



COMPARATIVE EVALUATION OF FUZZY INFERENCE SYSTEM, SUPPORT VECTOR MACHINE AND MULTILAYER FEED-FORWARD NEURAL NETWORK IN MAKING DISCRETIONARY LANE CHANGING DECISIONS

E. Balal, R.L. Cheu**

Abstract: This paper compares Fuzzy Inference System (FIS), Support Vector Machine (SVM) and MultiLayer Feed-forward neural network (MLF) in modeling a driver's decision when making a discretionary lane changing move on a freeway. The FIS model has been developed and published in an earlier work by the authors, whereas the SVM and MLF models are newly developed in this research. The FIS, SVM and MLF models use the same four inputs: the gap between the subject vehicle and the leading vehicle in the original lane, the gap between the subject vehicle and the leading vehicle in the destination lane, the gap between the subject vehicle and the trailing vehicle in the destination lane, and the distance between the preceding and trailing vehicles in the destination lane. The models give a binary decision of "no, stay in the same lane" or "yes, move to the destination lane now". These models were trained and then tested with the Next Generation SIMulation (NGSIM) vehicle trajectory data. The results have shown that the FIS has the highest accuracies in making correct lane changing decisions. It recommends "yes, move to the destination lane now" with 82.2% accuracy, and "no, stay in the same lane" with 99.5% accuracy. The SVM model also outperformed the traditional gap acceptance model which was used as the benchmark. However, the MLF model was not as accurate as the gap acceptance model.

Key words: *fuzzy inference system, multilayer feed-forward neural network, support vector machine, lane change, gap acceptance*

Received: February 20, 2018

DOI: 10.14311/NNW.2018.28.021

Revised and accepted: August 13, 2018

1. Introduction

Lane change is one of the most frequently executed actions when driving on freeways. A driver performs his/her lane changing actions by first assessing the position

*Esmail Balal; Ruey Long Cheu – Corresponding author; Dept. of Civil Engineering, The University of Texas at El Paso, 500 W. University Ave., El Paso, TX 79968, Email: ebalal@utep.edu, rcheu@utep.edu

of his/her vehicle (the subject vehicle) relative to the surrounding vehicles, determine if it is safe to move, followed by steering and accelerating/decelerating the subject vehicle from the original lane to the destination lane.

Lane changes in freeways may be categorized into either mandatory or discretionary. A driver makes a Mandatory Lane Change (MLC) when he/she is trying to move his/her vehicle to the correct lane in anticipation of the next left or right-turn, or to avoid a lane closure immediately downstream. The driver is said to execute a Discretionary Lane Change (DLC) if he/she desires a faster speed, greater following distance, further line of sight, better ride quality, etc. in the destination lane [4, 28]. This models tested in this paper deals only with DLC.

A lane change may be modeled as a four-step process: (1) motivation; (2) selection of destination lane; (3) seeking for opportunity to initiate a move the from the original lane to the destination lane; and (4) steer the vehicle from the original lane to the destination lane. The four steps are separated by times t_1 , t_2 , t_3 , t_4 and t_5 , respectively. At t_1 , the driver of the subject vehicle S (simply known as the subject driver) begins to feel the need to move away from the original lane. The subject driver starts looking for a destination lane (either the left lane or the right lane of the original lane) at t_2 . Between t_2 and t_3 , the driver decides which destination lane he/she should move to. Actively, from t_3 onwards, the driver waits for an opportunity to move S to the destination lane. The driver starts to steer the subject vehicle towards the destination lane at t_4 . The lane changing move is completed at t_5 . Three models, namely Fuzzy Inference Systems (FIS), Support Vector Machines (SVM) and MultiLayer Feedforward neural network (MLF) models have been developed to replicate human decisions between t_3 and t_4 . The models answer the question “Is it time to start moving to the destination lane?” The answer is either “no, stay in the same lane” or “yes, move to the destination lane now”.

In traffic flow theory, this lane changing decision problem has long been analyzed as a gap acceptance problem [4]. In fact, this problem may also be viewed as a binary pattern classification problem. That is, a pattern classifier classifies input data (the relative positions between the S and the surrounding vehicles) into one of the two classes: (“no, stay in the same lane” or “yes, move to the destination lane now”).

The objective of this paper is to compare the performances of four lane changing decision models: FIS, SVM, MLF, and Gap Acceptance Model (GAM) [4, 7]. The FIS and GAM have been developed in earlier research [2, 7] while the SVM and MLF models will be developed in present work. The model development and comparative evaluation will be performed using the vehicle trajectory data from the well-known Next Generation SIMulation (NGSIM) database [5, 6].

This paper is organized as follows. After this introduction, a brief review of lane changing models is provided. The driving scenarios in which the lane changing decision models are developed and applied are specified. The next section presents the FIS, SVM, MLF and GAM approaches. This is followed by the description of the data used and the model development processes. The results of the applications of the FIS, SVM, MLF and GAM to a test data set are next presented, compared and discussed, before making a conclusion.

2. Review of lane changing models

The lane changing models has been reviewed extensively by [15,28]. Readers may also find the authors' review reports in [1,2]. The purpose of literature review, in the context of this paper, is to select the best model as the benchmark of comparative evaluation. Based on the literature review, the authors have identified several issues when selecting the benchmarking model:

1. Types of lane change: Although most of the researchers classified lane changes into MLC and DLC, some researchers did not make this distinction, or classified lane changes into more categories.
2. Lane changing process: Some authors do not discretize a lane changing event into the sequential four-step process.
3. Input variables: Many variables have been used as inputs to the different lane changing models. Only [2] conducted a stated preference survey on drivers to find out what decision variables they always used in making lane changing moves.
4. Model replication: The authors of many lane changing studies did not report the drivers' decision logic in sufficient detail, or provide the numerical values of the model's internal parameters. This has made the authors impossible to implement such models with a new set of data for comparative evaluation.

The authors have found that only the GAM built into the TRANSLMODLER microscopic traffic simulation tool [4] met the requirement of and was implementable in this research. The developer of the GAM has described the model's logic and parameters in [4] and [7]. Therefore, GAM is adopted as the benchmark.

3. Lane changing setting

Fig. 1 illustrates a typical lane changing scenario involving up to five vehicles. The subject vehicle (S) is moving one lane to the right, from the original lane to the destination lane. Vehicles LB, TB, LA and TA denote the leading vehicle before lane change, trailing vehicle before lane change, leading vehicle after lane change and trailing vehicle after lane change, respectively.

