This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



# Fine-grained Abnormal Driving Behaviors Detection and Identification with Smartphones

Jiadi Yu,Zhongyang Chen, Yanmin Zhu, Yingying Chen, Linghe Kong and Minglu Li

**Abstract**—Real-time abnormal driving behaviors monitoring is a corner stone to improving driving safety. Existing works on driving behaviors monitoring using smartphones only provide a coarse-grained result, i.e. distinguishing abnormal driving behaviors from normal ones. To improve drivers' awareness of their driving habits so as to prevent potential car accidents, we need to consider a fine-grained monitoring approach, which not only detects abnormal driving behaviors but also identifies specific types of abnormal driving behaviors, i.e. *Weaving*, *Swerving*, *Sideslipping*, *Fast U-turn*, *Turning with a wide radius* and *Sudden braking*. Through empirical studies of the 6-month driving traces collected from real driving environments, we find that all of the six types of driving behaviors have their unique patterns on acceleration and orientation. Recognizing this observation, we further propose a fine-grained abnormal *D*riving behavior *D*etection and i*D*entification system, $D^3$, to perform real-time high-accurate abnormal driving behaviors monitoring using smartphone sensors. We extract effective features to capture the patterns of abnormal driving behaviors. After that, two machine learning methods, *Support Vector Machine* (SVM) and *Neuron Networks* (NN), are employed respectively to train the features and output a classifier model which conducts fine-grained abnormal driving behaviors detection and identification. From results of extensive experiments with 20 volunteers driving for another 4 months in real driving environments, we show that $D^3$ achieves an average total accuracy of $95.36\%$ with SVM classifier model, and $96.88\%$ with NN classifier model.

**Index Terms**—Abnormal driving behavior, Accelerometer, Orientation sensor, Support Vector Machine (SVM), Neuron Networks (NN)

## 1 Introduction

According to the statistics from World Health Organization (WHO), traffic accidents have become one of the top 10 leading causes of death in the world[1]. Specifically, traffic accidents claimed nearly 3500 lives each day in 2014. Studies show that most traffic accidents are caused by human factors, e.g. drivers' abnormal driving behaviors [2]. Therefore, it is necessary to detect drivers' abnormal driving behaviors to alert the drivers or report Transportation Bureau to record them.

Although there has been works[3][4][5] on abnormal driving behaviors detection, the focus is on detecting driver's status based on pre-deployed infrastructure, such as alcohol sensor, infrared sensor and cameras, which incur high installation cost. Since smartphones have received increasing popularities over the recent years and blended into our daily lives, more and more smartphone-based vehicular applications[6][7][8][21] are developed in Intelligent Transportation System. Driving behavior analysis is also a popular direction of smartphone-based vehicular applications. However, existing works[9][10] on driving behaviors detection using smartphones can only provide a coarse-grained result using thresholds, i.e. distinguishing abnormal driving behaviors from normal ones. Since thresholds may be affected by car type and sensors' sensitivity, they cannot accurately distinguish the differences in various driving behavioral patterns. Therefore, Those solutions cannot provide fine-grained identification, i.e. identifying specific types of driving behaviors.

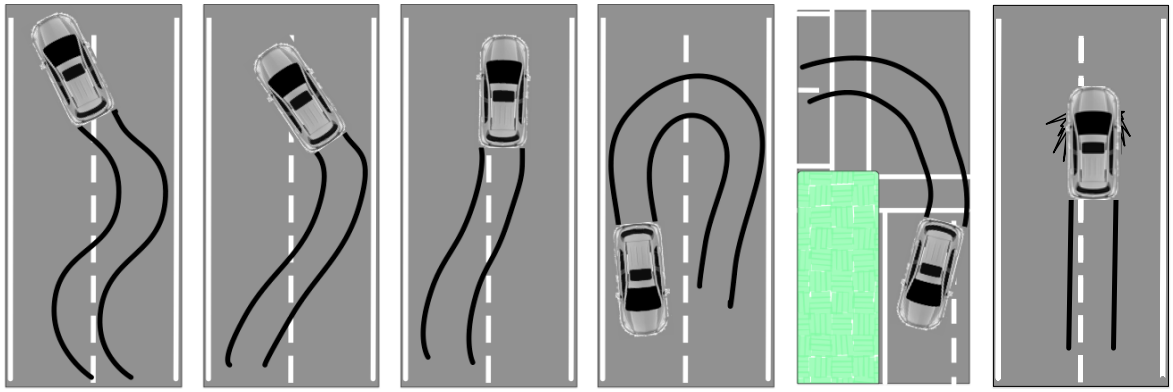
Moving along this direction, we need to consider a fine-grained abnormal driving behaviors monitoring approach, which uses smartphone sensors to not only detect abnormal driving behaviors but also identify specific types of the driving behaviors without requiring any additional hardwares. The fine-grained abnormal driving behaviors monitoring is able to improve drivers' awareness of their driving habits as most of the drivers are over-confident and not aware of their reckless driving habits. Additionally, some abnormal driving behaviors are unapparent and easy to be ignored by drivers. If we can identify drivers' abnormal driving behaviors automatically, the drivers can be aware of their bad driving habits, so that they can correct them, helping to prevent potential car accidents. Furthermore, if the results of the monitoring could be passed back to a central server, they could be used by the police to detect drunken-driving automatically or Vehicle Insurance Company to analyze the policyholders' driving habits.

According to [11], there are six types of abnormal driving behaviors defined, and they are illustrated in Fig.1. *Weaving* (Fig.1(a)) is driving alternately toward one side of the lane and then the other, i.e. serpentine driving or driving in S-shape; *Swerving* (Fig.1(b)) is making an abrupt redirection when driving along a generally straight course; *Sideslipping* (Fig.1(c)) is when driving in a generally straight line, but deviating from the normal driving direction; *Fast U-turn* (Fig.1(d)) is a fast turning in U-shape, i.e. turning round (180 degrees) quickly and then driving along the opposite direction; *Turning with a wide radius* (Fig.3(e)) is turning

- *J. Yu, Z. Chen, Y. Zhu, L. Kong and M. Li are with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.*
  *E-mail: {jiadiyu, wodoo2474, yzhu, linghe.kong mlli}@sjtu.edu.cn*
- *Y. Chen is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA.*
  *E-mail: yingying.chen@stevens.edu*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing





(a) (b) (c) (d) (e) (f)

Fig. 1. Six types of abnormal driving behaviors: (a) Weaving, (b) Swerving, (c) Sideslipping, (d) Fast U-turn, (e) Turning with a wide radius, (f) Sudden braking.

cross an intersection at such an extremely high speed that the car would drive along a curve with a big radius, and the vehicle sometimes appears to drift outside of the lane, or into another line; *Sudden braking* (Fig.3(f)) is when the driver slams on the brake and the vehicle's speed falls down sharply in a very short period of time.

This work uses smartphone sensing and machine learning techniques. By extracting unique features from the readings of smartphone sensors, we can detect and identify the six types of abnormal driving behaviors above. To realize a fine-grained abnormal driving behaviors detection and identification, we face the following great challenges. First, patterns of driving behaviors need to be identified from readings of smartphone sensors. Second, the noise of smartphone sensors' readings should be removed. Finally, the solution should be lightweight and computational feasible on smartphones.

In this paper, we first set out to investigate effective features from smartphone sensors' readings that are able to depict each type of abnormal driving behavior. Through empirical studies of the 6-month driving traces collected from smartphone sensors of 20 drivers in a real driving environment, we find that each type of abnormal driving behaviors has its unique patterns on readings from accelerometers and orientation sensors. By extracting unique features from readings of smartphones' accelerometer and orientation sensor, we first identify 16 representative basic features to capture the patterns of driving behaviors, then generate 136 polynomial features based on the 16 features, and obtain 152 features in total. Then, we train those features through two machine learning methods respectively, *Support Vector Machine* (SVM) and *Neuron Networks* (NN), to generate a classifier model which could clearly identify each of driving behaviors (i.e. the normal driving behaviors as well as the six types of abnormal ones). Based on the classifier model, we propose an abnormal *D*riving behavior *D*etection and i*D*entification system, $D^3$, which can realize a fine-grained abnormal driving behaviors detection and identification in real-time using smartphone sensors. Our prototype implementation of $D^3$ on Android-based mobile devices verifies the feasibility of using $D^3$ in real driving environments.

We highlight our main contributions as follows:

- We identify 16 representative basic features and 136 polynomial features to capture the patterns of abnormal driving behaviors by empirically analyzing the 6-month driving traces collected from real driving environments.
- We use two machine learning method respectively, SVM and NN, to train the features of driving behaviors and obtain a classifier model which can not only distinguish abnormal driving behaviors from normal ones but also identify specific types of abnormal driving behavior.
- We propose a fine-grained abnormal driving behaviors detection and identification system, $D^3$, to perform real-time high-accurate abnormal driving behaviors monitoring with smartphones. The fine-grained system can inform drivers of their abnormal driving behaviors which otherwise may be ignored by them so as to improve their awareness of driving habits.
- We conduct extensive experiments in real driving environments. The result shows that in real driving environments, $D^3$ can identify specific types of abnormal driving behaviors in real time with an average total accuracy of 95.36% with SVM classifier model, and 96.88% with NN classifier model.

The rest of the paper is organized as follows: The related work is reviewed in Section 2. In Section 3, we analyze the acceleration and orientation patterns of the six specific types of abnormal driving behaviors from smartphone sensors' readings. We present the design details of our abnormal driving behaviors detection and identification system, $D^3$, in Section 4. We evaluate the performance of $D^3$ and present the results in Section 5. Finally, we give the conclusion remarks in Section 6.

