

APP: Augmented Proactive Perception for Driving Hazards with Sparse GPS Trace

Siqian Yang

Department of Computer Science and Technology,
Tongji University
Shanghai, China
youngfourkings@gmail.com

Hongzi Zhu*

Department of Computer Science and Engineering,
Shanghai Jiaotong University
Shanghai, China
hongzi@cs.sjtu.edu.cn

Cheng Wang*

Department of Computer Science and Technology,
Tongji University
Shanghai, China
chengwang@tongji.edu.cn

Changjun Jiang

Department of Computer Science and Technology,
Tongji University
Shanghai, China
cjjiang@tongji.edu.cn

ABSTRACT

Driving safety is a persistent concern for urban dwellers who spend hours driving on road in ordinary daily life. Traditional driving hazard detection solutions heavily rely on onboard sensors (e.g., front and rear radars, cameras) with limited sensing range. In this article, we propose a proactive hazard warning system, called *APP*, which aims to alert drivers when there are vehicles with dangerous behaviors nearby. To this end, *APP* incorporates several basic techniques (e.g., tensor decomposition, similarity comparison) to estimate behavioral data of a driver based on sparse sampled GPS trace at first. Then, with the estimated unlabelled data, potential dangerous behaviors of a particular vehicle are identified and recognized with a Gaussian Mixture Model (GMM) based approach. We have implemented and evaluated our system with a dataset collected for 30 days from over 13,676 taxicabs. Our method shows on average 81% accuracy in potential dangerous behavior recognition.

CCS CONCEPTS

• **Mathematics of computing** → **Exploratory data analysis**; • **Networks** → *Network protocol design*;

ACM Reference Format:

Siqian Yang, Cheng Wang, Hongzi Zhu, and Changjun Jiang. 2019. APP: Augmented Proactive Perception for Driving Hazards with Sparse GPS Trace. In *Proceedings of the Twentieth International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'19)*. ACM, 10 pages.

*Corresponding Author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiHoc'19, July 2019, Catania, Italy

© 2019 Association for Computing Machinery.

ACM ISBN 123-4567-24-567/08/06...\$15.00

DOI: 10.1145/3323679.3326500

1 INTRODUCTION

According to the Global Status Report on Road Safety 2015 [15] by World Health Organization (WHO), traffic accidents take more than 1.2 million lives each year. Various potential factors, e.g., unskilled, drowsy, drunk, and drugged, lead to the tragedies and can be reflected to dangerous driving behaviors, e.g., speeding, hog overtaking, and sudden braking. Therefore, to perceive such dangerous driving behaviors (or hazards) in advance is key to avoiding traffic accidents and improving road safety. For instance, when a car driven by an unskilled driver approaches, even though the driver currently behaves fine, each neighboring vehicle will get alerted by messages, reminding drivers to pay more attention to this car; or when a driver is about to get exhausted and fall asleep, he/she gets refreshed by a piece of music automatically played.

Such applications pose several rigid requirements to a hazard perception system as follows: 1) *Being proactive*: severe traffic accidents lead to injuries or even deceases. For driving safety, it is better to actively identify potential driving hazards in advance and to take corresponding actions in time. The system should be proactive and try to avoid a driving hazard rather than respond to a hazard after it has happened. 2) *Reliability*: in normal driving scenarios, drivers cannot be all eyes and all ears for all surrounding situations at all time. It is very likely that blind spots exist during driving. The systems should be reliable and be able to perceive hazards in all directions for all time under all circumstances. 3) *Low deployment and usage costs*: the system should rely on cheap devices or devices that users already have. Furthermore, as the service is provided to users on a daily basis, the usage cost should be low.

In the literature, there exist a number of driving behavior recognition schemes. One category is based on high-sensitive video devices, which are mounted on vehicles to capture the driving behaviors of urban drivers [27] or the mobilities of surrounding vehicles. By analyzing head movement video data with image analysis method, e.g., convolutional neural network (CNN) [26], drowsy driving can be recognized. Moreover, an Event Data Recorder (EDR) can provide the driving sight data, which can be used to locate a vehicle's traffic lane and distinguish the behaviors of its front vehicles.

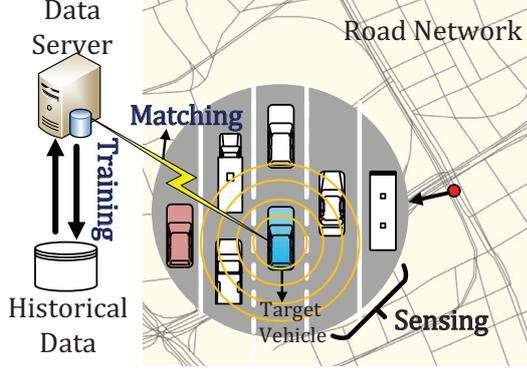


Figure 1: System Scenario

However, the results of these methods are affected by various factors, e.g., obstruction, video quality, weather, and camera's view angle. Furthermore, dedicated devices bring installation cost. Another category is based on the data of motion sensors, e.g., accelerometers, gyroscopes and magnetometers, which are utilized to determine that the driving state is safe or dangerous [12][28][6]. These data can reveal a vehicle's driving state, e.g., left-turn, right-turn, sudden braking and lane-change [24]. With the help of classification algorithms, e.g., Dynamic Time Wrapping (DTW) [12][18], Support Vector Machine (SVM) [6], and Hidden Markov Model (HMM) [2], typical and aggressive drivers can be identified. Nevertheless, this type of methods requires fine-grained sensor data (constant monitoring) and can only monitor the own vehicle of drivers. In addition, [21] gives a driving behavior analysis method by identifying speed-related and direction-related two operations from GPS traces through representation learning approach. Besides, all mentioned approaches are not proactive, which can only detect dangerous behaviors after they happen.

In this paper, we propose a proactive hazard warning system, called *APP*, which can be implemented on the smartphone as an application without any hardware modification. *APP* can alert drivers when there are vehicles with dangerous behaviors around. As illustrated in Fig. 1, in *APP*, vehicles report their traveling information to a server via wireless communications at a low frequency (i.e., the data is sparse in temporal and spatial dimensions). If there is a vehicle (e.g., the left-most vehicle) traveling with abnormal behaviors, the target vehicle (e.g., the vehicle in the center) will be informed by the server. The core idea of *APP* is for the server to estimate behavioral data of a driver based on the sparse GPS trace of him/her and other related drivers. With the estimated data, potential dangerous behaviors of a particular vehicle are identified and recognized. Finally, warning messages are pushed to vehicles in the vicinity of the identified vehicle.

