{11}	9
T_{12}	7
T_{13}	5
T_{14}	3
T_{15}	2
T_{16}	1

The relationship between total number of terminal nodes $|\widetilde{T_k}|$ and estimated misclassification rate for both the training and testing data is presented in figure 4.1. As shown in figure 4.1, the estimated misclassification rate for training data $R(T_k)$ decreased sharply as the tree initially increased in size, then decreased slowly. The estimated misclassification rate for testing data $R^{ts}(T_k)$ also initially decreased sharply, but after reaching its minimum value at 18 terminal nodes, the rate began to climb as the tree size grew. Thus, the tree T_8 with 18 terminal nodes was selected as the correct size decision tree model for predicting merge and non-merge events.

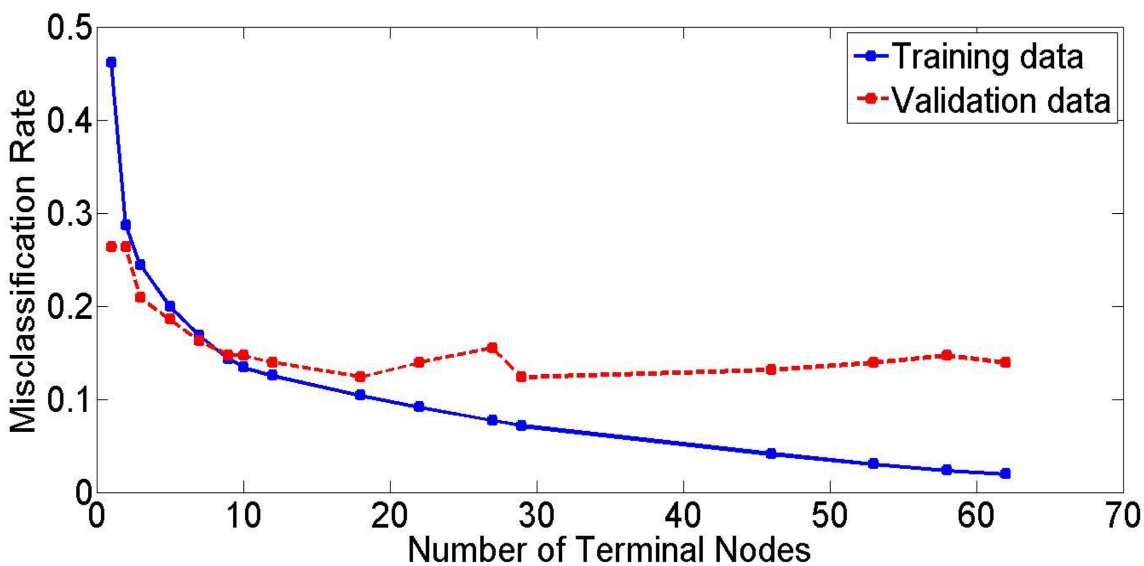


Figure 4.1 Relationship between total number of terminal nodes and misclassification rate

The tree structure is presented in figure 4.2, where terminal nodes are represented by shaded squares and decision nodes are represented by circles. The number of observations, class labels, and predictive accuracies for the terminal nodes are displayed beneath the terminal nodes. Node 1 was first split using the relative speed between the lead and merging vehicles, ΔV_{lead}.

