

Article

# Accident Prediction System Based on Hidden Markov Model for Vehicular Ad-Hoc Network in Urban Environments

Nyothiri Aung<sup>1</sup>, Weidong Zhang<sup>2,\*</sup>, Sahraoui Dhelim<sup>1</sup> and Yibo Ai<sup>2</sup>

<sup>1</sup> School of Computer Science and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China; nyothiriaung@xs.ustb.edu.cn (N.A.); sahraoui.dhelim@xs.ustb.edu.cn (S.D.)

<sup>2</sup> National Center for Materials Service Safety, University of Science and Technology Beijing, Beijing 100083, China; ybai@ustb.edu.cn

\* Correspondence: zwd@ustb.edu.cn

Received: 22 October 2018; Accepted: 5 December 2018; Published: 7 December 2018



**Abstract:** With the emergence of autonomous vehicles and internet of vehicles (IoV), future roads of smart cities will have a combination of autonomous and automated vehicles with regular vehicles that require human operators. To ensure the safety of the road commuters in such a network, it is imperative to enhance the performance of Advanced Driver Assistance Systems (ADAS). Real-time driving risk prediction is a fundamental part of an ADAS. Many driving risk prediction systems have been proposed. However, most of them are based only on vehicle's velocity. But in most of the accident scenarios, other factors are also involved, such as weather conditions or driver fatigue. In this paper, we proposed an accident prediction system for Vehicular ad hoc networks (VANETs) in urban environments, in which we considered the crash risk as a latent variable that can be observed using multi-observation such as velocity, weather condition, risk location, nearby vehicles density and driver fatigue. A Hidden Markov Model (HMM) was used to model the correlation between these observations and the latent variable. Simulation results showed that the proposed system has a better performance in terms of sensitivity and precision compared to state of the art single factor schemes.

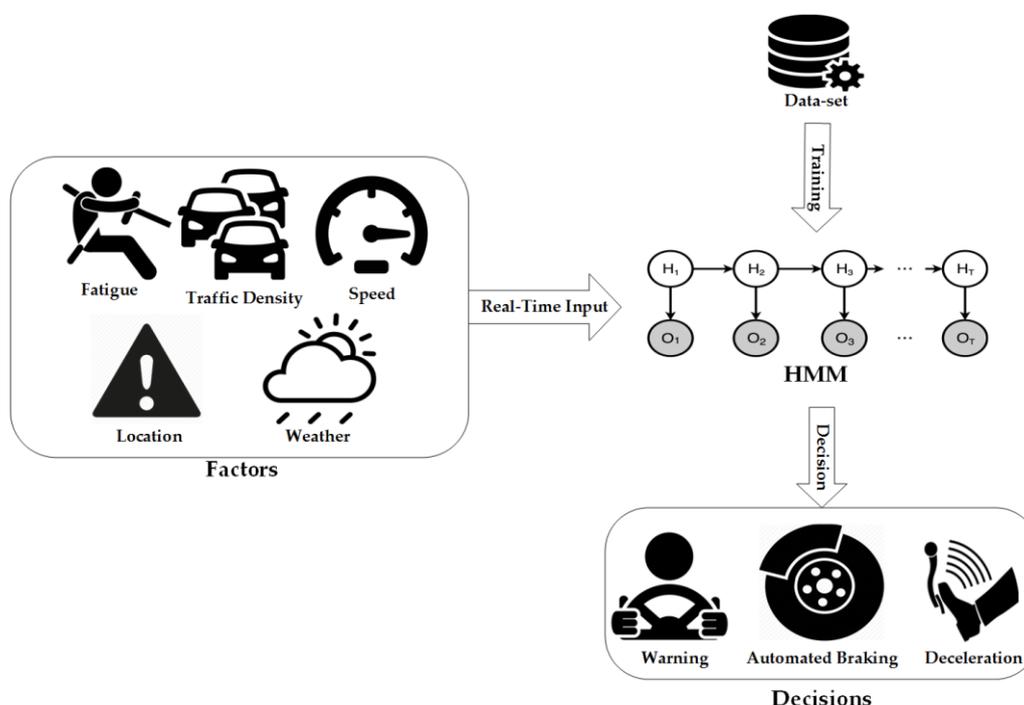
**Keywords:** accident prediction system; driver assistance system; hidden markov model; VANET; ITS; HMM; ADAS

## 1. Introduction

Besides its enormous economic losses, according to World Health Organization (WHO) 1.24 million people die, and tens of millions are injured every year as a result of road traffic accidents [1], which will make road crashes the 7th leading cause of death by 2030 [2].

An accident prediction system (APS) is an automobile safety system that is used to detect any potential collision: it either warns the driver or acts autonomously by steering or braking without the driver's intervention in case of an imminent collision. There are two types of APS(s), hardware-based and software-based; the former one is realized by placing a variety of sensors and cameras in the vehicles, based on the sensor's aggregated data the potential crash is predicted, while the latter's decision is based on the predictions of a software. Vehicular ad hoc networks (VANETs) have gained considerable attention in the past few years. VANET is a network of connected vehicles, in which vehicles are equipped with wireless communication devices that enable them to drive collaboratively and exchange short messages. VANET has many traffic-related applications such as traffic control [3], congestion avoidance [4] and parking lot management [5]. In the literature of VANET, many APS(s) have been proposed. Most of these are solely based on the vehicle's velocity. However, the weather

conditions, driver fatigue and other factors also need to be considered in crash scenarios. The Hidden Markov Model (HMM) has been applied in a wide variety of areas, such as speech recognition, handwriting recognition, activity recognition and vehicle route prediction. In this paper, we proposed a new APS for VANET in an urban environment in which we considered the crash risk as a latent variable that can be observed using multi observations such as velocity, weather, crash location, nearby vehicles density and driver fatigue. HMM was used to model the correlation between these observations and the crash risk, see Figure 1. HMM is well suited for such scenarios due to its efficient learning, as the model learning takes place directly from the raw sequence data. As far as we know, our system is the first that incorporates many crash-related factors to model the crash scenario using multilayered HMM. Aiming to give a holistic representation of the accident, the crash factors were represented as HMM observations, and the risk level as HMM states. We trained the model based on an accidents data-set, as well as other driving scenarios. Simulation results showed that the proposed system has a better performance in terms of sensitivity and precision compared to state of the art single factor schemes.



**Figure 1.** The proposed system.

Our contributions can be summarized as follows:

- We proposed a new APS based on VANET and HMM, in which the crash risk was considered as a latent variable.
- Unlike other schemes, besides the velocity, the proposed system also considers other factors that may cause the crash.
- The proposed system was modeled as a weighted multi-observation layer HMM rather than the conventional signal layer HMM.
- The proposed system was validated by means of extensive simulation on a map of London city.
- Simulation results show the high sensitivity and precision of the proposed system.

The rest of the paper is organized as follows:

Section 2 highlights the related works in the recent years, Section 3 presents preliminaries about HMM, Section 4 sets up the theoretical foundation of the proposed system, whereas Section 5 gives

details about the implementation of the system. In Section 6, we discuss the performance evaluation and simulation results; we conclude the paper and present future directions in Section 7.

## 2. Related Work

Some early works have presented a mathematical model that expressed the relationship between accident frequencies in a given site as a function of traffic flow and other location characteristics [6–8]. Many APSs have been proposed in the last few years. The works in [9–11] surveyed the recent techniques in the literature of collision prediction and avoidance using VANET. However, most of the state-of-the-art schemes are based only on the vehicle's velocity, or another factor solely. In this section, we will present some of the recently proposed APSs.