## 2 Related Work

In this section, we review the existing works on driving behaviors detection, which can be categorized as follows.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



***Detection using pre-deployed infrastructure***: [3] uses an EGG equipment which samples the driver's EGG signals to detect drowsiness during car driving. [12] uses infrared sensors monitoring the driver's head movement to detect drowsy driving. [13] captures the driver's facial images using a camera to detect whether the driver is drowsy driving by image processing. In [4], GPS, cameras, alcohol sensor and accelerometer sensor are used to detect driver's status of drunk, fatigued, or reckless. However, the solutions all rely on pre-deployed infrastructures and additional hardwares that incur installation cost. Moreover, those additional hardwares could suffer the difference of day and night, bad weather condition and high maintenance cost.

***Detection using smartphone sensors***: To eliminate the need of pre-deployed infrastructures and additional hardwares, recent studies concentrate on using smartphones to detect abnormal driving behaviors. In particular, [14] uses accelerometers, magnetometers and GPS sensors to determine whether high-risk motorcycle maneuvers or accidents occur. [15] uses accelerometers, gyroscopes and magnetometers to estimate a driver's driving style as *Safe* or *Unsafe*. [9][10] use accelerometers to detect drunk driving and sudden driving maneuver, respectively. The works are similar in that they perform a coarse-grained driving behavior detection which uses some thresholds to find out abnormal driving behaviors. Nevertheless, thresholds may be affected by car type and sensors' sensitivity so that they cannot accurately distinguish the differences in various driving behavioral patterns. Therefore, none of existing works can realize fine-grained identification.

Our work uses smartphone sensing and machine learning techniques to realize a fine-grained abnormal driving behaviors detection and identification. Although machine learning technique already is used to some activity recognition work [18][19][20], our work is first to identify driving activities using machine learning technique. In [18][19][20], since activities are instantaneous, pattern of activities is simple. So features of activities' pattern would be identified easily. However, in real driving environments, since the time duration of some driving behavior is long, not instantaneous, such as Weaving, the system need to determine the beginning and ending of the driving behavior first. Extracting and selecting effective features of each type of abnormal driving behavior would be more complex.

## 3 Driving Behavior Characterization

In this section, we first describe the data collection process for driving behavior samples from real driving environments. Then we analyze patterns of each type of driving behavior from smartphone sensors' readings.

### 3.1 Collecting Data from Smartphone Sensors

We develop an Andriod-based App to collect readings from the 3-axis accelerometer and the 3-axis orientation sensor. We align the two coordinate systems in the smartphone and in the vehicle by making the accelerometer's y-axis along the moving direction of the vehicle. Therefore, we could monitor the vehicle's acceleration and orientation by retrieving readings from the smartphone's accelerometer and orientation sensor.

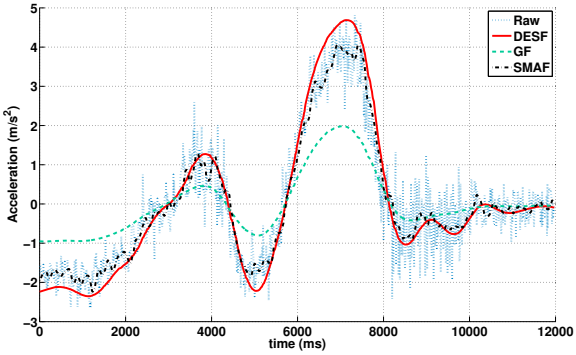



Fig. 2. Raw acceleration and the resulted acceleration after applying SMAF, DESF and GF

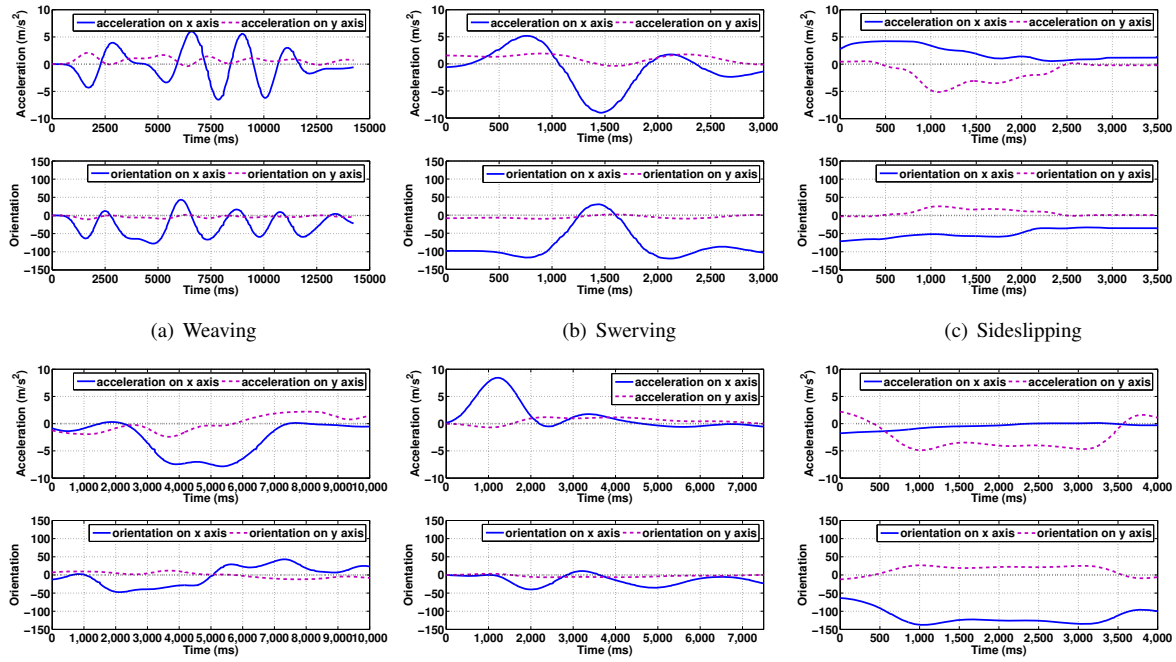
We collect traces from the accelerometers and orientation sensors' readings on 20 drivers with distinct vehicles from Jan. 11 to July 12, 2014. Each driver fixes a smartphone along with a Car Digital Video Recorder (DVR) in his/her vehicle within daily natural driving. The smartphone and Car DVR record the sensors' readings and all objective driving behaviors, respectively. The 20 drivers keep collecting data in their daily driving, including commute to work, shopping, touring and so on. Those 20 drivers live in different communities and they have different commute routes. On average, each driver may drive 60 to 80 kilometers per day. 20 smartphones of 5 different types are used in our data collection, i.e. Huawei Honor3C, ZTE U809, SAMSUNG Nexus3, SAMSUNG Nexus4 and HTC sprint, four devices for each type. After that, we ask 9 experienced drivers to watch the videos recorded by the Car DVR and recognize all types of abnormal driving behaviors from the 6-month traces, i.e. *Weaving*, *Swerving*, *Sideslipping*, *Fast U-turn*, *Turning with a wide radius* or *Sudden braking*. In total, we obtain 4029 samples of abnormal driving behaviors from the collected traces, which is viewed as the ground truth of abnormal driving behaviors.

### 3.2 Low-pass Filtering

Due to environmental dynamics, there are many high frequency noises in our collected raw data which has an impact on analyzing the dynamics of different driving behaviors. Thus we conduct low-pass filtering to the collected raw data first to remove the high frequency noise and yet capture the statistical features presented in the traces. There are various low-pass filters that have different performances. In this paper, we experiment with three low-pass filters in our dataset, i.e. simple moving average filter (SMAF), dynamic exponential smoothing filter (DESF) and Gaussian filter (GF) [16], and compare their performances to finally select one that efficiently eliminate high frequency noises while preserving the effective dynamics of different driving behaviors to the utmost extent. Fig.2 shows an example including the raw acceleration reading from smartphones'

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing






(a) Weaving (b) Swerving (c) Sideslipping

(d) Fast U-turn (e) Turning with a wide radius (f) Sudden braking

Fig. 3. The acceleration and orientation patterns of the six types of abnormal driving behaviors from an accelerometer and an orientation sensor's readings.

accelerometer and the resulted accelerations after applying the three low-pass filters. It can be seen that SMAF cannot perfectly remove high frequency noises and GF cannot preserve effective features of driving behaviors. Compared with the rough trend of the acceleration preserved by SMAF and GF, DESF is able to not only return the nicely fitted curve but also preserve effective features of different driving behaviors. This is because DESF is an exponential smoother that changes its smoothing factor dynamically according to previous samples. Hence, we select DESF as the low-pass filter used in our $D^3$ to remove high frequency noises as well as preserve the effective dynamics of driving behaviors to the utmost extent.

### 3.3 Analyzing Patterns of Abnormal Driving Behaviors

After high frequency noises are removed in the collected data using the low-pass filter, we can analyze the acceleration and orientation patterns of each type of abnormal driving behaviors. Let $acc_x$ and $acc_y$ be the acceleration on x-axis and y-axis, respectively. Let $ori_x$ and $ori_y$ be the orientation on x-axis and y-axis, respectively.

#### 3.3.1 Weaving

Fig.3(a) shows the acceleration and orientation patterns of weaving from an accelerometer and orientation sensor's readings. We observe from this figure that there is a drastic fluctuation on $acc_x$ and this fluctuation continues for a period of time, while $acc_y$ keeps smooth. Thus, both the standard deviation and the range of $acc_x$ are very large and the time duration is long. The mean value of $acc_x$ is around zero. In addition, the orientation values have similar patterns as acceleration values.

#### 3.3.2 Swerving

Fig.3(b) shows the acceleration and orientation patterns of swerving. Since swerving is an abrupt, instant behavior, the time duration is very short. When swerving occurs, there is a great peak on both $acc_x$ and $ori_x$. Thus, the range and standard deviation of both $acc_x$ and $ori_x$ are large, and the mean value is not near zero. In addition, both $acc_y$ and $ori_y$ are flat during swerving.