One main challenge is to estimate a single driver's accurate behavioral data from sparse GPS samples. For example, Fig. 2 illustrates the behavioral data of five taxicabs in a day. As shown in the sub-figure marked *A*, reported behavioral data of two taxicabs are very coarse-grained due to communication overhead. In addition, tortuous road segments

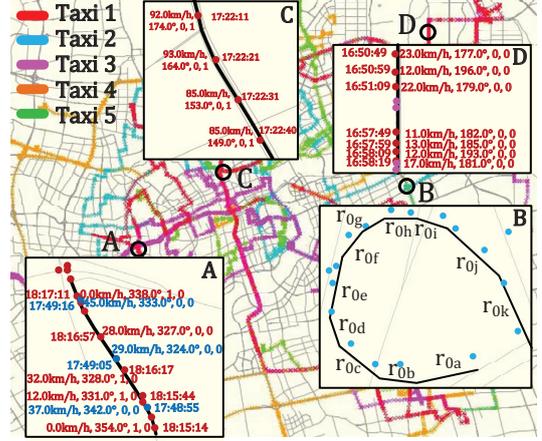


Figure 2: Local Examples of the Sparse Challenge

influences the accuracy of the driving direction estimation, as illustrated in the sub-figure marked *B* of Fig. 2. In order to tackle this challenge, *APP* considers the temporal-spatial correlation of its own behavioral data and other drivers'. Specifically, four tensors are constructed to store four driving behavior attributes, i.e., acceleration, brake, direction, and velocity. Then, a tensor decomposition approach is utilized to fill the non-zero entries. For the extreme situations where there is no sufficient data available on a road segment, *APP* tries its best to estimate the behavioral data by utilizing the road characteristics.

Another challenge is to recognize dangerous driving behaviors only from the behavioral data with no ground-truth dangerous behaviors labelled for training. Since it is hard to demarcate the border of the dangerous behaviors, we introduce the Gaussian Mixture Model (GMM), which has an appealing feature of soft assignment. In addition, we assume the distribution of abnormal data follows a Gaussian model, because most drivers act the similar dangerous behaviors by a same psychological motivator: trying faster. In *APP*, a GMM based dangerous behavior recognition method is proposed to distinguish those drivers with bad habits.

We implement the *APP* system and conduct extensive trace-driven simulations on the behavioral dataset of 13,676 taxicabs collected in a metropolis. The results show that *APP* can achieve an 81% accuracy on average in recognizing potential dangerous behaviors. In addition, a demo dataset provided by an insurance company is utilized to validate the performance of dangerous vehicle recognition in the evaluation. At last, we introduce a discussion about cutting down the server's reaction time by a broadcasting scheme under the hybrid wireless communication architecture.

The rest of the paper is organized as follows: Section 3 describes an overview of our *APP* system. In Section 4, Section 5, we introduce two basic modules in the system: source data training, and behavior matching, respectively. Section 6 gives the performances of the system. We discuss the implementation of our system under future vehicular network in Section 7. Section 8 reviews the related literatures and Section 9 concludes.

2 SYSTEM MODEL AND DESIGN GOALS

In this section, we introduce the main components and the scenarios considered in our system and list the design goals.

2.1 System Model

There are two main entities involved in the system as follows:

- **The server.** The server collects various data and provides services to drivers of these vehicles via wireless communication. It is often the case that a driver constantly accesses to such a server with his/her smart device for certain services, e.g., online navigation.
- **Vehicles.** A vehicle is equipped with multiple sensors, e.g., a GPS receiver, motion sensors, etc., used to monitor its behavior on the road. The coarse-grained sensory data is packaged into the GPS trace and reported to the server via wireless communication.
- **Wireless communication.** In APP, at current stage, data transmission between the server and vehicles are based on cellular networks such as 3G/4G/5G. Due to the communication cost, vehicles report their GPS trace data at a low frequency. In the future, new techniques such as V2X technology (vehicle-to-vehicle, vehicle-to-infrastructure, and vehicle-to-pedestrian) would also be considered.

2.2 Design Goals

Three goals motivate us to design our system.

- Proactively alert urban drivers of potential dangerous neighbor vehicles via historical GPS trace data on the road.
- Recognize the dangerous behavior accurately and in time due to fast mobility and ephemeral neighboring relationship among the vehicles on the road.
- Low communication cost is essential to provide the service due to the limitation of network traffic.

3 OVERVIEW OF APP SYSTEM

In this section, we take an overview of the system framework. As shown in Fig. 3, the system consists of two modules: training module, and matching module. The training module is to fill the driving behaviors at each road segment via the rough historical GPS trace data, which is taken charge of by the data sever. In the model, a four-dimension coordinate (acceleration, brake, direction, and velocity) is constructed from the mapped data to describe the driving behaviors on each road segment. After tensor recovery, we obtain an amount of drivers' behaviors on each road segment during a certain time period, which facilitates to tell the dangerous behaviors from the normal ones. In the matching module, a GMM based approach is utilized to cluster the recovered data on each road segment and time slot into two categories: dangerous and normal. Meanwhile, the historical behavior data trains the parameters in the GMM to identify dangerous risks when an unknown data stream comes. This step is applied on the sever side. Thus, the sever feedbacks dangerous vehicles nearby to the target one through the real-time collected behavior data. With the cooperation of these two modules and cellular network architecture, as shown in

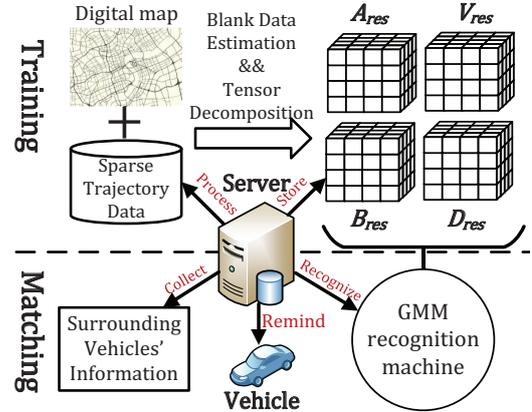


Figure 3: System Framework

Fig. 3, the system can send warnings to urban drivers when there are potential dangers among surrounding vehicles.

4 SOURCE DATA TRAINING

In this section, we first give the problem description in the training of the source GPS trace data. Then, we show the solutions step by step.

4.1 Problem Description

An item in the source historical GPS trace data mainly contains six elements: GPS point, time, instant velocity, direction, brake, and highway. Specially, highway and brake are Boolean values to reflect whether a vehicle is on highway and braking state, respectively. For example, Fig. 2 labels several items. The basic step is to match the raw GPS data onto each road segment in a digital map. In this paper, the road network is a set of road segments, and each one is one-way and contains only two intersections. As Fig. 4(a) shows, the roads are divided into segments by red hollow points.

The biggest challenge is data coverage. The historical GPS trace data is far away to provide enough driving information at a certain time period on each road segment, which obstructs recognizing the dangerous behaviors since the abnormal behaviors are in the minority. For instance, Fig. 2 shows five cabs' trajectories during a day, where different colors denote different cabs. The data was recorded almost every 10 seconds. Obviously, the trajectory is clear (could tell the driving route), but it is not real to make the cab's data cover all the time and map. What is worse, there are several road segments with no data mapped. Besides, the changing direction of a road segment affects the vehicles' direction data which makes it hard to excavate precise behaviors, e.g., r_0 in the lower right of Fig. 2 and r_1 in the upper left of Fig. 4(a). Hence, we propose a hybrid solution, including direction data refinement, blank data estimation and tensor decomposition to enhance the dataset with precise items.