### 2.1. Velocity Based Approaches

Sam et al. [12] proposed a VANET-based vehicle control system. The system was implemented by wirelessly connecting the vehicle nodes, pedestrian body units and the roadside units. Based on the current speed of the vehicle and pedestrians' positions, the system can detect potential accidents and act accordingly. In [13], an urban traffic accident analysis was done using support vector machines (SVM) with Gaussian kernel. The proposed system considers two major factors to predict the collision, speed and alcohol level. Similarly, in [14], a method for early car accident detection was proposed. This system used previous velocity data with a support vector machine (SVM) to predict accidents.

### 2.2. Traffic Density Based Approaches

In [15], a collision warning that avoids traffic rear-ends accidents was proposed, in which all the vehicles were embedded with sensing and communication models. The sensing module can help vehicles sense the surrounding environment and detect nearby vehicle traffic density to predict crashes yielded by sudden braking. Iqbal et al. [16] proposed a crash detection system to prevent and mitigate adjacent vehicle collisions by providing warning information of on-road vehicles and possible collisions. A dynamic Bayesian network (DBN) was used to fuse multiple sensors' data. By using an on-board camera, the system gathered the ego-motion characteristics of the oncoming vehicles. The ego vehicle's characteristics, such as speed, acceleration and heading, were captured using the inertial measurement unit (IMU) that was integrated with the camera, and subsequently, fed into a gradient feature based classifier. In [17], big data analysis was used to analyze traffic data and ultimately predict crashes. This was done by dividing the roadway into segments. They designed a real-time Big Data analysis system that received online streamed data from vehicles on the road in addition to real-time average speed data from vehicles detectors on the roadside. The system predicted accidents before they happened using Naive Bayes (NB) and Distributed Random Forest (DRF) classifiers. Moreover, the system streamed the data for each segment along with other features, such as number of vehicles.

### 2.3. Driver Fatigue Based Approaches

Detecting the driver's physical and mental fatigue is a critical factor for ADS development. In [18], Arbabzadeh and Jafari presented a data-driven prediction model that could predict various traffic risks. The proposed model could be customized by including the driver's specific variables. In order to build a predictive model, the authors used the elastic net regularized multinomial logistic regression and applied it on data obtained from the Second Strategic Highway Research Program (SHRP 2). The work in [19] presented a real-time driving fatigue detection methodology based on Electroencephalographic (EEG) signals; the objective of such a system was to detect mental fatigue, thus preventing possible crashes. In [20], a crash prediction system is presented: the proposed system's decisions were based on the driver's behavioral and physiological features.

### 2.4. Location Based Approaches

To address intersection crash scenarios, Maile et al. [21] presented an intersection collision avoidance system based on Dedicated Short Range Communications (DSRC). The automated braking system was activated when a high risk of a potential accident was detected and the driver did not react to the displayed alert. Akhil et al. [22] proposed an accident detection system at intersections. This system was implemented by installing vehicle detectors near the intersections: when the vehicle crosses the intersection, it captured the detector’s ID along with other relevant kinematic data and passed it to the smartphone, which in turn forwarded it to the base station. Based on that, the base station predicted any potential crashes and warned the driver through their smartphone.

### 2.5. Weather Based Approaches

In [23], Bordel et al. defined a model that predicted the accidents in mountain roads by analyzing the sensed weather data obtained from the weather forecasting office. The proposed model was based on Taylor’s series and multivariate functions. The authors validated their model by applying it to real data obtained from Valais city’s road network. Lu et al. [24] proposed a traffic accident prediction model that used weather conditions to create a state matrix that represents the traffic state. The model was based on the convolutional neural network. In [24], a new road traffic accident prediction model (TAP-CNN) was established by using traffic accident influencing factors, such as traffic flow, weather and light to build a state matrix to describe the traffic state and CNN model.

## 3. Preliminaries

HMM is a statistical model that is used to model time series data, in which the current state of the system is hidden and follows the Markovian process. Usually, this model is applied when we can only get observations related to the current state of the system, rather than visible system status like simple Markov models. Figure 2 shows an example of the graphical representation of an HMM. HMMs mainly consist of two parts: a set of states and a set of observations. In Figure 2, the states are denoted by  $(Q_1, Q_2, Q_3)$ , and the observations by  $(O_1, O_2, O_3, O_4)$ .

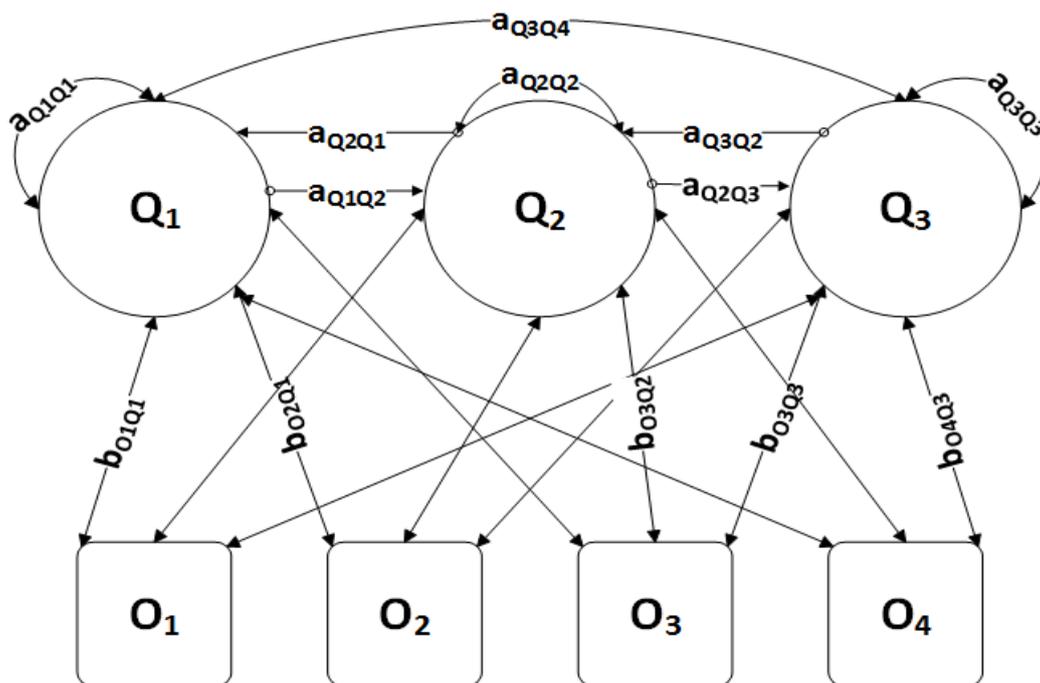


Figure 2. Example of Graphical Representation of Hidden Markov Model (HMM).

### 3.1. Notations

In the literature about HMM, different works used different notations of the HMM’s parameters. For the sake of readability we have listed all the notations that we will use later in this paper in Table 1.

**Table 1.** Notation.

Symbol	Description
$\pi$	Initial state distribution
$A$	State transition probability matrix (transition matrix)
$a_{i,j}$	The probability of being in state $i$ at the time $t$ and transfer to state $j$ at the time $t+1$
$B$	Observation probability matrix (emission matrix)
$b_{i,j}$	The probability of being in state $j$ during the observation $O_i$
$\lambda(\pi, A, B)$	The HMM parameters
$O = (O_0, O_1 \dots O_t)$	The observed sequence
$T$	Length of the observation sequence
$N$	Number of states in the model
$M$	Number of observations

### 3.2. Observation Evaluation

Given a model  $\lambda$  and an observation sequence  $O$ , the observation evaluation problem is to calculate  $P(O|\lambda)$ , that is to determine the probability of observing the sequence  $O$  given the model  $\lambda$ . It can be solved using the forward-backward procedure described in the next section.