This result further supports the finding from the Bayes classification model that ΔV_{lead} was the most relevant driver feature in making merging decisions. The decision making process of the decision tree model is intuitive. For example, as shown by terminal node t_8, a driver merged if the merging vehicle was slower ($\Delta V_{lead} \geq 0\ m/s$) or slightly faster ($0 > \Delta V_{lead} \geq -2.7\ m/s$) than the lead vehicle and both the lead and lag gap were large ($D_{lag} \geq 2.4\ m$, $D_{lead} \geq 7.6\ m$). In contrast, terminal node t_7 can be interpreted in natural language in the following manner: if the merging vehicle was much faster ($\Delta V_{lead} < -2.7\ m/s$) than the lead vehicle and the lead gap was small ($D_{lead} < 8.9\ m$), then the driver did not merge. For terminal node t_{14}, if the merging vehicle speed was much greater ($\Delta V_{lead} \geq -2.7\ m/s$) than the lead vehicle, if the lead gap was large ($D_{lead} \geq 8.9\ m$), the distance from the beginning of the merge lane was far ($S \geq 138.7\ m$), and the even lag gap was not too large ($D_{lag} \geq 0.76\ m$), then the driver decided to merge, since driver merge behavior become more aggressive upon approaching the end of the merge lane. These rules generated by the decision tree are representative of everyday driving experiences. The predictive results of the decision tree are presented in table 4.4. Again, the predictive accuracy for the validation data was close to that of the training data (87% non-merge and 93% merge), indicating that the model did not overfit the training data. The performance of the test data is also shown in table 4.4. The accuracy for both merge and non-merge events was above 80%. However, a higher accuracy for non-merge events is desirable for the lane changing assistance system.

Table 4.4 Accuracy of decision tree for validation and test data

Decision	Validation data		Test data	
	Observations	**Accuracy**	**Observations**	**Accuracy**
Non-merge	73	89.0%	459	84.3%
Merge	56	85.7%	208	80.8%