4.2 Behavior Definition

Since the elements in an GPS trace item is able to describe a vehicle's traveling state at a moment, its driving behavior could be figured out by several items. For example, in the

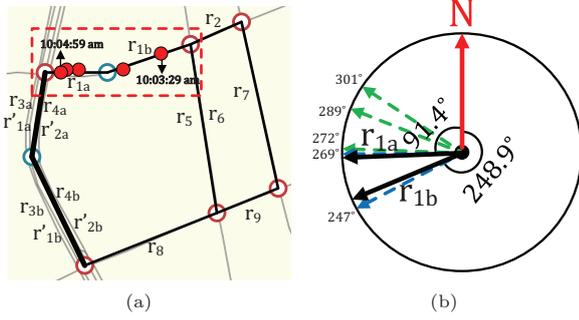


Figure 4: Road Segment

subfigure marked C of Fig. 2, the item at 17:22:21 reflects the vehicle was overspeed for an instant, while the four items show its driving behavior at this road segment is illegal. Here, we introduce four attributes to describe the driving behavior based on map-matching data: velocity, acceleration, direction, and brake.

- **Velocity:** The instant velocity data is attached to each item. For a vehicle, velocity is directly related to safety. Both driving too fast and too slow are contrary to normal behavior. In this paper, we define velocity attribute as the average speed on a road segment.
- **Acceleration:** Besides the speed, big acceleration also brings dangers. We define the acceleration attribute as the average acceleration on a road segment.
- **Direction:** The direction is defined as the clockwise rotation angle between the road (or vehicle) direction and the North of map. In Fig. 4(a), the red points in the rectangle are map-matching points of a taxi from 10 am to 11 am at April 1st, while their corresponding directions are displayed in Fig. 4(b). In the figure, the direction of r_{1a} is 268.6° , and the green dotted arrows are the points mapped on r_{1a} . From this observation, we find that there are small deviations between the directions of the vehicle and the road, which meets the common sense. But if the vehicle changes lanes frequently, deviations will become large. Meanwhile, the behavior will threaten others' safety. Hence, we describe the direction attribute by calculating the variance of the differences between the record directions and the direction of a road segment. The detail is shown in Section 4.3.
- **Brake:** There is a brake tag in a data item, which reflects whether the target vehicle is under braking condition or not. On a road segment, we define the brake attribute as the frequency of the braking conditions.

4.3 Direction Data Refinement

As mentioned in Section 4.1, the value of direction attribute is affected by the changes of the road direction. In this paper, we give a refinement method to obtain a more-accurate value. Three steps are contained: a) split a road segment into several straight sub-segments; b) calibrate the direction value on each sub-segment; c) combine the results together.

In a digital map, a road consists of several GPS points. We filter two types of points: cross and inflection ones. The

cross point, denoted as intersection in physically, is utilized to divide a road into road segments, while an inflection point separates a road segment into two sub-segments with different directions. The cross points are easy to obtain. Besides, we find the inflection points of a road segment using following equation.

$$|\Theta(p_i, p_b) - \Theta(p_i, p_e)| > \varepsilon, \quad (1)$$

where $\Theta(p_i, p_j)$ denotes the oblique angle of the line segment with ending points p_i and p_j , and ε denotes the bias threshold.

For example, as illustrated in Fig. 4(a), the hollow points in red are cross points, while those in blue are inflection points. Hence, the roads in the area are divided into 9 road segments and 16 sub-segments. r_i denotes the i -th urban road segment, while r'_i denotes the i -th highway road segment.

Then, on each sub-segment, the mapped points calculate the differences between their directions and the sub-segment's direction. Fig. 4(b) shows the example data. On sub-segment r_{1a} , there are three mapped points, and their directions are 272° , 289° , and 301° , respectively. The absolute differences are 3.4, 20.4 and 32.4. By the same logic, that of the points on r_{1b} is 20.1 and 1.9. At last, we calculate the mean square deviation of the differences to obtain the value of direction attribute. In the example, it is about 377. In addition, as displayed in the bottom right of Fig. 2, definition in Eq. (1) also could handle the special case that the road segment is an arc (appears in highway).

4.4 Data Filling

After raw data preprocessing, including map-matching and direction data refinement, we obtain the driving behavior data from four dimensions on each road segment at a certain time slot. However, on most road segments, the scale of the behavior data are not enough to distinguish dangerous behaviors from normal ones. To solve this problem, we apply a tensor decomposition based method to enhance behavior data on each road segment.

4.4.1 Tensor Construction. Each action in the behavior data is determined by a 3-D coordinate (u, r, t) , where u , r and t denote the driver, the road segment, and the time slot, respectively. Since a tensor describes linear relations between scalars and geometric vectors, we can link the driving behaviors (scalars) with the inherent property elements (vector) by tensors. Hence, four 3-D tensors (\mathcal{A} , \mathcal{B} , \mathcal{D} , and \mathcal{V}) are constructed to store the behavior data of different attributes. The size of each tensor is $M \times N \times L$, where M , N , and L denote the number of drivers, the road segments and the time slots, respectively. In the evaluation, we set the time slot to be one hour.

4.4.2 Blank Data Estimation. After construction, we find that there are several road segments with no data mapped due to the sparsity of the source data. In the evaluation, the count is 185 in total 478 road segments. To solve the problem, we fill the blank data by comparing the Euclidean distances between the target road segment and the others with the similar shape. A feature vector \mathbf{f} is included to

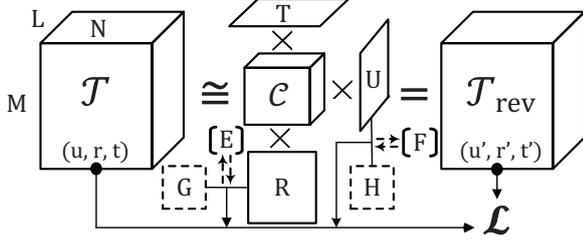


Figure 5: Description of Tensor Decomposition

collect the intrinsic properties for each road segment. The vector consists of three elements: average direction, length, and intersection type. Then, we update the each tensor by following equation:

$$\mathcal{T}(:, i, :) = \frac{\sum_{j=1 \& j \neq i}^N \mathbf{f}_i \mathbf{f}_j \cdot \mathcal{T}(:, j, :)}{\sum_{j=1 \& j \neq i}^N \mathbf{f}_i \mathbf{f}_j}, \quad (2)$$

where labels i and j denote two different segments, and $\mathbf{f}_i \mathbf{f}_j$ denotes the Euclidean distance between \mathbf{f}_i and \mathbf{f}_j .