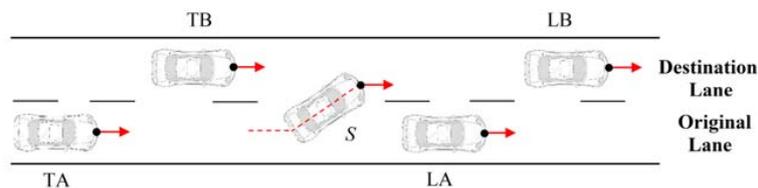


Fig. 1 Lane changing scenario.

Based on the outcome of a questionnaire (stated preference) survey on drivers, [2] have identified four variables that were used most frequently by drivers in making

decisions on discretionary lane changing moves on freeways. These four variables are:

- G_{LB} = Gap between vehicle S and vehicle LB (m),
- G_{LA} = Gap between vehicle S and vehicle LA (m),
- G_{TA} = Gap between vehicle S and vehicle TA (m),
- D = Distance between vehicle LA and vehicle TA (m).

The idea behind the lane changing decision models implemented in this paper is derived from the “blind spot monitoring/warning system” which has been used in many new vehicles. Such “passive” system relies on sensors instrumented at the perimeters of the subject vehicle S to detect any vehicle present in the subject driver’s blind spots, and warn the subject driver of the existence of this vehicle. An improved and “active” version, which may be called “lane changing advisory system”, uses sensors instrumented in the subject vehicle alone to measure G_{LB} , G_{LA} , G_{TA} and D as inputs. The proposed system is a stand-alone system and is technically deployable immediately. The new system will work as follows:

1. The system will be activated once the subject driver activates the turn indicator;
2. The sensors start to continuously measure and calculate the values of G_{LB} , G_{LA} , G_{TA} and D ;
3. The lane changing decision model uses the four inputs to recommend “yes, move to the destination lane now” or “no, stay in the same lane”;
4. The recommendation is then communicated visually (by an icon in the instrument panel) or acoustically (by voice or beeping sound) to the subject driver;
5. As soon as the subject driver switches off the turn indicator (after the driver has completed or abandoned the lane change), the system will stop functioning

The system may also be programmed to function as part of autonomous vehicles.

4. Modeling methodologies

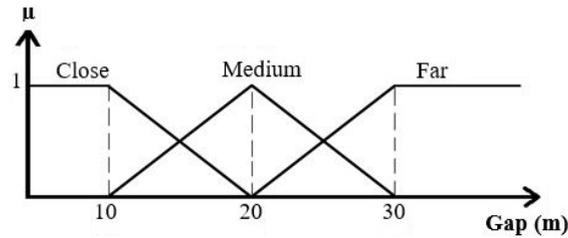
Fuzzy logic, SVM and MLF have been applied to transportation engineering problems, see for examples [24, 25] and [12]. This section reviews these three methodologies in the context of making lane changing decisions.

4.1 Fuzzy Inference System

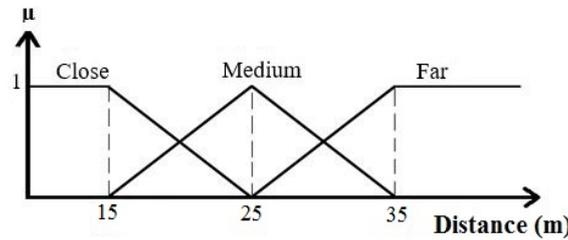
Fuzzy logic was first presented by [27] as a concept to manipulate imprecise information. Fuzzy logic has been applied to solve a wide range of engineering problems associated with estimation, control, pattern recognition and decision making with uncertain data. Fuzzy Inference System (FIS) applies the fuzzy logic/reasoning concept to model the human decision making process in four sequential stages: (i) fuzzification; (ii) inference; (iii) composition; and (iv) defuzzification [10]. The

fuzzification stage uses fuzzy sets and fuzzy membership functions to map crisp inputs into fuzzy membership values. The inference stage feeds the fuzzified inputs to IF-THEN fuzzy rules to generate fuzzified outputs. The outputs of multiple rules are combined into one or a few fuzzified outputs in the composition stage. Finally, the defuzzification stage converts the fuzzified outputs back to crisp values. Variants of FIS have been applied to model different aspects of lane changing process [3, 8, 16], but with lesser degree of success compared to [2].

A fuzzy set is a set of several linguistic values that a variable may take on. For example, we may use $\tilde{G}_{TA} = (\text{Close}, \text{Medium}, \text{Far})$ to denote that variable G_{TA} may take on linguistic value of close, medium or far. We say that $\tilde{G}_{TA} = (\text{Close}, \text{Medium}, \text{Far})$ is a fuzzy set of three (linguistic values). Similarly, a fuzzy set of three for D may be $\tilde{D} = (\text{Close}, \text{Medium}, \text{Far})$ while a fuzzy set of two for C (which denotes change lane) may be $\tilde{C} = (\text{No}, \text{Yes})$. Each linguistic value in a fuzzy set has its unique fuzzy membership function. The purpose of these functions is to convert crisp input data values into fuzzy membership values (also known as the degrees of membership) between 0 and 1. The fuzzy membership functions for \tilde{G}_{TA} and \tilde{D} may be defined as in Fig. 2(a) and Fig. 2(b) respectively.



(a) Gap



(b) Distance

Fig. 2 Fuzzy membership functions.

Fuzzy inference rules are usually written in the IF-THEN format. If the antecedent of a rule has multiple fuzzified variables, the fuzzified variables may be combined with the “AND” and/or “OR” operator. A simple example is

$$\text{IF } [(\tilde{G}_{FA} \text{ is Close}) \text{ AND } (\tilde{D} \text{ is Close})] \text{ THEN } (\tilde{C} \text{ is No})$$

The above rule combines fuzzified inputs of G_{FA} and D to infer a fuzzified output of C . Internally, the logical operator AND combines the membership values of $(\tilde{G}_{FA} \text{ is Close})$ and $(\tilde{D} \text{ is Close})$, and assign a membership value to $(\tilde{C} \text{ is No})$.