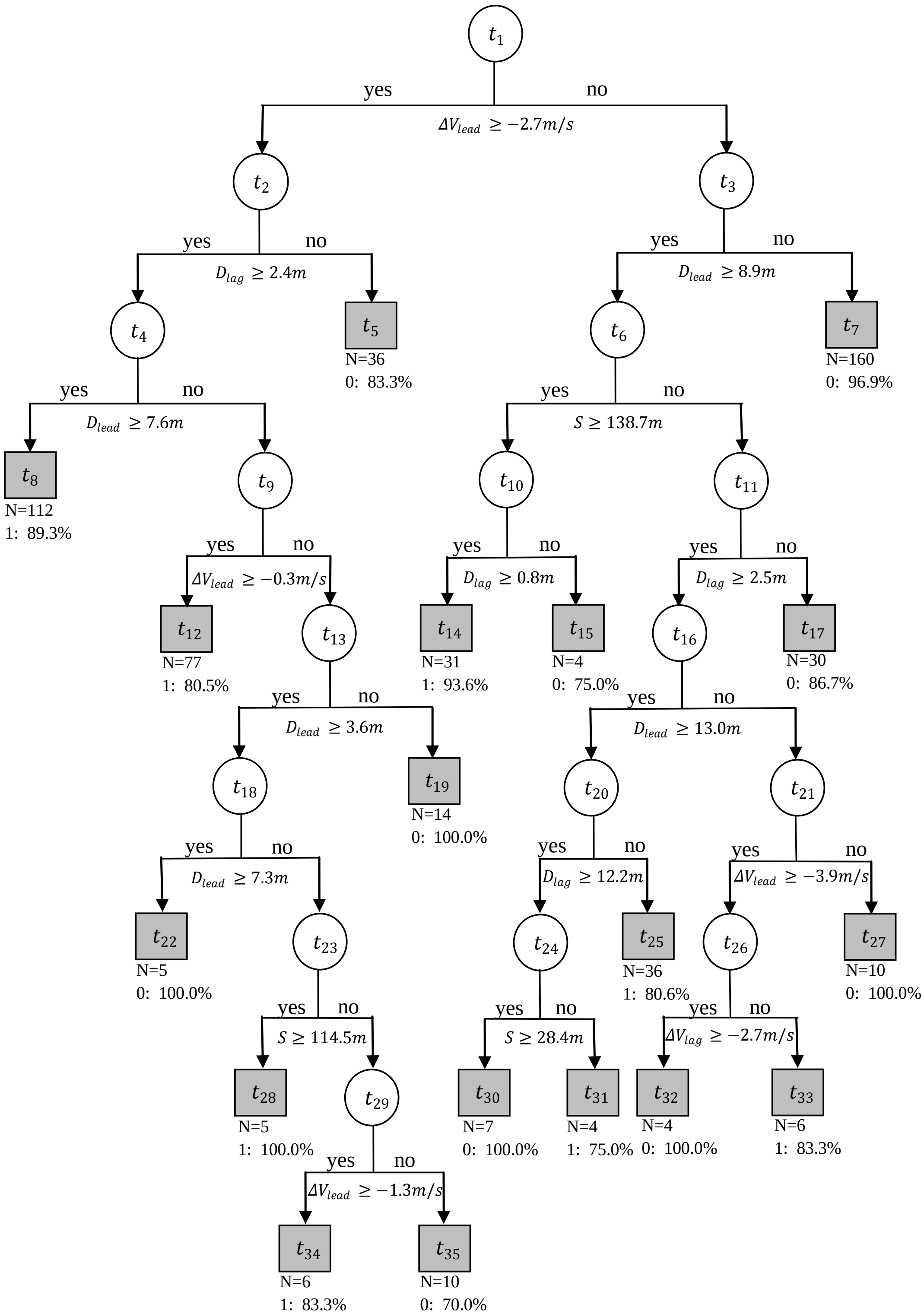


Figure 4.2 Decision tree model structure

4.3. Combining Classifiers

The Bayes classifier and decision tree models were combined using the majority voting rule. The resulting model was tested using the test data, and the results are shown in table 4.5. The accuracy for non-merge events improved to 94.3%, while the accuracy for merge events dropped slightly to 79.3%. As previously discussed, the accuracy of non-merge events is more critical than merge events for safety applications such as the lane change assistance system. Misclassifying a merge event as a non-merge event would result in a lost opportunity to merge, but would not have a negative safety impact. The misclassification parameter $\frac{\lambda_{21}}{\lambda_{12}}$ can be adjusted to give greater weight to the predictive accuracy of non-merge events. To illustrate, the model performance for $\frac{\lambda_{21}}{\lambda_{12}} = 2$ and 5 is shown in table 4.6. The predictive accuracy for non-merge events increased, while the accuracy for merge events decreased as the model became more conservative.

Table 4.5 Accuracy of combined classifier for test data

Decision	Observations	Accuracy
Non-merge	459	94.3%
Merge	208	79.3%

Table 4.6 Sensitivity of combined classifier to misclassification weights

Decision	Observations	$\frac{\lambda_{21}}{\lambda_{12}} = 2$	$\frac{\lambda_{21}}{\lambda_{12}} = 5$
		Accuracy	**Accuracy**
Non-merge	459	95.4%	96.7%
Merge	208	73.6%	49.5%

4.4. Performance of Other Models

Two models from the literature, the genetic fuzzy model and the binary logit model, were also evaluated for comparison. They were estimated using the same dataset and the same set of variables. The coefficients of the binary logit model are presented in table 4.7. For the genetic fuzzy system, a total of 120 rules were generated from the training data. The performance of the two models relative to the test data is shown in table 4.8. Both models performed poorly in comparison to the classifier models developed in this study. The low accuracy for non-merge events is a concern relative to real-time lane changing assistance systems. However, it is noted that the estimated logit model was a binary logit model and was based on existing research. In the future, advanced discrete choice models could be developed to increase predictive accuracy.