4.4.3 Tensor Decomposition. To provide enough behavior data on a road segment, we decompose the tensors by Tucker Decomposition to fill the empty entries [7]. A target behavior tensor \mathcal{T} is decomposed into four parts: core tensor \mathcal{C} , and three factor matrices U , R and T [19], as following equation shows:

$$\mathcal{T}_{rev} = \mathcal{C} \times U_1 \times R_2 \times T_3,$$

where 1, 2 and 3 denote the multiply order, and \mathcal{T}_{rev} denotes the recovery tensor.

The entries in each part are initialized with Xavier initialization. Then, during every iteration, a recovery tensor is constructed by the multiplication of these four parts. Through the comparison of the non-zero entries between the source tensor and the recovery one by an objective function, the values in each part will be converged. At last, the source tensor is recovered by additional estimated entries. Fig. 5 gives the main process of the decomposition, where \mathcal{L} denotes the objective function. Each non-zero entry (u, r, t) in source tensor \mathcal{T} is compared with corresponding entry (u', r', t') until convergence (the difference is below a threshold). Besides, the physical meaning of the decomposition is to estimate a driver's four dimension behavior by considering different factors, e.g., driver, road, and time. Thus, we get four full tensors \mathcal{A}_{rev} , \mathcal{B}_{rev} , \mathcal{D}_{rev} , and \mathcal{V}_{rev} .

4.4.4 Check Matrix. Nevertheless, there are irrational values among the filled entries in the tensors, due to the neglect of additional limitations, e.g., environment and familiarity. To guarantee the accuracy of the recover entries, two check metrics are involved in the tensor decomposition: traffic environment matrix E and driver familiarity matrix F . In the system, the digital map is divided into 4 zones to fill the check matrices with non-zero entries. E collects the traffic flow at each zone during a time slot, which size is $L \times 4$. Meanwhile, F counts the drivers' appearance at each zone, which size is $M \times 4$. As Fig. 5 shows, the objective function of each tensor is updated with two additional items: $\|F - U \times G\|^2$

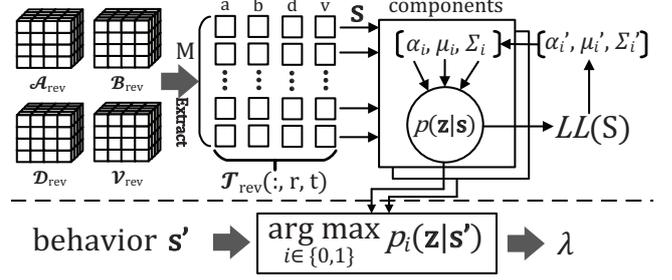


Figure 6: Description of GMM

and $\|E - T \times H\|^2$, where $\|\cdot\|^2$ denotes the L_2 norm, and G , H are auxiliary matrices.

5 BEHAVIOR MATCHING

According to the processed unlabelled behavior data, the problem here is how to define the dangerous tags, and identify an unknown behavior on a road segment at a certain time slot. Hence, two main steps are included in the matching module: distinguish and recognize the dangerous behaviors.

5.1 Danger Extraction

In the first step, we cluster the behavior data on each road segment into two categories: dangerous and normal. Since it is hard to demarcate the border of the dangerous behaviors, Gaussian Mixture Model (GMM) is applied in the system due to its soft assignment feature. In addition, the distribution of the nature world usually can be described by Gaussian model, including the vehicular network.

The basic idea of GMM clustering method is to regard the distribution of the source data as the mixture of several Gaussian distributions, and separate each out by the parameters trained through Expectation Maximization (EM) algorithm [10]. Fig. 6 displays the process. From the recovered behavior data, each tensor extracts a vector with M elements, which means a behavior on a road segment during a certain time slot. Then, we construct a behavior matrix S , which size is $4 \times M$. Each row vector in S respects a driver's driving behaviors, named *behavior vector*. Since two categories (dangerous and normal) are enough in the system, we mix 2 Gaussian distributions, called components, to fit the distribution of the behavior vectors. A component consists of three parameters α , μ , Σ and a probability $p(z|\mathbf{s})$, where α , μ , and Σ denote the mixture coefficient, center vector, and covariance matrix, respectively. Specially, $\sum_{i=0}^1 \alpha = 1$ and covariance matrix is a 4×4 matrix. z is a random variable to denote the tag of the component of a vector \mathbf{s} , where $z \in \{0, 1\}$. The calculation of $p(z|\mathbf{s})$ is based on Bayesian theorem, combining with mentioned three parameters. In the first iteration, $\alpha_0 = \alpha_1 = 0.5$, and two center vectors are random selected from the behavior matrix. Meanwhile, the covariance matrices Σ_0 and Σ_1 are initialized as identity matrices. Besides, as Fig. 6 shown, a likelihood function $LL(S)$, integrated with $p(z|\mathbf{s})$, is utilized to train the parameters in each component. The training process is handled by EM algorithm, which object is to maximize the likelihood

function during every iteration. After convergence, we obtain the probability distributions for the dangerous and normal behaviors at each road segment during a certain time slot, which separates these two behaviors out.

5.2 Danger Recognition

With the parameters obtained in the distinguish step, the system could recognize an unknown behavior vector, shown in the lower part of Fig. 6. When a new behavior vector \mathbf{s}' comes, the system use λ to label the behavior according to the result of $\arg \min_{i \in \{0,1\}} p_i(z|\mathbf{s}')$. The physical meaning is to put the behavior into the corresponding categories (clustered in Section 5.1) with the maximum probability.

6 EVALUATION

We first describe the evaluation methodology, including dataset description, metrics definition and experimental settings. Then, we show the choices of the parameters in the system. At last, we validate the efficiency of our system from two impacts: data sparseness, and road types.

6.1 Methodology

6.1.1 Dataset Description. We give a brief introduction of the datasets, digital map, and ground truth.

Source Dataset: The source dataset records 13,676 taxis' running items from April 1st to Apr. 30th at 2015 in Shanghai, which size is about 300G. Each item stores the information of highway, GPS point, record time, direction, instant velocity, brake state and so on. Besides, the average record time interval is about 10 seconds.

Validation Dataset The validation dataset is used to test the effectiveness of GMM model described in Section 5. The dataset is a demo set collected by an insurance company from Jul. 4th to Dec. 31st at 2016 in Shanghai, including 100 vehicles' traveling data, and its size is about 4MB. In the dataset, each item not only includes the elements similar to the source dataset, but also contains an additional element named *insurance compensation ratio*. The compensation ratio of insurance could reflect whether the vehicle has suffered accidents or not.

Digital Map: The map we utilized in the experiment is generated by OpenStreetMap [8], which is the area with the red dotted borders in Fig. 7(a). Besides, 478 road segments are contained in the map.

