

Research Article

A Red-Light Running Prevention System Based on Artificial Neural Network and Vehicle Trajectory Data

Pengfei Li,^{1,2} Yan Li,³ and Xiucheng Guo¹

¹ Transportation School, Southeast University, 2 Sipailou, Nanjing 210096, China

² School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA

³ Highway School, Chang'an University, The Middle Section of Southern Second Ring Road, Xi'an 710064, China

Correspondence should be addressed to Yan Li; lyan@chd.edu.cn

Received 14 July 2014; Accepted 5 October 2014; Published 4 November 2014

Academic Editor: Xiaobei Jiang

Copyright © 2014 Pengfei Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The high frequency of red-light running and complex driving behaviors at the yellow onset at intersections cannot be explained solely by the dilemma zone and vehicle kinematics. In this paper, the author presented a red-light running prevention system which was based on artificial neural networks (ANNs) to approximate the complex driver behaviors during yellow and all-red clearance and serve as the basis of an innovative red-light running prevention system. The artificial neural network and vehicle trajectory are applied to identify the potential red-light runners. The ANN training time was also acceptable and its predicting accurate rate was over 80%. Lastly, a prototype red-light running prevention system with the trained ANN model was described. This new system can be directly retrofitted into the existing traffic signal systems.

1. Introduction

Motorists face indecisiveness during the yellow and all-red clearance at signalized intersections. It is a composite result of the incompatible reactions to the changes of signal indicators and random safety perceptions among motorists. Such indecisiveness is a leading cause for signal violations at intersections. According to a research conducted by the University of California, Berkeley, 2.5 million accidents or 40% of all reported crashes in US were considered related to intersections in 2004 and 20% of these intersection accidents were signal-related [1], which can be interpreted a \$13 billion loss annually [2]. In order to prevent the signal-related accidents, it is necessary to study the driver behaviors at intersections. Compared to the driver behaviors at other road segments (e.g., freeway or arterial links), the complexity of driver behaviors around intersections is that, during the yellow and all-red clearances, a driver will have to not only respond to signal changes but also interact with other adjacent vehicles to ensure safe maneuvers. As such, the driver behaviors at intersections can hardly be represented with analytical models.

The most severe problem caused by the signal violation is red-light running (RLR). The RLR is defined as a situation

that approaching vehicles attempt to cross intersections during all-red or red phases. A RLR may be caused either by the driver's misperception of signal settings or simply by being distracted. Dilemma zone (DZ) is an area where the vehicles face indecisiveness of whether to stop or go at the yellow onset and it is commonly considered the primary reason for the RLR problem. Therefore, most of the RLR prevention systems in the past focus on modeling the driver behaviors in the DZ and countermeasures to protect the vehicles in DZ. Although success has been reported, some research also reported high RLR occurrences at congested and therefore low-speed intersections, where the DZ problem hardly exists [3, 4]. This finding implies that the RLR problem cannot be well addressed solely by mitigating the DZ issues. At congested intersections, the drivers may be distracted or just lose their respect to signal after excessive delays. They might determine to cross the intersection during the yellow, even though there is a RLR risk, so as to avoid further waiting. These intuitive explanations have little to do with the dilemma zone but significantly contribute to the RLR problem. After an extensive literature review, we concluded that there are no RLR prevention systems that could address the aforementioned situations because nearly all the existing

RLR prevention systems, or collision avoidance systems, were based on vehicles' kinematics which did not take into account all possible reasons for the RLR issues. The RLR prevention system developed in this paper was based on the ANN technology. The ANN technology has been widely used to approximate complex system behaviors. In our system, variants of ANN networks were extensively trained to approximate the driver behaviors during yellow and all-red at intersections and the trained ANN model was used to predict if an approaching vehicle would become a red-light runner and take some preventive measures accordingly.

The development of ANN was initially inspired by understanding biological learning systems, such as human brains, but has been divided into two groups at present: one focuses on using ANNs to model biological process and the other focuses on developing effective machine learning algorithms [5]. The ANN is one of the most commonly used methods to approximate behaviors of complex systems. Typical ANNs are composed of a web of interconnected "neurons" (also called "nodes" or "processing units" in other literature) (see Figure 1). Mathematically, an ANN network is intended to approximate complex systems with weighted composition of relative simple functions which are further approximated with weighted composition of other simple functions and so on. A well-designed ANN should be able to contain sufficient system complexity.

In this paper, the authors first analyzed several ANN models to approximate the driver behaviors regarding the RLR problem and then illustrated the potential of using the ANN model to address the RLR problem. Second, a conceptual RLR prevention system was designed and evaluated. This paper is structured as follows: in the first section, the authors explained the structures of several popular ANN network variants, their advantages, and possible shortcomings; secondly, based on the vehicle trajectories data, the authors designed, trained, and then selected the most efficient ANN models as the fundamental predicting model to present the driver behaviors during the yellow and all-red clearance; this model served as the fundamental model in predicting possible RLR events. Lastly, the authors also discussed how to develop and retrofit this new system into the existing traffic signal systems.

2. Literature Review

The literature review was divided into two groups and reviewed, respectively: literature on the ANN and literature on the ANN's application to the traffic studies.

There is rich literature on the ANN theories and new research efforts are still being dedicated to this research today. Therefore this review can only cover a small portion of all related literature. As far as model specification is concerned, the ANNs are determined by three kinds of parameters:

- (i) the interconnection pattern between different layers of neurons;
- (ii) the activation function that converts a neuron's weighted input to its output activation;
- (iii) the learning process for updating the weights of the interconnections.

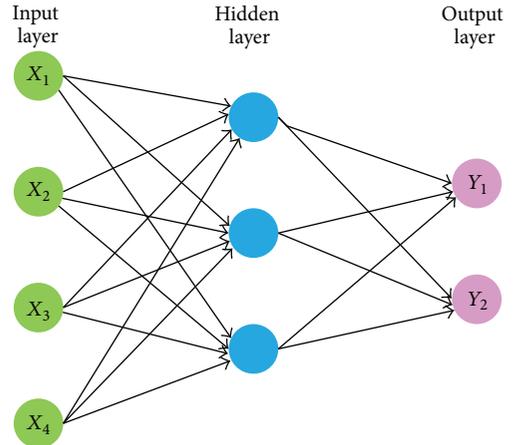


FIGURE 1: An illustration of classic three-layer neural network structure.

