Smart Diagnosis: Deep Learning Boosted Driver Inattention Detection and Abnormal Driving Prediction

Landu Jiang, Wen Xie, Dian Zhang, *Member, IEEE*, and Tao Gu, *Senior Member, IEEE*

***Abstract*—Inattentive driving is one of the high-risk factors that causes a large number of traffic accidents every year. In this article, we aim to detect driver inattention leveraging on large-scale vehicle trajectory data while at the same time explore how do these inattentive events affect driver behaviors and what following reactions they may cause, especially, for commercial vehicles. Specifically, the proposed system targets four most commonly occurring critical inattentive events, including smoking, phone call, turning back, and yawning. By applying a deep convolutional neural network (CNN) (Inception v3) with two data augmentation routines—Mixup and synthetic minority oversampling technique (Smote), we are able to balance the training data distribution and improve the generalization of the classification model. Then, based on the output derived from the inattention detection combining with point of interest (POI) and climate data, a long short-term memory (LSTM)-based model is deployed to predict driver upcoming abnormal operations on road (due to inattention) which may result in potential dangerous driving conditions, such as sudden acceleration/deceleration, aggressive left/right lane change, etc. To evaluate our proposed system, we collect more than 120 000 real-world driving traces from over 200 drivers. The experimental results show that our model achieves a weight accuracy (WA) of 92.27% for inattentive driving detection and a WA of 91.67% for abnormal driving prediction, demonstrating its great potential of shaping good driving habits and promoting road safety.**

***Index Terms*—Data augmentation, deep learning, driver safety, inattentive driving, large-scale data.**

I. Introduction

INATTENTIVE driving poses a major threat to road safety and has raised great interest in both academia and industry. Report in [1] shows that there were nearly 3000 fatal accidents every year in the United State from 2015 to 2019, and driver inattention accounted for 9% of all fatal accidents. Inattentive driving may lead to failure to notice traffic signs or yield traffic patterns, thus, drivers may overreact when making turns or changing lanes. Especially, for commercial vehicle drivers, due to the level of vehicle drivability and complex driving conditions (e.g., frequent night driving), traffic accidents may occur more often than private vehicle drivers, causing serious loss of lives and property. Therefore, monitoring and understanding drivers' inattentive behaviors are essential to road safety measurement, shaping good driving habits, and preventing road accidents.

To enhance driver safety, in-vehicle cameras have been widely deployed for monitoring driver's status in [2]–[4]. These vision-based systems can accurately detect whether a driver is in an unsafe driving state (i.e., inattentiveness) by tracking facial states of the driver, but it will incur installation costs and the detection accuracy may be affected by lighting conditions (e.g., may not work well for night driving). You *et al.* [5] presented a less expensive way to identify critical driving events (e.g., careless/aggressive lane changes or speeding and drowsiness/unfocused driving) by leveraging smartphone cameras and inertial sensors. However, the system may fail to capture objects if the phone is not properly placed in the vehicle [5]. In addition, we have seen great success in in-vehicle human activity recognition based on wireless sensing technologies, such as Wi-Fi channel state information (CSI) and acoustic signals (e.g., ultrasound) [6]–[10]. Though these methods can achieve high recognition accuracy, they usually require extra infrastructures and time-consuming calibration to update the signal fingerprints. What is more, there have been a variety of driving safety systems that use smart sensors or wrist-worn devices. SMARTwheel [11] offers a safety product that has the linear potentiometers embedded in the steering wheel to determine whether a driver has their hands on or off the wheel. The wrist-worn devices like smartwatches have been applied to detect driver hand motion and dangerous in-vehicle activities [6]. The main disadvantage of such systems is that they still miss certain of unsafe events when the driving inattention has no hand motion involved.

On the other hand, driving trajectory analysis and vehicle dynamic measurement have been the fundamental tools that allow us to evaluate interactions between drivers, vehicles, and road zones [12]–[14]. Though a number of approaches have used embedded sensors (e.g., accelerometer, gyroscope, and GPS) on smart devices (e.g., smartphones) to track driving conditions [15], [16] as well as monitor driver smartphone use (e.g., calling, texting, and reading) to prevent

Manuscript received April 18, 2021; revised July 15, 2021; accepted July 24, 2021. Date of publication August 10, 2021; date of current version March 7, 2022. This work was supported in part by NSFC under Grant 61872247; in part by the Shenzhen Peacock Talent Grant under Grant 827-000175; in part by the Guangdong Natural Science Fund under Grant 2019A1515011064; in part by the Lingnan University Direct Grant under Grant DR21A6; and in part by the Lingnan Research Seed Fund under Grant 102363. *(Landu Jiang and Wen Xie contributed equally to this article.) (Corresponding author: Dian Zhang.)*

Landu Jiang, Wen Xie, and Dian Zhang are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: landu.jiang@mail.mcgill.ca; 1810273027@szu.edu.cn; serena.dian@gmail.com).

Tao Gu is with the Department of Computing, Macquarie University, Sydney, NSW 2109, Australia (e-mail: tao.gu@mq.edu.au).

Digital Object Identifier 10.1109/JIOT.2021.3103852

distractions [17]–[19], other unsafe driving activities are not fully investigated. If a driver is or intends to be involved in any inattention or distraction, there could be a distinct change in vehicle movements (e.g., frequent steering-wheel correction activity) [20], [21]. Such changes can be derived from the vehicle trajectory and thus, be used as soft hints indicating inappropriate driving behaviors. More importantly, it is worth noting that in-vehicle inattentive activities are very likely to have an impact on the driver following operations on the road [22]. For example, when a driver is distracted due to smoking or phone call, he/she may not be able to react properly to the upcoming traffic alerts (e.g., sudden acceleration/hard braking and changing the lane aggressively), resulting in abnormal vehicle trajectory patterns. Predicting these unsafe driving conditions triggered by driver inattention in advance is desired, unfortunately, few work has been proposed addressing the issue.

Inspired by this, in this article, we present an inattentive driving analytics system to detect driver inattention while at the same time explore how these activities affect operations during driving, leveraging on large-scale vehicle trajectory data, such as GPS trace, acceleration, speed, distance, and point-of-interest (POI) data. To achieve this goal, three modules are presented, including unsafe events spotting, inattentive driving recognition, and abnormal operation prediction. We first adopt random forest (RF) to spot the unsafe events and then apply a deep convolutional neural network (CNN) model (Inception v3 with Mixup) to identify driver inattentive behaviors. In addition, we use the synthetic minority oversampling technique (Smote) algorithm to augment the data and adopt the Mixup method to train the linear boundary avoiding model overfitting. Finally, a long short-term memory (LSTM)-based model is deployed to investigate the driver upcoming operations (affected by inattentive behaviors) and then predict dangerous driving events in advance (e.g., abnormal acceleration/deceleration and abnormal left/right lane change). Based on the vehicle trajectory data collected from long-term onboard diagnostics (OBDs) devices, the proposed models can successfully identify four most commonly occurring inattentive driving events (e.g., smoking, phone call, turning back, and yawning) and predict potential abnormal unsafe operations accordingly.

The main contributions of this article are summarized as follows.

1) In this article, we present an inattentive driving analytics system that leverages vehicle trajectory data to detect and study driver inattention, especially, for commercial vehicles. The proposed system not only targets identifying in-vehicle inattentive driving activities but also focuses on predicting potential abnormal unsafe operations that may caused by driver inattention.
2) To detect and recognize inattentive driving conditions, we apply Inception v3 with the Mixup method to train a classifier model not only solving the problem of discrete features and overfitting but also distinguishing different types of driver inattentiveness. The Smote algorithm is also adopted to balance positive and negative samples and address the data sparsity.
3) To study the correlations between inattentive driving and abnormal operations, we leverage an LSTM-based model by considering external features (i.e., the type of inattentiveness, POI, and weather) to predict and avoid potential unsafe driving conditions in advance.
4) We conduct extensive experiments based on large-scale commercial vehicle trajectory data. Our solution outperforms other models which achieves a weighted accuracy at 92.27% for inattentive driving recognition and at 91.67% for abnormal operation prediction, which demonstrate that the proposed models are robust and effective.

To the best of our knowledge, our system is the first one that detects and recognizes driver inattentive behaviors by leveraging models based on large-scale trajectory data, while at the same time exploring the correlations between driving inattention and abnormal/unsafe driver behaviors to predict potential dangerous driving events (i.e., aggressive driving). It is worth noting that the proposed models are also capable of processing real-time data obtained from smart/onboard devices (e.g., smartphones) which have GPS and IMU sensors embedded. Therefore, this work has great potentials to assist the driver in real driving condition and promote road safety.

The remainder of this article is organized as follows. We describe the related work in Section II. We then introduce the design consideration in Section III and present the system design in Section IV. We conduct our experiments in Section V and then discuss the system latency and how can it be extended in Section VI. Finally, we conclude this article and discuss the future work in Section VII.

II. Related Work

Inattentive/aggressive driving detection basically can be classified into four categories: 1) vision-based detection; 2) wireless signal-based detection; 3) wearable sensing-based detection; and 4) IMU-based detection.