Ground Truth: In the experiment, we manually mark 10 taxis' behaviors in the source dataset as the ground truth. 3 of them are selected from a blacklist with 27 taxis, which is published by the Traffic Management Bureau. The other 7 taxis are those with the maximum mapped items in the selected map. To mark the behavior, we first give a threshold parameter ω . A behavior will be regarded as dangerous, if any one value in the behavior vector does not satisfy following equation,

$$1 - \omega \leq \frac{\chi}{\chi_{avg}} \leq 1 + \omega, \quad (3)$$

where $\omega \in (0, 1)$ and χ denotes the element (a , b , d or v) in the behavior vector (shown in Fig. 6). From the observations of the behavior vectors on each road segment, we find

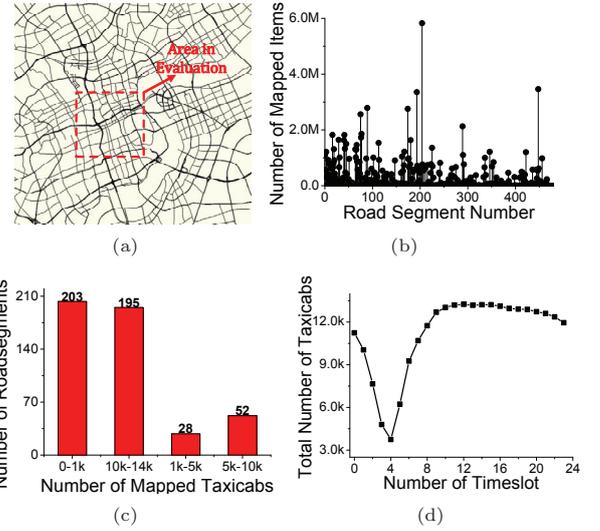


Figure 7: Results of Map-Matching

when $\omega = 0.1$, most remaining behaviors filtered by Eq. (3) can be considered as dangerous acts in common sense. Then, we carefully refine the two behavior sets to construct a test dataset. Besides, in the validation dataset, we regard those with non-zero compensation ratio as the dangerous vehicles.

6.1.2 Metrics Definition. We evaluate the performance of dangerous behavior recognition from two metrics: precision and recall. The calculation of precision is defined as $\frac{tp}{tp + fp}$, and that of recall is $\frac{tp}{tp + fn}$, where tp , tn , fp and fn denote the true positives, the true negatives, the false positives and the false negatives, respectively. Specifically, the true positives are the count that right recognizing dangerous behaviors.

6.1.3 Experimental Settings. The training and matching steps are implemented in C++ on a server with 32GB memory and Intel(R) Xeon(R) E5-2665 at 2.4GHz.

6.2 Data Processing Results and Parameter Choice

We first show the results in source data processing, then we give the choice of the parameters in our system.

6.2.1 Map-Matching. The fundamental step is map-matching. We utilize the algorithm in [16]. The total count of the mapped items is 120,172,806. Fig. 7 gives the detailed results. Fig. 7(b) shows the number of the items mapped on each road segment, while Fig. 7(c) displays its distribution on the road segments. Besides, Fig. 7(d) illustrates the taxis mapped on each time slot. Obviously, the counts drop to a minimum value at about 4 am (4th time slot).

6.2.2 Data Filling. Table 1 displays the statistical information of the behavior attribute tensors and matrices described in Section 4.4. Note that the duration of a time slot is set

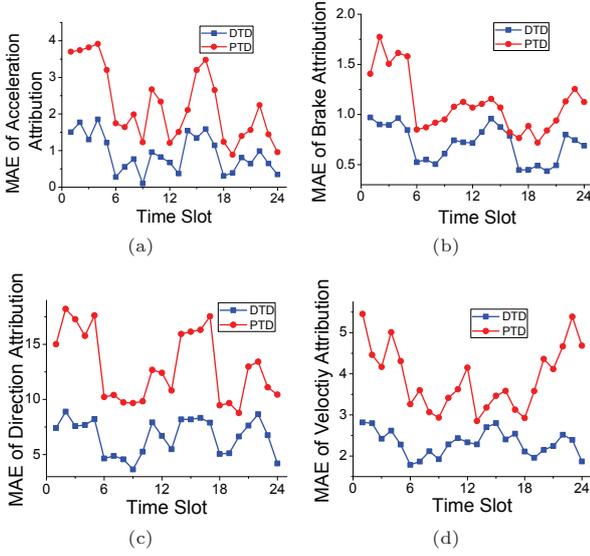


Figure 8: MAE between PTD and DTD

to be one hour. Since the size of four tensors is the same ($13,676 \times 478 \times 24$), we let \mathcal{T} denote the target attribute tensor. The ratio of the non-zero entries in tensor \mathcal{T} is 11.57%. Moreover, we find that there are 185 road segments with no data recorded. We fill these zero entries with blank data estimation, which is defined in Eq. (2). Thus, the non-zero ratio of each tensor increases to 18.88%.

Table 1: Statistics of the Tensor/Matrix

Tensor or Matrix	Size	Non-zero ratio
\mathcal{T}	$13,676 \times 478 \times 24$	11.57%
E	$13,676 \times 4$	100%
F	4×24	100%

Note: \mathcal{T} denotes the tensor (\mathcal{A} , \mathcal{B} , \mathcal{D} , or \mathcal{V}), while E and F denote the check matrices.

After decomposition, the zero entries in each tensor are filled with estimated values. We validate the performance of two check matrices using the mean absolute error (MAE) compared with the non-zero entries of a marked taxicab, which is selected as the ground truth. First, we decompose the four attribute tensors with and without check matrices, respectively. Then, MAE is calculated between the real non-zero values in the original tensors and the estimated values in the recovery ones. In addition, as Table 1 illustrates, the size of the matrix E is $13,676 \times 4$, while that of the matrix F is 4×24 . We name the decomposition method integrated with check matrices derivative method (DTD), while the primary method (PTD) is without check matrices. The MAEs between the PTD and the DTD of four attribute tensors are shown in Fig. 8. Clearly, the check matrices give a great favor in the tensor decomposition.

6.2.3 Parameter Settings. The threshold ε in Eq. (1) is set to be 1° . Hence, the 478 road segments in the selected map can

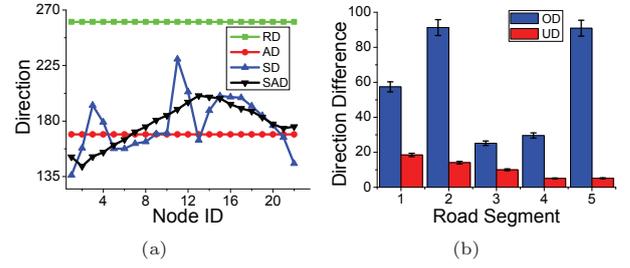


Figure 9: Results of Direction Refinement

be divided into 1,870 sub-segments. We evaluate the performance of direction data refinement using the average value of the taxicabs' directions mapped on a road segment as the reference. Fig. 9(a) illustrates the directions of a road segment with 21 sub-segments and that of the taxicabs mapped on it. Note that labels "RD", "AD", "SD" and "SAD" denote road segment direction, average direction, segmental direction and segmental average direction, respectively. Note that "RD" and "SD" (calculate the direction only between beginning node and end node) come from the digital map, while "AD" and "SAD" are calculated (obtain the average value from the direction data) by the taxicabs' data. Obviously, only using the directions of the road segments may cause errors in the direction attribute calculation. Fig. 9(b) compares the original difference (OD) and updated difference (UD) by 5 road segments in the map. Original difference means the absolute average difference between "RD" and "AD", while updated difference denotes that between "SD" and "SAD".