According to the interconnection pattern between ANN neurons, the ANNs can be divided into feedforward and recurrent neural networks (RNNs). The connections between neurons in feedforward ANNs are acyclic like in Figure 2(a); a variant of feedforward ANN is the neural network with shortcuts in which some connections skip intermediate layer(s) like in Figure 2(b). In contrast, the connections in the RNNs can form circles and therefore use internal memory to process the inputs series as in Figure 2(c).

The *feedforward neural network* is relatively simple and commonly applied to various fields. McCulloch and Pitts are recognized as the founder of the ANN concept and designed the first neural network by combining many simple processing units together to increase in computational power [6]. The first influential feedforward neural network model can be traced back to the "perceptron" defined by Rosenblatt [7], which is a linear classifier by calculating a linear composition of the inputs with (1) and then outputting 1 if the calculation is greater than certain threshold, otherwise -1. Training a perceptron is a process of choosing values for the weights in the space H of all possible weight vectors:

$$o(x_1, x_2, \dots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n > 0 \\ -1 & \text{otherwise,} \end{cases} \quad (1)$$

where w_i is the weight that determines the contribution of input x_i .

From (1), the original perceptron is single-layer and can only express linear decision surface and the inputs have to be linearly separable. To overcome these shortcomings, the perceptron was extended to multiple layers, or the multilayer perceptron (MLP). The major difference between the original perceptron and MLP is that each neuron's output in MLP is

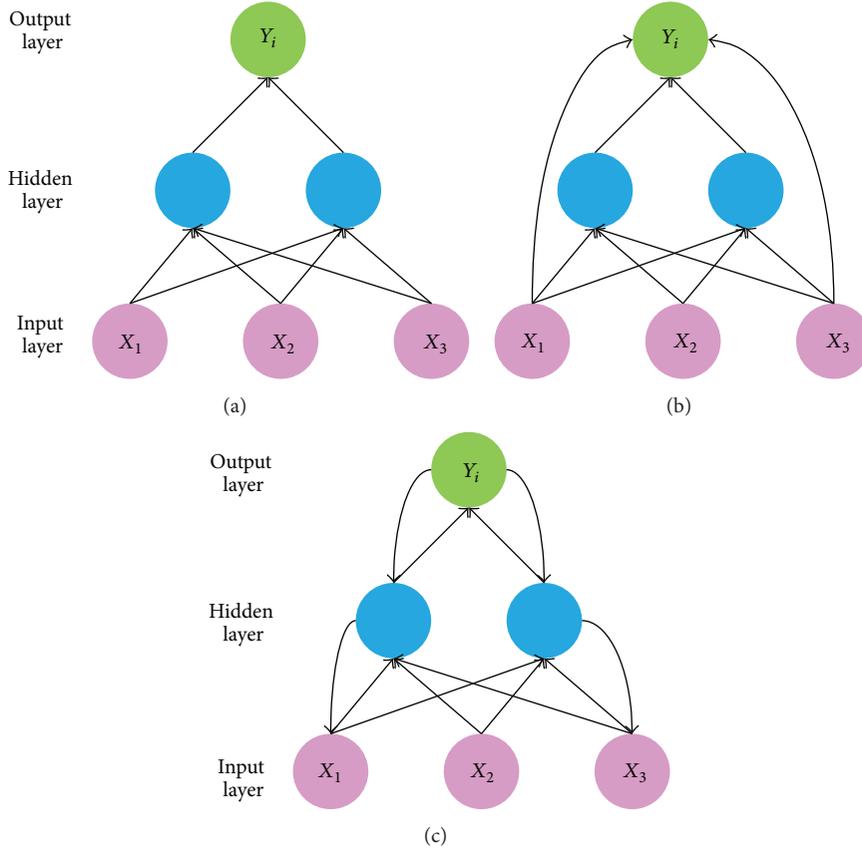


FIGURE 2: Three types of neural network connections.

a nonlinear and differentiable function (namely, activation function) of its inputs [8]. The MLP's nonlinear feature allows for representing more complex systems. Later, Werbos [9] and Rumelhart et al. [10] developed efficient backpropagation training algorithms for the MLP which significantly extend the MLP's applicability in various fields.

It is apparent that the feedforward neural network treats all the data as new and cannot discover the possible temporal dependence between samples. This shortcoming sometimes needs a feedforward neural network to be extended to a rather large scale to approximate complex systems. In other words, the feedforward neural network has a memoryless structure. By contrast the RNN allows for the internal feedback and is more appropriate to solve certain types of dynamic problems. Jordan introduced the first RNN which feeds the outputs back to the input vector with time delay [11]. In other words, the RNN output at time t will be used as part of input information at $t + 1$. Mathematically, the outputs of a three-layer Jordan network with m , q , and n neurons on the input layer, hidden layer, and output layer, respectively, are as follows:

$$o_{t+1,j} = F \left(\beta_{j,0} + \sum_{h=1}^q \beta_{j,h} G(\gamma_{h,0} + x'_t \gamma_h + o'_t \delta_h) \right), \quad (2)$$

$$j = 1, 2, \dots, n,$$

where x'_t, o'_t are vectors of input and output at time t ; δ_h is the vector of the connection weights between h th hidden neurons and input neurons which receive lagged outputs; $\beta_j = (\beta_{j,1}, \beta_{j,2}, \dots, \beta_{j,q})'$ is the vector of the connection weights between the j th output neuron and all q hidden neurons; $\gamma_h = (\gamma_{h,1}, \gamma_{h,2}, \dots, \gamma_{h,m})'$ is the vector of the connection weights between the h th hidden neuron and all m input neurons; F and G are the activation functions on the output layer and hidden layer, respectively; and $\beta_{j,0}, \gamma_{h,0}$ are biased terms to add the flexibility of activation functions.

Similarly, Elman designed a RNN that the hidden neurons are connected to input neurons with time delay as in (3) [12]. Consider

$$o_{t+1,j} = F \left(\beta_{j,0} + \sum_{h=1}^q \beta_{j,h} a_{t,h} \right), \quad j = 1, 2, \dots, n, \quad (3)$$

$$a_{t+1,h} = G(\gamma_{h,0} + x'_t \gamma_h + a'_t \delta_h), \quad h = 1, 2, \dots, q,$$

where $a_t = (a_{t,1}, a_{t,2}, \dots, a_{t,q})'$ is the vector of lagged hidden-neuron activations; δ_h is the connection weights between the h th hidden neuron and all the inputs which receive lagged hidden neuron activations.