#### 3.2.1. Forward Procedure

The so-called forward procedure or  $\alpha$ -pass is used to determine the likelihood of a given observation sequence. In other words, the probability of being in a given state at a certain time, given the history of evidence:

$$\alpha_t(i) = P(O_0, O_1, \dots, O_t, x_t = q_i | \lambda) \tag{1}$$

where:  $x_t$  is the current state. and  $\alpha_t(i)$  is the probability of seeing the sequence  $O = (O_0, O_1 \dots O_t)$  and being in state  $q_i$  at time  $t$ . The crucial point here is that  $\alpha$ -pass can be calculated recursively, as shown in Algorithm 1:

**Algorithm 1** Forward Procedure

---

```

for  $i = 0$  to  $N - 1$  do
     $\alpha_0(i) \leftarrow \pi_i b_i(O_0)$ 
end for
for  $i = 0$  to  $N - 1$  do
    for  $t = 1$  to  $t = T - 1$  do
         $\alpha_t(i) = \left[ \sum_{j=0}^{N-1} \alpha_{t-1}(j) a_{ji} \right] b_i(O_t)$ 
    end for
end for

```

---

#### 3.2.2. Backward Procedure

The backward procedure or  $\beta$ -pass is similar to the  $\alpha$ -pass discussed above. The only difference between them is that  $\beta$ -pass starts at the end and works back toward the beginning:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_{T-1} | x_t = q_i, \lambda) \tag{2}$$

It can also be computed recursively (and efficiently), as shown in Algorithm 2.

In short,  $\alpha_t(i)$  calculates the relevant probability up to time  $t$ , and  $\beta_t(i)$  calculates the relevant probability after time  $t$ .

---

**Algorithm 2** Backward procedure

---

```

for  $i = 0$  to  $N - 1$  do
     $\beta_{T-1}(i) \leftarrow 1$ 
end for
for  $i = 0$  to  $N-1$  do
    for  $t = T - 2$  to  $0$  do
         $\beta_t(i) = \sum_{j=0}^{N-1} a_{ij}b_j(O_{t+1})\beta_{t+1}(j)$ 
    end for
end for
    
```

---

**4. System Modeling**

As mentioned earlier, the proposed APS was modeled as an HMM, where the states represent the crash risk level and the HMM observations represent the crash factors.

*4.1. HMM Parameters*

The proposed collision avoidance scheme was modeled as an HMM, in which the hidden states represent the risk levels of a potential crash and the observation space depended on different observation factors. A single element in the observation space is a combination of different factors, these factors, when combined together, depict a holistic representation of the crash environment at a given time, see Table 2.

**Table 2.** Observation factors.

Factor	Value Range
States	Negligible, Low, Moderate, High, Very high, Deadly
S	Very Slow, Slow, Medium, High, Very high, Extreme
L	Safe, Normal, Dangerous, Deadly
W	Clear, Sunny, Rainy, Foggy, Snowing
V	Low, Medium, High
D	Fresh, Medium, Tired

Where S, L, W, V and D stand for current vehicle Speed, current Location crash risk, Weather condition, Vehicles density and Driver fatigue, respectively. A single element  $O_x$  in the observation space was defined by the five components S, L, W, V and D, as shown in (3).

$$O_x = \{S, L, W, V, D\} \tag{3}$$

Example:

$$O_2 = \{S = \textit{Medium}, L = \textit{Safe}, W = \textit{Rainy}, V = \textit{High}, D = \textit{Tired}\}$$

In the example, the observation  $O_2$  takes a place when the current speed is medium, the current location never witnessed a crash according to the previous statistics, the weather is rainy, the density of nearby vehicles is high and the driver is tired. The graphical representation of the HMM is shown in Figure 3. After listing all the elements yielded from the combination of observation factors, the size of the observation space is 1080 observations, and they form a JEPD (Jointly Exhaustive Pairwise Disjoint)

group. During its journey in the urban environment, a vehicle can match only one observation from the total 1080 observations at a given time. After a change of one of the five factors, the vehicle will switch to the new corresponding observation.

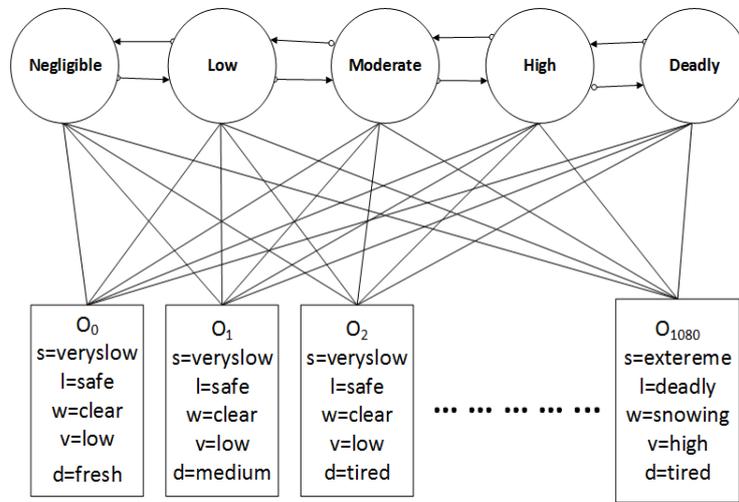


Figure 3. Graphical representation of HMM after listing all the observations combination.

#### 4.2. Probability Fusion

Each observation factor has its separate emission probability matrix, which we denote as  $B_S$ ,  $B_L$ ,  $B_W$ ,  $B_V$  and  $B_D$ . They represent the emission matrix of Speed, Location, Weather, Vehicles density and Driver fatigue level, respectively.

$$B_S = \begin{pmatrix} P_{\{verySlow,Negligible\}} & \cdots & P_{\{verySlow,Deadly\}} \\ \vdots & \ddots & \vdots \\ P_{\{Extreme,negligible\}} & \cdots & P_{\{Extreme,Deadly\}} \end{pmatrix}$$

$$B_L = \begin{pmatrix} P_{\{Low,Negligible\}} & \cdots & P_{\{Low,Deadly\}} \\ \vdots & \ddots & \vdots \\ P_{\{High,Negligible\}} & \cdots & P_{\{High,Deadly\}} \end{pmatrix}$$

$$B_W = \begin{pmatrix} P_{\{Clear,Negligible\}} & \cdots & P_{\{Clear,Deadly\}} \\ \vdots & \ddots & \vdots \\ P_{\{Snowing,Negligible\}} & \cdots & P_{\{Snowing,Deadly\}} \end{pmatrix}$$

$$B_V = \begin{pmatrix} P_{\{Low,Negligible\}} & \cdots & P_{\{Low,Deadly\}} \\ \vdots & \ddots & \vdots \\ P_{\{High,Negligible\}} & \cdots & P_{\{High,Deadly\}} \end{pmatrix}$$

$$B_D = \begin{pmatrix} P_{\{Fresh,Negligible\}} & \cdots & P_{\{Fresh,Deadly\}} \\ \vdots & \ddots & \vdots \\ P_{\{Tired,Negligible\}} & \cdots & P_{\{Tired,Deadly\}} \end{pmatrix}$$

where:

$$\sum_{j=0}^{|B_x|} B_x(i, j) = 1 | x \in \{S, L, W, V, D\} \tag{4}$$

After calculating the separated emission matrices, the weighted mean formula in (5) as applied to get the fused probability.