Table 4.7 Coefficients of binary logit model

Variable	**Coefficient**	***p*-value**
ΔV_{lead}(m/s)	0.163	<.0001
ΔV_{lag}(m/s)	0.070	0.0043
D_{lead}(m)	0.061	<.0001
D_{lag} (m)	0.003	0.4721
S(m)	-0.004	<.0001
Intercept	1.967	<.0001

* Not significant at 0.05 significance level

Table 4.8 Predicted results for genetic fuzzy and binary logit models

Decision	**Observed**	**Accuracy**	
		Genetic Fuzzy	**Binary Logit**
Non-merge	459	73.6%	20.9%
Merge	208	71.6%	95.7%

Chapter 5 Conclusions

In this report, a lane changing assistance system for mandatory lane changes at lane drops was developed using the Bayes classifier and decision tree methods. The publicly available NGSIM vehicle trajectory dataset was used for model development and testing. The NGSIM dataset consisted of traffic conditions approaching congestion and congested conditions. The model employed factors such as vehicle speeds relative to lead and lag vehicles in the target lane, lead and lag gap distances, and the distance from the beginning of the merge lane. Previous research focused on the development of models for use in microscopic simulation, whereas the current study focused on the design of a lane changing assistance system. One main difference between simulation and lane change assistance system applications is a discrepancy in the relative importance of misclassification between merge and non-merge decisions. In a simulation model, the effect of a non-merge event misclassified as a merge event affects only the mobility measures. The same misclassification in a lane change assistance system, however, would affect safety and the likelihood of a traffic crash.

The combined classifier that used both the Bayes classifier and decision tree models generated high predictive accuracy for critical non-merge events. The cost of misclassification was a surrogate for driver conservativeness. The greater the cost, the more conservative or less aggressive a driver was in pursuing the gap to change lanes. By assigning values of 1, 2, and 5 to the cost of misclassification, the classifier produced accuracies of 94.3%, 95.4% and 96.7% for non-merge events, and 79.3%, 73.6% and 49.5% for merge events. As the cost of misclassification increased, the accuracy for non-merge events also increased, but the accuracy for merge events decreased. Although this report illustrated the performance of two additional models from the literature, the genetic fuzzy system and binary logit, these models, as proposed in the literature, were targeted at microscopic simulation.

References

1. Herman, R. and G. H. Weiss. 1961. "Comments on the highway crossing problem." *Operation Research 9*, 828-840.

2. Drew, D. R., L. R. LaMotte, J. H. Buhr, and J. A. Wattleworth. 1967. "Gap acceptance in the freeway merging process." *Texas Transportation Institute,* 430-2.

3. Miller, A. J. 1972. "Nine estimators of gap acceptance parameters." In *proceedings of the 5th International Symposium on the Theory of Traffic Flow and Transportation*, 215-235.

4. Daganzo, C. F. 1981. "Estimation of gap acceptance parameters within and across the population from direct roadside observation." *Transportation Research Part B, 15B*, 1-15.

5. Gipps, P. G. 1986. "A model for the structure of lane changing decisions." *Transportation Research 20B*(*5*), 403-414.

6. Kita, H. 1993. "Effect of merging lane length on the merging behavior at expressway on-ramps." *Transportation and Traffic Theory*, 37-51.

7. Yang, Q. and H. N. Koutsopoulos. 1996. "A microscopic traffic simulator for evaluation of dynamic traffic management systems." *Transportation Research Part C4*, 113-129.

8. Ahmed, K. I., M. E. Ben-Akiva, H. N. Koutsopoulos, and R. G. Mishalani. 1996. "Models of freeway lane changing and gap acceptance behavior." *Transportation and Traffic Theory*, 501-515.

9. Kita, H. 1999. "A merging-giveway interaction model of cars in a merging section: a game theoretic analysis." *Transportation Research Part 33 A*, 305-312.

10. Hidas, P. 2005. "Modeling vehicle interactions in microscopic simulation of merging and weaving." *Transportation Research Part C 13*, 37-62.

11. Toledo, T., H. N. Koutsopoulos, and M. Ben-Akiva. 2007. "Integrated driving behavior modeling." *Transportation Research Part C 15*, 96-112.

12. Meng, Q. and J. Weng. 2012. "A classification and regression tree approach for predicting drivers' merging behavior in short-term work zone merging areas." *Journal of Transportation Engineering, ASCE,* ISSN 0733-947X.