Obviously, (3) and (4) can be extended to feed multiple previous step output vectors observations back to the input vectors.

2.2. Choice of Activation Functions. The activation functions in neurons are the building blocks of an ANN model. Similar to the neurons in a biology system, the activation function determines whether a neuron should be turned on or off according to the inputs. In a simple form, such on/off response can be represented with threshold functions, also known as a Heaviside function in the ANN literature as follows:

$$G(\gamma_{h,0} + x'_t \gamma_h) = \begin{cases} 1, & \text{if } \gamma_{h,0} + x'_t \gamma_h \geq 0 \\ 0, & \text{if } \gamma_{h,0} + x'_t \gamma_h < 0 \end{cases}, \quad (4)$$

where c is the threshold and the remaining variables are defined previously. In some complex systems, the neurons may also need to be bounded real values. It is common to select sigmoid (S-shaped) and squashing (bounded) activation functions. It is also required that an activation function is bounded and differentiable. The most used two sigmoid functions in the ANN models are the logistic function and hyperbolic tangent (Tanh) function. Equations (5) and (6) are their mathematical expressions:

$$G(\gamma_{h,0} + x'_t \gamma_h) = \frac{1}{1 + e^{-(\gamma_{h,0} + x'_t \gamma_h)}}. \quad (5)$$

$$G(\gamma_{h,0} + x'_t \gamma_h) = \frac{e^{(\gamma_{h,0} + x'_t \gamma_h)} - e^{-(\gamma_{h,0} + x'_t \gamma_h)}}{e^{(\gamma_{h,0} + x'_t \gamma_h)} + e^{-(\gamma_{h,0} + x'_t \gamma_h)}}. \quad (6)$$

2.3. Learning Process to Update the Weights of Interconnections. Training ANNs can be divided into supervised training and unsupervised training. The supervised learning needs pairs of training samples and each pair is composed of inputs and desired outputs (i.e., observations). The learning process is to adjust the interconnection weights to reduce the difference between the inferred outputs from the ANN model and the actual observations whereas the unsupervised learning is to find hidden structure in unlabeled data with, for example, statistical inference. In the context of this paper, the authors only review part of influential supervised learning algorithms.

To effectively approximate the complex systems, the interconnection weights in the ANNs have to be estimated with the existing observations. A simple example with only one single target output y and the network function $y = f_{G,q}(x; \theta)$ is used to illustrate how to update the weights. θ is the vector of interconnection weights.

After the activation G and the structure of hidden layers are determined and a training sample of T observations is given, the optimal θ can be obtained by minimizing the mean squared error (MSE) in (7), which can be obtained with the first order differentiation of (7) (i.e., (8) and (9)):

$$\frac{1}{T} \sum_{i=1}^T (y - f_{G,q}(x; \theta))^2, \quad (7)$$

$$E [\nabla f_{G,q}(x; \theta) (y - f_{G,q}(x; \theta))] = 0, \quad (8)$$

where $f_{G,q}(x; \theta)$ is the gradient vector of $f_{G,q}$ with respect to θ . Rumelhart et al. designed a recursive gradient-descent-based algorithm to estimate the $\hat{\theta}$ as follows [10]:

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \eta_t \nabla f_{G,q}(x; \hat{\theta}_t) [y_t - f_{G,q}(x; \hat{\theta}_t)], \quad (9)$$

where η_t is the learning rate and (9) is so called backpropagation algorithm and is a generalized form of the ‘‘delta rule’’ of single-layer perceptron model [5]. Some famous variants of the backpropagation algorithm include the *Quickprop* algorithm [13] and the Resilient Backpropagation (*RPROP*), an adaptive learning algorithm [14].

2.4. Complexity Regularization. Given a fixed training sample, an ANN with excessive hidden neurons will overfit the data whereas the ANN with insufficient neurons cannot capture all of systems’ properties and become unstable. In analogy to the linear programming problem, the excessive neurons issue is like more equations than the variables whereas the insufficient neurons issue is like more variables than equations.

One idea is to begin an ANN with zero hidden neurons instead of fixing the ANN structure at first and then insert hidden neurons as needed until the MSE can be reduced to an acceptable level. One of commonly accepted such algorithms is the cascade-correlation (CC) developed by Fahlman and Lebiere [15]. The initial CC neuron network contains zero hidden neuron and therefore it is likely that the target MSE cannot be reached even with a large size of training data. Secondly, several so-called candidate neurons are created and they are only connected to all input neurons and existing hidden neurons with random weights; the third step is to train the weights of neuron candidates to maximize the correlation between the candidate hidden neurons’ activations and overall network errors, which is calculated with (10). Thirdly select the candidate neuron with the highest correlation, freeze its connection weights (i.e., unchangeable during the later training process) to the input neurons, and connect it to the output neurons with random connect weights. At this point, the original CC network grows by one more neuron and lastly the new CC network is trained again to minimize the MSE. If the target MSE is reached, then the training process ends; otherwise, go back to step 2 and repeat until the target MSE is reached. Obviously, the final CC network contains multiple single-neuron hidden layers:

$$C = \frac{\sum_o \left| \sum_p (h_p - \langle h \rangle) (e_{op} - \langle e_o \rangle) \right|}{\sum_o \sum_p (e_{op} - \langle e_o \rangle)^2}, \quad (10)$$

where h is the hidden neuron activation; e is the network error; $\langle h \rangle$, $\langle e_o \rangle$ are means.

As for the ANN’s applications to the traffic studies, it is still in its infancy. Lu et al. developed a neural network based tool to filter and mining the highly skewed traffic data [16]; Huang utilized the wavelet neural network to forecast the traffic flow and the results reveal the forecasting accuracy was improved compared to the traditional methods [17]; Chong et al. deployed the feedforward neural network to train the driver in simulation based on the naturalistic data and the results showed that the driving behavior is closer to the actual observation than the traditional car-following models [18]. Jia et al. trained an ANN-based car-following model with the data collected via a five-wheel system. The inputs include speed of following vehicle, relative speed, relative distance, and desired speed. The output vector includes the

acceleration of the following vehicle [19]. Panwai and Dia developed a similar backpropagation-neural-network-based car-following model in AIMSUN, commercial microscopic software. Compared with the embedded car-following model in AIMSUN, the new car-following model is 20% better in terms of errors reduction [20].