In a FIS, there can be many rules. The number of rules depends on the number of fuzzified variables in the antecedent of the rules, and the possible combinations of linguistic values of these variables. In the above example, the two antecedent variables each has a fuzzy set of three, i.e., $\tilde{G}_{TA} = (\text{Close}, \text{Medium}, \text{Far})$ and $\tilde{D} = (\text{Close}, \text{Medium}, \text{Far})$. Therefore, there are nine possible combinations of the two fuzzified inputs, resulting in nine rules. The collection of rules in a FIS is called “rule base”. In the composition stage, the fuzzified outputs of all the rules are combined, so that each linguistic value of the fuzzified output variable has a single membership value. Finally, the defuzzification stage combines multiple memberships of the different linguistic values, and “defuzzified” into a single, crisp number. More descriptions of FIS operations and its variants can be found in [10].

4.2 Support Vector Machine

Support Vector Machine (SVM) is a binary pattern classifier based on the statistical learning technique. Since its introduction by [22], SVM has been used in several transportation applications [11], beginning with [26].

A binary SVM classifier places an input vector into one of two classes, using a decision boundary that has been determined to minimize the classification errors. For data that can be separated into two distinct classes by a plane in the SVM’s dimensional space, the SVM’s learning algorithm calculates an optimal plane that separates the two classes of training vectors by the maximum margin (see Fig. 3). Consider the problem of lane changing decision where X_i is the i -th input vector with n dimension $\mathbf{X}_i = \{x_{i1}, \dots, x_{in}\}$ (in our case $n = 4$, for four input variables). The SVM’s operates by using a vector $\mathbf{W} = \{w_1, \dots, w_n\}$ and a scalar b , such that

$$f(\mathbf{X}_i) = \text{sign}(\mathbf{W} \cdot \mathbf{X}_i + b).$$

The lane changing advisory system may use positive $f(\mathbf{X}_i)$ value to represent “yes, move to the destination lane now” while negative $f(\mathbf{X}_i)$ value as “no, stay in the

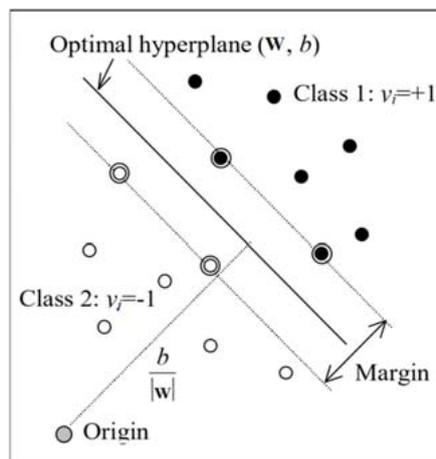


Fig. 3 Two-dimensional SVM binary classification.

same lane". As shown in Fig. 3, the optimal hyperplane that separates positive and negative values of $f(\mathbf{X}_i)$ is defined by the linear equation

$$\mathbf{W} \cdot \mathbf{X}_i + b = 0. \quad (1)$$

The objective of SVM learning, which is supervised, is to obtain \mathbf{W} and b . Consider a set of training data which has l vectors denoted by (\mathbf{X}_i, y_i) , where $i = 1, \dots, l$; $y_i \in \{+1, -1\}$, the hyperplane equation may be rewritten as

$$y_j (\mathbf{W} \cdot \mathbf{X}_i + b) \geq 1.$$

It can be shown that the margin, ρ , defined as the distance between the closest training vectors on both sides of (1), is given by $\rho = \|\mathbf{W}\|/2$ (Schlkoph et al., 2002). The solution that minimizes ρ which has the objective function

$$\max \left[\sum_{j=1}^l \alpha_j - \frac{1}{2} \sum_{k=1}^l \sum_{j=1}^l \alpha_j \alpha_k y_j y_k (\mathbf{X}_j \cdot \mathbf{X}_k) \right]. \quad (2)$$

In the solution, the training vectors that have $\alpha_j > 0$ are termed support vectors. Then,

$$\mathbf{W} = \sum_{j=1}^l \alpha_j \mathbf{X}_j y_j,$$

b can subsequently be obtained. The linear SVM classifier becomes

$$f(\mathbf{X}_i) = \text{sign}(\mathbf{W} \cdot \mathbf{X}_i + b) = \text{sign} \left[\sum_{\forall j, \alpha_j > 0} y_j \alpha_j (\mathbf{X}_i \cdot \mathbf{X}_j) + b \right].$$

For non-linearly separable data, the SVM uses the "kernel method" to map the training data from the original dimensional space $\mathbf{X}_i \in R^n$ into a higher dimensional space $\Phi : R^n \rightarrow H$. An optimal hyperplane can then be drawn in H , to separate the two classes, using the training data in the form of $\Phi(\mathbf{X}_j)$ instead of \mathbf{X}_j . However, to avoid the relatively long computation time of $\Phi(\mathbf{X}_j) \cdot \Phi(\mathbf{X}_k)$ that replaces $(\mathbf{X}_j \cdot \mathbf{X}_k)$ in (2), it is simpler to replace $(\mathbf{X}_j \cdot \mathbf{X}_k)$ with a kernel function $K(\mathbf{X}_j \cdot \mathbf{X}_k)$. With the use of kernel function, the non-linear SVM classifier becomes

$$f(\mathbf{X}_i) = \text{sign} \left[\sum_{\forall j, \alpha_j > 0} y_j \alpha_j K(\mathbf{X}_i \cdot \mathbf{X}_j) + b \right].$$

To achieve the best classification results, the designer of SVM needs to experiment with linear and non-linear SVM and in the case of non-linear SVM the various kernel functions.

4.3 Multilayer Feed-Forward Neural Network

MultiLayer Feed-forward neural network (MLF) is an Artificial Neural Network (ANN) pattern classifier. ANNs are modeled after the physical architecture of the

human brain and have been used in many transportation applications [11]. Among the many types of ANN, the MLF is most frequently used.

The structure of a typical MLF is shown in Fig. 4. This MLF has three layers: namely the input, hidden and output layers. The neurons in the input, hidden and output layers are referred to as the input, hidden and output neurons respectively. Pairs of neurons between two adjacent layers are connected. The n input neurons receive the i -th input vector $\mathbf{X}_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ (in our case $n = 4$). Each hidden neuron, say neuron j , has a weight vector $\mathbf{W}_j = \{w_{1j}, \dots, w_{ij}, \dots, w_{nj}\}$. Each component of \mathbf{W}_j is the weight of the connection between an input neuron i and the hidden neuron j . At hidden neuron j , the dot product of \mathbf{X}_i and \mathbf{W}_j is computed. A bias term θ_j that is related to this hidden neuron j is added to the dot product, resulting in $\mathbf{X}_i \cdot \mathbf{W}_j + \theta_j$. The outcome is then passed through a transfer function to produce an output $f(\mathbf{X}_i \cdot \mathbf{W}_j + \theta_j)$ of hidden neuron j .