#### 3.3.3 Sideslipping

Fig.3(c) shows the acceleration and orientation patterns of sideslipping. When sideslipping occurs, $acc_y$ falls down sharply. Thus, the minimum value and mean value of $acc_y$ are negative, and the range of $acc_y$ is large. In addition, $acc_x$ in sideslipping is not near zero. If the vehicle slips toward the right side, $acc_x$ would be around a positive value, while if left, then negative. The mean value of $acc_x$ thus is not near zero. When it comes to orientation, there are no obvious changes. Moreover, since sideslipping is an abrupt driving behavior, the time duration is short.

#### 3.3.4 Fast U-turn

Fig.3(d) shows the acceleration and orientation patterns of fast U-turn. When a driver turns right or left fast in U-shape, $acc_x$ rises quickly to a very high value or drops fast to a very low value, respectively. Moreover, the value would last for a period of time. The standard deviation of $acc_x$ thus is large on the beginning and ending of a fast U-turn, the mean value of $acc_x$ is far from zero and the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



range of $acc_x$ is large. When it comes to $acc_y$, there are no obvious changes. Moreover, $ori_x$ would pass over the zero point. Specifically, $ori_x$ would change either from positive to negative or from negative to positive, depending on the original driving direction. Thus, the standard deviation and value range of $ori_x$ would be large. The mean values in first half and second half of $ori_x$ would be of opposite sign, i.e. one positive and the other negative. It may take a period of time to finish a fast U-turn, so its time duration is long.

#### 3.3.5 Turning with a wide radius

The acceleration and orientation patterns of turning with a wide radius are shown in Fig.3(e). When turning at an extremely high speed, $acc_x$ sees a high magnitude for a period of time, while the $acc_y$ is around zero. Thus, the mean value of $acc_x$ is far from zero and the standard deviation of $acc_x$ is large. When it comes to orientation, $ori_x$ sees a fluctuation, while $ori_y$ keeps smooth. The standard deviation of $ori_x$ thus is relatively large, and the mean value of $ori_x$ is not near zero since the driving direction is changed. It may take a period of time to finish a turning with a wide radius, so the time duration is long.

#### 3.3.6 Sudden braking

Fig.3(f) shows the acceleration and orientation patterns of sudden braking. When a vehicle brakes suddenly, $acc_x$ remains flat while $acc_y$ sharply downs and keeps negative for some time. Thus, the standard deviation and value range of $acc_x$ are small. On $acc_y$, the standard deviation is large at the beginning and ending of a sudden braking and the range of $acc_y$ is large. Moreover, there are no obvious changes on both $ori_x$ and $ori_y$. Since sudden braking is an abrupt driving behavior, the time duration is short.

#### 3.3.7 Normal Driving Behavior

Normal driving behavior means smooth and safe driving with few and small fluctuations. Since there are few drastic actions in a normal driving behavior, the values on both $acc_x$ and $acc_y$ are not very large. So the mean, standard deviation, maximum and minimum values in acceleration on x-/y-axis are near zero. When it comes to orientation, a normal driving behavior presents smooth most of time. So the standard deviation and range of orientation are small.

Based on the analysis above, we find that each driving behavior has its unique features, e.g. standard deviation, mean, maximum, minimum, value range on $acc_x$, $acc_y$, $ori_x$ and $ori_y$, as well as the time duration. Therefore, we could use those features to identify specific types of abnormal driving behaviors using machine learning techniques.

## 4 System Design

In this section, we present the design of our proposed system, $D^3$, which detects abnormal driving behaviors from normal ones and identifies different abnormal types using smartphone sensors. $D^3$ does not depend on any pre-deployed infrastructures and additional hardwares.

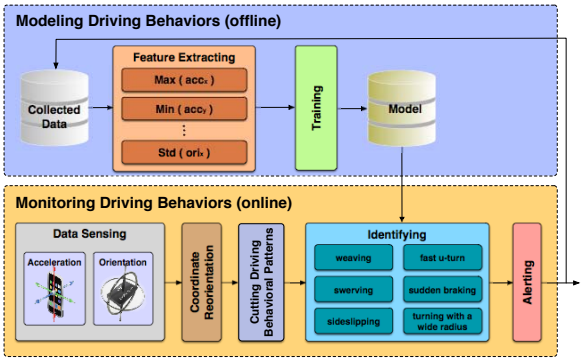



Fig. 4. System architecture.

### 4.1 Overview

In our system, $D^3$, abnormal driving behaviors could be detected and identified by smartphones according to readings from accelerometers and orientation sensors. Fig.4 shows the architecture of $D^3$. The whole system is separated into offline part-*Modeling Driving Behaviors* and online part-*Monitoring Driving Behaviors*.

In the offline part, *Modeling Driving Behaviors*, $D^3$ trains a classifier model using machine learning techniques based on the collected data, which could identify specific types of driving behaviors. In the *Feature Extracting*, effective features are extracted from specific types of driving behavioral patterns on acceleration and orientation. Afterwards, the features are trained in the *Training* and a classifier model would be generated which can realize fine-grained identification for various types of driving behaviors. Finally, the classifier model is output and stored to *Model Database*.

The online part, *Monitoring Driving Behaviors*, is installed on smartphones which senses real-time vehicular dynamics to detect and identify abnormal driving behaviors. $D^3$ first senses the vehicles' acceleration and orientation by smartphones' accelerometers and orientation sensors. After getting real-time readings from the accelerometer and the orientation sensor, the *Coordinate Reorientation* is performed to align the smartphone's coordinate system with the vehicle's using the method in [6][7][21]. Then, in the *Cutting Driving Behavioral Patterns*, the beginning and ending of a driving behavior are found out from accelerometer and orientation sensor's readings. Afterwards, in *Identifying*, $D^3$ extracts features from patterns of the driving behaviors, then identifies whether one of the abnormal driving behaviors occurs based on the classifier model trained in *Modeling Driving Behaviors*. Finally, if any of the abnormal driving behaviors were identified, a warning message would be sent to receivers by the *Alerting*.

### 4.2 Extracting and Selecting Effective Features

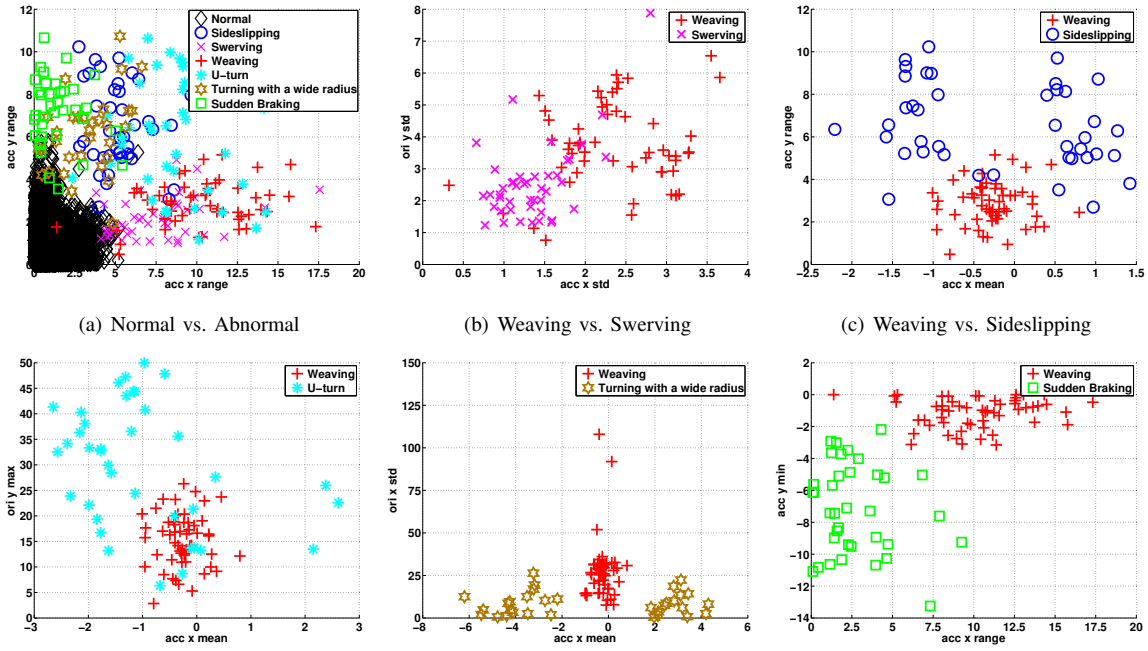
In $D^3$, we use machine learning techniques to identify fine-grained abnormal driving behaviors. The process of feature extraction and selection is discussed in the following.

#### 4.2.1 Feature Extraction

When machine learning algorithms are processed, representative tuple of features rather than raw data is a more

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing






(a) Normal vs. Abnormal (b) Weaving vs. Swerving (c) Weaving vs. Sideslipping

(d) Weaving vs. Fast U-turn (e) Weaving vs. Turning with a wide radius (f) Weaving vs. Sudden braking

Fig. 5. Some effective features for identifying normal driving behavior from abnormal ones and weaving behavior from other five abnormal driving behaviors.

effective input. Thus, it is necessary to extract effective features from driving behavioral patterns. According to the analysis in Section 3, each driving behavior has its unique patterns on $acc_x$, $acc_y$, $ori_x$, $ori_y$ and time duration ($t$). The main difference between various driving behaviors lies in the maximum, minimum, value range, mean, and standard deviation of $acc_x$, $acc_y$, $ori_x$ and $ori_y$ and $t$. Therefore, those values can be used as features for training. However, not all of them are equally effective for abnormal driving behaviors' detection and identification.

### 4.2.2 Feature Selection

In order to select the really effective features, we analyze the collected traces. Fig.5 shows some of the effective features which distinguish abnormal driving behaviors from normal ones and distinguish weaving from the other five abnormal driving behaviors.