3. Significance of the Research

Many severe crashes occur at signalized intersections today due to signal violations and so it is important to study the red-light running prevention at intersections. Compared with the other traffic segments, the driver behaviors close to signalized intersections are more difficult to represent by models due to the random individual vehicle's decision, intensive interactions between vehicles, and the complex feedback mechanism between vehicles and the traffic signal system. The driver behaviors study at signalized intersections will help identify the possible reasons of certain unsafe vehicle maneuvers, such as the RLR, and eventually help develop countermeasures to mitigate the safety hazards.

In the past, it is commonly assumed that the RLR is caused by the dilemma zone and therefore most of the related research of the red-light running issue focused on how to minimize the vehicles in the dilemma zone. However, some recent research on vehicle trajectories during yellow and all-red clearance reveals that vehicles may still run red lights at low speeds in which the dilemma zone issue hardly exists. This finding implies that there must be other factors than the dilemma zone to contribute to the red-light running. Intuitively, drivers at congested intersections may be more likely to take the RLR risk to cross the intersection in order to avoid further waiting. Or the drivers may be just distracted and fail to observe the traffic lights. Obviously, the reasons for RLR are complex and difficult to be represented with the traditional modeling method. Meanwhile, although it is difficult to precisely analyze the reasons for the individual RLRs, the red-light runners may still share some common kinematic patterns, such as shorter headways from their leading vehicles or faster speeds at yellow onsets. These kinematic features can be retrieved from the vehicle trajectory data collected via the radar, video imaging detectors, or the connected vehicle technology in the future.

In this paper, the authors explored various ANN networks to approximate the driver behaviors during the yellow and all-red clearances. The inputs of ANNs (i.e., vehicles' kinematic features) were captured and calculated based on the vehicles' trajectories during the yellow and all-red clearance. The well trained ANN model then served as the fundamental predictive model to identify the possible red-light runners. The collision avoidance measures were activated then to avoid potential crashes.

4. Methodology

4.1. Problem Representation

4.1.1. ANN Model Inputs. It is assumed that a potential RLR event begins at the yellow onset when a driver has to decide

whether to cross or stop according to his safety perception. The safety perception is psychological and determined by many factors. However, only the four most important factors were taken into account as the ANN input candidates: distance-to-intersection (DTI), travel speed (v_i), the number of front vehicles before the stop line, and headway from the leading vehicle (h_i). These parameters were reportedly effective to represent a driver's status at the yellow onset in other literature [21]. No lane changing was considered in the fact that it was rare according to our field observations.

4.1.2. ANN Model Outputs. In general, there are two possible time moments able to be used to tell an occurrence of RLR: the all-red onset and at the all-red end. If the all-red onset is used, a subject vehicle would be considered a red-light runner when it has not reached the stop line but cannot completely stop according to its distance to the stop line, speed, and maximal possible deceleration. If the all-red end is used, a subject vehicle is considered a red-light runner when it is still within the intersection when the all-red clearance expires. In the all-red-end-based method, two factors are considered relevant regarding the ANN outputs, the DTI and speed.

Using the all-red onset would have to assume that driver becomes the red-light runner only when it cannot stop. However, this assumption is questionable because a slow driver may still want to take the RLR risk to cross the intersection or it may just be distracted and become a red-light runner. Therefore, in this paper, a vehicle's status at the all-red clearance end was used to measure the RLR event. Two types of outputs were used: (1) classifier: a vehicle was labeled as a run-light runner if it was still within the intersection at the all-red clearance end, regardless where it is exactly; (2) the vehicle's location and speed were observed within the intersection at the all-red end. These two output variants were evaluated, respectively, and compared later.

4.2. ANN Model Design

4.2.1. ANN Structure. Since the driver behaviors during the yellow and all-red are independent from cycle to cycle (i.e., the drivers are not aware of other vehicles' maneuvers in previous cycles), the feedforward neuron network was selected due to its memoryless property. In terms of the number of hidden neurons, two ANN structures were used and compared: standard feedforward neuron network with multiple hidden layers and the cascade-correlation (CC) neuron network which iteratively inserts new hidden neurons until the CC can achieve the desired MSE.

4.2.2. Training Algorithms. A variant of the generic backpropagation algorithm is used to train the ANNs, namely, *QuickProp*. The readers can refer to the literature [13] for more details. The weights were updated once with (9) after all the sample pairs were fed in. In other words, the standard backpropagation algorithm in this paper was offline and the weights were updated only once after each epoch.

4.3. Experiments Design. Although the RLR event is a severe safety problem, its occurrence is relatively rare at most

TABLE 1: Illustration of vehicle trajectory data snapshots.

Time	Phase	Veh_ID	Link	DTI	Speed	Headway	Front_veh
17:00:05	1_Y_start	8	1	97	17.1	2.6	2
17:00:11	1_AR_end	8	1	9	7.1	0.0	0

intersections in reality. In order to capture real RLR samples, it would take several months to collect data and the data processing would be tedious too. As such, we determined using microscopic simulation to create RLR samples. The challenge of using simulation to train the ANN network was how to adjust the simulation settings so that the real driver behaviors could be accurately reflected in simulation. In this paper, we used PTV VISSIM simulation engine and carefully calibrated the VISSIM’s car-following model and stop-or-go responses to signal changes with the vehicle trajectory data at the Peppers Ferry intersection in Christiansburg, Virginia, which was collected with a high performance data acquisition system. The data were choreographed and recorded by a customized hardware package. The data included synchronized vehicle trajectories, signal phases states, and error messages and were stored at 20 HZ to a binary file. The sensing system was composed of radar, signal sniffer, and video imaging systems. Table 1 illustrates vehicle’s trajectory data snapshots at the yellow onset and after all-red clearance.