As described in Section 5.1, we set $\alpha_0 = \alpha_1 = 0.5$ when we initialize the parameters in GMM. Meanwhile, two center vectors are random selected among 13,676 behaviors on each road segment at a certain time slot. The covariance matrices Σ_0 and Σ_1 are initialized as the identity matrices, which have the same size 2×2 . Fig. 10(a) shows the average percentages of the dangerous taxicabs at each time slot after the clustering of GMM based method. We find that less dangerous behaviors are occurred during rush hour than others. Moreover, to validate the credibility of the GMM-based approach in the system, we first do not input the behavior data of the 10 marked taxicabs into the GMM in the danger extraction step. After a trained GMM obtained, we analyze its performance among the total behaviors of the marked taxicabs, which is shown in Fig. 10(b). The average precision that a taxicab's dangerous behavior can be right recognized is about 81%.

6.2.4 Dangerous Vehicle Recognition. In this section, we give the results about the dangerous vehicle recognition by the GMM based method described in Section 5. As mentioned in Section 6.1.1, a validation dataset is included in the evaluation. The dataset contains 6,370 behaviors of 100 vehicles. Fig. 11(a) shows the count distribution of the vehicles' behaviors. The minimum count is 1, while the maximum is 223. During the recognition, we utilize the GMM based method to distinguish each behavior. Then, for each vehicle, their behaviors are classified into two categories: dangerous and normal. Thus, if the count of the dangerous one is bigger

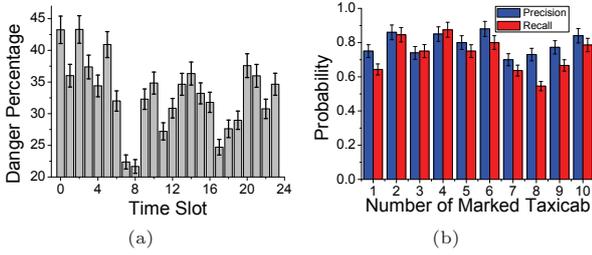


Figure 10: Dangerous Behavior Recognition Results

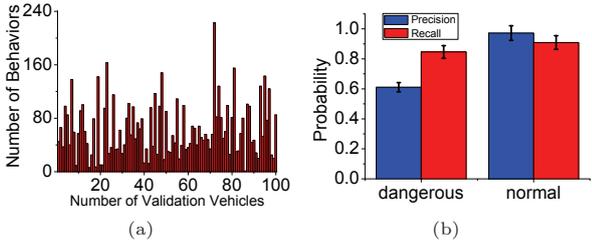


Figure 11: Dangerous Behavior Recognition Results

than the normal one, the vehicle will be marked “dangerous”. Otherwise, we mark it “normal”.

Specifically, as the ground truth mentioned in Section 6.1.1, 13 vehicles’ compensation ratio are not equal to zero, which are regarded as dangerous vehicles. Fig. 11(b) gives the recognition results of the dangerous vehicles in the validation dataset. We find that most (over 80%) dangerous vehicles could be recognized. Meanwhile, the precision of the normal vehicles recognition is high (about 97%), which means our system gives a strict bound for those vehicles with safety behaviors.

6.3 Case Study

In this section, we design two different experiments to study the impacts of the data sparseness and the road type on the system, respectively.

6.3.1 Impact of Data Sparseness. In our system, the data sparseness is described as the non-zero ratio in the source historical dataset. We adjust the ratio by random removing the non-zero entries in the original behavior tensors. 5 levels are divided in the experiment: 20%, 40%, 60%, 80% and 100%. Each level reflects the ratio of the remaining non-zero entries in original ones. For example, when the level is 20%, the non-zero entries ratio is $11.57\% \times 20\% = 2.314\%$ in the original tensors. Then, we calculate the recall and precision of a marked taxicab’s total behaviors. The result is shown in Fig. 12(a). It is easy to observe that the behavior recognition tends to random distribution when the data is too sparse.

6.3.2 Impact of Road Types. Next, we keep the original training tensor and compare the performances under urban street and highway two different road types. In the experiment, we select two road segments with different types (street or highway), and regard a marked taxicab’s behaviors on both ones

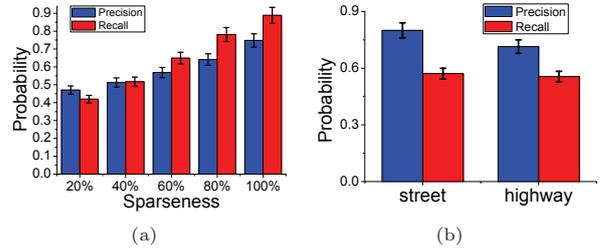


Figure 12: Results of Case Study

as the ground truth. The result is displayed in Fig. 12(b). The performance in highway shows more recall but less precision due to the high mobility.

7 DISCUSSION

The cellular network architecture has its drawbacks, e.g., overload. The control center should collect information items of the vehicles with extra network traffic. Besides, through the explosion of the V2X technology, vehicular ad-hoc network (VANET) [1][4] can support the communications among vehicles in the foreseeable future. If a vehicle can proactively sense its neighbors’ driving behaviors, it will avoid unnecessary network traffic and reduce service latency. Hence, we discuss to extend our network architecture for future traffic environment.

A general hybrid network architecture can be generated. As shown in the left part of Fig. 13, the server and the vehicles construct a central network architecture, while a distributed network architecture is built among vehicles. The central architecture collects each vehicle’s reported data, and feedbacks the dangerous vehicles’ plate licenses. In the distributed network, a vehicular ad-hoc network (VANET) is built among vehicles, which makes the vehicles sense the surroundings and generates behavior vectors by themselves. The prospective communication schedule under the architecture is shown in the right side of Fig. 13. A vehicle node broadcasts the state beacon periodically. Meanwhile, it reports to the server with the behavior vectors, which is obtained from the collected state items. Two packages are involved:

- **Broadcast beacon**, which contains plate license, GPS points, time, brake state, direction, and velocity;
- **Report package**, which consists of plate licenses and corresponding behavior vectors of the vehicles around.