Fig.5(a) shows the difference between normal and abnormal driving behaviors in a 2-dimensional features tuple (i.e. $range_{acc,x}$ and $range_{acc,y}$). It can be seen that the two features can clearly discriminate normal and abnormal driving behaviors. Therefore, we manage to distinguish abnormal driving behaviors from normal ones with only two features.

In fact, additionally to the two features shown in Fig.5(a), some other combinations of a 2-dimensional features tuple (i.e. any 2 out of $t$, $max_{ori,x}$, $max_{ori,y}$, $\sigma_{ori,x}$, $\sigma_{ori,y}$, $\sigma_{acc,x}$, $range_{acc,x}$, $min_{acc,y}$ and $range_{acc,y}$) also manage to distinguish abnormal driving behaviors from normal ones.

Although we can distinguish abnormal driving behaviors from normal ones using a 2-dimensional features tuple, we fail to differentiate the six types of abnormal behaviors from each other only using 2-dimensional features. As the example shown in Fig.5(a), the six types of abnormal driving behaviors are mixed with each other. Nevertheless, they could be differentiated pairwise with a 2-dimensional features tuple. In other words, although the six abnormal driving behaviors cannot be differentiated from each other at the same time, any two among them can be differentiated intuitively by a 2-dimensional features tuple. Taking weaving for example (see Fig.5(b) - Fig.5(f)), weaving can be distinguished from the other five abnormal driving behaviors using a 2-dimensional features tuple. For instance, in Fig.5(b), weaving and swerving can be discriminated from

TABLE 1
Basic Features

| Feature | Description |
|---|---|
| $range_{acc,x}$ | subtraction of maximum minus minimum value of $acc_x$ |
| $range_{acc,y}$ | subtraction of maximum minus minimum value of $acc_y$ |
| $\sigma_{acc,x}$ | standard deviation of $acc_x$ |
| $\sigma_{acc,y}$ | standard deviation of $acc_y$ |
| $\sigma_{ori,x}$ | standard deviation of $ori_x$ |
| $\sigma_{ori,y}$ | standard deviation of $ori_y$ |
| $\mu_{acc,x}$ | mean value of $acc_x$ |
| $\mu_{acc,y}$ | mean value of $acc_y$ |
| $\mu_{ori,x}$ | mean value of $ori_x$ |
| $\mu_{ori,y}$ | mean value of $ori_y$ |
| $\mu_{acc,x,1}$ | mean value of $1^{st}$ half of $acc_x$ |
| $\mu_{acc,x,2}$ | mean value of $2^{nd}$ half of $acc_x$ |
| $max_{ori,x}$ | maximum value of $ori_x$ |
| $max_{ori,y}$ | maximum value of $ori_y$ |
| $min_{acc,y}$ | minimum value of $acc_y$ |
| $t$ | time duration between the begining and the ending of a driving behavior |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



each other using $\sigma_{ori,y}$ and $\sigma_{acc,x}$. Similarly, other abnormal driving behaviors can also be pairwise discriminated using 2-dimensional features tuples.

Based on the collected traces, we investigate all possible pairwise cases. In each case, we find out several effective features conductive to distinguishing one driving behavior from another. Finally, we identify 16 effective features that are able to capture the patterns of different types of abnormal driving behaviors, as listed in TABLE 1.

In depth analysis, we find that, in some cases, the decision boundary, i.e. the boundary that divides two types of abnormal driving behaviors into seperate areas, is some form of a circle or an ellipse, such as the boundary in Fig.5(c) and Fig.5(d). In these cases, it is better to add some polynomial features to build a better classifier model so that it could distinguish the two abnormal driving behaviors with a non-linear decision boundary. Taking Fig.5(c) for example, we have two basic features $\mu_{acc,x}$ and $range_{acc,y}$ to distinguish weaving samples from sideslipping samples. Since the expected decision boundary which distinguishes the two types of samples looks similar to a circle that weaving samples are in the circle and sideslipping samples are outside the circle. In this case, we need some polynomial features rather than only the basic two features, $\mu_{acc,x}$ and $range_{acc,y}$, to generate such a circular boundary through machine learning. Thus we generate three additional polynomial features, $\mu^2_{acc,x}$, $range^2_{acc,y}$ and $\mu_{acc,x} \times range_{acc,y}$ to build a better classifier model.

We only add quadratic items as new features and omit the higher order ones (e.g. the cubic features $\mu^3_{acc,x}$, $range^3_{acc,y}$, $\mu^2_{acc,x} \times range_{acc,y}$, $\mu_{acc,x} \times range^2_{acc,y}$ and some more higher order ones.) This is because firstly quadratic items (i.e. squared or product items) are able to generate circular decision boundary. And secondly, although higher order items can also generate circular decision boundary, it may lead to over fitting phenomenon after learning from the training set. In other words, we may get an over fitting classifier model that suits every sample perfectly in the training set but have low performance for newly coming samples in the test set or in real time.

Based on the above analysis, we select 152 features in total to build a classifier model. Among the 152 features, 16 are basic features as listed in TABLE 1, next 16 are the squared versions of the basic features, and the rest are the products of any two different basic features.

### 4.3 Training a Fine-grained Classifier Model to Identify Abnormal Driving Behaviors

After feature extracting, we obtain a tuple of features for each driving behavior. Then a classifier model is trained based on the tuples for all driving behaviors through machine learning techniques [17] to identify various driving behaviors. We use two multi-class machine learning methods, i.e. *Support Vector Machine* (SVM) [18][19] and *Neural Networks* (NN) to train the classifier model. For each driving behavior, the input into SVM is in the form of <16-dimensional features, label>, where the 16-dimensional features are the basic features tuples obtained from the *Feature Extracting* and the label is the type of the driving behavior. The input into NN is in the form of <152-dimensional features, label>, where the 152-dimensional features are the basic as well as polynomial features mentioned in Section 4.2.

#### 4.3.1 SVM

The cores in SVM are the *kernel* and the *similarity function*. A *kernel* is a landmark, and the *similarity function* computes the similarity between an input example and the kernels. Specifically, assume that our training set contains $m$ samples, and each sample are 16-dimensional (i.e. the 16-dimensional features), denoted by

$$\begin{aligned} x^{(i)} &= (x^{(i)}_1, x^{(i)}_2, \cdots, x^{(i)}_{16}), \\ i &= 1, 2, \cdots, m, \end{aligned} \tag{1}$$

where $x^{(i)}$ is the $i^{th}$ sample, and $x^{(i)}_j$ means the $j^{th}$ feature of $x^{(i)}$. When SVM starts, all input samples $(x^{(1)}, x^{(2)}, \cdots, x^{(m)})$ are selected as kernels, recorded as $l^{(1)}, l^{(2)}, \cdots, l^{(m)}$. Note that $x^{(i)} = l^{(i)}$ for $i = 1, 2, \cdots, m$. Afterwards, for each sample, SVM computes its similarity between the kernels by

$$\begin{aligned} f^{(i)}_j &= e^{-\frac{\|x^{(i)} - l^{(j)}\|^2}{2\sigma^2}}, \\ i, j &= 1, 2, \cdots, m, \end{aligned} \tag{2}$$

where $f^{(i)}_j$ is the similarity between input sample $x^{(i)}$ and the kernel $l^{(j)}$, $\sigma$ is a parameter defined manually, and $\|x^{(i)} - l^{(j)}\|^2$ is the distance between $x^{(i)}$ and $l^{(j)}$ calculated by

$$\begin{aligned} \|x^{(i)} - l^{(j)}\|^2 &= \sum_{k=1}^{16} (x^{(i)}_k - l^{(j)}_k)^2, \\ i, j &= 1, 2, \cdots, m. \end{aligned} \tag{3}$$

In SVM, $m$ 16-dimensional input samples (i.e. $x^{(1)}, x^{(2)}, \cdots, x^{(m)}$) would be converted into $m$ m-dimentional similarity features (i.e. $f^{(1)}, f^{(2)}, \cdots, f^{(m)}$), since for each $x^{(i)}$, the similarity between $x^{(i)}$ and any $l^{(j)}$ in $l^{(1)}, l^{(2)}, \cdots, l^{(m)}$ are calculated by Equation 2. With the new features $f = (f^{(1)}, f^{(2)}, \cdots, f^{(m)})$, a cost function $J(\theta)$ (see Equation 4) calculated from $f$ would be minimized to find optimal $\theta$.

$$J(\theta) = C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{m} \theta_j^2, \tag{4}$$

where $C$ is a parameter defined manually, $y^{(i)}$ is the label of $i^{th}$ input example (i.e. the label of $x^{(i)}$), $\theta^T$ means $\theta$ transpose and

$$\begin{aligned} cost_1(\theta^T f^{(i)}) &= \log(\frac{1}{1 + e^{-\theta^T f^{(i)}}}), \\ cost_0(\theta^T f^{(i)}) &= \log(1 - \frac{1}{1 + e^{-\theta^T f^{(i)}}}), \end{aligned} \tag{5}$$

and

$$\theta^T f^{(i)} = \theta_1 f^{(i)}_1 + \theta_2 f^{(i)}_2 + \cdots + \theta_m f^{(i)}_m. \tag{6}$$

The classifier model is finally determined by the optimal $\theta$. In a word, SVM trains the inputs and then output a classifier

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



model which conducts fine-grained identification to the six types of abnormal driving behaviors.

In SVM, the similarity or distance between two sampling points are used to determine whether the two sampling points belong to the same abnormal driving behavioral type. If the 136 polynomial features are used as inputs in addition to the 16 basic features, it heavily increases the computational cost of building the classifier model, but cannot obviously improve the performance. Hence, we only use the 16 basic features as input and it is unnecessary to use the additional 136 polynomial features.