A. Vision-Based Detection

Onboard cameras are widely used for real-time monitoring to determine whether the driver is distracted or fatigue [23]. However, it will incur additional costs and require specialized infrastructures in the vehicle. Meanwhile, smartphones are considered as an alternative option [24]–[26] to track driver's head position and eye directions. You *et al.* [5] used computer vision and machine learning algorithms on the phone to detect driver inattention using the front-facing camera while at the same time tracking road conditions using the rear-facing camera. Ji *et al.* [23] proposed a nonintrusive vision-based prototype using remotely located cameras equipped with active infrared radiation (IR) illuminators to acquire video frames of the driver. Moreover, with the rise of deep learning, a number of systems leverage the CNN model to detection distraction driving [27]–[29]. Streiffer *et al.* [30] proposed a multimodal analytics platform—DarNet, which could detect driver in-vehicle activities, such as normal driving, talking, eating, reaching, etc. They employed a CNN model for video analysis collected from an inward-facing camera and recurrent neural network (RNN) model for the data from mobile embedded sensors. Xing *et al.* [31] applied and evaluated three different pretrained CNN models, including AlexNet, GoogLeNet, and ResNet50, focusing on seven common driving behaviors (e.g., secondary tasks). Despite the privacy issues, most vision-based approaches, however, are not always stable due to different setup conditions (e.g., weather, lighting, and the device placement).

B. Wireless Signal-Based Detection

Many systems using wireless signals have been offered in driver behavior monitoring. Xie *et al.* [32] proposed a real-time drowsy driving detection system based on audio devices embedded in smartphones. It can detect nodding, yawning, and abnormal steering of commercial vehicles in real time. Xu *et al.* [9] leveraged existing audio devices on smartphones to realize early recognition of inattentive driving events. Yang *et al.* [33] leveraged the existing infrastructure of the car to detect driver phone use by estimating the range between the phone and car speakers. In addition, there has been active research work using Wi-Fi CSI for human motion sensing, WiCAR [8] employed a multiadversarial domain adaptation network model to recognize various driver inattentive activities, while WiFind [10] applied Hilbert–Huang transform (HHT)-based pattern extracting method to enhance driver fatigue detection. Although these methods achieved promising results, they are susceptible to the change of testing environments, such as environmental noise, changes in vehicle personnel, and window conditions.

C. Wearable Sensing-Based Detection

Inattentive driving detection using wrist-worn devices (e.g., smartwatch) emerges as a hot topic in recent years. Several systems have been offered to track driver hands position and analyze in-vehicle behaviors for preventing unsafe driving [34]. Bi *et al.* [35] presented SafeWatch, a system based on smartphone and smartwatches to estimate the posture of the driver's forearm and the positions of the driver hands. SafeWatch could sense the motion of the vehicle and the driver's hands with two watches worn on both hands of the driver. Jiang *et al.* [6] focused on driver's right-hand motion to recognize secondary tasks and potential dangerous behaviors. But it only tracks one single-hand movement and the basic assumption is the movement always starts at the steering wheel. Moreover, Lee *et al.* [36] also proposed a standalone system that detects driver drowsiness using smartwatch only. The system is able to determine the drowsiness level based on different types of driver behaviors computed from the motion data. Thus, it is no doubt that wearable sensing if capable of detecting unsafe driving activities. However, drivers still need to wear the device on their hands, and one single smart device cannot capture all types of unsafe events.

D. IMU-Based Detection

A number of IMU-based systems using sensors on cars/smartphones have been proposed for unsafe/inattentive driving detection. Wang *et al.* [37] used embedded sensors in smartphones (i.e., accelerometers and gyroscopes) capturing the differences of centripetal acceleration readings during turns. Based on the information, it can estimate the position of the smartphone in the vehicle—at driver side or passenger side, thus, prevent the phone calling/texting if there is a high chance that the driver is holding the device. Johnson and Rajamani [38] also proposed a method for the localization of a smartphone inside a vehicle using the motion data acquired from the user's phone. In addition, Chen *et al.* [39] provided a smartphone-based system for fine-grained abnormal driving behavior recognition. By leveraging the support vector machine (SVM), it is able to distinguish six different driving patterns accurately. Bhaskar [40] presented EyeAwake, a drowsy driving detection system that employs a set of sensors, including an IR detector, accelerometer, thermistor, and phototransistor. EyeAwake detects driver drowsiness by checking human behavioral and physiological characteristics, such as eye blinking rate, unnatural head nodding/swaying, breathing rate, and heart rate. Toward the most related work in detecting driver distraction, Ahmed *et al.* [18] developed a simulation platform—CAREN to study the impact of smartphone use on driver operations. The testing data from smartphone sensors (i.e., accelerometer and gyroscope) shows that when a driver is holding the phone and attempting to text, call, or read the massage, the signal patterns will have subtle changes that are distinct from normal/safe driving states [41].

Though the above approaches well examined the capability of different smart sensing techniques as well as IMU-based platforms, only a few specific inattentive activities (i.e., the smartphone uses related events) have been addressed. Different from existing approaches, in this article, we leverage trajectory data to identify a number of driver distractions that most commonly occurred in the vehicle. In addition, we further explore the driver reaction caused by such inattentive activities, thus, the proposed model is able to predict the upcoming abnormal behaviors in advance and has great potential to reduce the risk of traffic accidents.

III. Design Consideration

A. Background and Motivation

As the incidence of traffic accidents increases year by year, safe driving has become one of the most concerned topics of intelligent transportation systems. It is desirable to study and analyze unsafe driving behaviors, especially, for commercial vehicles which are more likely to cause fatal accidents since it needs a longer breaking distance to fully stop them due to the large momentum and inertia [42]. Generally, drivers that operate commercial vehicles may experience more complex driving conditions than passenger cars which may cause distractions, such as receiving more phone calls/orders or taking care of passengers [43]. In addition, they may also become fatigue after long-distance driving or working at midnight as shown in Fig. 2. An efficient behavior analysis system could help commercial vehicle drivers to raise risk awareness and shape good driving habits to avoid the accident.

Moving along this direction, in this article, we leverage large-scale trajectory data to study unsafe driving events especially, for commercial vehicle drivers. Unlike existing driver alert systems [18], [37], our approach does not only focus on detecting inattentive activities but also aims to predict potential dangerous driving conditions in advance. Once drivers' unsafe behaviors are identified, there will be a high chance that an accident may occur in a few seconds. Thus, predicting such dangerous conditions is important and necessary, which can prompt the vehicle safety systems (e.g., electronic stability control, intelligent speed adaptation, and collision avoidance system) to better react to incoming operations accordingly and avoid traffic accidents. For example, when the driver is inattentive (e.g., on the phone) and suddenly notice that there are people crossing the road ahead, the driver may behave aggressively to avoid collision while the system should be able to automatically adjusts the torsion force of the steering wheel and prepare for hard braking.

B. Challenges

Analyzing driver behaviors leveraging large-scale trajectory data is very challenging. First, the total number of inattentive events in one single path is limited and each path may contain different types of driving inattention events. Thus, the density of the target samples (i.e., inattentive driving) in the data set is sparse and the amount of data in each class is not equally distributed—imbalanced. The existence of unsafe trajectories only accounts for 48% of the entire trajectory collection, and the average number of points with GPS alerts (for unsafe driving) is less than 8% of each unsafe trajectory. Because the driver performs normally (safely) most of the time while the unsafe driving events only occur accidentally. Second, due to the huge amount of trajectory data, there are missing GPS points on certain trajectories and many GPS points are not correctly aligned on the path. Third, other factors on road may also affect the performance of driving and result in different dangerous driving conditions especially, when the driver is inattentive (as discussed in Section V-E). In order to handle such issues and build a robust model, data augmentation and interpolation schemes are required. In addition, the proposed system should also take the external data such as POIs into account for driver operation/reaction prediction after an inattentive driving event is identified.

C. Inattentive Driving and Abnormal Driving Conditions Scenarios

In this article, we aim to propose an inattentive driving analytics system to detect driving inattentive and warn the driver of abnormal driving in the inattentive driving in advance. Especially, we not only detect whether the driver is inattentive while driving but also combine the driver's inattentive behavior and POI, weather to further predict the driver's abnormal driving condition. We tested four common inattentive behaviors: 1) smoking; 2) phone call; 3) turning back; and 4) yawning, as illustrated in Fig. 1. The abnormal driving conditions are abnormal acceleration/deceleration and abnormal lane change left/right.

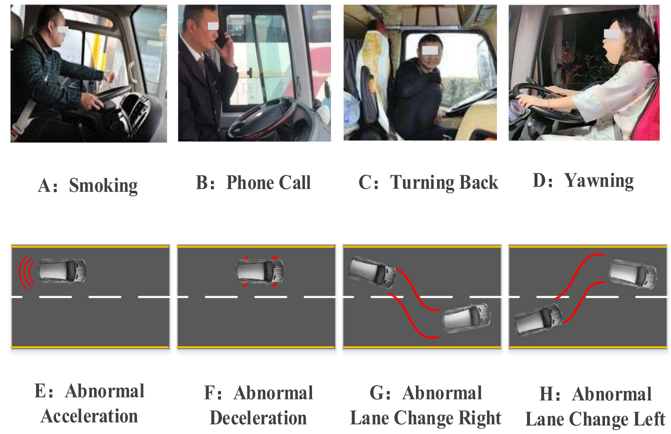


Fig. 1. Inattentive driving and abnormal driving conditions. (A)–(D) represent smoking, phone call, turning back, and yawn, respectively. (E)–(H) represent abnormal acceleration, abnormal deceleration, abnormal lane change right, and abnormal lane change left, respectively.

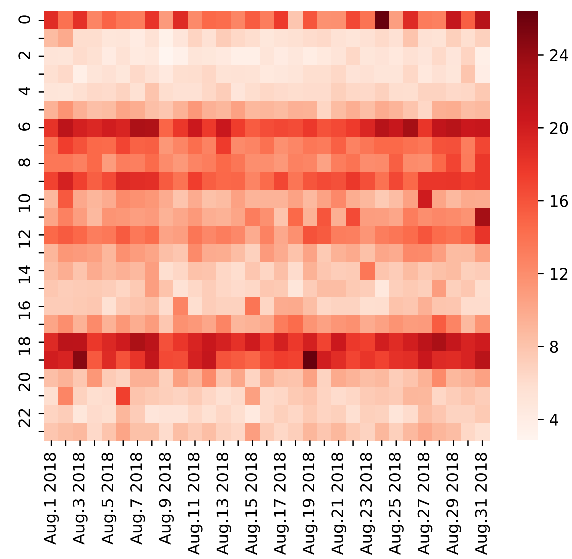


Fig. 2. Distribution of the number of inattentive trajectories in the city in August 2018.