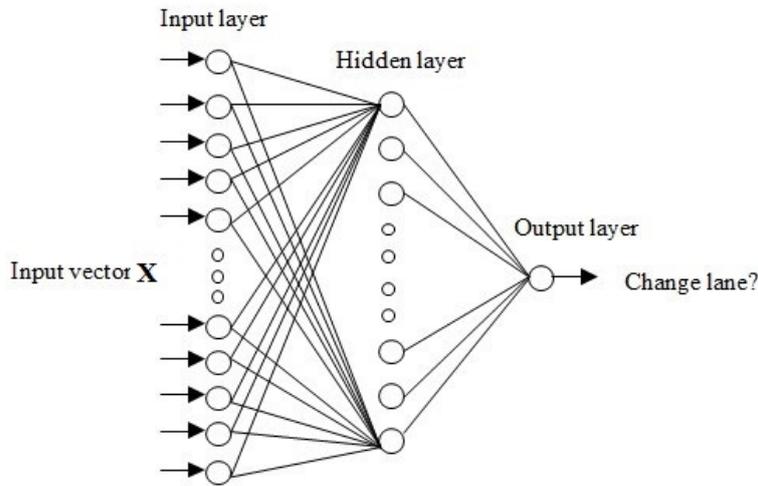


Fig. 4 Architecture of MLF for lane changing decisions.

Two commonly used transfer functions are the sigmoid or logsig function

$$f(\mathbf{X}_i \cdot \mathbf{W}_j + \theta_j) = \frac{1}{1 + \exp[-(\mathbf{X}_i \cdot \mathbf{W}_j + \theta_j)]},$$

and the tansig function

$$f(\mathbf{X}_i \cdot \mathbf{W}_j + \theta_j) = \frac{2}{1 + \exp[-2(\mathbf{X}_i \cdot \mathbf{W}_j + \theta_j)]} - 1.$$

The tansig function is an approximation of tanh function but with improved computational efficiency. When a MLF receives an input vector \mathbf{X}_i , the above operations are performed for all the hidden neurons. Once this is completed, similar operations are repeated using the outputs of the hidden neurons as inputs to the next layer. For the lane changing decision model, the MLF has only one output neuron that produces a value between 0 and +1 (if the logsig transfer function is used),

or between -1 and $+1$ (if the tansig transfer function is used). This continuous output value is then mapped into a binary value indicating the recommendation of “yes, move to the destination lane now” or “no, stay in the same lane”.

The training of MLF is to determine the weights and bias of the hidden neurons and the output neuron, using the training vectors. The common iterative algorithm used in the determination is the backpropagation algorithm [9]. To optimize the classification accuracy, the designers usually train several MLFs, each with a different number of hidden neurons, and pick the MLF that has the highest accuracy.

4.4 Gap Acceptance Model

The Gap Acceptance Model (GAM) is the lane changing decision model in the TRANSMODELER microscopic traffic simulation tool [4]. GAM recommends “no, stay in the same lane” or “yes, move to the destination lane now” by comparing G_{TA} , and G_{LA} against their respective critical gaps, G_{TA}^{\min} and G_{LA}^{\min} [7]. The decision rule is:

IF $[(G_{LA} \geq G_{TA}^{\min}) \text{ AND } (G_{LA} \geq G_{LA}^{\min})]$ THEN (“yes, move to the destination lane now”) ELSE (“no, stay in the same lane”)

The critical gap equations are

$$G_{LA}^{\min} = \exp [b_{0,LA} + b_{1,LA} \cdot \max(0, V_{LA} - V) + b_{2,LA} \cdot \min(0, V - V_{PA}) + b_{3,LA} V_{LA} + \alpha_{LA} v + \varepsilon_{LA}],$$

$$G_{TA}^{\min} = \exp [b_{0,TA} + b_{1,TA} \cdot \max(0, V_{TA} - V) + b_{2,TA} V_{TA} - \alpha_{TA} v + \varepsilon_{TA}]$$

in which V_{TA} and V_{LA} are the speed of vehicles TA and LA, respectively. Subject driver’s inter-driver heterogeneity is accounted for by the ε_{LA} and ε_{TA} terms, which were assumed to follow normal distributions, i.e., $\varepsilon_{LA} \sim N(0, \sigma_{LA}^2)$, $\varepsilon_{TA} \sim N(0, \sigma_{TA}^2)$. The v term, which accounts for intra-diver heterogeneity, follows a truncated standard normal distribution $v \sim N(0, 1)$, $-3 \leq v \leq 3$. The rest of the terms $\{b_{0,LA}, b_{1,LA}, b_{2,LA}, b_{3,LA}, \alpha_{LA}, b_{0,TA}, b_{1,TA}, b_{2,TA}, \alpha_{TA}\}$ are coefficients which were calibrated by the model developer, also with the NGSIM data [4].

5. Data

The well-known NGSIM data were used in this comparative evaluation. The data collected at two freeway sites were used: one for calibrating/training the models and one for testing the models. The characteristics of the data are described in Tab. I. The data, consisted of the x-y coordinates of vehicles at 0.1 second intervals, were processed, and the steps are briefly described as follows:

- Only passenger cars originally travelled in the through lanes that changed lane once were considered as subject vehicle S . This condition is to filter out vehicles that may not make DLCs, and other type of vehicles that may have different lane changing behavior.

- For each identified S , the time t_4 , the onset of a lane changing move, was identified as the instance when the front center of S first reached or exceeded lateral velocity of 0.2 m/s. This lateral velocity criterion was from [23].
- At t_4 , vehicles LB, TB, LA, TA, were identified.
- The four input variables were calculated at 0.5 second increment, before and after t_4 . The procedure of averaging vehicle coordinates and deriving the input variables at 0.5 second intervals followed the steps recommended by [18] and was consistent with [21].
- The processed input variables at the same time interval were assembled to form a vector $\{G_{LB}, G_{LA}, G_{TA}, D\}$. For each S , the Observed Maneuver (OM) was set to $OM = 1$ at t_4 , and $OM = 0$ at all other time intervals. $OM = \{0, 1\}$ were used as the “ground truth” in the “supervised” training of FIS, SVM and MLF, and in the calculation of classification accuracies of FIS, SVM, MLF and GA.