From this network architecture, the control center could receive more information of the vehicles with less network traffic on a target road segment. However, there remains a problem that insufficient realtime information cannot describe accurate state of a vehicle. This affects the precision of the recognition directly. Through a small experiment, we find the relationship between the behavior similarity and the information item count by cosine similarity, shown in Table 2. From the table, we observe that when the number of the items collected is bigger than 6, the similarity is over 90%.

Then, we apply a simulation to find optimal parameters under this network architecture. Two simulators SUMO [3] and NS-3 [17] are utilized to simulate the car-following

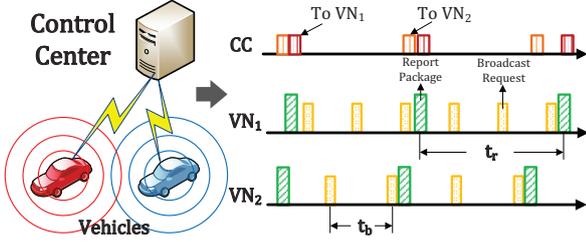


Figure 13: Hybrid Structure and Broadcast Schedule

Table 2: Relationship Between Similarity and Items

Items	2	3	4	5	6+
Similarity	57.2%	59.4%	88.9%	89.2%	92.1%

Note: The first row denotes the count of collected information items, while the second row denotes the similarity between the estimated behavior and the ground truth.

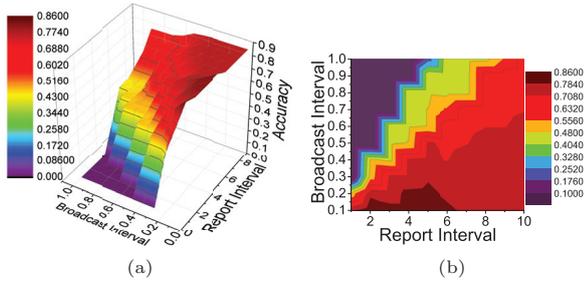


Figure 14: Performance Under Hybrid Architecture

mobility and wireless communication. In the simulation, we select a road segment with the most points mapped, and simulate the taxicabs' mobility during 24 time slots according to the map-matching data. Besides, 10 simulator vehicles are added into the simulation on each time slot. 3 of them have dangerous behaviors and other 7 are normal. The range of the broadcast interval is selected from 0.1s to 1s, while that of the report interval is from 1s to 10s. Especially, we use a base station as the control center. The communication protocol between the center and the vehicles is set to be LTE [5], while we utilize IEEE 802.11p protocol [14] to support V2V communication. Fig. 14 shows the relationship between the average accuracy of dangerous behavior recognition and two parameters (broadcast interval and report interval) in the wireless communication. Obviously, long report interval and short broadcast interval result in high recognition accuracy. Moreover, we find when $p_b = 0.2s$ and $p_r = 2s$, the average recognition accuracy is still outstanding (over than 75%).

In the simulation, the reaction latency of the control center side is over 3s under traditional central network architecture, since it should generate and recognize the behavior vectors from the collected messages of the vehicles' instantaneous states, then feedback the results. The latency reduces to about 2s in the hybrid architecture due to omitting the behavior vector generation step on the control center.

8 RELATED WORK

The researches on assisting driving safety mainly utilize GPS or multi-sensors to capture a vehicle's current driving state, and provides services to the drivers, e.g., Advanced Driving Assistant Systems (ADASs). The studies focus on two types of data: trajectory data [29] and sensor data.

Trajectory data is a trace of geographical points with chronologically ordered (collected by GPS). Several work are devoted to preprocess trajectory data by noise filtering, stay point detection, compression, segmentation, and map-matching. With the help of a digital map [8] and mentioned basic preprocessing operators, approaches are proposed for path inference [23], and traffic time estimation [22], and etc. However, these methods can only tell the coarse-grained traveling actions, e.g., left-turn, right-turn and stop place, which are not enable to distinguish the detailed actions, e.g., line-change, overspeed, braking and etc. Specially, the drivers' social relationship [25] can be added into trajectory study to excavate their driving skills. But it is hard to obtain the comprehensive trajectory data and corresponding social data of most drivers.

To measure the precise driving state, additional sensor devices are introduced into investigation, which are mainly divided into two parts: camera and other sensors, e.g., accelerometer, gyroscope, and magnetometer. *Video-based approaches* belong to an independent research field, which deal with the visible dangers by image processing techniques. You et al. [27] proposed an application to alert the drowsy drivers by dual cameras on smartphones. *Sensor-based approaches* use sensors to record the vehicles' consecutive realtime performances, e.g., direction, and acceleration. In the early study, Vlad Coroama [9] built a system named "Smart Tachograph" with sensors to analyze the individual traffic costs. Nowadays, machine learning algorithms, e.g., DTW, SVM, and NMF, are hot applied on the data to achieve driving behavior monitoring, estimation, and prediction. Aoude et al. [2] proposed a classification method based on HMM and SVM to identify the driver as compliant or violating at urban intersection. Eren et al. [12] and Smith et al. [18] tried to understand the drivers' behaviors by DTW. Yu et al. [28] estimated vehicle speed only using sensor data without GPS information. Chen et al. [6] proposed a realtime driving behaviors monitoring method combining with SVM to detect and extract fine-grained abnormal behaviors. Deng et al. [11] utilized NMF to predict realtime traffic on high-fidelity spatiotemporal sensor datasets. Moreover, driving behavior visualization[13] is a rising research direction, which is to combine sensors' record with deep learning algorithms, e.g., deep sparse autoencoder (DSAE), to give a color pattern of the driving state. In addition, [21] gives a driving behavior analysis method by identifying speed-related and direction-related two operations from GPS traces through representation learning approach. The main goal of these studies is to detect the dangerous driving behaviors for a single target vehicle using precise sensor data. But the dangers from surrounding vehicles are not taken into account.

Besides, several researchers try to solve the problem by wireless communication. Takaaki Umedu et al. [20] proposed a Dangerous Vehicle Detection Protocol (DVDP). The target

vehicle utilizes the collected information to estimate the velocities of surrounding vehicles. However, only realtime speed is insufficient to reflect the surrounding dangers.

To the best of our knowledge, our work first extracts and recognizes potential driving dangers nearby from sparse behavioral records, and extends it to future vehicular network.

9 CONCLUSION

In this paper, we propose a system, named APP, to proactively alert the urban driver of the dangerous vehicles around. Further, the system is capable to recognize the dangerous behaviors via the rough historical behavioral data under the cellular network architecture. Two modules are designed to provide the service: the training module is designed to enhance the behavioral data from the sparse source GPS trace; the matching module is to extract dangerous behaviors, and recognize the unknown behavior data stream. Besides, we discuss the implementation of our system under a hybrid network architecture, which is suit for future vehicular network. As an evaluation, our system shows on average 81% accuracy in dangerous driving behavior recognition.

In the future work, we will try to model the detail behaviors in each GPS sampling interval, since 10s is too long in raw GPS trace. Besides, further to cut down the service response is also an emergency task.