13. Hou, Y., P. Edara, and C. Sun. 2012. "A Genetic Fuzzy System for Modeling Mandatory Lane Changing." In *Proceedings of the 15th International IEEE Conference on Intelligent Transportation Systems*.

14. Federal Highway Administration. 2011. Next Generation Simulation Fact Sheet. ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm.

15. Choudhury, C., V. Ramanujam, and M. Ben-Akiva. 2009. "Modeling acceleration decisions for freeway merges." *Transportation Research Record,* 2124, 45-57.

16. Yeo, H., A. Skabardonis, J. Halkias, J. Colyar, and V. Alexiadis. 2008. "Oversaturated freeway flow algorithm for use in next generation simulation." *Transportation Research Record,* 2088, 68-79.

17. Cambridge Systematics, Inc. 2005. "NGSIM U.S. 101 Data Analysis Summery Report." Washington, D.C., Federal Highway Administration.

18. Punzo, V., M. T. Borzacchiello, and B. Ciuffo. 2009. "Estimation of vehicle trajectories from observed discrete positions and next-generation simulation program (NGSIM) data." *Proceedings of the 88th Annual Meeting of the Transportation Research Board*.

19. Kesting, A. and M. Treiber. 2008. "Calibrating car-following models using trajectory data: a methodological study." *Proceedings of the 87th Annual Meeting of the Transportation Research Board*.

20. Duret, A., C. Buisson, and N. Chiabaut. 2008. "Estimating individual speed-spacing relationship and assessing the Newell's car-following model ability to reproduce trajectories." *Proceedings of the 87th Annual Meeting of the Transportation Research Board*.

21. Ossen, S. and S. P. Hoogendoorn. 2008. "Validity of trajectory-based calibration approach of car-following models in the presence of measurement errors." Proceedings of the 87th Annual Meeting of the Transportation Research Board.

22. Punzo, V., D. J. Formisano, and V. Torrieri. 2005. "Nonstationary kalman filter for estimation of accurate and consistent car-following data." *Transportation Research Record,* 1934, 3-12.

23. Ma, X. and I. Andreasson. 2007. "Statistical analysis of driver behavior data in different regimes of the car-following stage." *Transportation Research Record,* 2018, 87-96.

24. Hastie, T., R. Tibshirani, and J. Friedman. 2001. "The Elements of Statistical Learning. 2nd edition." Springer.

25. Martin, T., P. Harten, P., D. Young, E. Muratov, A. Golbraikh, H. Zhu, and A. Tropsha. 2012. "Does rational selection of training and test sets improve the outcome of QSAR modeling?" *Journal of Chemical Information and Modeling*, *52*(10), 2570-2578.

26. Edara, P., D. Teodorovic, and H. Baik. 2007. "Using Neural Networks to Model Intercity Mode Choice." *Intelligent Engineering Systems through Artificial Neural Networks*, *17*, 143-148. New York: ASME Press.

27. Theodoridis, S. and K. Koutroumbas. 2006. *Pattern Recognition.* 3rd ed. Academic Press.

28. Buturovic, L. J. 1993. "Improving k-nearest neighbor density and error estimates." *Pattern Recognition*, 26(4), 611-616.

29. Maunder, C. R. F. 1997. *Algebraic Topology*. New York: Dover.

30. Domeniconi, C., D. Gunopoulos, and J. Peng. 2005. "Large margin nearest neighbor classifiers." *IEEE Transactions on Neural Networks, 16*(4), 899-909.

31. Russel, S. and P. Norvig. 2003. *Artificial Intelligence A Modern Approach.* 2nd edition. New Jersey: Prentice Hall, 653-663.

32. Shannon, C. and W. Weaver. 1949. *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press.

33. Breiman, L., J. Friedman, R. Olshen, and C. Stone. 1984. *Classification and Regression Tree.* Belmont, CA: Wadsworth International, 66-75.