More details of data processing are documented in [1, 2]. Although the NGSIM data is not as comprehensive and high resolution as the data acquired in and in-laboratory diver simulator [17], it was collected in the real driving environment involving a relatively larger sample.

Name of data set	Dataset A	Dataset B
<i>Site, date and time</i>		
Name of highway	I-80 Freeway	U.S. Highway 101
City, state	Emeryville, California	Los Angeles, California
Length of site	1,650 ft (503 m)	2,100 ft (640 m)
Traffic flow direction	northbound	southbound
Upstream interchange	Powell Street	Ventura Boulevard
Downstream interchange	Ashby Street	Cahuenga Boulevard
Total no. of lanes	6	6
Date of video recording	April 13, 2005	June 15, 2005
Time of video recording	4:00-4:15 p.m.	7:50-8:05 a.m.
Reference	(Cambridge, 2005)	(Cambridge, 2005)
<i>No. of subject vehicles</i>		
Discretionary lane change	163	171
No lane change	3,202	2,612
Total	3,365	2,783
<i>No. of vectors in data file</i>		
Discretionary lane change	163	171
No lane change	232,493	209,681
Total	232,656	209,852

Tab. I Characteristics of data used.

6. Model development

The development and calibration of FIS (which has been reported in [2]), the learning of SVMs, and the training of MLFs are described in this section. All these models have four input variables: G_{LB} , G_{LA} , G_{TA} and D and one binary output variable C .

6.1 Fuzzy Inference System

The development of the FIS can be traced back to 2014 when [1] first studied the statistical properties and correlations of 10 discretionary lane changing parameters. Subsequently, [2] used the findings of four of the 10 parameters to design the FIS. [2] presented the development, calibration and test of the FIS. Vechione et al. (2018) performed hypothesis tests that proved that the properties of three out of the four lane changing parameters (used as inputs to the FIS) were not significantly different between discretionary and mandatory lane changes.

MATLAB's Fuzzy Logic Designer App [14] was used to implement the FIS [2]. There were four input variables to the FIS. Each variable had a fuzzy set of three, i.e., (Close, Medium, Far). The FIS has only one output variable C (which denoted "change lane"). It has a fuzzy set of two, i.e., $\tilde{C} = \{\text{Yes}, \text{No}\}$. The membership functions for G_{LB} , G_{LA} and G_{TA} are shown in Fig. 2(a) while the membership functions for D are shown in Fig. 2(b). These membership functions were selected based on engineering judgement, after examining the probability distributions of the parameters [2]. The antecedent of each rule consisted of the fuzzified values of the four input variables, while the output of each rule was the fuzzified value of \tilde{C} . For examples:

IF [(\tilde{G}_{TA} is Close) AND (\tilde{G}_{LA} is Close) AND (\tilde{D} is Close) AND (\tilde{G}_{LB} is Close)]
THEN (\tilde{C} is No)

IF [(\tilde{G}_{TA} is Close) AND (\tilde{G}_{LA} is Far) AND (\tilde{D} is Far) AND (\tilde{G}_{LB} is Close)]
THEN (\tilde{C} is Yes)

The rule base had up to 81 rules. However, certain combinations of linguistic values were infeasible. After removing the 30 rules which had infeasible combinations of linguistic values, the rule base was left with 51 rules. The Mamdani max-min method [19] was adopted to infer a fuzzified output value among the 51 rules. This composition method is described as follows. First, for each rule, the four membership values in the antecedent were obtained from the respective membership functions. Then, the minimum of the four membership values in the antecedent was taken as the output C of that rule. Finally, the maximum C value among the 51 rules was assigned to C^* .

Next, the defuzzification stage converted $C^* \in [0, 1]$ to either "yes, move to the destination lane now" or "no, stay in the same lane" by comparing C^* against a threshold value τ , to come out with a FIS's Recommendation, denoted by FR :s 9/4/2018

$$FR = \begin{cases} 1 \text{ for "yes, change lane"} & \text{if } C^* \geq \tau \\ 0 \text{ for "no, do not change lane"} & \text{if } C^* < \tau \end{cases} .$$

The fuzzy rules and the process of calibrating $\tau = 0.5$ with Dataset A has been described in [2].

6.2 Support Vector Machine

Several SVMs were implemented in MATLAB's Pattern Classification Learner App. The following six SVMs were trained using Dataset A. The differences between these SVMs lie in their kernel functions. The σ values in the Gaussian kernel functions have been built into the MATLAB's Pattern Classification Learner App. The default values were used for the SVMs with Gaussian kernel.

Linear SVM:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \mathbf{X}_i \cdot \mathbf{X}_j.$$

Quadratic SVM:

$$K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^2.$$

Cubic SVM:

$$K(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i \cdot \mathbf{X}_j + 1)^3.$$

Course Gaussian SVM:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \exp \left[- \left(\frac{\|\mathbf{X}_i - \mathbf{X}_j\|}{2\sigma^2} \right) \right] \mathbf{X}_i \cdot \mathbf{X}_j \text{ with } \sigma = \frac{\sqrt{2}}{4}.$$

Medium Gaussian SVM:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \exp \left[- \left(\frac{\|\mathbf{X}_i - \mathbf{X}_j\|}{2\sigma^2} \right) \right] \mathbf{X}_i \cdot \mathbf{X}_j \text{ with } \sigma = \sqrt{2}.$$

Fine Gaussian SVM:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \exp \left[- \left(\frac{\|\mathbf{X}_i - \mathbf{X}_j\|}{2\sigma^2} \right) \right] \mathbf{X}_i \cdot \mathbf{X}_j \text{ with } \sigma = 4\sqrt{2}.$$

The training of these SVMs using MATLAB's Pattern Classification Learner App was straight forward. The program automatically gives a "confusion matrix" similar to the classification matrix in [16], when a trained SVM is applied to a data set.