Each vehicle’s trajectory and its stop-or-go decision at the yellow onset were summarized. Then vehicles’ speed distribution, average headway, still headway, acceleration distribution, and other information were summarized [22]. Then they all were input into the simulation settings. After these adjustments, vehicles’ behaviors in simulation were very close to the field observation. However, it was found that few RLR events occurred in simulation and therefore we further reduced the drivers’ attention to their front vehicles and to traffic signals. This change generated more red-light runners in simulation and significantly increased RLR samples for the following ANN training.

In reality, either current vehicle trajectory detectors or future connected vehicle technology has a discovery range from 200 meters to 400 meters [23]. Therefore only those vehicles whose distance to the stop line was less than 100 meters were monitored and the status (i.e., DTI_i , v_i , and h_i) of each monitored vehicle was archived at the all-red end if it was within the range during the yellow and all red. The 100-meter area can be translated into 5.5 seconds to 6 seconds to stop line where drivers’ indecisiveness begins at the yellow onset.

The speed limit and traffic volume were 60 km per hour and average 1500 vph on the link and two simulation runs were conducted and lasted until 300 RLR events were captured.

Faster training can be achieved by normalizing the inputs and outputs. The captured vehicles’ DTI was normalized by $DTI_N = DTI/L_d$ where the L_d is the length of discovering area and 100 meters in this paper. Similarly, the speed, headway, and the number of front vehicles were also be normalized in a form of $a_N = (a - a_{min})/(a_{max} - a_{min})$ where the a , a_{min} , and a_{max} are the original value, the observed maximal value,

and observed minimal value, respectively. The data from one simulation run were used to train the ANNs and the data from the other independent simulation run were to validate the training effects and prevent the overfitting issue.

5. ANN Training and Results Evaluation

Multiple experiments were conducted and the results were compared to determine the best ANN model to predict the individual vehicle’s RLR possibilities. The ANN training process is usually long but once the training is finished, the well trained ANN model is essentially an analytical model and so it is fast enough for all kinds of online applications.

5.1. Scenario One: Input Data Are Combined with Red-Light Runners and Regular Vehicles

Step 1. Train and compare various ANNs with different compositions of input variables, output variables, and network structures. The training algorithm was the standard backpropagation algorithm as in (9) with the learning rate 0.7 and the stopping MSE was 0.005. The activation functions were set as the Tanh functions (6) for both hidden neurons and output neurons. Preliminarily, sixteen options were generated with various compositions of inputs and outputs. The underlying rationale was that some input variables may contribute more to the RLR problem than the others and it is needed to only capture the most important factors to avoid overcomplicating the problem. In addition, the output variants are useful for various collision avoidance strategies.

Given that we had little prior knowledge about how many hidden layers and neurons of the MLP network were sufficient to approximate the RLR problem, it was wise to start with the cascade-correlation (CC) network which gradually adds hidden neurons while learning and the final CC structure can help us to better understand the ANN’s necessary complexity. Table 2 describes the configurations of all the sixteen options. After some preliminary tests, the maximum of hidden neurons in the CC model was set as 100 because more neurons made the training excessively long with only limited further MSE reduction. The MLP structure was designed as three hidden layers and each hidden layer contains 10 hidden neurons.

Table 3 is the ranking in the minimum MSE (i.e., the effectiveness of approximation). From Table 3, only Options 8 and 16 could reach the target MSE (0.005) and therefore be selected as the candidate model and then go to the next step: model validation. The remaining options stagnated before reaching the desired 0.005. Figure 3 reveals the learning trends of Option 8 and Option 16. Option 8 and Option 16

TABLE 2: Configurations of preliminary twelve ANNs.

	Input: yellow onset				Output: all-red end				ANN structure	
	DTI	Speed	Headway	Number of front veh.	DTI	Speed	Classifier	Fixed-structure	Dynamic cascade-correlation	
Option 1	X	X			X	X		X		
Option 2	X	X			X	X				X
Option 3	X	X	X		X	X		X		
Option 4	X	X	X		X	X				X
Option 5	X	X		X	X	X		X		
Option 6	X	X		X	X	X				X
Option 7	X	X	X	X	X	X		X		
Option 8	X	X	X	X	X	X				X
Option 9	X	X					X	X		
Option 10	X	X					X			X
Option 11	X	X	X				X	X		
Option 12	X	X	X				X			X
Option 13	X	X		X			X	X		
Option 14	X	X		X			X			X
Option 15	X	X	X	X			X	X		
Option 16	X	X	X	X			X			X

TABLE 3: MSE ranking among various options.

Option	8	16	2	14	6	4	12	10	15	9	1	5	7	13	3	11
Min MSE	0.005	0.005	0.0058	0.0067	0.0087	0.0089	0.0089	0.0095	0.0098	0.0225	0.0243	0.0252	0.026	0.0268	0.0274	0.0274

had no overfitting issues before reaching the target MSE since the test MSEs kept decreasing in the training process.

Step 2 (model validation with a new set of data). After the models were trained and two candidate models were selected, these two ANN models were coded into the RLR prevention system to serve as the fundamental predictive model. In the next section, the development of the RLR prevention system is described in detail. Before the system was deployed in the real world, the ANN model was first tested to justify its accuracy in predicting red light runners according to their kinematic patterns at the yellow onset. The RLR prevention will not work unless the red-light runner prediction is accurate enough. Two types of errors were evaluated: Type I error: a regular vehicle was reported as a red light runner; Type II error: a red-light runner fails to be predicted. The new set of data contains 1500 samples which includes 1450 regular vehicles and 150 red-light runners. Table 4 reveals the results of Type I and Type II errors.

From Table 4, it seems that both ANN models had low rates of false alarms (i.e., Type I error) but were not effective in predicting the red-light runners (i.e., high rates of Type II error). It makes sense because the vast majority of data was composed of regular vehicles and therefore the ANN models overwhelmingly learn the patterns of regular vehicles compared to the red-light runners. Therefore if the mixed data (regular and red-light runner) were used for training, the false alarm rate was low whereas the accurate rate of predicting RLR events was low due to lack of enough samples. In order to improve the RLR predicting effectiveness, Scenario

Two was designed which only contains the red-light runner data.