The weights were calculated according to the previous statistics about accidents causes. Each weight will determine the impact of the current factor on the overall observation. The more the factor is involved as one of the causes of the accident according to the previous traffic collision reconstruction statistics, the more its weight increases (see Section 5 for more about weights estimations):

$$B_{O_x S_j} = (W_s B_s[S_x, j]) + (W_L B_L[L_x, j]) + (W_w B_w[W_x, j]) + (W_v B_v[V_x, j]) + (W_D B_D[D_x, j]) \quad (5)$$

where:

$$O_x = \{S_x, L_x, W_x, V_x, D_x\}$$

And where  $W_s, W_L, W_W, W_V$  and  $W_D$ , are the weights of each factor.

### 4.3. Training HMM

To train the model, the Baum-Welch algorithm [25] was used to determine the HMM parameters  $\lambda(\pi, A, B)$  given a sequence of observations. The following inputs were fed into the Baum-Welch algorithm to measure the HMM parameters: (1) the extracted information from [26] accidents dataset that contained accident information (for more details, see Section 5); (2) the observed sequences of each vehicle from the simulation logs; each vehicle recorded its current observation during its journey. The sequence of the observations was recorded in the vehicle’s log along with the corresponding timestamp. After the initial estimation of the HMM parameters, the Baum-Welch algorithm refined these estimations every time it read an input from the observation sequence. It did that using the  $\alpha$ -pass and  $\beta$ -pass procedures that we have defined in Section 3. In this context, we define another two variables:  $\gamma_t(i)$  (see (6)) and  $\varepsilon_t(i, j)$  (see (7)), used in the Baum-Welch algorithm:

$$\gamma_t(i) = P(X_t = q_i | O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=0}^{N-1} \alpha_t(j)\beta_t(j)} \quad (6)$$

where  $\gamma_t(i)$  is the probability of being in state  $q_i$  at time  $t$  given the observed sequence  $O$  and with the parameters  $\lambda(\pi, A, B)$ .

$$\varepsilon_t(i, j) = P(X_t = q_i, X_{t+1} = q_j | O, \lambda) = \frac{\alpha_t(i)a_{i,j}\beta_{t+1}(j)b_j(O_{t+1})}{\sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \alpha_t(i)\beta_{t+1}(i)b_j(O_{t+1})} \quad (7)$$

$\varepsilon_t(i, j)$  is the probability of being in state  $q_i$  and  $q_j$  at time  $t$  and  $t + 1$  respectively given the observed sequence  $O$  and the parameters  $\lambda(\pi, A, B)$ .

Using  $\gamma_t(i)$  and  $\varepsilon_t(i, j)$ , the parameters  $\lambda(\pi, A, B)$  were updated every time we parsed a new element from  $O$ ; we denoted the updated parameters as  $\lambda(\pi^*, A^*, B^*)$ , see (8), (9) and (10):

$$\pi_i^* = \gamma_1(i) \quad (8)$$

$$a_{i,j}^* = \frac{\sum_{t=1}^{T-1} \varepsilon_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (9)$$

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T-1} \delta \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad (10)$$

where:

$$\delta = \begin{cases} 1 & \text{if } y_t = v_k \\ 0 & \text{otherwise} \end{cases}$$

## 5. Implementation

In this section, we present the training data-set details and the simulation of the proposed system:

### 5.1. Training Data-Set

To train our system, we have used the governmental public dataset in [26], which provided detailed data about the circumstances road accidents that took place in United Kingdom during the period from 1/1/1979 until 31/12/2016. However, since the road infrastructure changes frequently, we only used the data from 1/1/2005 until 31/12/2016. A total of 1,325,255 accidents took place during the span of 11 years; the dataset provided information about the accidents' circumstances, in the following form:

General data: general information about the accident circumstances such as:

- Accident Severity (fatal, serious, slight).
- Accident time (time, date, day of the week,)

Speed data:

- Estimated speed during the accidents
- Whether the road limit was exceeded
- Road speed limit

Location data:

- Coordinates (Latitude, Longitude)
- Grid reference coordinates (Location Easting OSGR, Location Northing OSGR)
- Weather Conditions (fine no high winds, raining no high winds, snowing no high winds, fine and high winds, raining and high winds, snowing and high winds, fog or mist . . . etc.)
- Light Conditions (daylight, darkness-lights lit, darkness-lights unlit, darkness-no lighting, darkness-lighting unknown)
- Road surface (dry, wet or damp, snow, frost or ice, flood over 3cm' deep', oil or diesel, mud)
- Road Type (roundabout, dual carriageway . . . etc.)

Driver data: information about the driver:

- Driver's age
- Journey purpose of driver
- Driver blood alcohol level
- Driver's health condition

Vehicles data: information about the vehicles that were involved in the accident:

- Number of vehicles involved in the accident
- Vehicle type and propulsion code
- Vehicle reference and engine capacity

Casualty data: information about the accident casualty:

- Casualty severity
- Casualties ages
- Casualty type

for a complete list of the available information in the dataset see [26].

### 5.2. Simulation Map

The used map to conduct the simulation was obtained from OpenStreetMap [27] (<https://www.openstreetmap.org>). We extracted an area of 3X4 km from London city as shown in Figure 4. The raw map data was downloaded in the form of OpenStreetMap file format (osm), which contained all the information about the map’s road network (road type, speed limitation and other details). As shown in Figure 5, the road network file and the simulated traffic were obtained by applying utilities of SUMO (Simulation of Urban Mobility) [27] (mainly netconvert, polyconvert and randomTrips.py). For more details about traffic generation see [28].

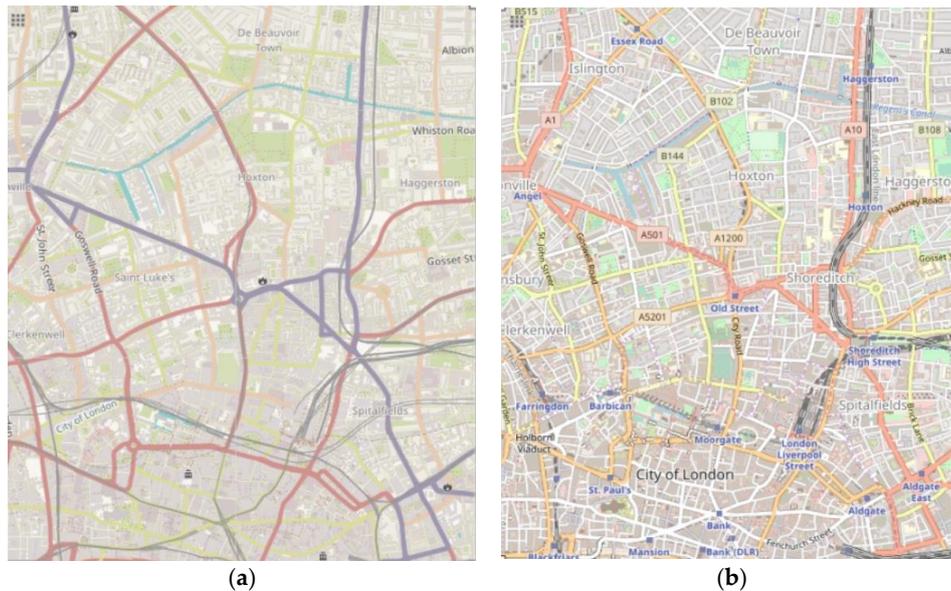


Figure 4. The simulated map: (a) transportation view (b) full view.