6.3 Multilayer Feed-Forward Neural Network

Eleven MLFs were trained with MATLAB's Neural Net Pattern Recognition App [14]. Because Dataset A has many more vectors with OM = 0 compared to OM = 1, the vectors with OM = 1 were replicated multiple times such that both classes in the MLF training data set had approximately the same number of vectors. The 11 MLFs trained each has a different number of hidden neurons, from 5 to 15. Each MLF was trained with the Levenberg-Marquardt algorithm for up to 1000 iterations with mean-squared-error set as the stopping criterion. That is, training was stopped when the mean-squared-error started to increase after reaching a minimum value. For each MLF, different combinations of logsig and tansig transfer functions were

used for the hidden layer and the output layer. Note that, when the logsig transfer function was used for the output neuron, the MLF output was in the range of $C = (0, 1)$. Therefore, the desired outputs in the training vectors were set to 0 and 1 for $OM = 0$ and $OM = 1$ respectively. Similarly, during classification, output values of less than 0.5 are mapped into “no, stay in the same lane”, while values of at least 0.5 are mapped into “yes, move to the destination lane now”. On the other hand, when the tansig transfer function was used for the output neuron, the MLF’s output was in the range of $C = (-1, 1)$, and the desired outputs during training were set to $(-1, 1)$. It followed that negative MLF’s classification outputs were mapped into “no, stay in the same lane”, while positive classification outputs were mapped into “yes, move to the destination lane now”. It was found that, for all the MLFs with a fixed number of hidden neurons, using the tansig transfer function in the hidden neurons and the logsig transfer function in the output neuron produced the highest classification accuracy. Therefore, the test results of MLFs with tansig transfer function in the hidden neurons and the logsig transfer function in the output neuron were reported subsequently.

7. Results

This section presents the results of the applications of the four lane changing decision models when applied to Dataset B. The FIS and SVM were calibrated or trained with Dataset A, while the MLFs were trained with the modified Dataset A (with increased number of $OM = 1$ vectors). The GAM was calibrated earlier with the NGSIM data as reported in [4]. The results of each model, when tested with the vectors in Dataset B, are presented in the form of classification matrix [16]. Because there are relatively many more vectors in Dataset B with $OM = 0$ compared to $OM = 1$, the decision accuracy for these two classes of data were computed separately. Tab. II(a) shows the classification matrix obtained from FIS with $\tau = 0.5$. The FIS achieves accuracies of 82.5% for $OM = 1$ and 5% for $OM = 0$.

Fig. 5(a) plots the rates of correct decision of the SVM with the six different kernel functions. Bars of $OM = 1$ and $OM = 0$, along with the percent correct decisions are plotted for each kernel function. None of the kernel functions has the highest rates of correct decisions for both $OM = 1$ and $OM = 0$. While the cubic polynomial kernel function has the highest accuracy for $OM = 1$, the fine Gaussian kernel function has the highest accuracy for $OM = 0$. There are two types of errors in the classification matrix. Type I error occurs when the SVM’s Recommendation (SR) is “yes, move to the destination lane now” but there was no observed maneuver in Dataset B, i.e., $SR = 1|OM = 0$. If the subject driver or vehicle follows the SVM’s recommendation and move into the destination lane, he/she may create an unsafe maneuver or even a collision. Type II error occurs when the SVM recommends “no, stay in the same lane” but in Dataset B there was an observed maneuver, i.e., $SR = 0|OM = 1$. If the driver of the subject vehicle follows the SVM’s recommendation and stay in the original lane, he/she simply gives up a lane changing opportunity, and wait until the next $SR = 1$. The negative consequence of the Type I error is worse than the Type II error. Therefore, priority is given to the model with the highest accuracy for classifying vectors with $OM = 0$. With this criterion, the SVM with fine Gaussian kernel function (denoted as SVM-

(a) FIS		FIS Recommendation, FR			
		Yes, move into the destination lane now FR = 1	No, stay in the same lane FR = 0	Total	Accuracy (%)
Observed	Changed lane OM = 1	141	30	171	82.5 %
Maneuver OM	Did not change lane OM = 0	1,020	208,661	209,681	99.5 %
Total		1,161	208,691	209,852	

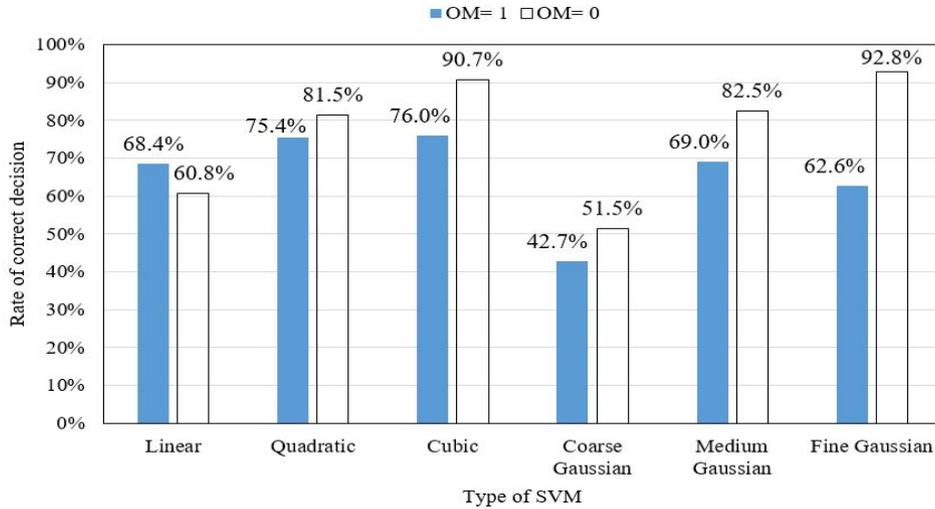
(b) SVM-FG		SVM-FG Recommendation, SR			
		Yes, move into the destination lane now SR = 1	No, stay in the same lane SR = 0	Total	Accuracy (%)
Observed	Changed lane OM = 1	107	64	171	62.6 %
Maneuver OM	Did not change lane OM = 0	15,045	194,636	209,681	92.8 %
Total		16,206	194,700	209,852	

(c) MLF-6		MLF-6 Recommendation, MR			
		Yes, move into the destination lane now MR = 1	No, stay in the same lane MR = 0	Total	Accuracy (%)
Observed	Changed lane OM = 1	92	79	171	52.6 %
Maneuver OM	Did not change lane OM = 0	121,425	88,256	209,681	42.0 %
Total		125,717	88,335	209,852	