5.2. Scenario Two: Input Data Only Contains the Red-Light Runners. Similarly, Scenario Two was also divided into two steps. The Options 9~16 in Table 2 were no longer suitable since all the data were for red-light runners. From the previous experience in Scenario One, all four relevant inputs were selected and the vehicle's location at the all-red end was selected as the ANN output. We compared the four relevant four inputs of regular vehicles and red-light runners and displayed the results in Figure 4. It seems that most red-light runners were 50 meters to 130 meters away from intersection at the yellow onset which means 3 to 6 seconds to the intersection. It was also found that the RLR vehicles tended to have slightly higher speeds, shorter headways, and fewer front vehicles at the yellow onset. These phenomena make sense because the RLR vehicles are intuitively more aggressive than regular vehicles and these findings also supported our fundamental speculations that the RLR vehicles could be distinguished from regular vehicles according to their features at the yellow onset.

Learning from Scenario One, we found that the number of neurons should be at least 100 in order to capture the key patterns of red-light runners and reduce the MSE to the desired level. After several preliminary experiments, fully connected and with five hidden layers with 20 neurons on each hidden layer. The activation functions of all neurons were the symmetric sigmoid as in (6).

TABLE 4: Results of data validation in scenario one.

		Observed	
		Positive	Negative
Calculated (Option 2)	Positive	11%	3%
	Negative	89%	97%
Calculated (Option 16)	Positive	11%	3%
	Negative	89%	97%

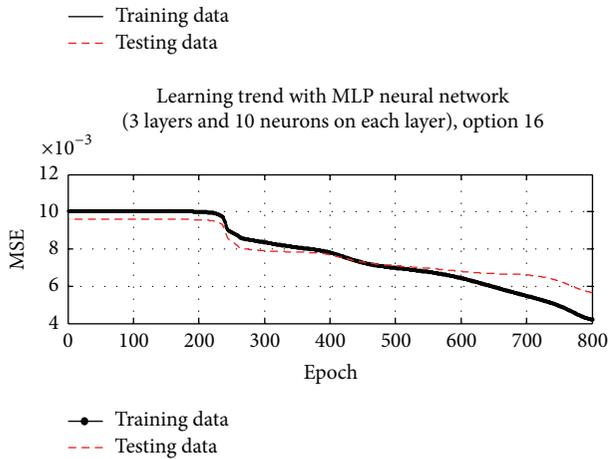
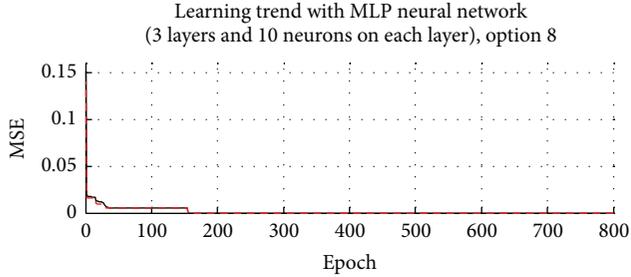


FIGURE 3: Training trend under the Option 8 and Option 16 models.

Step 1 (collected 300 RLR event samples). Consider that the sample size was smaller and therefore the target MSE was reduced to 0. Figure 6 reveals that model in Figure 5 converged quickly and then became stagnated while the test MSE begins to increase. Therefore the final ANN model was selected at the 7000th epoch.

Step 2 (model validation with a new set of mixed data containing 300 new RLR events and 7,000 regular vehicles). Table 5 shows the predicting accuracies of the trained ANN models. Compared to Table 4, the new ANN model could significantly reduce Type I and Type II errors in the RLR prediction. This makes sense because the ANN was trained with RLR samples only and therefore the accuracy of predicting RLR events would clearly increase. Meanwhile, since this is a binary identification problem, the regular vehicles' identification accuracy will also be increased accordingly. The total training time was about two and half hours with a standard desktop PC, which was acceptable.

With all RLR samples, we further plot identified (blue in Figure 7) and unidentified samples (red in Figure 7),

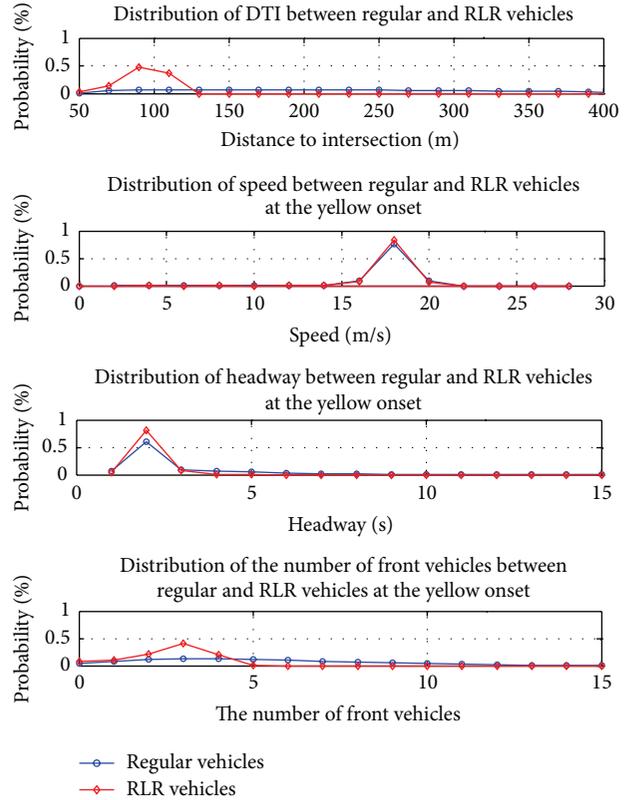


FIGURE 4: Comparison between regular vehicles and RLR vehicles.

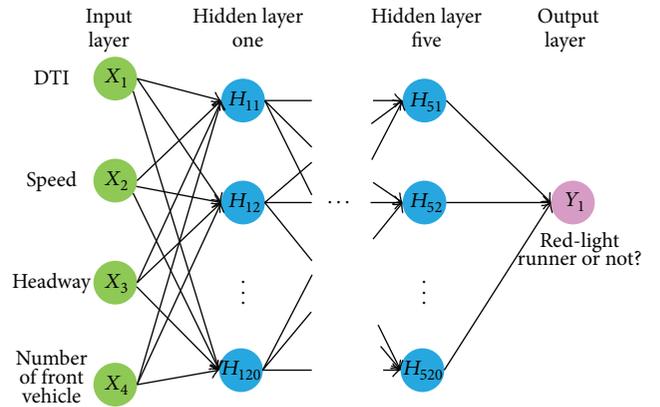


FIGURE 5: Structure of ANN network for training.

respectively, to seek the dominant factors in identifying the RLR vehicles. However, from Figure 7, none of the four factors were statistically effective to separate identified and unidentified groups. Therefore, the ANN model should not be further simplified such as excluding some selecting inputs. Otherwise the predicting accuracy of RLR vehicles would deteriorate.