1) Smoking (A): Smoking is a common in-vehicle activity for drivers who are addicted to smoking. Because there are some drivers who think that smoking can relieve driving fatigue, but in fact this produces hallucinations.

2) Phone Call (B): Commercial vehicle drivers use mobile phones more frequently than private cars. According to a survey, 86.3% of drivers believe that reading text messages and smoking on their mobile phones will filter out the troubles of long-distance driving. Due to the danger of using mobile phones while driving, both the United States and Canada prohibit the use of mobile phones while driving [44].

3) Tuning Back (C): When the commercial vehicle reaches the destination or a certain POI area, the driver's eyes will turn to the road signs or landmark buildings outside the vehicle to identify the direction of the destination.

4) Yawning (D): Yawning is the most common in-vehicle activity for drivers. As commercial vehicles require longer driving time, drivers are more prone to fatigue.

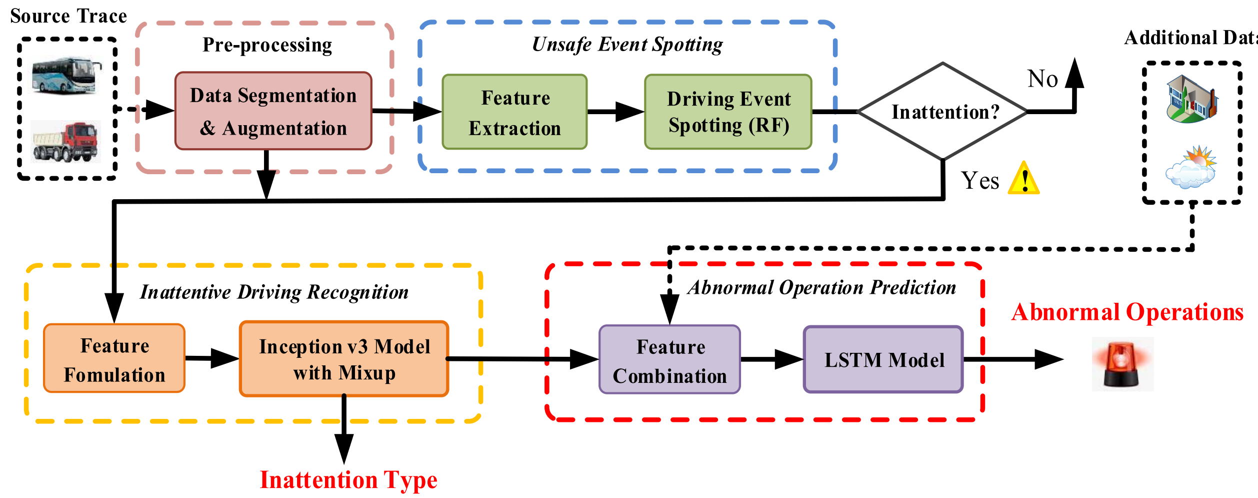


Fig. 3. System overview. Inattention type includes smoking, phone call, turning back, and yawning. Abnormal operation includes sudden acceleration/deceleration and aggressive left/right lane change.

5) Abnormal Acceleration/Deceleration (E and F): After the driver's inattentive behavior, it will affect the normal control and judgment. According to the report [1], if a vehicle is moving forward at a speed of 60 km/h and suddenly decelerates, under the action of inertia, the distance that the vehicle slides forward is proportional to the mass, and the resulting loss is huge.

6) Abnormal Lane Change Left/Right (G and H): Changing lanes in normal driving may not have adverse effects. However, the left and right lane changes caused by slamming the steering wheel after an inattentive behavior are prone to the danger of rollover, leading to traffic accidents.

It is worth noting that inattentive driving behaviors are all unintentional behaviors under the driver's ideology, while aggressive driving is some abnormal operations caused by the driver's stimulation of certain distracted behaviors.

IV. System Design

In this section, we give a detailed discussion of the system architecture and algorithms for inattentive driving analysis.

A. System Overview

The proposed system mainly includes four components: 1) data preprocessing; 2) unsafe event spotting; 3) inattentive driving recognition; and 4) abnormal operation prediction. As illustrated in Fig. 3.

The trajectory data in the real road driving environments may contain ambiguity, uncertainty, and noise, and only a limited number of driving cycles contain the target events (i.e., unsafe driving). To address these issues, we first employ data segmentation and augmentation to format and enrich the original data set. We then proceed the event spotting module to obtain interval candidates from the entire trajectory and filters out the input if it does not belong to the task domain. When a target period of inattentiveness is spotted/detected, the classification module leveraging deep CNN model—Inception v3 with Mixup method will be triggered to recognize the exact type of driver inattention. Once the type of an inattentive driving event is identified, the system is able to predict the driver upcoming reactions by using a well-trained LSTM-based model while at the same time considering the external features (e.g., POI and weather), thereby reducing the risk of potential dangerous driving conditions.

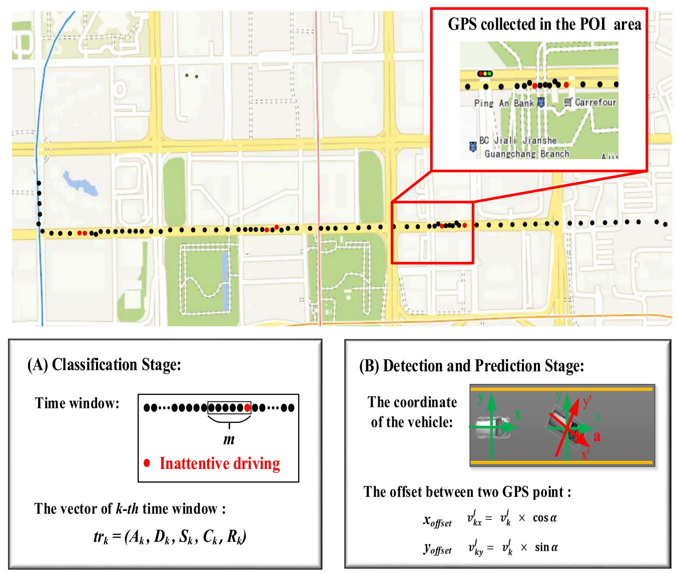


Fig. 4. Data preprocessing. (A) is the trajectory classification stage, which mainly divides the trajectory into time windows with the same length, and then extracts the basic features of each time windows, including acceleration $\mathcal{A}_k$, distance $\mathcal{D}_k$, speed $\mathcal{S}_k$, altitude difference $\mathcal{C}_k$, POI $\mathcal{P}_k$, weather $\mathcal{W}_k$, and peak period $\mathcal{R}_k$. (B) is mainly to extract the speed offset x_{offset} and y_{offset} between two GPS points in the abnormal time windows. The red box represents the distribution of GPS points on the trajectory when the vehicle is driving in an area (like a commercial area).

B. Data Preprocessing

1) Data Segmentation: In order to capture the target events, we employ a sliding window with a fixed length m (a time window contains m GPS cells) along with each trajectory for feature extraction as shown in Fig. 4. The length of the window buffer m is set as 6 which has been proved to be effective in the test and the detection latency is discussed in Section VI-A. In one time window, each GPS cell contains latitude, longitude, and altitude. The definition of the jth GPS cell GPS^j is

$$\text{GPS}^j = \left[\text{lat}^j, \text{lon}^j, \text{alt}^j\right]. \tag{1}$$

Therefore, the kth time window of the trajectory can be expressed as a vector tr_k, which contains a group of latitude, longitude, and altitude from GPS cells

$$\text{tr}_k = \left[\text{GPS}_k^1, \text{GPS}_k^j, \ldots, \text{GPS}_k^m\right]\left(1 \leqq j \leqq m\right) \tag{2}$$

then one driving path can be presented as $\mathcal{T}$

$$\mathcal{T} = [\text{tr}_1, \text{tr}_2, \ldots, \text{tr}_k, \ldots, \text{tr}_n]\left(1 \leqq k \leqq n\right) \tag{3}$$

where n represents the total number of time windows from one single driving path. The stride between time windows is one GPS cell (2 s), and each selected time window (interval) will be used as the input that fed into modules at the next step.

2) Data Augmentation: The traditional data augmentation solution is to replicate the minority samples repeatedly, with the same number of samples in different categories. But such a method can also cause overfitting of the training model. In our design, the Smote [45] is used to oversample the elements in the data set $\mathcal{T}$ that reaches a balanced distribution, the basic steps of the Smote algorithm are as follows.

1) Take a random sample from the minority class tr_p in the set $\mathcal{T}$, and use the Euclidean distance to calculate the distance from this sample tr_p to all samples in the set. Then select κ nearest neighbor samples of this type, that is, κ with smaller Euclidean distance.
2) Set a sampling ratio according to the unbalanced ratio of the sample to determine the sampling magnification β. Then, randomly select one sample among the κ neighbors of this type of sample tr_p, assuming that the selected nearest neighbor is $\widehat{\text{tr}_p}$ as the basis for subsequent calculations.
3) For $\widehat{\text{tr}_p}$ selected in (2), construct a new sample tr_{p_new} with the original sample according to formula

$$\text{tr}_{p\_\text{new}} = \text{tr}_p + \text{rand}(0, 1) \times \left(\widehat{\text{tr}_p} - \text{tr}_p\right) \tag{4}$$

so, the augmentation collection of samples of the smote algorithm

$$\widehat{\mathcal{T}} = \text{Smote}(\mathcal{T}). \tag{5}$$

In the Smote algorithm, the ratio parameter is used to set the number of positive and negative samples so that the ratio reaches 1:1.