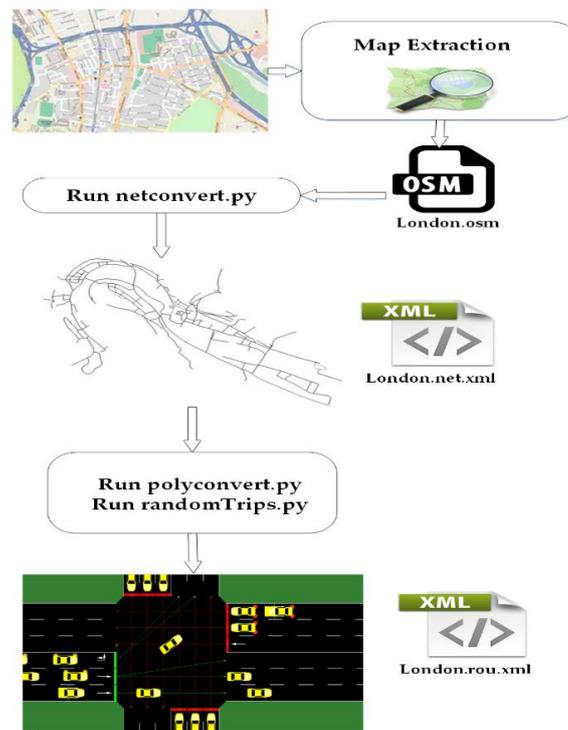


Figure 5. Trip generation using SUMO and OSM.

### 5.3. Training the System

#### 5.3.1. Ranges Mapping

The dataset values were mapped to our model's values range:

**Velocity:** different roads and even different lanes have a different speed limit. Therefore, the speed was mapped in accordance with the current lane's maximum speed limit specified in the dataset. For example, if the maximum speed limit of the current lane was 100 km/h the current speed was within 0–20 km/h, and was therefore considered to be very slow, as we have chosen 5 values for the speed (very slow, slow, medium, high and very high). If the speed was within 20–40 km/h, it was classified as slow and so on and so forth.

**Location:** As most of the crashes usually take place in the intersections, the intersections were regarded as distinct locations. Other than that, two accidents were said to take place in the same location if the distance between their coordinates was under 10 m and the coordinates were within the same lane.

**Driver fatigue:** In real scenarios, driver fatigue status could be obtained by connecting the driver's Body Area Network (BAN) sensors with the accident prediction system. Other methods could also be used, such as analyzing eye movement to detect the level of sleepiness. However, in our case, we assumed that driver fatigue was directly related to driving time; thus, the longer the trip (driving time) in the simulation, the more tired the driver.

#### 5.3.2. Weights Optimization

As mentioned in Section 3, each factor was associated with a weight that will determine its importance compared to other factors.

To determine their values, we extracted the percentage of each accident cause from the dataset of all accidents that lie within the simulation area. The more the factor caused crashes, the more its weight increased. After each iteration, the weights combination was optimized accordingly. The optimized weights are presented in Table 3.

**Table 3.** Observation factors weights.

Factor (Reason)	Weight (Percentage)
Vehicle speed ( $W_S$ )	48.3%
Location dangerous level ( $W_L$ )	12.9%
Weather conditions ( $W_W$ )	18%
Vehicle density ( $W_V$ )	15.8%
Driver fatigue ( $W_D$ )	5%

### 5.4. Traffic Simulation

Traffic was generated using the SUMO traffic simulator, while Traci [29] was used to synchronize the generated traffic by SUMO and the vehicular behavior in Veins [28], details of the simulation tools are shown in Table 4. As mentioned earlier, a road network from London city was used in the simulation. However, since the road network was extracted from a real map, we could not simulate high velocity (100–150 km/h) on most of the lanes, as they have a speed limit. Therefore, we changed the road properties using SUMO to allow such high velocities. The vehicles started their trips from random starting entries. For each density, vehicles were generated until the required density was reached. Each vehicle broadcasted beacon message regularly to announce its presence. The message contained the vehicle's GPS coordinates. Every vehicle maintained a linked list of its current neighbors (nearby vehicles) and the distance that separated it from each neighbor. When a neighbor vehicle leaves its communication range, that vehicle's ID was removed from the neighbor list (see Figure 6). The presented results in Section 6 are the average of 100 simulation runs for each value.

Table 4. Simulation tools.

Parameters	Description
Network Simulator	Omnet++ 5 [30]
Traffic Simulator	Sumo 0.27.1 [28]
Map Information	Openstreetmap [27]
Simulated Location	London
Simulated area	3X4 km

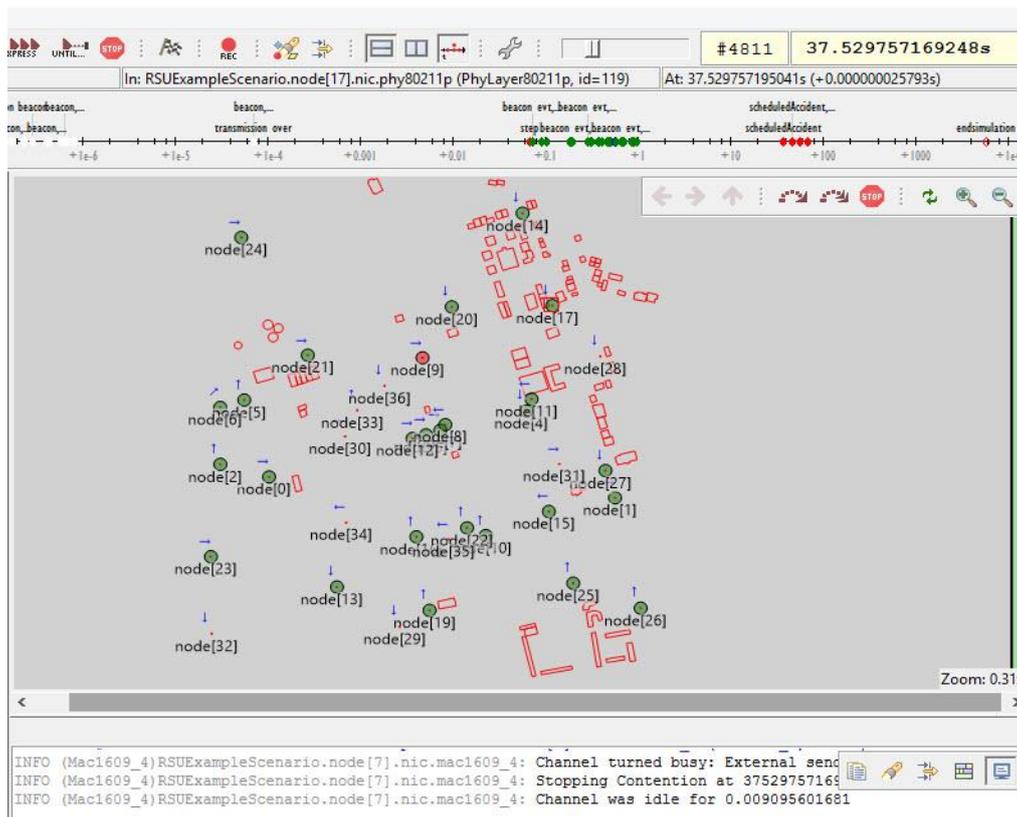


Figure 6. Simulation visualization on Omnet++ IDE.

### 5.5. V2V Communication Parameters

The wireless communication model parameters are detailed in Table 5.

Table 5. Wireless Communication Parameter.