#### 4.3.2 NN

NN method mimics the learning process of a human brain, which is it accepts a group of inputs and then do some computations through a series of neurons to output results. The NN method performs better as the number of features increases. In the NN algorithm, there are several layers, and each layer has several computational units or neurons to do various computations based on the inputs from the former layer. The first layer is also known as the input layer, in our method, there are 152 neurons in the first layer, and each neuron contains the value of a feature. The last layer is known as the output layer, there are 7 units in the output layer, each represents a type of driving behavior (i.e. 6 abnormal type and 1 normal type). The layers between input and output layers are called hidden layers. Assume there are $L$ hidden layers (i.e. $h^{(1)}, h^{(2)}, \cdots, h^{(L)}$) and in each hidden layer there are $S$ units (e.g. in $h^{(1)}$, there are units named $h_1^{(1)}, h_2^{(1)}, \cdots, h_S^{(1)}$).

The $i^{th}$ units in first hidden layer can be calculated through:

$$h_i^{(1)} = g(\Theta_{i,0}^{(1)} x_0 + \Theta_{i,1}^{1} x_{(1)} + \cdots + \Theta_{i,152}^{(1)} x_{152}), \quad i = 1, 2, \cdots, S. \tag{7}$$

where $\Theta^{(1)}$ is a $S \times 153$ dimensional matrix of weights that controls function mapping from input layer to the first hidden layer, $x$ is a 153 dimensional vector that contains the 152 features and one bias bit $x_0 = 1$, and $g()$ is a computational function in any unit in the hidden layers (i.e. also called activation of the unit), where

$$g(z) = \frac{1}{1 + e^{-z}}, \tag{8}$$

where $z$ is the input of $g()$ and $z$ is $\Theta_{i,0}^{(1)} x_0 + \Theta_{i,1}^{1} x_{(1)} + \cdots + \Theta_{i,152}^{(1)} x_{152}$ in Equation 7.

After we obtain the values in the units in the first layer, we can forward propagate the values and obtain the units in the next hidden layer until we reach to the last layer. The unit $i$ in $(j+1)^{th}$ hidden layer $h_i^{(j+1)}$ can be calculated based on the units in $j^{th}$ layer according to

$$h_i^{(j+1)} = g(\Theta_{i,0}^{(j+1)} h_0^{(j)} + \Theta_{i,1}^{(j+1)} h_1^{(j)} + \cdots + \Theta_{i,S}^{(j+1)} h_S^{(j)}), \quad i = 1, 2, \cdots, S, \tag{9}$$

where $\Theta^{(j+1)}$ is a $S \times (S+1)$ dimensional matrix of weights that controls function mapping from layer $j$ to layer $j+1$, $h_1^{(j)}, h_2^{(j)}, \cdots, h_S^{(j)}$ is the units' values in layer $j$ and also act as inputs into layer $j+1$ and $h_0^{(j)}$ is a bias bit additionally in the input from layer $j$ to layer $j+1$ which is equal to 1. $g()$ is the same function as in Equation 8.

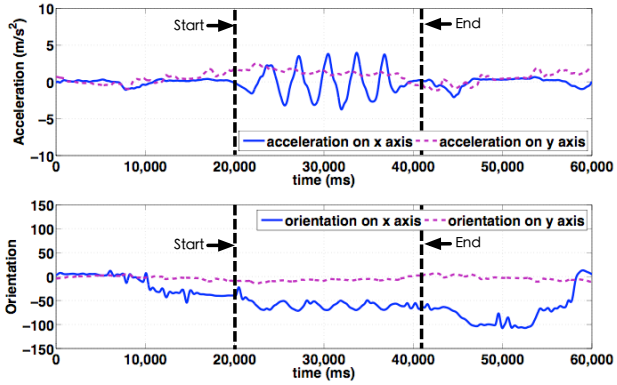



Fig. 6. The acceleration and orientation patterns of one minute driving behaviors.

The units in the output layer can be calculated from the units in the $L^{th}$ hidden layer:

$$h_i^{(out)} = g(\Theta_{i,0}^{(out)} h_0^{(L)} + \Theta_{i,1}^{(out)} h_1^{(L)} + \cdots + \Theta_{i,S}^{(out)} h_S^{(L)}), \quad i = 1, 2, \cdots, 7. \tag{10}$$

After we obtain the values in the output layer, we are able to calculate the cost function in NN as following:

$$J(\Theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{7} y_k^{(i)} log(h_\Theta(x^{(i)}))_k + (1 - y_k^{(i)}) log(1 - (h_\Theta(x^{(i)}))_k)], \tag{11}$$

where $m$ is the number of samples in the training set, $y_k^{(i)}$ is the label of the $i^{th}$ sample (i.e. if the $i^{th}$ sample belongs to $k^{th}$ abnormal type, $y_k^{(i)} = 1$, otherwise 0), $x^{(i)}$ is the $i^{th}$ sample in the training set and $(h_\Theta(x^{(i)}))_k$ is the value of $k^{th}$ output unit on input $x^{(i)}$. $(h_\Theta(x^{(i)}))_k$ is a function of parameter $\Theta$, and $\Theta$ can be obtained by unrolling all matrices between layers in the network and then concatenate all of them. In NN algorithm, we can find an optimal $\Theta$ which minimizes the cost function $J(\Theta)$. The classifier model then can be built based on the optimal $\Theta$.

### 4.4 Detecting and Identifying Abnormal Driving Behaviors

After we obtain a classifier model, we are able to detect and identify abnormal driving behaviors in real driving environments using the model. In order to identify current driving behavior using the model, we should input features extracted from patterns of a driving behavior. $D^3$ thus need to determine the beginning and ending of the driving behavior first, i.e. cutting patterns of the driving behavior. Fig.6 shows the readings from a smartphone' accelerometer and orientation sensor on x-axis and y-axis in a one minute driving, which contains a weaving behavior. In Fig.6, the weaving behavior is sensed from its beginning to ending.

The method of sensing the beginning and ending of a driving behavior is proposed based on an analysis on the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



acceleration and orientation patterns of all types of driving behaviors. Specifically, when an abnormal driving behavior begins, the standard deviation of either the acceleration or the orientation values sharply rise to and keep a relatively high value until the driving behavior ends, while in most normal driving behaviors, the standard deviation always presents as low and smooth. Moreover, during an abnormal driving behavior, the magnitude of acceleration on either x-axis or y-axis presents an extremely high value, as illustrated in Section 3. But when driving normally, the magnitude of accelerations seldomly reaches to such a high value.

Therefore, It is simple but effective that we monitor the standard deviation of acceleration and orientation as well as the magnitude of acceleration of the vehicle from smartphone sensors to cut patterns of driving behaviors. In real driving environments, we retrieve readings from smartphones' accelerometers and orientation sensors and then compute their standard deviation as well as mean value in a small window. If a vehicle is under normal driving, $D^3$ compares the standard deviation and the mean value with some thresholds to determine whether an abnormal driving behavior begins. The window size and thresholds can be learned from the collected data. After the beginning of a driving behavior is found out, $D^3$ continues to check the standard deviation and mean value to determine whether the driving behavior ends.

After cutting patterns of a driving behavior, effective features can be extracted from the driving behavioral patterns and then sent to the classifier model. Finally, the model outputs a fine-grained identification result. If the result denotes the normal driving behavior, it is ignored, and if it denotes any one of abnormal driving behaviors, $D^3$ sends a warning message.

The warning message will be sent both to the driver and to some remote receivers. The drivers can be aware of their bad driving habits from the warning messages along their driving, so that they can make targeted corrections of their bad driving habits, helping to prevent potential car accidents. Furthermore, the identification results can be passed back to a remote central server. On that case, the remote server can call the police automatically once a traffic accident occurs, which may save lives in case the victims have difficulty doing so by themselves (i.e. in case they lose consciousness or cannot move). In addition, the automatic recorded warning messages along with the specific car/driver may help identifying the driver in hit-and-run accidents. Those warning messages stored in the central server may also help the vehicle insurance company to analyze the policyholders' driving habits, so that the insurance company may offer a more preferential car insurance policy to drivers with good driving habits, and a harsh policy to those with a long history of warning message lists.

### 4.5 Complexity Analysis

We assume that the amount of real-time sensing data from accelerometer and orientation is $N$. The calculation

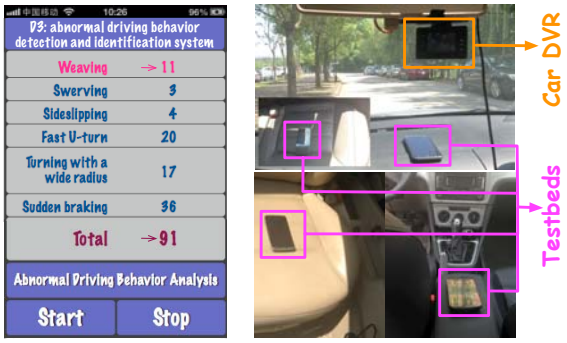



Fig. 7. User interface of $D^3$ and testbeds.

overhead of $D^3$ can be divided into the following parts.

*Building the Classifier Model*: The classifier model (either with SVM or NN machine learning algorithm) is built off-line and it is built only once so that it does not contribute to the computational complexity of real-time identification process.

*Low-pass Filtering*: Low-pass Filtering needs linear computational complexity with the amount of real-time sensing data from accelerometer and orientation, $N$. In other words, it has a complexity of $O(N)$.

*Feature Extracting*: Extracting features from each input data needs constant computation. Hence, for $N$ inputs, it is $O(N)$.

*Identification by the Classifier Model*: The computation from one input features tuple to an output identification result in the model is constant. Thus, for $N$ input tuples, it has a complexity of $O(N)$.

Based on the above analysis, the computational complexity of $D^3$ is $O(N)$ to detect and identify different abnormal driving behaviors.

## 5 Evaluations

In this section, we first present the prototype of $D^3$, then evaluate the performance of $D^3$ in real driving environments.