C. Event Spotting

1) Feature Extraction: At this stage, the basic features are derived from each time window, including average acceleration $\mathcal{A}$, average distance $\mathcal{D}$, average speed $\mathcal{S}$, and altitude difference $\mathcal{C}$. In addition, as shown in Fig. 2, the occurrence of the inattentive event is highly related to the specific time periods of the day such as the early morning period or the midnight period. Thus, we add the time period $\mathcal{R}$ as an additional feature in driving inattention detection. Therefore, the feature vector of the kth time windows tr_k^s can be formulated as $\text{tr}_k^s = [\mathcal{A}_k, \mathcal{D}_k, \mathcal{S}_k, \mathcal{C}_k, \mathcal{R}_k]$. Then, the input data set of the model can be defined as $\widetilde{\mathcal{T}^s} = [\text{tr}_1^s, \text{tr}_2^s, \ldots, \text{tr}_k^s, \ldots, \text{tr}_n^s]$, where n represents the number of time windows.

2) Driving Event Spotting: As discussed in Section III-B, only a limited number of time periods in the entire trajectory may be involved in unsafe driving. To distinguish the target interval, we use the RF [46] algorithm to make a binary decision based on the input data $\widetilde{\mathcal{T}^s}$. There are two types of labels for the input data, namely, alarm and normal. Alarm refers to the unsafe driving pattern detected and normal means no target events found during this time period. Applying RF-based event spotting not only allows us to filter out nonrelated data and increase the training speed (RF outperforms other methods as shown in Fig. 7) but also enable the system maintain the minimum functionality to alert drivers when other components (i.e., inattention recognition and abnormal operation prediction) cannot work properly.

The main steps of the RF model used in event spotting include the following. First, there are N samples randomly selected with replacement in the sample set $\widetilde{\mathcal{T}^s}$. These N samples are used to train a decision tree as the samples at the root node of the decision tree. Each sample has five attributes, and h out of these five attributes are selected ($h << 5$). Then, from the h attributes, information gain is used to select one attribute as the split attribute of the node. Finally, in the process of forming the decision tree, each node is split according to the previous step. Until it cannot be divided. Eventually constitute an RF model.

D. Inattentive Driving Recognition

This module consists of two phases: 1) feature formulation and 2) event recognition. The recognition model proposed in this article is a deep CNN, which is composed of Inception v3 [47] and Mixup method [48].

1) Feature Formulation: At this stage, first, we set the previous GPS cell GPS^{j-1} before the jth GPS cell as a reference point and calculate $\text{GPS}^j(1 \leqq j \leqq m)$ and GPS^{j-1} velocity offset along the x-axis and y-axis. Assuming that the angle between GPS^j and GPS^{j-1} is α, the velocity offset along the x- and y-axis is defined as

$$\nu_{k_x}^j = \nu_k^j \times \cos\alpha \tag{6}$$

$$\nu_{k_y}^j = \nu_k^j \times \sin\alpha \tag{7}$$

where ν_k^j represents the instantaneous velocity of the vehicle when it reaches the jth position.

This three vectors ν_k^j, $\nu_{k_x}^j$, and $\nu_{k_y}^j (1 \leqq j \leqq m)$ of the kth time windows form a 3-D vector. The vector of the kth time windows tr_k^o is defined as

$$\text{tr}_k^o = \begin{pmatrix} \nu_k^1, & \nu_k^2, & \ldots, & \nu_k^j, & \ldots, & \nu_k^m \\ \nu_{k_x}^1 & \nu_{k_x}^2, & \ldots, & \nu_{k_x}^j, & \ldots, & \nu_{k_x}^m \\ \nu_{k_y}^1 & \nu_{k_y}^2, & \ldots, & \nu_{k_y}^j, & \ldots, & \nu_{k_y}^m \end{pmatrix} \tag{8}$$

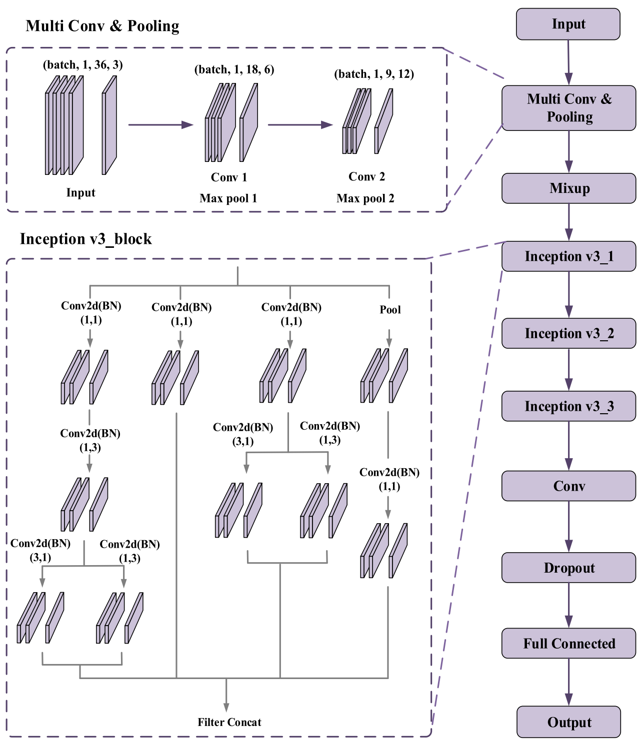


Fig. 5. Inception v3 with mixup model.

so, one driving path can be formed as $\widetilde{\mathcal{T}^o}$

$$\widetilde{\mathcal{T}^o} = \left[\mathrm{tr}_1^o, \mathrm{tr}_2^o, \ldots, \mathrm{tr}_k^o, \ldots, \mathrm{tr}_n^o\right]\left(1 \leqq k \leqq n\right) \tag{9}$$

where n represents the number of time windows in one single driving path.

2) Inception V3: Inception v3 is designed to solve the difference in the proportion of the prominent part of the image [49]. It learns the image data set by setting multiple convolution kernels at the same time and has a good fit. It is worth noting that in the time windows set, the distribution of alarm points for each time window is also very sparse. This causes the choice of convolution kernel to be more biased. If the CNN is simply used, overfitting will occur and it will be difficult to transmit the gradient update to the entire network. Inception v3 adds batch normalization (BN) [50] to each batch of data on the basis of Inception, which can reduce the dependency on parameter initialization, thereby improving the training speed and preventing network saturation. The BN is defined as

$$\widehat{b^{(s)}} = \frac{b^{(s)} - E\left[b^{(s)}\right]}{\sqrt{\mathrm{Var}\left(b^{(s)}\right)}} \tag{10}$$

where $E[b^{(s)}]$ refers to the average value of neuron in the sth batch of data; and $Var(b^{(s)})$ refers to the standard deviation of the input value of each neuron in the sth batch of training data.

The data set $\mathcal{O}$ is used to train. The dimension of each vector o is $3 \times 1 \times m$. Therefore, we change the three initial convolution kernels of Inception v3 and set them to 1×1, 1×1, and 1×1, respectively. The structure is shown in Fig. 5. Adding the 1×1 convolutional layer in front of the 1×3, 3×1 two convolution kernels will help limit the number of channels and improve computing power. After the Inception v3 network, the convolution results of four different convolution kernels can be obtained, then filter concatenate them to the next network.

3) Mixup: To improve the generalization ability of the model and reduce overfitting. We use the Mixup method to expand the virtual samples between different categories. We first choose two different types of samples and labels $(\mathrm{batch}_{\mathrm{tr}_i^o}, \mathrm{batch}_{L_i})$ and $(\mathrm{batch}_{\mathrm{tr}_j^o}, \mathrm{batch}_{L_j})$ then generate mixed samples and labels $(\mathrm{mixbat}_{\mathrm{tr}^o}, \mathrm{batch}_L)$

$$\begin{aligned}\mathrm{mixbat}_{\mathrm{tr}^o} &= \lambda \times \mathrm{batch}_{\mathrm{tr}_i^o} + (1-\lambda) \times \mathrm{batch}_{\mathrm{tr}_j^o}\\ \mathrm{mixbat}_L &= \lambda \times \mathrm{batch}_{L_i} + (1-\lambda) \times \mathrm{batch}_{L_j}\end{aligned} \tag{11}$$

where λ represents the mixing coefficient, λ—Beta$(\alpha,\ \alpha)$, $\alpha\epsilon(0, \infty)$, α controls the intensity of interpolation between the feature–target vector.

Since minimizing the average error of training data is the purpose of neural networks. Such learning rule is also called the principle of empirical risk minimization (ERM) [51]. However, because the driving habits of each driver are different, it leads to the appearance of adversarial samples (that is, the test distribution is slightly different from the training sample data). The traditional ERM method has less ability on interpretation and generalization in this case. Therefore, the data enhancement method Mixup [48] proposes vicinal risk minimization (VRM) by expanding the data of the same type in the data set

$$\begin{aligned}&\mu\left(\mathrm{mixbat}_{\mathrm{tr}^o}, \mathrm{mixbat}_L | \mathrm{batch}_{\mathrm{tr}_i^o}, \mathrm{batch}_{\mathrm{tr}_j^o}\right)\\ &\quad= \frac{1}{n}\sum_j^n \underset{\lambda}{E}[\delta(\mathrm{mixbat}_{\mathrm{tr}^o}, \mathrm{mixbat}_L)]\end{aligned} \tag{12}$$

where $\mathrm{mixbat}_{\mathrm{tr}^o}$ and mixbat_L represents the samples and label enhanced by Mixup. $\mathrm{batch}_{\mathrm{tr}_i^o}$ and $\mathrm{batch}_{\mathrm{tr}_j^o}$ represent the source sample and label, respectively. When $\alpha'0$, it returns to the ERM.