Parameter	Value
PHY model	802.11 p
Channel frequency	5.890e9 Hz
Propagation model	Two ray model
MAC model	EDCA
Propagation distance	150 m
Maximum hop count	10
Fading model	Jakes model rayleigh fading
Shadowing model	LogNormal
Antenna model	Omnidirectional
Transmission power	20 mW

## 6. Performance Evaluation

In order to test the effectiveness of the proposed scheme, we measured its ability to detect potential crashes. The overall rating of the system depended on the sensitivity (11) (true positive rate) and precision (12) (positive predictive value); the former was used to measure the proportion of positive alarms that were correctly identified, while the latter as used to measure the proportion of negatives that were correctly identified. Technically speaking, the scenario manager scheduled accidents and we measured the ability of the individual nodes to detect this potential crash and cooperatively warn each other.

$$Sensitivity = \frac{TP}{(TP + FN)} \quad (11)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (12)$$

$$Performance = \frac{(Sensitivity + Precision)}{2} \quad (13)$$

- True positive (TP): the scenario manager launched the crash and the observed vehicle could detect it.
- False positive (FP): the scenario manager did not launch the crash but the observed vehicle falsely detected a crash.
- False negative (FN): the scenario manager launched the crash but the observed vehicle did not detect it.

### 6.1. Metrics

- Velocity vs Sensitivity: in this test, we changed the velocity values to test and compared the sensitivity of the schemes.
- Velocity vs Precision: in this test, we changed the velocity values to test and compared the precision of the schemes.
- Velocity vs Performance: to measure the performance of the scheme when changing the velocity value.
- Density vs Sensitivity: in this test, we changed vehicles density to test and compared the sensitivity of the schemes.
- Density vs Precision: in this test, we changed vehicles density to test and compared the precision of the schemes.
- Density vs Performance: to measure the performance of the schemes when changing the vehicles density.

### 6.2. Baselines

Since many of the state-of-the-art ADS(s) are essentially based on the velocity and vehicles' density to detect a potential crash, the following systems were compared in term of performance:

**Velocity-Based:** Collision avoidance system based only on vehicles' velocity, such as the one proposed by Sam et al. [12], where the vehicles (also can be applied with pedestrians) contently broadcast their position to nearby vehicles. Any potential crash is detected based on the estimated speed (by computing GPS coordinates) and vehicles direction.

**Density-Based:** Collision avoidance system based only on nearby vehicles density, such as the one proposed by Lv et al. [15], where rear-end accidents is detected by measuring the distance between vehicles based on frequently exchanged bacon messages.

**HMM-Mean:** The proposed system, but with equal factors weights: 25% for each factor.

**HMM-Optimized:** The proposed system, but with optimized factors weights as shown in Section 5.

### 6.3. Simulation Results

Figure 7 shows how the four systems (HMM-Mean, HMM-Optimized, Sam et al. and Lv et al.) responded in terms of sensitivity when the velocity increases within the range of 20 to 150 km/h. The system from Lv et al. had the worst sensitivity, because it is based only on the traffic density. Thus it failed to detect any accident related to the velocity increase. All of the three other systems' sensitivity was directly proportional to the velocity increase but in different rates, from 20 to 70 km/h. HMM-Optimized obviously performed better than HMM-Mean and Velocity-Based; the latter one sensitivity was the lowest one because it considers velocity only and no other factors, and at this level (20–70 km/h) the accident cause is due to other factors than speed in most cases. However, when the velocity exceeded 70 km/h, its performance surpassed the other two systems and became slightly better than HMM-Optimized and much better than HMM-Mean, which had the worst performance compared to HMM-Optimized and Sam's scheme in the range of 70 to 150 km/h. Although HMM-Mean considers many factors, it had a low sensitivity because it had equal weights for all the factors. Thus, it neglected the high velocity that eventually causes the crash and treats it like any other factor.

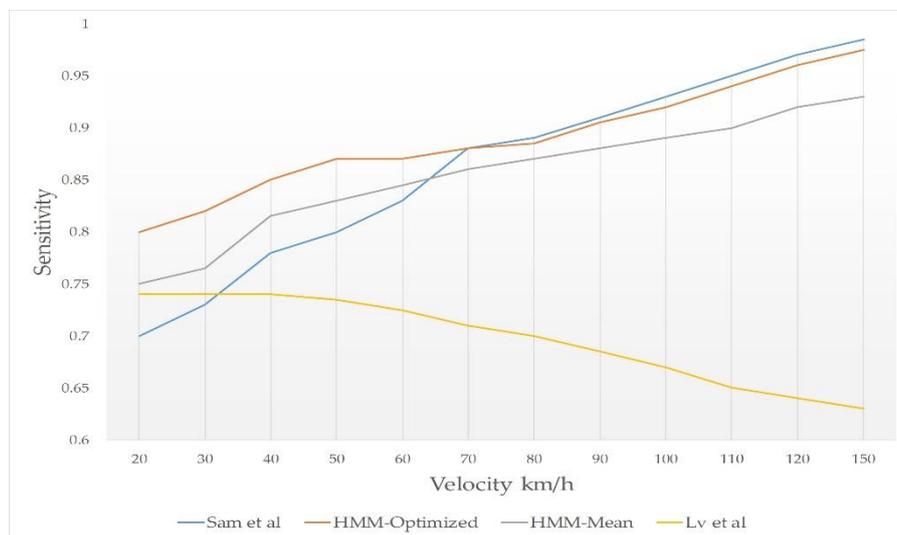


Figure 7. Velocity versus Sensitivity.

However, the cost of the high sensitivity that the velocity-based system provides is its low precision, which became obvious within the range of 70 to 150 km/h in Figure 8. As shown in Figure 8, Sam's scheme had the worst precision compared to HMM-Optimized, which had the highest precision. That is because velocity-based systems generated more FPs at high speeds. In other words, the more the speed increases, the more it generates alerts, regardless of the accuracy of these alerts. The overall performance, which is the average of sensitivity and precision, is presented in Figure 9, from which we can observe that HMM-Optimized surpasses HMM-Mean and Sam's scheme all over the graph for all values (20–150 km/h), while the other two, interchangeably compete with only slight differences.

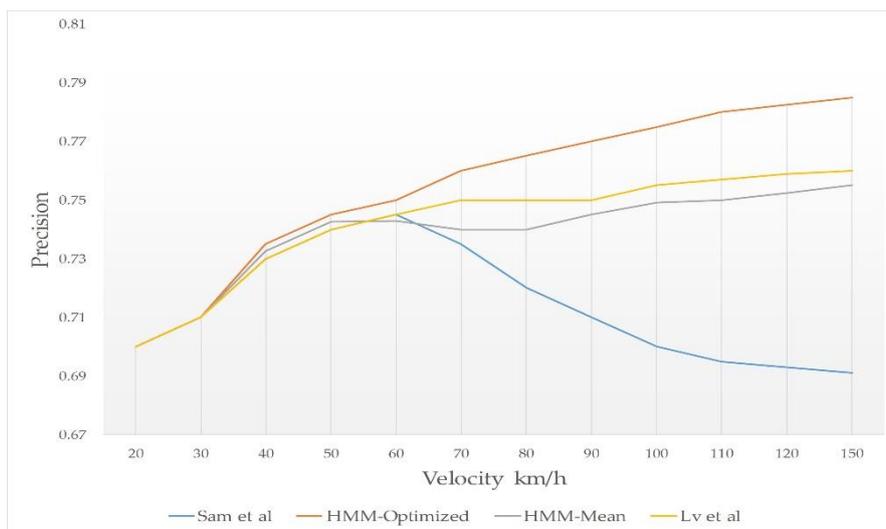


Figure 8. Velocity versus Precision.