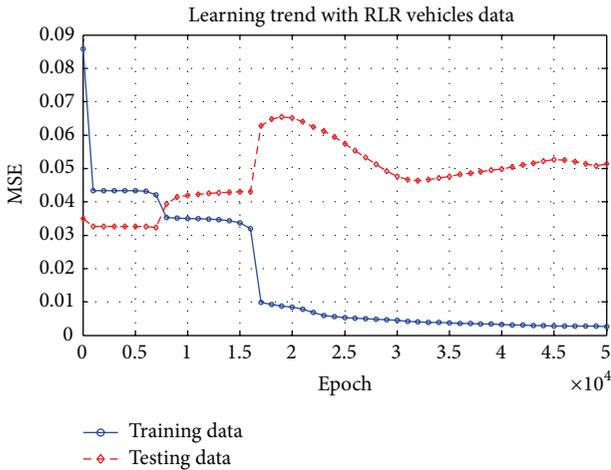


FIGURE 6: Training trend with the red-light running data.

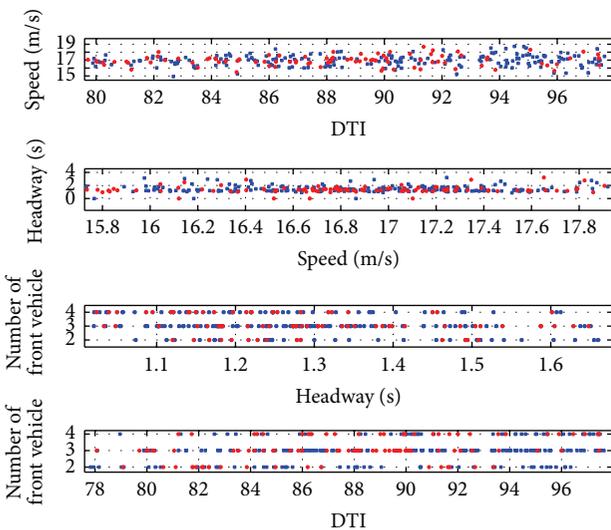


FIGURE 7: Plot of identified and unidentified RLR vehicles.

6. Red-Light Running Prevention System

The challenge of developing such a system is that the ANN network will not be supported by any commercial signal control equipment at this stage and therefore some interfacing equipment must be designed to retrofit this new system into the existing traffic signal systems. Nowadays, most traffic signal controllers in the field are compliant with the National Transportation Communication for ITS protocols (NTCIP) [24]. Through the serial port or Ethernet port on a signal controller, it is possible to override the current timings to prevent the possible RLR-related collisions, such as extending the all-red clearance or extending the current green. As in Figure 8, after the ANN model is trained, the ANN model will be ported to a hardened computer and become a module of the RLR prevention system. The hardened computer will also be connected to a vehicle trajectory detector located at the far end of intersections via the standard Ethernet, such as trajectory radar [25]. The radar

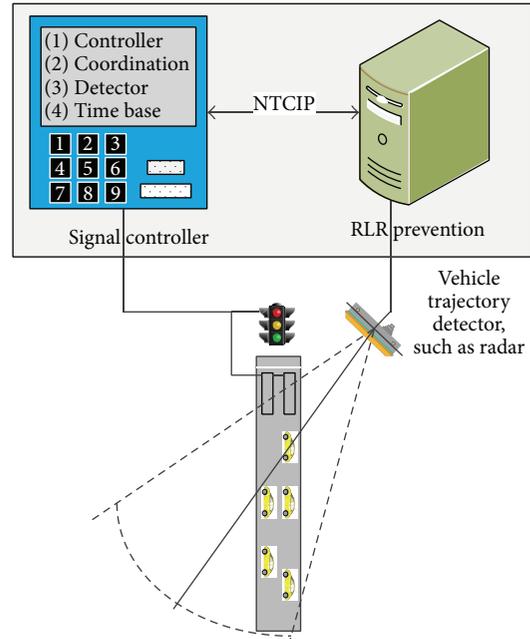


FIGURE 8: Architecture of the new RLR prevention system.

will keep monitoring approaching vehicles and record their speeds, accelerations, and distance. At the same time, the hardened computer also keeps retrieving the signal phasing message from signal controller via the NTCIP 1202 object, such as “OID: 1.3.6.1.4.1.1206.4.2.1.1.4.1.4.1” (Phase Group One Green Status) to understand the phasing status. When a green phase is about to end, the RLR prevention system will examine the approaching vehicles’ speeds, distances to stop line, headway, and other kinematic parameters and then predict if the number of potential red-light runners is beyond a threshold with the ANN model. If so, the computer will send a “hold” NTCIP message to the signal controller, “OID: 1.3.6.1.4.1.1206.4.2.1.1.5.1.4.1” (Phase Hold) to override the current timing and extend the current green for another several seconds. After each extension, there is a minimum time interval for another green extension. At the same time, if the computer finds there are vehicles still within the intersection or some aggressive vehicles are impossible to completely stop after the all-red clearance, the computer will issue another “Phase Hold” command to prevent vehicles on other approaches from entering the intersection. Through these two types of safety countermeasures, the RLR event can be substantially reduced. In practice, the radar detector may lose tracking vehicles when they are totally stopped. However, the latest radio detection product can identify and estimate the vehicles’ trajectories with satisfaction.

The ANN model should be retrained periodically, such as every three months, after sufficient new RLR samples are collected in the field. This way will ensure the system’s effectiveness to the continually evolving traffic patterns.

TABLE 5: Results of data validation in scenario two.