E. Abnormal Driving Prediction

At this stage, we predict the abnormal driving operation by leveraging the information derived from inattention event recognition combined with the external data.

1) Feature Combination: To formulate the input vector of the prediction model, five basic features are considered at each GPS point. The feature vector of the jth GPS point in the time window k can be expressed as

$$\mathrm{GPS}_k^j = \left[v_k^j, \mathcal{N}_k^j, \mathcal{P}_k^j, \mathcal{W}_k^j, \mathcal{L}_k\right] \tag{13}$$

where v_k^j, and $\mathcal{N}_k^j$ represent the velocity and the heading angle change of the vehicle. In addition, the external factors $\mathcal{P}_k^j$ (POI) and meteorological $\mathcal{W}_k^j$ are taking into account as the existence of contextual data (the data can be obtained from public websites, such as Google map and weather underground),[1] and $\mathcal{L}_k$ represents the label which is the type of inattentive detection in the kth time window.

[1] https://www.wunderground.com/

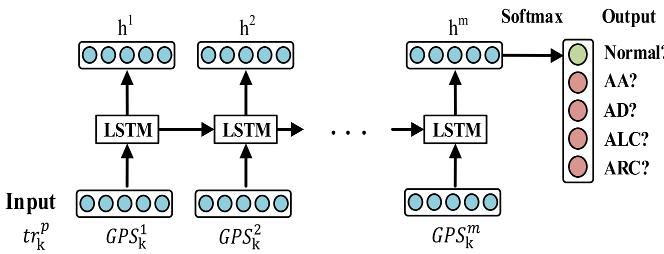


Fig. 6. LSTM-based abnormal operation prediction model.

2) LSTM-Based Prediction: As illustrated in Fig. 6, we are able to formulate the vector tr_k^p (one time window) using GPS_k^j as the input to the LSTM model, and each input vector tr_j^p has a specified size of 5 (the length of time window $m = 5$). Moreover, we collect the event at the next GPS point (2 s later) as the ground truth and define the label as $\mathcal{D}^{k+1}$ (normal or abnormal). Then, the vector of all GPS points of the time window k is

$$\mathrm{tr}_k^p = \left[\mathrm{GPS}_k^1, \mathrm{GPS}_k^2, \ldots, \mathrm{GPS}_k^j, \ldots, \mathrm{GPS}_k^m\right] \tag{14}$$

and the data set of input vectors is defined as

$$\widetilde{\mathcal{T}^p} = \left[\mathrm{tr}_1^p, \ldots, \mathrm{tr}_k^p, \ldots, \mathrm{tr}_n^p\right]. \tag{15}$$

Then, we use the data from $\widetilde{\mathcal{T}^p}$ for LSTM model training, and finally, get the prediction of abnormal operation type.

According to the input vector tr_k^p, we initialize the model with five LSTM cells, and the last element h^m is selected as the output to the softmax layer. At the final step (the classification layer), five classes are specified (i.e., normal driving, abnormal acceleration, abnormal deceleration, abnormal left lane change, and abnormal right lane change) by passing through a fully connected layer.

V. Evaluation

In this section, we conduct extensive experiments to evaluate algorithms and models that proposed in this article for inattentive driving analytics. We first present the experiment setup and data sets and then introduce the baselines used for comparisons of each module. Moreover, we summarize the results of unsafe event spotting, inattentive driving recognition, and abnormal driving prediction, a detailed discussion is also provided.

A. Experimental Setup

We conduct experiments using a large-scale vehicle trajectory data set provided by the commercial insurance company. The company installed OBD-II devices and cameras on the insured vehicles to record driving dynamics and critical events. The time period of data collection is one year—from September 1, 2017 to September 30, 2018. The data set contains over 120 000 driving trajectories captured from nearly 200 vehicles. In the training phase, we are able to derive more than 100 000 driving trajectory vectors for unsafe event spotting, inattentive driving recognition, and abnormal operation prediction. We implement our deep model using PyTorch which is a high-level neural network framework based on the Torch library. The model training phase was done using a server with NVIDIA GeForce GTX 1080 Ti GPU and 64-GB CPU memory. For the optimization method, we use a momentum of 0.9 and a basic learning rate of 0.01. In addition, we set the parameters of the LSTM model epoch = 500, batch = 128, and the optimizer chooses Adam.

TABLE I
Running Time of Each Model for Event Spotting

	Models				
	GBDT	KNN	SVM	LR	RF
ET (s)	**12.074**	**0.016**	**0.632**	**0.003**	**0.028**

B. Baseline and Metrics

We evaluate the performance of each module in our proposed system (i.e., unsafe events spotting, inattentiveness detection, and dangerous event prediction) by using different baselines as follows.

1) *Unsafe Event Spotting:* In this module, we test a number of popular machine learning algorithms that are widely used for classification and regression, such as gradient boost decision tree (GBDT) [52], SVM [53], k-nearest neighbor (KNN) [54], and linear regression (LR) [55].
2) *Inattentive Driving Recognition:* We compare the proposed method (Inception v3) with the classic version of the CNN model.
3) *Abnormal Driving Prediction:* In this module, we use the D^3 system [39] as the baseline, which is an SVM-based model that can train the features and output a classifier to perform the fine-grained identification.

In addition, we use accuracy, precision, recall, and f1 scores to evaluate methods for unsafe event spotting (i.e., RF, GBDT, SVM, KNN, and LR). For multiclass classification—inattentive driving recognition and abnormal driving prediction, weighted average of accuracy (WA), precision (WP), recall (WR), and f1 scores (WF) are applied

$$\mathrm{WA} = \sum_{i=1}^{N} \frac{\mathrm{TP}_i + \mathrm{FN}_i}{\sum_{j=1}^{N} \mathrm{TP}_j + \mathrm{FN}_j} \times \mathrm{accuracy}_i$$
$$\mathrm{WP} = \frac{\sum_{i=1}^{N} \mathrm{TP}_i}{\sum_{j=1}^{N} \mathrm{TP}_j + \mathrm{FP}_j}$$
$$\mathrm{WR} = \frac{\sum_{i=1}^{N} \mathrm{TP}_i}{\sum_{j=1}^{N} \mathrm{TP}_j + \mathrm{FN}_j}, \ \mathrm{WF} = \frac{2 \times \mathrm{WP} \times \mathrm{WR}}{\mathrm{WP} \times \mathrm{WR}} \tag{16}$$

where WP is the sum of true positives across all classes divided by the sum of true positives and false positives across all classes, while WR is the sum of true positives across all classes divided by the sum of true positives and false negatives across all classes. Then, we can calculate the WA and WF accordingly. What is more, the confusion matrix and histogram are also used in the evaluation to present the classification results.

C. Unsafe Event Spotting

As shown in Fig. 7, the RF model outperformed other machine learning algorithms that reached accuracy, precision, recall, and f1-score at 90.92%, 92.25%, 91.76%, and 91.00%,

TABLE II
COMPARE INCEPTION V3 WITH MIXUP AND BASELINES

Model	$\widehat{\mathcal{T}^o}$				
	WA (%)	WP (%)	WR (%)	WF (%)	ET(s)
SVM	76.24	75.26	76.35	76.35	0.946
GBDT	82.56	71.23	72.10	72.31	0.795
DNN	81.45	82.25	81.02	81.45	0.778
CNN	**80.43**	**79.76**	**80.70**	**79.79**	**0.298**
Inception v3	**89.13**	**88.86**	**89.05**	**88.69**	**0.371**
Inception v3 with Mixup	**92.27**	**92.30**	**92.27**	**92.27**	**0.384**

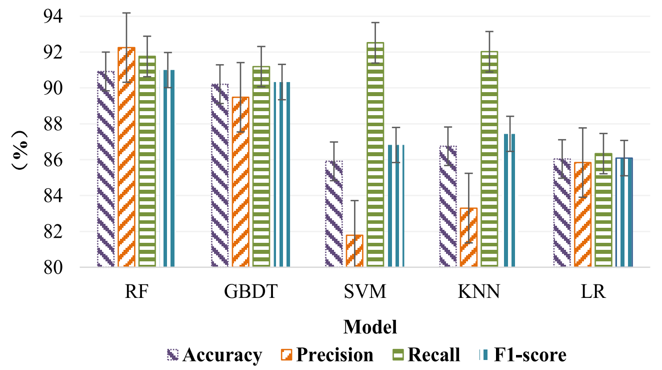


Fig. 7. Accuracy, precision, recall, and F1-score between RF and other machine learning algorithms.

TABLE III
PARAMETERS OF THE PROPOSED CNN MODEL

Type	Input size	Output size
conv 1	36x1x3	18x1x8
conv 2	18x1x8	18x1x64
Inception b1	18x1x64	9x1x64
Inception b2	9x1x64	9x1x128
Inception b3	9x1x128	5x1x128
Fully-connected	5x1x128	5x1x256

respectively. As we use the Smote algorithm to balance the data sets, we are able to train a more generalized model by artificially synthesizing and adding new samples that benefits not only unsafe event spotting but also inattentive driving recognition. As a result, the digital ratio between the training sample, the verification sample, and the test sample is 8:1:1. For other models (GBDT, SVM, KNN, and LR), we conducted experiments according to the settings in the original or highly related paper.