### 5.1 Prototype

We implement $D^3$ as an Android App and install it on smartphones (listed in Section 3.1). Fig.7 shows the user interface of $D^3$ and testbeds in vehicles. $D^3$ is running by 20 drivers with distinct vehicles in real driving environments to collect the data for evaluation. Meanwhile, Car DVRs are used to record driving behaviors and 9 experienced drivers are asked to recognize abnormal driving behaviors as ground truth. After a 4-month data collection (i.e. July 21 to Nov. 30, 2014, using the same method of collecting data as described in Section 3.1), we obtain a test set with 3141 abnormal driving behaviors to evaluate the performance of $D^3$.

### 5.2 Metrics

To evaluate the performance of $D^3$, we define the following metrics based on the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



- ***Accuracy***: The probability that the identification of a behavior is the same as the ground truth.
- ***Precision***: The probability that the identifications for behavior A is exactly A in ground truth.
- ***Recall***: The probability that all behavior A in ground truth are identified as A.
- ***False Positive Rate (FPR)***: the probability that a behavior of type Not A is identified as A.

In the following subsections, we investigate the impact of various factors to $D^3$ and present the details.

### 5.3 Overall Performance

The performance of $D^3$ is evaluated by three levels, i.e. total accuracy, detecting abnormal vs. normal driving behaviors and identifying fine-grained driving behaviors.

#### 5.3.1 Total Accuracy

Total accuracy is the ratio of correct identifications to total identifications, containing identifications for the six types of abnormal driving behaviors as well as the normal. The total accuracy for each driver is evaluated respectively in TABLE 2. It can be seen that all of the 20 drivers achieve high total accuracies. Among the 20 drivers, the lowest total accuracy is 92.44% with SVM classifier model, and 93.56% with NN classifier model. On average, $D^3$ achieves a total accuracy of 95.36% with SVM classifier model, and 96.88% with NN classifier model.

#### 5.3.2 Detecting the Abnormal vs. the Normal

In this level, we treat all types of abnormal driving behaviors as one type (i.e. *Abnormal*), and merely identify whether a driving behavior is abnormal or normal. As is shown in TABLE 3, $D^3$ performs so excellent that almost all abnormal driving behaviors are identified. In other words, $D^3$ could identify abnormal driving behaviors vs. normal ones with a recall of 99.84% with SVM, and 99.92% with NN. In addition, none of normal driving behaviors is identified as abnormal one, i.e. with 100% precision and 0 FPR using either SVM or NN classifier model.

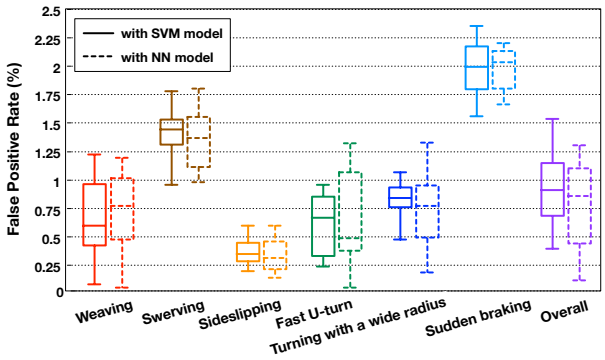



Fig. 8. Box plot of FPR of identifying specific types of driving behaviors.

#### 5.3.3 Identifying Abnormal Driving Behaviors

$D^3$ also realizes fine-grained identification, i.e., discriminates *Weaving*, *Swerving*, *Sideslipping*, *Fast U-turn*, *Turning with a wide radius* and *Sudden braking*. TABLE 3 shows the identification results. With SVM classifier model, the accuracy for identifying each of the six abnormal driving behaviors is no less than 94%, the precision is above 85%, and the recall is more than 70%. And with NN classifier model, $D^3$ achieves an accuracy of no less than 95%, a precision of above 90% and a recall of more than 85%. The FPRs for identifying all types of abnormal

TABLE 2
Total accuracy in 20 drivers' experiments

| Dirver | 1 | | 2 | | 3 | | 4 | | 5 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Total Accuracy (%) | SVM | NN | SVM | NN | SVM | NN | SVM | NN | SVM | NN |
| | 98.66 | 98.72 | 96.43 | 98.23 | 95.29 | 96.40 | 95.61 | 95.58 | 97.13 | 96.91 |
| Driver | 6 | | 7 | | 8 | | 9 | | 10 | |
| Total Accuracy (%) | SVM | NN | SVM | NN | SVM | NN | SVM | NN | SVM | NN |
| | 94.55 | 96.37 | 97.83 | 98.49 | 99.07 | 98.52 | 98.37 | 99.12 | 92.44 | 95.35 |
| Driver | 11 | | 12 | | 13 | | 14 | | 15 | |
| Total Accuracy (%) | SVM | NN | SVM | NN | SVM | NN | SVM | NN | SVM | NN |
| | 93.46 | 95.31 | 96.30 | 99.24 | 94.02 | 96.10 | 99.59 | 98.67 | 91.35 | 94.23 |
| Driver | 16 | | 17 | | 18 | | 19 | | 20 | |
| Total Accuracy (%) | SVM | NN | SVM | NN | SVM | NN | SVM | NN | SVM | NN |
| | 94.50 | 96.21 | 92.86 | 93.56 | 94.68 | 96.48 | 95.49 | 97.44 | 95.43 | 96.57 |

TABLE 3
Accuracy evaluation

| Behavior | Accuracy(%) | | Precision(%) | | Recall(%) | | FPR(%) | |
|---|---|---|---|---|---|---|---|---|
| | SVM | NN | SVM | NN | SVM | NN | SVM | NN |
| Normal | 99.84 | 99.92 | 98.80 | 99.22 | 100.00 | 100.00 | 0.19 | 0.10 |
| Abnormal | 94.81 | 96.23 | 100.00 | 100.00 | 99.80 | 99.67 | 0.00 | 0.00 |
| Weaving | 98.43 | 98.56 | 92.55 | 95.67 | 87.87 | 92.25 | 0.63 | 0.33 |
| Swerving | 97.94 | 98.33 | 92.29 | 95.47 | 94.15 | 95.20 | 1.39 | 0.76 |
| Sideslipping | 98.60 | 98.49 | 87.96 | 90.50 | 71.43 | 88.34 | 0.37 | 0.08 |
| Fast U-turn | 98.49 | 98.78 | 85.71 | 92.73 | 76.00 | 86.55 | 0.54 | 0.29 |
| Turning with a wide radius | 98.68 | 98.67 | 89.30 | 93.61 | 92.72 | 92.58 | 0.86 | 0.82 |
| Sudden braking | 95.74 | 97.23 | 97.88 | 98.56 | 99.04 | 99.77 | 1.93 | 0.74 |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



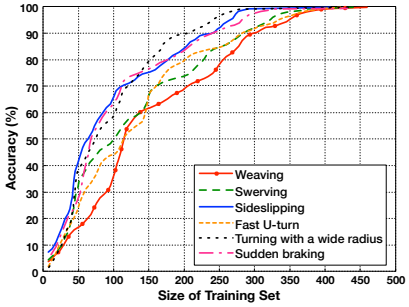



Fig. 9. Total accuracy under different sizes of training set with SVM classifier model.

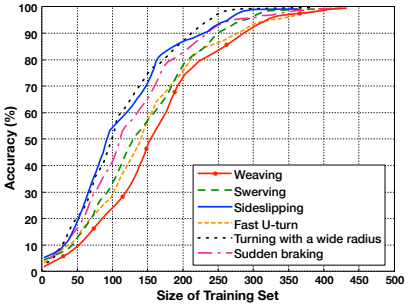



Fig. 10. Total accuracy under different sizes of training set with NN classifier model.

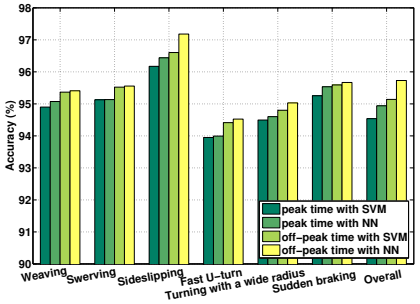



Fig. 11. Accuracy under different traffic conditions.

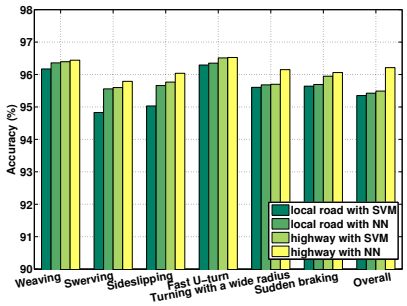



Fig. 12. Accuracy under different rode types.

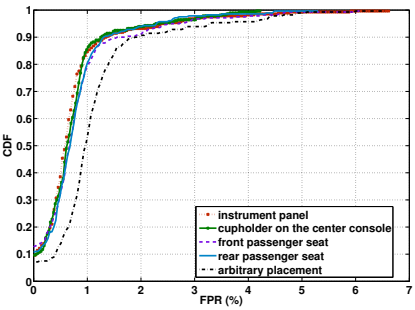



Fig. 13. CDF of FPR under different smartphone placements with SVM classifier model.

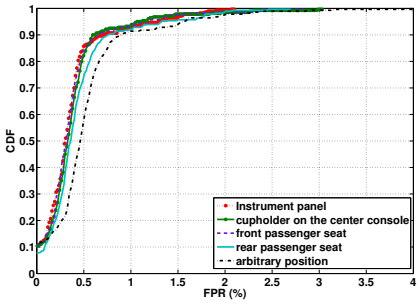



Fig. 14. CDF of FPR under different smartphone placements with NN classifier model.

driving behaviors are no more than 2% with both SVM and NN classifier models. The results show that $D^3$ is an high-accurate system to identify various abnormal driving behaviors.