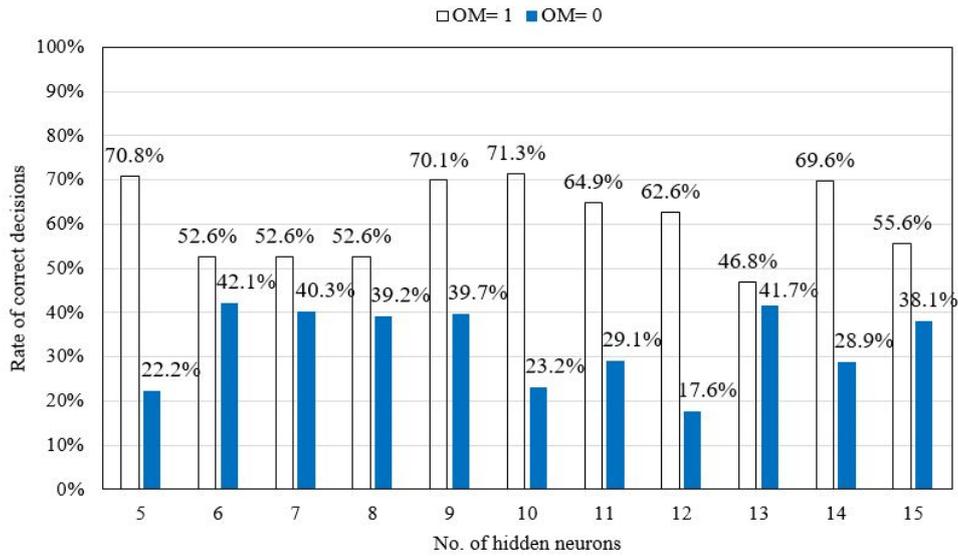
(d) GAM		GAM Recommendation, GR			
		Yes, move into the destination lane now GR = 1	No, stay in the same lane GR = 0	Total	Accuracy (%)
Observed	Changed lane OM = 1	100	71	171	58.5 %
Maneuver OM	Did not change lane OM = 0	69,810	139,871	209,681	66.7 %
Total		69,910	139,942	209,852	

Tab. II Classification matrix for Dataset B.

FG) was selected to represent the best SVM trained. The classification matrix for SVM-FG is shown in Tab. II(b). This SVM-FG achieves correct classification rates of 62.6% for OM = 1 and 92.8% for OM = 0, respectively.



(a) SVM



(b) MLF

Fig. 5 Classification accuracies of SVMs and MLFs with Dataset B.

Fig. 5(b) plots the rates of correct decisions, for the test vectors in Dataset B, for the MLFs with different number of hidden neurons. None of the MLFs has the highest accuracies for vectors that belong to OM = 0 and OM = 1 classes,

respectively. In general, the vectors for OM = 1 have a better accuracy (46.8 % to 71.3 %) than the vectors for OM= 2 (17.6 % to 42.1 %). One may argue that the higher accuracies for the OM = 1 vectors are because in the training data file, the OM = 1 vectors were replicated such that there were approximately equal number of OM = 1 and OM = 0 vectors. Because of the MLF’s training algorithm, such replication is necessary. Otherwise the MLFs will be trained to bias towards OM = 0 vectors. The FIS and SVMs do not face such an issue. Following the approach that selected SVM-FG, the rate of correct decisions for OM = 0 is more critical than OM = 1. The MLF with six hidden neurons has the best rate of correct decisions for OM = 0 vectors, at 42.1 %. It has a rate of correct decisions of 52.6 % for OM = 1 vectors. The training of this MLF was stopped after 13 iterations, with a mean-squared-error of 0.00805. This MLF, denoted as MLF-6, was selected for comparison of performance with FIS, SVM-FG and GAM. The classification matrix for MLF-6 is shown in Tab. II(c).

The GAM has a simple decision rule:

$$\text{IF } [(G_{TA} \geq G_{TA}^{\min}) \text{ AND } (G_{LA} \geq G_{LA}^{\min})] \text{ THEN } (GR = 1) \text{ ELSE } (GR = 0),$$

where GR denotes GAM’s Recommendation, GR= {0,1}, which corresponds to {“no, stay in the same lane”, “yes, move to the destination lane now”}. The equations for the critical gaps are:

$$G_{LA}^{\min} = \exp [1 + 1.541 \cdot \max (0, V_{LA} - V) + 0.621 \cdot \min (0, V - V_{PA}) + 0.130V_{LA} - 0.008v + \varepsilon_{LA}] \tag{3}$$

$$G_{TA}^{\min} = \exp [1.5 + 1.426 \cdot \max (0, V_{TA} - V) + 0.640V_{TA} - 0.205v + \varepsilon_{TA}], \tag{4}$$

and

$$\varepsilon_{LA} \sim N (0, 0.854), \quad \varepsilon_{TA} \sim N (0, 0.954).$$

Since GAM had been calibrated with NGSIM data, it was directly applied to Dataset B, and the resulting GR are used to compare against the performances of FIS, SVM and MLF. In (3) and (4), the coefficients were taken from [7]. Tab. II(d) shows the classification matrix obtained by applying GAM to Dataset B. The GAM achieved accuracies of 58.5 % for OM = 1 and 66.7 % for OM = 0.

Comparing the accuracies of FIS, SVM-FG, MLF-6 and GAM in classifying vectors in Dataset B with OM = 1 and OM = 0 respectively in the rightmost column in Tab. II, the FIS has the highest rates of correct decisions for vectors that belong to OM = 1 and OM = 0 , among the four models. The rate of correct decisions of FIS for OM = 1 is 82.5 % which is 19.9 % higher than the second place model, SVM-FG. The FIS also achieves the high accuracy of 99.5 % for the OM = 0 vectors, among the four models tested. The overall accuracies of SVM-FG, MLF-6 and GAM are not as satisfactory.

The good performance of the FIS may be attributed to its approach; that is, representing the human reasoning process through fuzzy sets, fuzzy membership functions, and fuzzy rules. The FIS’s decision making process can easily be understood by the users (drivers). The approaches used by MLFs and SVMs are data driven. The SVM learning algorithm constructs the hyperplane that separates the OM = 1 and OM = 0 vectors in the four dimensional space, or the kernel space.

The MLF backpropagation training algorithm adjusts the weight vectors and the bias values to map the four input values of each training vector with its desired output value. There is no clear explanation of the human reasoning logic that relates the input values to the output values in the SVMs and MLFs.

8. Conclusion

This research has developed, using the NGSIM vehicle trajectory data, a SVM (denoted by SVM-FG), and a MLF (denoted by MLF-6), for making discretionary lane changing decisions on freeways. This research has also implemented the existing FIS and TRANSMODELER's GAM, which was calibrated by their original developers using NGSIM data, for making decisions on lane changing move. The rates of correct decisions for "yes, move to the destination lane now" and "no, stay in the same lane" obtained by implementing the FIS, SVM-FG, MLF-6 and GAM, from an independent NGSIM test data set were compared. The results show that the FIS has a highest rates of correct decisions for both "yes, move to the destination lane now" at 82.5% and "no, stay in the same lane" at 99.5%. The SVM-FG has the next highest rates of correct decisions for both "yes, move to the destination lane now" and "no, stay in the same lane", at 62.6% and 92.8% respectively. Among the four modeling techniques tested, the FIS is hence recommended to model discretionary lane changing decisions on freeways.