The RF is an ensemble model of decision trees. The time complexity for building an RF is $O(u*v*n\log(n))$, where u is the number of decision trees (10 in our design), n is the number of records for training, and v is the number of attributes in the input data. While the computational complexity at test time for an RF is $O(u*D)$, where D is the maximum depth of the trees (no greater than 5 in our cases). In addition, we also measure the average execution time (ET) of RF and other models for one single input as shown in Table I. The RF model could perform an accurate and timely detection on unsafe driving events.

D. Inattentive Driving Recognition

In this section, we will evaluate our module *Inception v3 with Mixup* and compare it with other baselines based on the data set $\widehat{\mathcal{T}^o}$ discussed in Section IV-D1. The data set $\widehat{\mathcal{T}^o}$ contains more than 6000 samples, among which the ratio of smoking, phone call, turning back, and yawn is 2:3:2:3. The ratio of training samples, test samples, and validation samples is 8:1:1. We can see, from the Table II, our proposed model—Inception v3 with Mixup outperforms other methods.

Comparing to the classic machine learning methods (i.e., SVM and GBDT) and deep learning methods (i.e., CNN and Inception v3). We can observe that the Inception v3 with Mixup could achieve much better performance by considering the locality of the prominent features. In detail, taking the input data set $\widehat{\mathcal{T}^o}$ as an example (shown in Table II), the WA of Inception v3 with Mixup is relatively 17.37% higher than SVM, 21.36% higher than GBDT, 11.73% higher than DNN, and 12.83% higher than CNN. The main reason is that our model does not only expand the breadth of the network, focusing on the extraction of feature locality but also deepen the depth of the network.

Moreover, we choose a simplified Inception v3 structure since it mainly focuses on saving computational power while maintaining the accuracy by modifying the previous Inception Networks. The number of parameters/size of each layer in the Inception v3 model is illustrated in Table III, and the detail of each Inception block is shown in Fig. 5. The average ET of different models for a single event is presented in Table II. Though the time is longer than a standard CNN model, it is acceptable (still less than half a second) considering the recognition accuracy.

1) With/Without the Mixup: In order to further understand the effectiveness of the Mixup method in the deep learning model, we use model ablation to explore its functionality. It is worth noting that the initial convolution kernels of Inception v3 are 1×1, 1×1, and 1×1, respectively. With these parameters unchanged, the performance of the model without Mixup was tested. As shown in Table II, we can see that the model performance became 3.40% higher with Mixup. This is because Mixup reduces the sensitivity of adversarial samples. As shown in Fig. 8(a) and (b), it indicates the results of the category detection with or without Mixup. We can see that the classification detection combined with Mixup can better distinguish the categories of smoking and yawn. This is because Mixup performs data expansion on training set samples, that is, performs linear modeling on different types

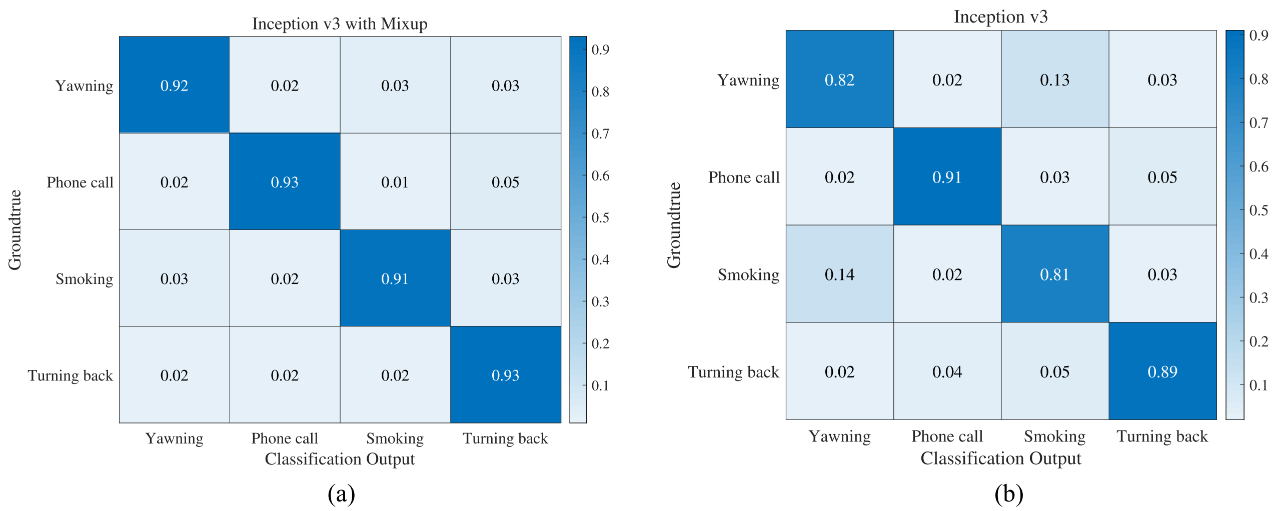


Fig. 8. Classification matrix of with or without mixup. (a) Confusion matrix of inception v3 with mixup. (b) Confusion matrix of inception v3.

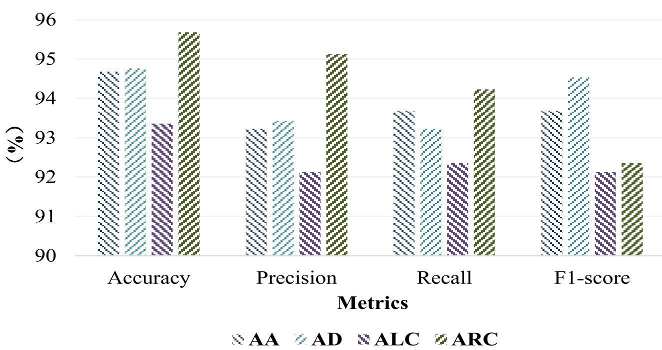


Fig. 9. WA, WP, WR, and WF of abnormal driving condition prediction.

TABLE IV
ABNORMAL DRIVING PREDICTION PERFORMANCE

Model	$\widetilde{\mathcal{T}^o}$				
	WA (%)	WP (%)	WP (%)	WF (%)	ET (s)
SVM	86.17	86.36	86.36	86.36	0.523
RNN	90.24	90.07	90.07	90.07	0.296
LSTM	**91.67**	**90.18**	**90.18**	**90.18**	**0.312**

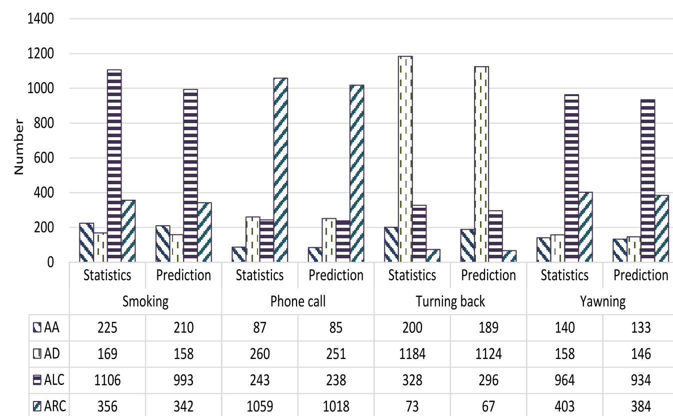


Fig. 10. Distribution of abnormal operations caused by each type of inattentive driving.

of samples to enhance the linear expression between training samples. In addition, Mixup improves the robustness of adversarial samples, especially, on large-scale data sets.

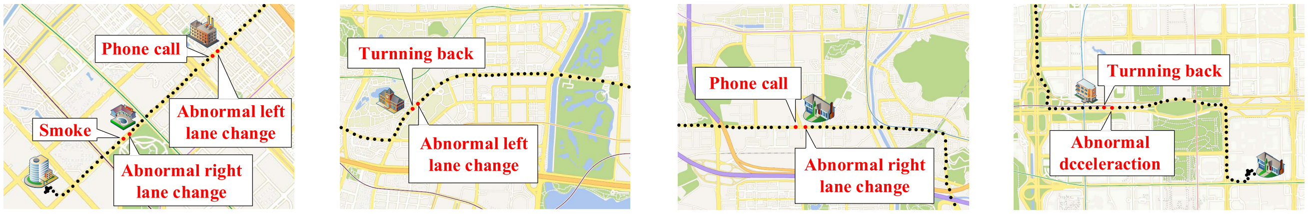
E. Abnormal Driving Prediction

Table IV shows the prediction results of our proposed LSTM-based model comparing with the baseline RNN [56] and SVM. The vector $\widehat{\mathcal{T}^p}$ is used as input to the LSTM module. We can see that our LSTM has the best performance among the other models. The details are as follows. LSTM outperforms RNN and SVM by 1.56% and 6.00%, respectively. The reason why our LSTM-based model is better than RNN is that it considers filtering of the past state in the time series, so as to select the state which has more effects on the current state, instead of simply using the most recent state.

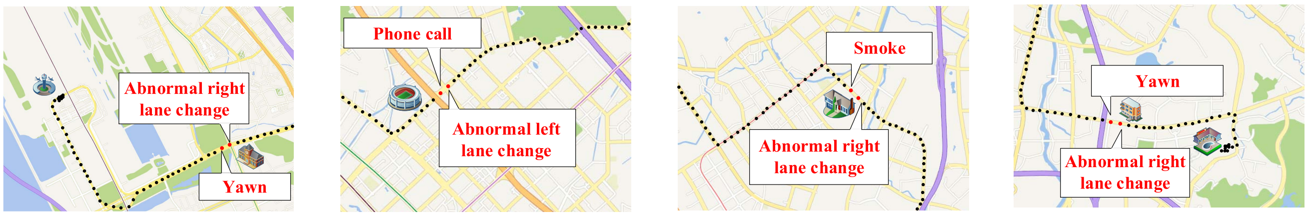
To better understand the computational complexity of our model, we summarize the number of parameters in the LSTM network. There are four neural network layers, including three gates—read/input, write/output, and forget, as well as one cell state. Thus, the total number should be $4 * (n * (m + 1) + n^2)$ based on the structure of LSTM. The average ET for abnormal driving prediction is shown in Table IV. We can see that the proposed model achieved a higher accuracy with just a few extra milliseconds.