10 ACKNOWLEDGE

The research of authors is partially supported by the National Natural Science Foundation of China under Grants 61772340 and 61420106010, the National Natural Science Foundation of China under Grants 61571331, Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China under Grant 151066, the Fundamental Research Funds for Central Universities under Grant kx0137020181527, National Key R&D Program Projects under application number SQ2018YFB210029, and 2017 CCF-IFAA RESEARCH FUND. Furthermore, we thank all anonymous reviewers for their insightful comments.

REFERENCES

- [1] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. 2014. A comprehensive survey on vehicular Ad Hoc network. *Journal of network and computer applications* 37 (2014), 380–392.
- [2] Georges S Aoude, Vishnu R Desaraju, Lauren H Stephens, and Jonathan P How. 2011. Behavior classification algorithms at intersections and validation using naturalistic data. In *Proc. IEEE IV 2011*.
- [3] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. 2011. SUMO—simulation of urban mobility: an overview. In *Proc. ThinkMind SIMUL 2011*.
- [4] Yuanguo Bi, Hangguan Shan, Xuemin Sherman Shen, Ning Wang, and Hai Zhao. 2016. A multi-hop broadcast protocol for emergency message dissemination in urban vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems (TITS)* 17, 3 (2016), 736–750.
- [5] Shanzhi Chen, Jinling Hu, Yan Shi, and Li Zhao. 2016. LTE-V: A TD-LTE-Based V2X Solution for Future Vehicular Network. *IEEE Internet of Things Journal* 3, 6 (2016), 997–1005.
- [6] Zhongyang Chen, Jiadi Yu, Yanmin Zhu, Yingying Chen, and Minglu Li. 2015. D3: Abnormal driving behaviors detection and identification using smartphone sensors. In *Proc. IEEE SECON 2015*.
- [7] Andrzej Cichocki. 2014. Era of big data processing: A new approach via tensor networks and tensor decompositions. *Computer Science* (2014).
- [8] OpenStreetMap Contributors. 2012. OpenStreetMap. URL <http://www.openstreetmap.org/> (2012).
- [9] Vlad Coroama. 2006. *The Smart Tachograph Individual Accounting of Traffic Costs and Its Implications*. Springer Berlin Heidelberg, 135–152 pages.
- [10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)* (1977), 1–38.
- [11] Dingxiong Deng, Cyrus Shahabi, Ugur Demiryurek, Linhong Zhu, Rose Yu, and Yan Liu. 2016. Latent space model for road networks to predict time-varying traffic. (2016).
- [12] H. Eren, S. Makinist, E. Akin, and A. Yilmaz. 2012. Estimating driving behavior by a smartphone. In *Proc. IEEE IV 2012*.
- [13] Hailong Liu, Tadahiro Taniguchi, Yusuke Tanaka, Kazuhito Takenaka, and Takashi Bando. 2017. Visualization of driving behavior based on hidden feature extraction by using deep learning. *IEEE Transactions on Intelligent Transportation Systems (TITS)* (2017).
- [14] Feng Lv, Hongzi Zhu, Hua Xue, Yanmin Zhu, Shan Chang, Mianxiong Dong, and Minglu Li. 2016. An Empirical Study on Urban IEEE 802.11p Vehicle-to-Vehicle Communication. In *Proc. IEEE SECON 2016*.
- [15] World Health Organization. 2015. *Global status report on road safety 2015*. World Health Organization.
- [16] Mohammed A Quddus, Washington Y Ochieng, and Robert B Noland. 2007. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies* 15, 5 (2007), 312–328.
- [17] George F Riley and Thomas R Henderson. 2010. The ns-3 network simulator. In *Modeling and tools for network simulation*. Springer, 15–34.
- [18] Jessy George Smith, Sai Kirthi Ponnuru, and Mandar Patil. 2016. Detection of aggressive driving behavior and fault behavior using pattern matching. In *Proc. IEEE ICACCI 2016*.
- [19] Huachun Tan, Guangdong Feng, Jianshuai Feng, Wuhong Wang, Yu-Jin Zhang, and Feng Li. 2013. A tensor-based method for missing traffic data completion. *Transportation Research Part C: Emerging Technologies* 28 (2013), 15–27.
- [20] Takaaki Umedu, Kumiko Isu, Teruo Higashino, and Chai-Keong Toh. 2010. An intervehicular-communication protocol for distributed detection of dangerous vehicles. *IEEE Transactions on Vehicular Technology (TVT)* 59, 2 (2010), 627–637.
- [21] Pengyan Wang, Fu Yanjie, Jiawei Zhang, Wang Pengfei, Zheng Yu, and Charu Aggarwal. 2018. You Are How You Drive: Peer and Temporal-Aware Representation Learning for Driving Behavior Analysis. In *Proc. ACM SIGKDD 2018*.
- [22] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In *Proc. ACM SIGKDD 2014*.
- [23] Hao Wu, Jianguyun Mao, Weiwei Sun, Baihua Zheng, Hanyuan Zhang, Ziyang Chen, and Wei Wang. 2016. Probabilistic robust route recovery with spatio-temporal dynamics. In *Proc. ACM SIGKDD 2016*.
- [24] Zhichen Wu, Jianda Li, Jiadi Yu, Yanmin Zhu, Guangtao Xue, and Minglu Li. 2016. L3: Sensing driving conditions for vehicle lane-level localization on highways. In *Proc. IEEE INFOCOM 2016*.
- [25] Tong Xu, Hengshu Zhu, Xiangyu Zhao, Qi Liu, Hao Zhong, Enhong Chen, and Hui Xiong. 2016. Taxi Driving Behavior Analysis in Latent Vehicle-to-Vehicle Networks: A Social Influence Perspective. In *Proc. ACM SIGKDD 2016*.
- [26] Shiyang Yan, Yuxuan Teng, Jeremy S Smith, and Bailing Zhang. 2016. Driver behavior recognition based on deep convolutional neural networks. In *Proc. IEEE ICNC-FSKD 2016*.
- [27] Chuang Wen You, Nicholas D. Lane, Fanglin Chen, Rui Wang, Zhenyu Chen, Thomas J. Bao, Martha Montes-De-Oca, Yuting Cheng, Mu Lin, and Lorenzo Torresani. 2013. CarSafe app: alerting drowsy and distracted drivers using dual cameras on smartphones. In *Proc. ACM MobiSys 2013*.
- [28] Jiadi Yu, Hongzi Zhu, Haofu Han, Yingying Jennifer Chen, Jie Yang, Yanmin Zhu, Zhongyang Chen, Guangtao Xue, and Minglu Li. 2015. SenSpeed: Sensing Driving Conditions to Estimate Vehicle Speed in Urban Environments. *IEEE Transactions on Mobile Computing (TMC)* 15, 1 (2015), 202–216.
- [29] Yu Zheng. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 29.