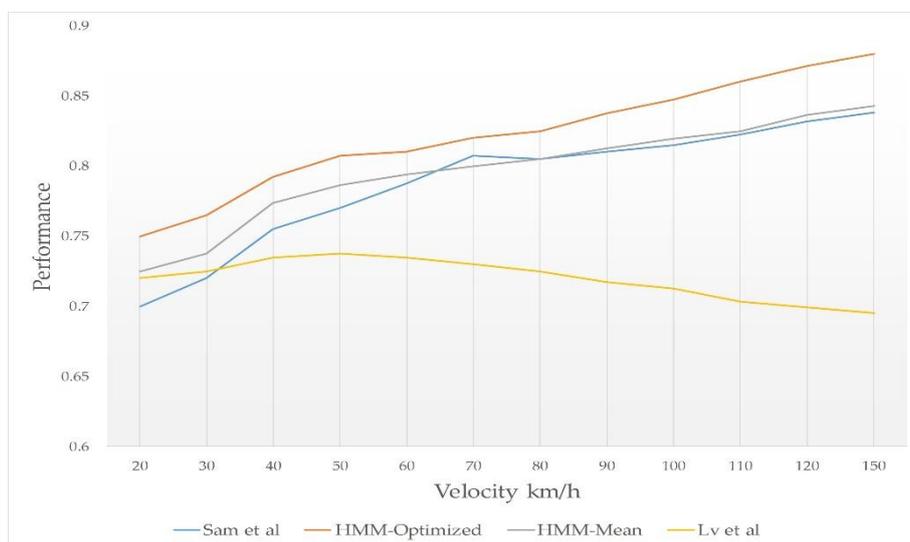


Figure 9. Velocity VS Performance.

Figures 10–12 present the result of the four schemes in term of sensitivity, precision and performance, respectively, with different vehicle densities (20–5000 vehicles/km<sup>2</sup>). In Figure 10, with the exception of the system from Sam et al., all of the other three systems increased in terms of sensitivity within the range of 20 to 300 vehicles/km<sup>2</sup>. HMM-Optimized had higher sensitivity (7.2 to 7.55) compared to HMM-Mean and Density-Based that had (0.685–0.74) and (0.659–0.75), respectively, while with high-density values (400–5000 vehicles/km<sup>2</sup>) Lv’s scheme surpassed HMM-Mean with a relatively big sensitivity difference, and also surpassed HMM-Optimized with a slight sensitivity difference. However, the superiority of Lv’s scheme in terms of sensitivity came at the cost of its low precision which is obvious in Figure 11. Despite its good precision within low dense environments (20–70 vehicles/km<sup>2</sup>), its precision dropped dramatically in highly dense environments (200–5000 vehicles/km<sup>2</sup>). That is because it generated more FP alerts at high density regardless of other factors. HMM-Optimized, on the other hand, maintained a stable precision in different densities. HMM-Mean had medium precision, as it considered other factors but with equal weights. In Figure 12, the overall performance is presented: we can clearly observe the robustness of our proposed system compared to its counterparts.

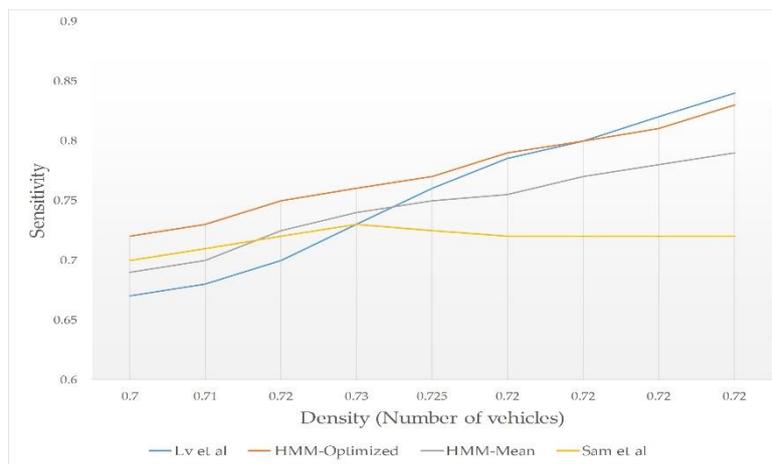


Figure 10. Density versus Sensitivity.

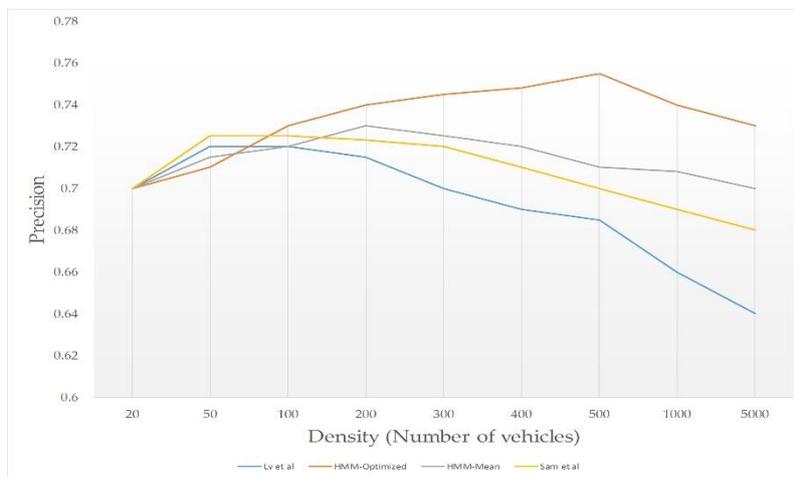


Figure 11. Density versus Precision.

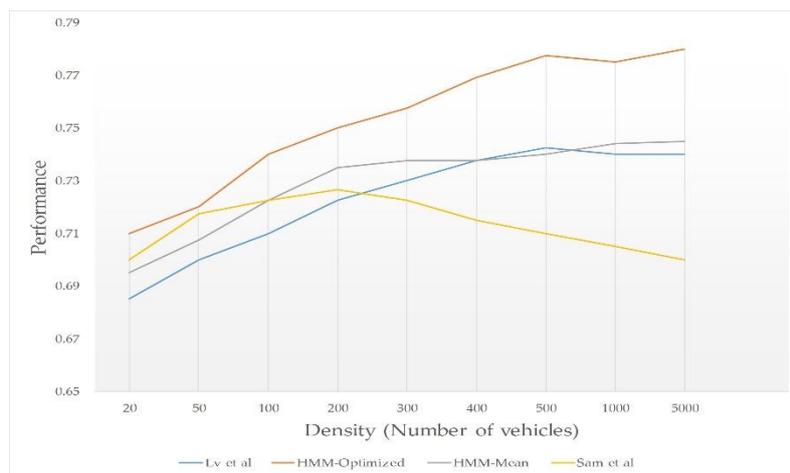


Figure 12. Density versus Performance.

## 7. Conclusions and Future Directions

In this paper, we proposed a new accident prediction system for VANET in urban environments, in which we have considered the risk of the crash as a latent variable that can be observed using multi-observation such as velocity, weather conditions, crash location, nearby vehicles density and

driver fatigue. A Hidden Markov Model was used to model the correlation between these observations and the latent variable. HMM is well suited for such scenarios due to its efficient learning, as the model learning takes place directly from the raw sequence data. The obtained results confirmed the effectiveness and robustness of the proposed scheme. The effectiveness of the scheme was tested based on its ability to detect potential crashes and the accuracy of its decisions. However, we did not address the case where these decisions are inaccurate, such as false positive alerts. Therefore, developing methods that can mitigate false positive decisions is our future direction.

**Author Contributions:** N.T., W.D.Z. and Y.B.A. conceived and designed the system; S.D. and N.T.A. Analyzed the data-set, mapped the values and performed the simulations. All the authors contributed in manuscript preparation.