1) With/Without External Features: In order to understand the effects of additional features (POI) on event prediction. The results show in Table V. The WA with/without external features are 91.67 and 89.23, respectively. This is because when the vehicle is driving to the POI area, the driver needs to observe the places on both sides of the road during driving, which leads to some inattentive driving behaviors. These inattentive behavior may cause the driver to perform dangerous driving operations.

In order to take a further look into the details, the prediction results for each type of abnormal driving are shown in Fig. 9, and all of them are over 92%. Moreover, we also investigate how different types of inattentive events affect the driver behaviors. As shown in Fig. 10, statistics represent the number of test sets, and predictions represent the correct number



(a)



(b)

Fig. 11. Trajectory of the (a) truck and (b) bus. (a) In the four parts of the truck's trajectory, the driver has abnormal driving behaviors due to inattentive driving. (b) In the four parts of the bus's trajectory, the driver has abnormal driving behaviors due to inattentive driving.

TABLE V
METRICS OF WITH/WITHOUT EXTERNAL FEATURE

	$\widetilde{\mathcal{T}}^o$			
	WA (%)	WP (%)	WP (%)	WF (%)
without POI	89.23	89.35	89.35	89.35
with POI	**91.67**	**90.18**	**90.18**	**90.18**

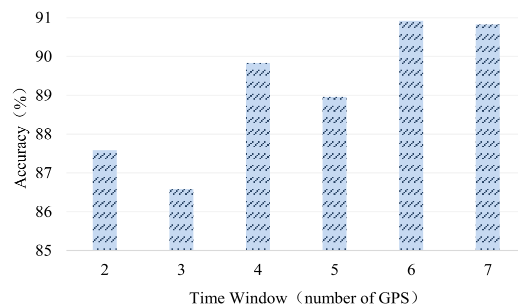


Fig. 12. Recognition accuracy of different time windows.

of predictions. The number of abnormal left lane changes due to the distracted behavior of the driver's smoking was 993, accounting for 58.31%. The number of abnormal right lane changes due to phone call is 1018, accounting for 63.94%. The number of abnormal deceleration operations due to the turning back is 1124, accounting for 67.06%. Due to the phone call, the number of abnormal left lane changes by the driver was 934, accounting for 58.48%. More importantly, by comparing the ground truth of abnormal event distribution with our prediction results in Fig. 10, it demonstrates that our proposed model can accurately infer the potential unsafe operations in advance based on driver current inattentive states.

We also study and analyze the correlations between inattentive driving and abnormal behaviors using real driving cases. Fig. 11(a) shows a trajectory sample of a truck from 1:45 to 3:13 in the morning and the destination is a construction area (i.e., warehouse and factory). The drivers always need to contact/call other people to arrange the deliver time or find the gate to enter, etc. Generally, the driver may use the right hand to hold the smartphone and only the left hand is on the steering wheel. Thus, the driver will do left turn more frequently due to human nature to avoid an obstacle on the road which results in abnormal left lane changes. Fig. 11(b) shows a trajectory sample of the van from 17:56 to 18:50 in the afternoon, and the vehicle traveled through a residential or commercial POI. The driver may get tired after a long time working, there will be a high chance of fatigue driving. Many drivers using cigarette for fighting fatigue (at least they believe it is useful) and they will use left hands (window side) to let the smoke vent out. Thus, the driver may only use the right hand to control the steering wheel which results in abnormal right lane changes.

VI. Discussion

In this section, we discuss the system latency of proposed models and how these models can be extended with real-time vehicle motion sensing based on IMU sensors for practical implications.

A. System Latency

In this work, we aim to build robust models which can be implemented into the real-time driving safety systems. For inattentive driving identification, the length of the sampling window (sliding window for data segmentation) plays a very important role in the latency optimization. It also affects whether the driver is able to have enough time to react on road. The size of the sampling window represents the number of GPS points in the given time period. Therefore, as shown in Fig. 12, we evaluated different window lengths from 2 points to 7 points. The results show that when the length of the sampling window is set as 6—it means six GPS cells are included (the time period between each pair of GPS points in the record is 2 s, thus, the time period of a time window is 10 s) which achieved the best accuracy. We acknowledge that most of the existing driver-assistance systems usually employ a 3-s threshold to alert the unsafe event [57]. However, driver inattention is a complex mental process that may occur before there is any actual behavior detected. Our purpose is to infer potential inattentive driving based on the data derived from the vehicle trajectory/dynamics, thus, it is not necessary to use 3 s as the latency limit but a tradeoff between the window length and the detection accuracy is appropriate.

In addition to the analysis of inattentive driving recognition, how far in advance that our model can predict the abnormal operations is also a critical issue worth discussing. In our design, the latency of the prediction model T_{latency} basically depends on the time interval of data (i.e., GPS points) collection T_c (2 s)

$$T_{\text{latency}} = T_c \times m \tag{17}$$

where m is the number of time intervals between a detected inattentive event and the abnormal operation that be affected. In this article, we set m as 1 and then T_{latency} is 2 s, or it is 2 s prior to the abnormal operation. Though we should predict the unsafe event as earlier as possible, if the time period is over 3 s (T_{latency} will be at least 4 s when m is beyond 2) there may be another unsafe behavior occurring during such time period.

B. Extended System With IMU Sensors

While our system is able to identify driver inattention and predict abnormal driving based on trajectory data, the models proposed in this work are also capable of processing real-time data obtained from smart/onboard devices (e.g., smartphones) which have GPS and IMU sensors embedded. More specifically, the input data/features for each module (i.e., event spotting, inattentive driving recognition, and abnormal operation prediction) have already been extracted from the original GPS trace after preprocessing. For example, the input vector for event spotting consists variables, such as average acceleration, average distance, average speed, etc. Then, we are using the vehicle velocity offset as the features for inattentive driving recognition and combining the velocity, the heading/moving angle, as well as the geographical (POI and meteorological) information for abnormal operation prediction. All of these input data/features can be obtained directly using GPS and IMU sensors (e.g., accelerometer and gyroscope). Consider that the proposed models do not require a high volume of space on the disk (i.e., the size of Inception v3 is only 989 kB—the largest one in the system), we are able to implement our proposed scheme in any off-the-shelf smart devices (i.e., smartphones) with embedded sensors for practical implications. In addition, by taking the advantage of a higher sampling rate (e.g., 20 Hz) of smart sensors, we are able to perform the fine-grained analysis on feature vectors (instead of 2 s per GPS point only) thus, further improve the system accuracy while at the same time minimize the detection and prediction latency.

VII. Conclusion

In this article, we propose an inattentive driving analytics system that detects driver inattentive behaviors and predicts abnormal driving operations to reduce the risk of traffic accidents. To achieve the goal, we propose a deep CNN (Inception v3), which can better extract the local features of the trajectory from the set filters of different widths. We use the Smote algorithm to increase the sample size to balance the sample and adopt the Mixup module to the trajectory data processing. In the evaluation of large-scale trajectory data, the overall performance of our model is promising. The experiment results show that the proposed model has great potential to prevent unsafe driving behaviors and thereby encourage us to further our study on aggressive driving style analysis and other types of in-vehicle activity monitoring.

Specifically, as discussed in Section VI-A, the proposed design is capable of processing real-time sensing data (e.g., acceleration and angular speed) obtained from embedded sensors on smart/onboard devices (e.g., smartphones). In the future, a hybrid system combining the smartphone GPS and IMU sensors, as well as wearable devices [6] is desirable for a comprehensive study of driver behaviors, which could provide a practical implementation that not only tracks vehicle dynamics on the road but also monitors driver real-time in-vehicle activities.

References

[1] P. Baranyi. (2019). *Distracted Driving in Fatal Crashes 2017*. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812700

[2] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan, "Drowsy driver detection through facial movement analysis," in *Proc. Int. Workshop Human-Comput. Interact.*, 2007, pp. 6–18.

[3] B. Bhowmick and K. C. Kumar, "Detection and classification of eye state in IR camera for driver drowsiness identification," in *Proc. IEEE Int. Conf. Signal Image Process. Appl.*, 2009, pp. 340–345.

[4] S. Al-Sultan, A. H. Al-Bayatti, and H. Zedan, "Context-aware driver behavior detection system in intelligent transportation systems," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4264–4275, Nov. 2013.

[5] C.-W. You *et al.*, "CarSafe App: Alerting drowsy and distracted drivers using dual cameras on smartphones," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services* 2013, pp. 13–26.

[6] L. Jiang, X. Lin, X. Liu, C. Bi, and G. Xing, "SafeDrive: Detecting distracted driving behaviors using Wrist-Worn devices," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 1–22, 2018.

[7] M. Raja, V. Ghaderi, and S. Sigg, "WiBot! In-vehicle behaviour and gesture recognition using wireless network edge," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 376–387.

[8] F. Wang, J. Liu, and W. Gong, "WiCAR: WiFi-based in-car activity recognition with multi-adversarial domain adaptation," in *Proc. IEEE/ACM 27th Int. Symp. Qual. Service*, 2019, pp. 1–10.

[9] X. Xu *et al.*, "ER: Early recognition of inattentive driving leveraging audio devices on smartphones," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[10] W. Jia, H. Peng, N. Ruan, Z. Tang, and W. Zhao, "WiFind: Driver fatigue detection with fine-grained Wi-Fi signal features," *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 269–282, Jun. 2020.