		Observed	
		Positive	Negative
Calculated	Positive	82%	18%
	Negative	17%	93%

7. Conclusions and Future Work

The red-light running is a leading cause for severe crashes at intersections and it has been assumed that the dilemma zone is the major reason for the RLR occurrence. However, recent research has revealed that the RLR occurrence is caused by not solely the dilemma zone but also many other factors. The complexity of modeling the RLR process is beyond most of the close-form analytical models. In this paper, the authors present the potential of the artificial neural networks to approximate the RLR process and predict the RLR occurrence according to vehicles' four statuses (DTI, speed, headway, and the number of front vehicles) at the yellow onset. This information can be obtained from the vehicle trajectory sensors or the connected vehicle technology in the future. From the multiple experiments, we concluded that using the data at the yellow onset as the input and the data at the all-red end as the output is the most effective while training the ANN networks. Using the well trained ANN model, we developed a prototype of RLR prevention system which can predict the potential red-light runners and take countermeasures accordingly.

The predicting accuracy is critical to the success of RLR prevention. With the properly selected inputs and outputs, the final ANN model can predict the RLR events with over 80% accuracy. Besides the two aforementioned countermeasures, other countermeasures are also possible, such as activating warning beacons to remind the potential red-light runners if necessary. Or, when the connected vehicle technology has a high market penetration in the future, the signal control system can send out customized warning messages to individual on-board units according to their specific status.

Although this new system has been proven effective in simulation, it must be further evaluated in the field before being fully deployed. Therefore we plan to deploy this prototype system in the field, fine-tune the ANN networks, and justify the systems' performance with the real RLR samples. Nevertheless, since the ANN model established in this paper does not depend on the vehicle kinematics or any traffic flow theories and the ANN model just retrieves the certain empirical features from the observed driver behaviors, it is expected that the performance of RLR prediction will not be significantly different between the simulation results and the filed observation data. In the future, we will plan to conduct more investigation on applying the artificial neural networks and other artificial intelligence technologies to the traffic safety improvement at signalized intersection.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the China Postdoctoral Science Foundation (no. 20110491333) and the National Nature Science Foundation of China (nos. 51208054 and 51408049). All the programs and codes in this paper are based on a widely used open source neural network library in native C++, namely, Fast Artificial Neural Network Library (<http://leenissen.dk/fann/wp/>). The authors would like to express their appreciation to the FANN developers. The writers are especially thankful to Dr. Vicki Neal and Dr. Zac Doerzaph at Virginia Tech Transportation Institute for providing accessibility to their intersection vehicle trajectory data.

References

- [1] D. C. J. Chang, L. Blincoe, R. Subramanian, and L. Lombardo, "CICAS-V research on comprehensive costs of intersection crashes," Tech. Rep. 07-0016, National Highway Traffic Safety Administration, Washington, DC, USA, 2007.
- [2] D. R. RagLand and A. A. Zabyshtny, "Intersection decision support project: taxonomy of crossing-path crashes at intersections using GES 2000 data," UCB-TSC-RR-2003-08, University of California, Berkeley, Calif, USA, 2000.
- [3] L. Zhang, L. Wang, K. Zhou, and W. B. Zhang, "Dynamic all-red extension at a signalized intersection: a framework of probabilistic modeling and performance evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 166–179, 2012.
- [4] L. Zhang, K. Zhou, W.-B. Zhang et al., *Empirical Observations of Red Light Running at Arterial Signalized Intersection*, Institute of Transportation Studies, University of California, Berkeley, California PATH, 2008.
- [5] T. M. Mitchell, *Machine Learning*, WCB/McGraw-Hill, Portland, Ore, USA, 1997.
- [6] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biology*, vol. 5, no. 4, pp. 115–133, 1943.
- [7] F. Rosenblatt, *The Perceptron, a Perceiving and Recognizing Automaton Project Para*, Cornell Aeronautical Laboratory, 1957.
- [8] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington, DC, USA, 1961.
- [9] P. J. Werbos, *Beyond regression: new tools for prediction and analysis in the behavioral sciences [Ph.D. thesis]*, Harvard University, Cambridge, Mass, USA, 1974.
- [10] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.
- [11] M. I. Jordan, "Serial order: a parallel distributed processing approach," Institute for Cognitive Science Report 8604, San Diego, Calif, USA, 1986.
- [12] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [13] S. Fahlman, "Faster-learning variations on back-propagation: an empirical study," in *Proceedings of the Connectionist Models Summer School*, Morgan-Kaufmann, 1988.
- [14] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the RPROP algorithm," in

- Proceedings of the IEEE International Conference on Neural Networks*, pp. 586–591, San Francisco, Calif, USA, April 1993.
- [15] S. E. Fahlman and C. Lebiere, “The cascade-correlation learning architecture,” in *Advances in Neural Information Processing Systems 2*, S. T. David, Ed., pp. 524–532, Morgan Kaufmann Publishers, Boston, Mass, USA, 1990.
- [16] J. Lu, S. Chen, W. Wang, and B. Ran, “Automatic traffic incident detection based on nFOIL,” *Expert Systems with Applications*, vol. 39, no. 7, pp. 6547–6556, 2012.
- [17] Y. Huang, “Traffic flow forecasting based on wavelet neural network and support vector machine,” *Journal of Computational Information Systems*, vol. 8, no. 8, pp. 3471–3478, 2012.
- [18] L. Chong, M. M. Abbas, and A. Medina, “Simulation of driver behavior with agent-based back-propagation neural network,” *Transportation Research Record*, no. 2249, pp. 44–51, 2011.
- [19] H. Jia, Z. Juan, and A. Ni, “Develop a car-following model using data collected by ‘five-wheel system,’” in *Proceedings of the Intelligent Transportation Systems*, IEEE, 2003.
- [20] S. Panwai and H. Dia, “Neural agent car-following models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 60–70, 2007.
- [21] L. Zhang, K. Zhou, W.-B. Zhang, and J. A. Misener, “Prediction of red light running based on statistics of discrete point sensors,” *Transportation Research Record*, vol. 2128, pp. 132–142, 2009.
- [22] P. Li and M. Abbas, “Stochastic dilemma hazard model at high-speed signalized intersections,” *Journal of Transportation Engineering*, vol. 136, no. 5, pp. 448–456, 2010.
- [23] Wavetronix LLC, Wavetronix SmartSensor, 2014, <http://www.wavetronix.com>.
- [24] NTCIP Committee, *The National Transportation Communications for ITS Protocol Online Resources*, 2012, <http://www.ntcip.org/>.
- [25] SmartMicro, *Smart Microwave Sensors*, 2012, <http://www.smartmicro.de/>.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