Moreover, we evaluate FPRs of identifying specific abnormal types. Fig.8 shows a box-plot of the FPRs for each type of abnormal driving behaviors and the overall FPR with SVM and NN classifier models. As is shown in the figure, with SVM classifier model, the highest FPR of identifying specific abnormal type is less than 2.5% and the overall FPR is around 0.9%. With NN classifier model, the highest FPR is less than 2% and the over all FPR is no more than 0.8%, which shows that $D^3$ could implement fine-grained identification with few false alarms. In addition, $D^3$ performs better when identifying weaving, sideslipping, turning with a wide radius and fast U-turn than identifying swerving and sudden braking. This is because the patterns of the former ones are more distinct than that of the latter. However, the performance of identifying swerving, sideslipping and turning with a wide radius is more stable than identifying other abnormal driving behaviors since they have smaller standard deviations. This is because the patterns of the former ones are more stable than that of the latter.

According to the analysis above, $D^3$ with both SVM and NN classifier models can achieve high performance in identifying different driving behaviors. It has a good total accuracy which is over 95%, an excellent performance in discriminate abnormal type from normal type (i.e. an accuracy of over 99%) and a good performance in identifying six different abnormal types. Moreover, compared with SVM classifier model, the NN classifier model helps $D^3$ to achieve a relatively better performance.

### 5.4 Impact of Training Set Size

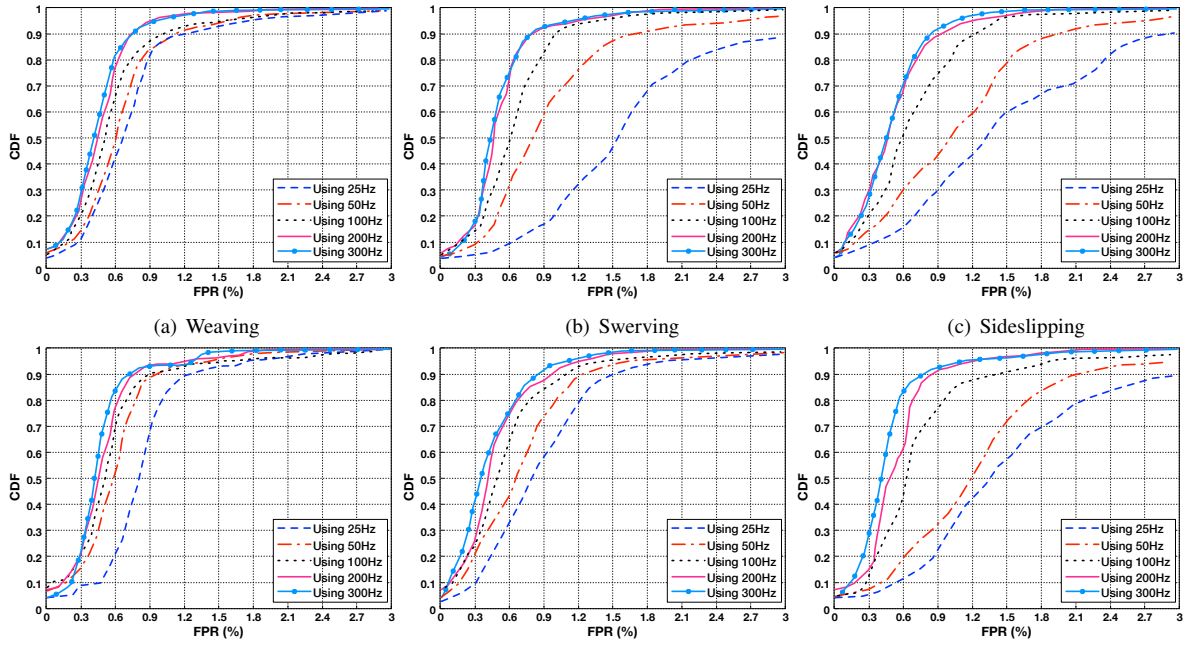
According to Section 3.1, we collect the traces for totally 4029 abnormal driving behaviors for training. The training set size (i.e. the number of training samples) may have an impact on the training results so that it may affect the performance of $D^3$. We thus evaluate the impact of the training set size. The results with SVM classifier model are shown in Fig.9, and with NN classifier model are shown in Fig.10. From the figures, we observe that the more training samples there are, the better performance $D^3$ has. With SVM classifier model, when we use 280 training samples for turning with a wide radius, sideslipping, 300 sudden braking samples, 350 swerving samples and 380 training samples for fast U-turn and weaving, respectively, $D^3$ could identify each specific type of driving behavior with an accuracy close to 100%. With NN classifier model, when we use 250 training samples for turning with a wide radius, 300 sideslipping, 350 sudden braking and swerving samples and 400 training samples for fast U-turn and weaving, respectively, $D^3$ could identify each specific type of driving behavior with an accuracy close to 100%. In order to guarantee the performance of $D^3$, we use as many training samples as possible.

### 5.5 Impact of Traffic Condition

The traffic conditions may affect the drivers' driving behaviors and further affect the performance of $D^3$. We analyze traces during peak time and off-peak time respectively to evaluate the impact of traffic conditions. Fig.11 shows the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing






(a) Weaving (b) Swerving (c) Sideslipping

(d) Fast U-turn (e) Turning with a wide radius (f) Sudden braking

Fig. 15. CDF of FPR using different sampling rates for each type of abnormal driving behaviors.

accuracies of identifying specific types of abnormal driving behaviors during peak and off-peak time. It can be seen that $D^3$ achieves good accuracy during both time periods, and the accuracy in off-peak time is slightly higher than that in peak time. Such results are consistent with both SVM and NN classifier models. This is because during peak time, the vehicles perform less drastic actions due to traffic jams. So some abnormal driving behaviors present restrained patterns during peak time. Different types of abnormal driving behaviors thus are much easier to be mistaken by each other and even be mistaken as normal driving behaviors. Nevertheless, during off-peak time, the patterns of all types of driving behaviors are performed more obvious. So different types of abnormal driving behaviors are more distinguishable. Moreover, the results with NN classifier model are sightly better than that with SVM classifier model.

### 5.6 Impact of Road Type

Drivers could perform abnormal driving behaviors on highway or local road, thus we further investigate the impact of the two road types on the performance of $D^3$. Fig.12 shows how road types affect the accuracy of identifying various types of abnormal driving behaviors. It can be seen that $D^3$ achieves good accuracy both on highway and local road, but the accuracy is slightly higher on highway than that on local road. Such results are consistent with both SVM and NN classifier models. This is because the better road condition on highway could reduce the fluctuations caused by bumpy surfaces. Since highway is more smooth and has less slopes compared with local road, there are less disturbances then. In addition, there are less curves and no traffic light stops on the highway, so when driving normally on the highway, drivers have less chance to perform drastic actions. As a result, $D^3$ can achieve a better performance on highway than that on local road. Moreover, the results with NN classifier model are sightly better than that with SVM classifier model.

### 5.7 Impact of Smartphone Placement

Smartphones could be arbitrarily placed in vehicles, we thus investigate the impact of smartphone placement. In our experiments with 20 vehicles, smartphones are fixed on instrument panel, cupholder on the center console, front passenger seat, or left rear passenger seat, where smartphone sensors' y-axis is aligned along the moving direction of vehicles, or on arbitrary placement (i.e. smartphones are put in the driver's pocket and its pose could be arbitrary). Fig.13 and Fig.14 show the CDF of FPRs of fine-grained identifications under different smartphone placements with SVM and NN classifier models. It can be seen that $D^3$ can achieve low FPRs under all smartphone placements, which shows $D^3$ performs excellent wherever the smartphone is placed in a vehicle. Although there is slightly higher FPR under arbitrary placement because of errors in the coordinate reorientation process, a FPR of less than 2% in 90% of the cases with SVM classifier model and less than 1% in 90% of the cases with NN classifier model is still a good result.

### 5.8 Impact of Smartphone Sensors' Sampling Rate

In $D^3$, keeping collecting sensing data from smartphones at a high sampling rate may result in heavy energy consumption. Since energy consumption is an important issue on smartphones, we prefer to a lower sampling rate in order to make $D^3$ more energy efficient. A low sampling rate, however, could have impact on the identification accuracy

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2618873, IEEE Transactions on Mobile Computing



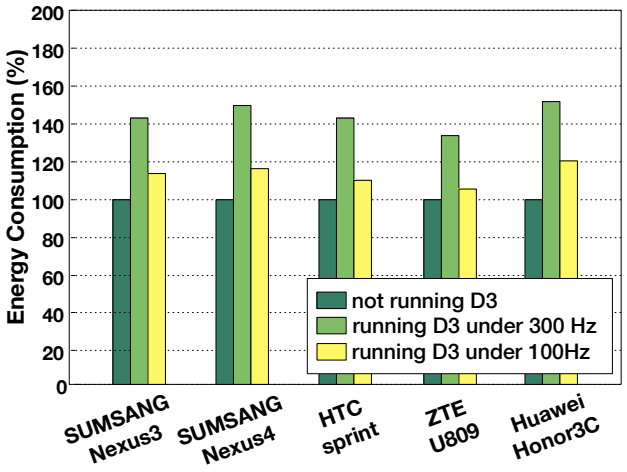



Fig. 16. Energy consumption on 5 types of devices with $D^3$ under different sampling rates and without running $D^3$.

of $D^3$. We thus further investigate the impact of lowering smartphone sensors' sampling rate on the accuracy of $D^3$.