[11] SMARTWheel. "The First Intelligent Steering Wheel Cover." [Online]. Available: https://smartwheelusa.com/ (Accessed: Feb. 12, 2021).

[12] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 596–614, Jun. 2011.

[13] B. Jacob and E. Violette, "Vehicle trajectory analysis: An advanced tool for road safety," *Procedia-Social Behav. Sci.*, vol. 48, pp. 1805–1814, Jun. 2012.

[14] C. Chen, D. Zhang, Y. Wang, and H. Huang, *Enabling Smart Urban Services with GPS Trajectory Data*. Singapore: Springer, 2021.

[15] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: Insights from a controlled field study," in *Proc. 11th Int. Conf. Mobile Ubiquitous Multimedia*, 2012, pp. 1–8.

[16] L. Jiang, X. Chen, and W. He, "SafeCam: Analyzing intersection-related driver behaviors using multi-sensor smartphones," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, 2016, pp. 1–9.

[17] M. Fazeen, B. Gozick, R. Dantu, M. Bhukhiya, and M. C. González, "Safe driving using mobile phones," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1462–1468, Sep. 2012.

[18] K. B. Ahmed, B. Goel, P. Bharti, S. Chellappan, and M. Bouhorma, "Leveraging smartphone sensors to detect distracted driving activities," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 9, pp. 3303–3312, Sep. 2019.

[19] Y. Wang *et al.*, "Determining driver phone use by exploiting smartphone integrated sensors," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 1965–1981, Aug. 2016.

[20] P. Subirats, Y. Goyat, B. Jacob, and E. Violette, "A new road safety indicator based on vehicle trajectory analysis," *Transp. Res. Procedia*, vol. 14, pp. 4267–4276, May 2016.

[21] Honda. "Driver Attention Monitor." [Online]. Available: https://www.hondainfocenter.com/2021/CR-V/Feature-Guide/Interior-Features/Driver-Attention-Monitor/ (Accessed: Feb. 12, 2021).

[22] P. Choudhary and N. R. Velaga, "Analysis of vehicle-based lateral performance measures during distracted driving due to phone use," *Transp. Res. F, Traffic Psychol. Behav.*, vol. 44, pp. 120–133, Jan. 2017.

[23] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 1052–1068, Jul. 2004.

[24] L. Xu, S. Li, K. Bian, T. Zhao, and W. Yan, "Sober-drive: A smartphone-assisted drowsy driving detection system," in *Proc. Int. Conf. Comput. Netw. Commun.*, 2014, pp. 398–402.

[25] M.-C. Chuang, R. Bala, E. A. Bernal, P. Paul, and A. Burry, "Estimating gaze direction of vehicle drivers using a smartphone camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 165–170.

[26] A. Ghosh, T. Chatterjee, S. Samanta, J. Aich, and S. Roy, "Distracted driving: A novel approach towards accident prevention," *Adv. Comput. Sci. Technol*, vol. 10, no. 8, pp. 2693–2705, 2017.

[27] D. Lee, S. Oh, S. Heo, and M. Hahn, "Drowsy driving detection based on the driver's head movement using infrared sensors," in *Proc. 2nd Int. Symp. Univers. Commun.*, 2008, pp. 231–236.

[28] R. Torres, O. Ohashi, E. Carvalho, and G. Pessin, "A deep learning approach to detect distracted drivers using a mobile phone," in *Proc. Int. Conf. Artif. Neural Netw.*, 2017, pp. 72–79.

[29] A. A. Assefa and T. Wenhong, "Bio-mechanical distracted driver recognition based on stacked autoencoder and convolutional neural network," in *Proc. IEEE 2nd Int. Conf. Inf. Commun. Signal Process.*, 2019, pp. 449–453.

[30] C. Streiffer, R. Raghavendra, T. Benson, and M. Srivatsa, "Darnet: A deep learning solution for distracted driving detection," in *Proc. 18th Acm/Ifip/Usenix Middleware Conf. Ind. Track*, 2017, pp. 22–28.

[31] Y. Xing, C. Lv, H. Wang, D. Cao, E. Velenis, and F.-Y. Wang, "Driver activity recognition for intelligent vehicles: A deep learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5379–5390, Jun. 2019.

[32] Y. Xie, F. Li, Y. Wu, S. Yang, and Y. Wang, "D3-Guard: Acoustic-based drowsy driving detection using smartphones," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1225–1233.

[33] J. Yang *et al.*, "Detecting driver phone use leveraging car speakers," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 97–108.

[34] B. Goel, A. K. Dey, P. Bharti, K. B. Ahmed, and S. Chellappan, "Detecting distracted driving using a Wrist-Worn wearable," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2018, pp. 233–238.

[35] C. Bi, J. Huang, G. Xing, L. Jiang, X. Liu, and M. Chen, "SafeWatch: A wearable hand motion tracking system for improving driving safety," *ACM Trans. Cyber Phys. Syst.*, vol. 4, no. 1, pp. 1–21, 2019.

[36] B.-L. Lee, B.-G. Lee, and W.-Y. Chung, "Standalone wearable driver drowsiness detection system in a smartwatch," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5444–5451, Jul. 2016.

[37] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing vehicle dynamics for determining driver phone use," in *Proc. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 41–54.

[38] G. Johnson and R. Rajamani, "Smartphone localization inside a moving car for prevention of distracted driving," *Veh. Syst. Dyn.*, vol. 58, no. 2, pp. 290–306, 2020.

[39] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li, "D3: Abnormal driving behaviors detection and identification using smartphone sensors," in *Proc. IEEE Int. Conf. Sens. Commun. Netw.*, 2015, pp. 524–532.

[40] A. Bhaskar, "EyeAwake: A cost effective drowsy driver alert and vehicle correction system," in *Proc. Int. Conf. Innovat. Inf. Embedded Commun. Syst. (ICIIECS)*, 2017, pp. 1–6.

[41] M. L. Watkins, I. A. Amaya, P. E. Keller, M. A. Hughes, and E. D. Beck, "Autonomous detection of distracted driving by cell phone," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, 2011, pp. 1960–1965.

[42] U.S. Department of Transportation, National Highway Traffic Safety Administration. "Distracted Driving." [Online]. Available: https://www.nhtsa.gov/risky-driving/distracted-driving (Accessed: Jan. 11, 2021).

[43] R. L. Olson, R. J. Hanowski, J. S. Hickman, and J. Bocanegra, "Driver distraction in commercial vehicle operations," U.S. Dept. Transp., Federal Motor Carrier Safety Admin., Washington, DC, USA, Rep. FMCSA-RRR-09-042, 2009.

[44] P. Schroeder, M. Wilbur, and R. Peña, "National survey on distracted driving attitudes and behaviors-2015," U.S. Dept. Transp., Nat. Highway Traffic Safety Admin., Washington, DC, USA, Rep. DOT HS 812 461, 2018.

[45] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 878–887.

[46] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vols. 2–3, pp. 18–22, Dec. 2002.

[47] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[48] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017. [Online]. Available: arXiv:1710.09412.

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[50] J. Bjorck, C. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," 2018. [Online]. Available: arXiv:1806.02375.

[51] G. Lugosi and K. Zeger, "Nonparametric estimation via empirical risk minimization," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 677–687, May 1995.

[52] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[53] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, 1999.

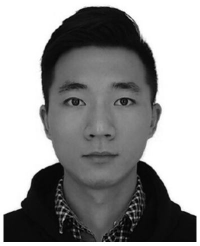
[54] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy K-nearest neighbor algorithm," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 4, pp. 580–585, Jul./Aug. 1985.

[55] J. Cohen, *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Mahwah, NJ, USA: Erlbaum Assoc., 2003.

[56] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. ICASSP*, 2013, pp. 6645–6649.

[57] NHTSA. "Breakthrough Research on Real-World Driver Behavior Released." [Online]. Available: https://www.nhtsa.gov/DrivingSafety/ (Accessed: Jan. 11, 2021).

Landu Jiang received the B.Eng. degree in information security engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010, the master's degree in computer science and minor in construction management from the University of Nebraska–Lincoln, Lincoln, NE, USA, in 2012, and the Ph.D. degree in computer science from McGill University, Montreal, QC, Canada, in 2018. He is currently a Research Associate Professor with Shenzhen University, Shenzhen, China. His research interests include ubiquitous computing, machine learning applications, mobile computing, computer vision, cyber-physical systems, and green energy solutions.



Wen Xie received the B.Eng. degree in computer science and engineering from the Guangzhou College, South China University of Technology, Guangzhou, China, in 2018, and the master's degree from the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China, in 2021.

His research interests include machine learning, neural networks, and Internet of Things.

Dian Zhang (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology (HKUST), Hong Kong, in 2010.

She worked as a Research Assistant Professor with the Fok Ying Tung Graduate School, HKUST. She also worked as an Associate Professor with Lingnan University, Hong Kong. She is currently an Associate Professor with Shenzhen University, Shenzhen, China. Her research interests include big data analytics and mobile computing.

Tao Gu (Senior Member, IEEE) received the B.Eng. degree in automatic control from Huazhong University of Science and Technology, Wuhan, China, the M.Sc. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, and the Ph.D. degree in computer science from the National University of Singapore, Singapore.

He is currently a Professor with the Department of Computing, Macquarie University, Sydney, NSW, Australia. His research interests include Internet of Things, ubiquitous computing, mobile computing, embedded AI, wireless sensor networks, and big data analytics.

Prof. Gu is currently serving as an Editor of *Interactive, Mobile, Wearable and Ubiquitous Technologies*, an Associate Editor of IEEE TRANSACTIONS ON MOBILE COMPUTING and IEEE INTERNET OF THINGS JOURNAL. Please find out more information at https://taogu.site.