**Funding:** This work is supported by the National Natural Science Foundation of China (Grant No 61273205), and the Fundamental Research Funds for Central Universities of China (No. FRF-BD-18-001A).

**Acknowledgments:** The authors would like to express their great gratitude to the anonymous reviewers that help to improve the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. World Health Organization. *Global Status Report on Road Safety: Supporting a Decade of Action*; World Health Organization: Geneva, Switzerland, 2013.
2. WHO Road Traffic Injuries. Available online: <http://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries> (accessed on 24 November 2018).
3. Jing, P.; Huang, H.; Chen, L. An Adaptive Traffic Signal Control in a Connected Vehicle Environment: A Systematic Review. *Information* **2017**, *8*, 101. [CrossRef]
4. Zhang, W.; Aung, N.; Dhelim, S.; Ai, Y. DIFTOS: A Distributed Infrastructure-Free Traffic Optimization System Based on Vehicular Ad Hoc Networks for Urban Environments. *Sensors* **2018**, *18*, 2567. [CrossRef] [PubMed]
5. Hartenstein, H.; Laberteaux, K.P. A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.* **2008**, *46*, 164–171. [CrossRef]
6. Chin, H.C.; Quddus, M.A. Applying the random effect negative binomial model to examine traffic accident occurrence at signalized intersections. *Accid. Anal. Prev.* **2003**, *35*, 253–259. [CrossRef]
7. Lord, D.; Persaud, B. Accident prediction models with and without trend: Application of the generalized estimating equations procedure. *Transp. Res. Rec. J. Transp. Res. Board* **2000**, 102–108. [CrossRef]
8. Poch, M.; Mannering, F. Negative binomial analysis of intersection-accident frequencies. *J. Transp. Eng.* **1996**, *122*, 105–113. [CrossRef]
9. Mukhtar, A.; Xia, L.; Tang, T.B. Vehicle Detection Techniques for Collision Avoidance Systems: A Review. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2318–2338. [CrossRef]
10. Kumar, K.P.; Evangelin, S.J.; Amudharani, V.; Inbavalli, P.; Suganya, R.; Prabu, U. Survey on Collision Avoidance in VANET. In Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)—ICARCSET '15, Eluru, India, 6–7 March 2015; ACM Press: New York, NY, USA, 2015; pp. 1–5.
11. Raut, S.B.; Malik, L.G. Survey on vehicle collision prediction in VANET. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICIC), Coimbatore, India, 18–20 December 2014; pp. 735–751. [CrossRef]
12. Sam, D.; Velanganni, C.; Evangelin, T.E. A vehicle control system using a time synchronized Hybrid VANET to reduce road accidents caused by human error. *Veh. Commun.* **2016**, *6*, 17–28. [CrossRef]
13. Sharma, B.; Katiyar, V.K.; Kumar, K. Kranti Kumar Traffic Accident Prediction Model Using Support Vector Machines with Gaussian Kernel. In *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*; Pant, M., Deep, K., Bansal, J.C., Nagar, A., Das, K.N., Eds.; Springer: Singapore, 2016; pp. 1–10, ISBN 978-981-10-0451-3.
14. Wu, Q.; Hui, L.C.K.; Yeung, C.Y.; Chim, T.W. Early car collision prediction in VANET. In *International Conference on Connected Vehicles and Expo (ICCVE)*; IEEE: Shenzhen, China, 2015; pp. 19–23.

15. Lv, H.; Xu, P.; Chen, H.; Zhou, B.; Ren, T.; Chen, Y. A novel rear-end collision warning algorithm in VANET. In Proceedings of the 8th IEEE International Conference on Communication Software and Networks (ICCSN), Beijing, China, 4–6 June 2016; IEEE: Beijing, China, 2016; pp. 4–6.
16. Iqbal, A.; Busso, C.; Gans, N.R. Adjacent Vehicle Collision Warning System using Image Sensor and Inertial Measurement Unit. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction—ICMI '15, Seattle, WA, USA, 9–13 November 2015; ACM Press: New York, NY, USA, 2015; pp. 291–298.
17. Al Najada, H.; Mahgoub, I. Anticipation and alert system of congestion and accidents in VANET using Big Data analysis for Intelligent Transportation Systems. In Proceedings of the IEEE Symposium Series on Computational Intelligence 2016 (SSCI), Athens, Greece, 6–9 December 2016; IEEE: Athens, Greece, 2016; pp. 6–9.
18. Arbabzadeh, N.; Jafari, M. A Data-Driven Approach for Driving Safety Risk Prediction Using Driver Behavior and Roadway Information Data. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 446–460. [[CrossRef](#)]
19. Wang, H.; Dragomir, A.; Abbasi, N.I.; Li, J.; Thakor, N.V.; Bezerianos, A. A novel real-time driving fatigue detection system based on wireless dry EEG. *Cogn. Neurodyn.* **2018**, 1–12. [[CrossRef](#)] [[PubMed](#)]
20. Ba, Y.; Zhang, W.; Wang, Q.; Zhou, R.; Ren, C. Crash prediction with behavioral and physiological features for advanced vehicle collision avoidance system. *Transp. Res. Part C Emerg. Technol.* **2017**, *74*, 22–33. [[CrossRef](#)]
21. Maile, M.; Chen, Q.; Brown, G.; Delgrossi, L. Intersection Collision Avoidance: From Driver Alerts to Vehicle Control. In Proceedings of the IEEE 81st Vehicular Technology Conference 2015 (VTC Spring), Glasgow, UK, 11–14 May 2015; IEEE: Glasgow, UK, 2015; pp. 11–14.
22. Akhil, M.; Vasudevan, N.; Ramanadhan, U.; Devassy, A.; Krishnaswamy, D.; Ramachandran, A. Collision avoidance at intersections using vehicle detectors and smartphones. In Proceedings of the International Conference on Connected Vehicles and Expo 2015 (ICCVE), Shenzhen, China, 19–23 October 2015; IEEE: Shenzhen, China, 2015; pp. 19–23.
23. Bordel, B.; Alcarria, R.; Rizzo, G.; Jara, A. Creating Predictive Models for Forecasting the Accident Rate in Mountain Roads Using VANETs. In *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*; Rocha, Á., Guarda, T., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 319–329, ISBN 978-3-319-73450-7.
24. Wenqi, L.; Dongyu, L.; Menghua, Y. A model of traffic accident prediction based on convolutional neural network. In Proceedings of the 2nd IEEE International Conference on Intelligent Transportation Engineering, ICITE 2017, Singapore, 1–3 September 2017; pp. 198–202.
25. Zhang, J.; Wang, F.; Zhong, Z.; Wang, S. Continuous Phase Modulation Classification via Baum-Welch Algorithm. *IEEE Commun. Lett.* **2018**, *22*, 1390–1393. [[CrossRef](#)]
26. UK Department for Transport Road Safety Data. Available online: <https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data> (accessed on 23 November 2018).
27. Haklay, M.; Weber, P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Comput.* **2008**, *7*, 12–18. [[CrossRef](#)]
28. Behrisch, M.; Bieker, L.; Erdmann, J.; Krajzewicz, D. Sumo-simulation of urban mobility. In Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain, 23–28 October 2011; Volume 42.
29. Wegener, A.; Piórkowski Michał and Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J.-P. TraCI: An interface for coupling road traffic and network simulators. In Proceedings of the 11th Communications and Networking Simulation Symposium, Ottawa, ON, Canada, 14–17 April 2008; pp. 155–163.
30. Varga, A.; Hornig, R. An overview of the OMNeT++ simulation environment. In Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, Marseille, France, 3–7 March 2008; p. 60.