Our original dataset is obtained using a sampling rate of $300Hz$. In order to obtain additional datasets with different sampling rates, we regularly drop some samples in the original dataset. For example, For a $150Hz$-sampling rate dataset, every other sampling point is discarded from the original dataset. Fig.15 shows the CDF of the FPRs (False Positive Rate) for each type of the six abnormal driving behaviors using different sampling rates from $300Hz$ to $25Hz$. All of the FPRs under different sampling rates are calculated based on the NN classifier model. It can be seen that the FPRs of swerving, sideslipping and sudden braking raise as the sampling rate decreases, while the FPRs of weaving, turning with a wide radius and fast U-turn do not significant change as the sampling rate decreases. This is because swerving, sideslipping and sudden braking are instant driving behaviors that continue a very short period of time, which may lead to lack of samples under a low sampling rate. However, it can be seen from the figure that we can reduce the sampling rate to at least $100Hz$, which guarantees the accuracy for identifying all abnormal driving behaviors to be over 95%.

We further perform an experiment to test the energy consumption of $D^3$ using our power meters [22][23]. All Apps and services (e.g. GPS or network) are closed during the experiment. Each time, three smartphones of the same type are fixed in a car for 2 days, one not running $D^3$ in standby mode, one running $D^3$ under a sampling rate of $300Hz$ and one running $D^3$ under $100Hz$. All smartphones are using the NN classifier model to identify different driving behaviors in this experiment. Fig.16 shows the energy consumption in the five types of smartphones (mentioned in Section 3.1) under the three conditions mentioned above. The energy consumption of the device without running $D^3$ is normalized to 100%. It can be seen from the figure that the energy consumption is significantly reduced under $100Hz$-sampling rate compared with $300Hz$-sampling rate.

Based on the analysis above, we finally employ a sampling rate of $100Hz$ in $D^3$, which enables $D^3$ to guarantee the accuracy for identifying all abnormal driving behaviors to be over 95%, and significantly reduce the energy consumption of $D^3$.

## 6 Conclusion

In this paper, we address the problem of performing abnormal driving behaviors detection (coarse-grained) and identification (fine-grained) to improve driving safety. In particular, we propose a system, $D^3$, to detect and identify specific types of abnormal driving behaviors by sensing the vehicle's acceleration and orientation using smartphone sensors. Compared with existing abnormal driving detection systems, $D^3$ not only implementes coarse-grained detections but also conducts fine-grained identifications. , i.e. *Weaving*, *Swerving*, *Sideslipping*, *Fast U-turn*, *Turning with a wide radius* and *Sudden braking*. To identify specific abnormal driving behaviors, $D^3$ trains a multi-class classifier model through Support Vector Machine (SVM) and Neuron Networks (NN) based on the acceleration and orientation patterns of specific types of driving behaviors. To obtain effective training inputs, we extract 16 basic features and 136 polynomial features from driving behavioral patterns collected from the 6-month driving traces in real driving environments. The extensive experiments driving in real driving environments in another 4 months show that $D^3$ achieves high accuracy when detecting and identifying abnormal driving behaviors.

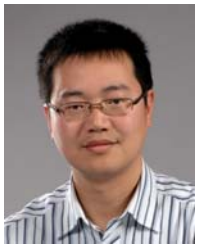
## References

[1] World.Health.Organisation. The top ten causes of death. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs310/en/
[2] C. Saiprasert and W. Pattara-Atikom, "Smartphone enabled dangerous driving report system," in *Proc. HICSS*, 2013, pp. 1231–1237.
[3] M. V. Yeo, X. Li, K. Shen, and E. P. Wilder-Smith, "Can svm be used for automatic eeg detection of drowsiness during car driving?" *Elsevier Safety Science*, vol. 47, pp. 115–124, 2009.
[4] S. Al-Sultan, A. H.Al-Bayatti, and H. Zedan, "Context-aware driver behavior detection system in intelligent transportaion system," *IEEE Trans. on Vehicular Technology*, vol. 62, pp. 4264–4275, 2013.
[5] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: insights from a controlled field study."
[6] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proc. ACM MobiSys*, 2013.
[7] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, Y. Zhu, G. Xue, and M. Li, "Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments," in *Proc. IEEE INFOCOM*, 2014.
[8] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Trans. on Sensor Networks*, vol. 6, no. 13, 2010.
[9] J. Dai, J. Teng, X. Bai, and Z. Shen, "Mobile phone based drunk driving detection," in *Proc. PervasiveHealth*, 2010, pp. 1–8.
[10] M. Fazeen, B. Gozick, R. Dantu, M. Bhukuiya, and M. C.Gonzalez, "Safe driving using mobile phones," *IEEE Trans. on Intelligent Transportation Systems*, vol. 13, pp. 1462–1468, 2012.
[11] U.S.NHTSA. The visual detection of dwi motorists. [Online]. Available: http://www.shippd.org/Alcohol/dwibooklet.pdf
[12] D. Lee, S. Oh, S. Heo, and M. Hahn, "Drowsy driving detection based on the driver's head movement using infrared sensors," in *Proc. IEEE ISUC*, 2008, pp. 231–236.
[13] M. Kaneda, H. Obara, and T. Nasu, "Adaptability to ambient light changes for drowsy driving detection using image processing," in *JSAE Review*, 1999, pp. 133–136.

[14] N. Condro, M.-H. Li, and R.-I. Chang, "Motosafe: Active safe system for digital forensics of motorcycle rider with android," *IJIEE*, vol. 2, no. 4, pp. 612–616, 2012.
[15] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Proc. IEEE IV*, 2012, pp. 234 – 239.
[16] Y. Wang, J. Yang, and Y. Chen, "Tracking human queues using single-point signal monitoring," in *Proc. ACM MobiSys*, 2014.
[17] P. Harrington, *Machine Learning in Action*. Manning Publications, 2012.
[18] Y. Guo, L. Yang, X. Ding, J. Han, and Y. Liu, "Opensesame: Unlocking smart phone through handshaking biometrics," in *Proc. IEEE INFOCOM*, 2013, pp. 365–369.
[19] C. Chang and C. Lin, "Libsvm: a library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.
[20] M. Shahzad, A. Liu, and A. Samuel, "Secure Unlocking of Mobile Touch Screen Devices by Simple Gestures - You can see it but you can not do it, " in *Proc. ACM MobiCom*, 2013, pp. 39–50.
[21] J. Yu, H. Zhu, H. Han, Y. Chen, J. Yang, Y. Zhu, G. Xue, and M. Li, "Senspeed: Sensing driving conditions to estimate vehicle speed in urban environments," in *IEEE Trans. on Mobile Computing*, vol. 15, no. 1, pp. 202–216 2016.
[22] H. Han, J. Yu, H. Zhu, Y. Chen, J. Yang, G. Xue, Y. Zhu, and M. Li, "E3: Energy-efficient engine for frame rate adaptation on smartphones," in *Proc. ACM SenSys*, 2013.
[23] J. Yu, H. Han, H. Zhu, Y. Chen, J. Yang, G. Xue, Y. Zhu, and M. Li, "Sensing Human-Screen Interaction for Energy-Efficient Frame Rate Adaptation on Smartphones," in *IEEE Trans. on Mobile Computing*, vol. 14, no. 8, pp. 1698–1711 2015.

**Jiadi Yu** received the Ph.D. degree in Computer Science from Shanghai Jiao Tong University, Shanghai, China, in 2007. He is currently an Associate Professor in Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Prior to joining Shanghai Jiao Tong University, he was a postdoctoral fellow in the Data Analysis and Information Security (DAISY) Laboratory at Stevens Institute of Technology from 2009 to 2011. His research interests include cyber security and privacy, mobile and pervasive computing, cloud computing and wireless sensor networks. He is a member of the IEEE and the IEEE Communication Society.

**Zhongyang Chen** received bachelor degree in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2013. She is a master student in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Her research interests include mobile computing and wireless sensor networks.



**Yanmin Zhu** received the Ph.D. from the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology in 2007. He is currently a Professor with the Department of Computer Science and Engineering at Shanghai Jiao Tong University. His research interests include wireless sensor networks and mobile computing. Before that, he was a Research Associate with the Department of Computing at Imperial College London. He is a member of the IEEE and the IEEE Communication Society.

**Yingying (Jennifer) Chen** is a Professor in the Department of Electrical and Computer Engineering at Stevens Institute of Technology. Her research interests include cyber security and privacy, mobile and pervasive computing, and mobile healthcare. She has published over 80 journals and referred conference papers in these areas. She received her Ph.D. degree in Computer Science from Rutgers University. Prior to joining Stevens, she was with Alcatel-Lucent. She is the recipient of the NSF CAREER Award and Google Faculty Research Award. She also received NJ Inventors Hall of Fame Innovator Award. She is the recipient of the Best Paper Award from ACM International Conference on Mobile Computing and Networking (MobiCom) 2011. She also received the IEEE Outstanding Contribution Award from IEEE New Jersey Coast Section each year 2005-2009. Her research has been reported in numerous media outlets including MIT Technology Review, Wall Street Journal, and National Public Radio. She is on the editorial boards of IEEE Transactions on Mobile Computing (IEEE TMC), IEEE Transactions on Wireless Communications (IEEE TWireless), and IEEE Network Magazine. She is a senior member of the IEEE.

**Linghe Kong** is currently an assistant professor with the Department of Computer Science and Engineering at Shanghai Jiao Tong University. From 2013 to 2015, he was a Postdoctoral Fellow at McGill University and a Postdoctoral Researcher at Singapore University of Technology and Design. He received his Ph.D. degree in Computer Science from Shanghai Jiao Tong University 2012, Master degree in Telecommunication from TELECOM SudParis 2007, and B. Eng. degree in Automation from Xidian University 2005. His research interests include wireless communications, sensor networks, mobile computing, and RFID.

**Minglu Li** graduated from the School of Electronic Technology, University of Information Engineering, in 1985 and received the Ph.D. degree in computer software from Shanghai Jiao Tong University (SJTU) in 1996. He is a full professor and the vice chair of the Department of Computer Science and Engineering and the director of Grid Computing Center of SJTU. Currently, his research interests include grid computing, services computing, and sensor networks.