References

- [1] BALAL E., CHEU R.L., SARKODIE-GYAN T., MIRAMONTES, J. Analysis of discretionary lane changing parameters on freeways. *International Journal of Transportation Science and Technology*, 2014, 3(3), pp. 277-296, doi:10.1260/2046-0430.3.3.277.
- [2] BALAL E., CHEU R.L., SARKODIE-GYAN T. A binary decision model for discretionary lane changing move based on fuzzy inference system. *Transportation Research Part C: Emerging Technologies*, 2016, 67, pp. 47-61, doi:10.1016/j.trc.2016.02.009.
- [3] BRACKSTONE W.J., MCDONALD M. Fuzzy sets and systems for a motorway microscopic simulation model. *Fuzzy Sets and Systems*, 2000, 116, pp. 65-76, doi: 10.1016/S0165-0114(99)00038-X.
- [4] CALIPER. *TransModeler User's Guide*. Version 2.6, Caliper Corp. Newton, MA. 2011.
- [5] CAMBRIDGE. NGSIM U.S. 101 *Data Analysis (7:50 a.m. to 8:05 a.m.) Summary Report*. Cambridge Systematic, Inc., Oakland, CA. 2005. Available from: <https://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm>.
- [6] CAMBRIDGE. NGSIM U.S. I-80 *Data Analysis (4:00 p.m. to 4:15 p.m.) Summary Report*. Cambridge Systematic, Inc., Oakland, CA. 2005. Available from: <https://www.fhwa.dot.gov/publications/research/operations/06137/index.cfm>.
- [7] CHOUDHURY C. *Modeling Driving Decisions with Latent Plans*. USA, 2007. Ph.D. thesis, MIT. Available from: <https://dspace.mit.edu/handle/1721.1/42220>.
- [8] DAS S., BOWLES B.A. Simulation of highway chaos using fuzzy logic. In: *Proceedings of the 18th International Conference of the North American Fuzzy Information Society (NAFIPS)*, New York, NY, USA, 1999, pp. 130-133.
- [9] HAYKIN S. *Neural Networks: A Comprehensive Foundation*. 2nd ed., Prentice Hall, New York, NY, USA, 1998.
- [10] JIANG J-S.R., SUN C-T., MIZUTANI E. *Neuro-Fuzzy and Soft Computing*. Prentice Hall, Upper Saddle River, NJ, 1997.

- [11] KARLAFTIS M.G., VLAHOIANNI E.I. Statistical methods versus neural networks in transportation research. *Transportation Research Part C: Emerging Technologies*, 2011, 19(3), pp. 387-399, doi:10.1016/j.trc.2010.10.004.
- [12] KUKLOVA, J., PRIBYL, O. Changeover from decision tree approach to fuzzy logic approach within highway management. *Neural Network World*, 2017, 2, pp. 181-196, DOI: 10.14311/NNW.2017.27.008.
- [13] MASSAHI A., CUTILLO M. A. Estimating the capacity impacts of urban street incidents. In: *The 96th Annual Meeting of the Transportation Research Board*. Washington D.C. USA, 2016.
- [14] MATHWORKS. *MATLAB Fuzzy Logic Toolbox User's Guide*. MathWorks, Inc., Natick, MA, 2014.
- [15] MORIDPOUR S., SARVI M., ROSE G. Lane changing models: a critical review. *Transportation Letters*, 2010, 2, pp. 157-173, doi:10.3328/TL.2010.02.03.157-173.
- [16] MORIDPOUR S., SARVI M., ROSE G., MAZLOUMI E. Lane-changing decision model for heavy vehicle drivers. *Journal of Intelligent Transportation Systems*, 2012. 16(1), pp. 24-35, doi:10.1080/15472450.2012.639640.
- [17] NOVOTNY S., BOUCHNER P. Advanced methodology for evaluation of driver's actual state with use of technical driving data. In: *8th WSEAS International Conference o Systems Theory and Scientific Computation (ISTAC'08)*, Rhodes, Greece, 2008, pp. 160-165.
- [18] PUNZO V., BOIZACCHIELLO M.T., CIUFFO B. On the assessment of vehicle trajectory accuracy and application to the Next Generation Simulation (NGSIM) program data. *Transportation Research Part C: Emerging Technologies*, 2011, 19, pp. 1243-1262, doi:10.1016/j.trc.2010.12.007.
- [19] ROSS T.J., *Fuzzy Logic with Engineering Applications*. 2nd Edition, John Wiley, 2004.
- [20] SCHLKOPH B., SMOLA A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.
- [21] SIUHI S., KASEKO M. Parametric study of stimulus-response behavior for car-following models. In: *The 89th Annual Meeting of the Transportation Research Board*, 2010, pp. 10-1179.
- [22] VAPNIK V. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [23] WANG Q., LI Z., LI L. Investigation of discretionary lane-change characteristics using Next-Generation Simulation data sets. *Journal of Intelligent Transportation Systems*, 2014. 18, pp. 246-253, doi:10.1080/15472450.2013.810994.
- [24] YANG M., CHEN C., WANG L., YAN X., ZHOU L. Bus arrival time prediction using support vector machine with genetic algorithm. *Neural Network World*, 2016, 3, pp. 205-217, doi: 10.14311/NNW.2016.26.011.
- [25] YU B., WANG Y.T., YAO J.B., WANG J.Y. A comparison of the performance of ANN and SVM for the prediction of traffic accident duration. *Neural Network World*, 2016, 3, pp. 271-287, doi:10.14311/NNW.2016.26.015.
- [26] YUAN F., CHEU R.L. Incident detection using support vector machines. *Transportation Research Part C: Emerging Technologies*, 2003, 11(3), pp. 309-328, doi:10.1016/S0968-090X(03)00020-2.
- [27] ZADEH L. A. Fuzzy sets. *Information Control*. 1965. 8, pp. 338-353.
- [28] ZHENG Z. Recent developments and research needs in modeling lane changing. *Transportation Research Part B: Methodological*, 2014, 60, pp. 16-32, doi:10.1016/j.trb.2013.11.009.