Driver Behavior Profiling Using Smartphones: A Low-Cost Platform for Driver Monitoring

German Castignani, Thierry Derrmann, Raphaël Frank, and Thomas Engel
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 4, rue Alphonse Weicker, L-2721 Luxembourg City, Luxembourg. E-mail: german.castignani@uni.lu

Abstract—Today's smartphones and mobile devices typically embed advanced motion sensors. Due to their increasing market penetration, there is a potential for the development of distributed sensing platforms. In particular, over the last few years there has been an increasing interest in monitoring vehicles and driving data, aiming to identify risky driving maneuvers and to improve driver efficiency. Such a driver profiling system can be useful in fleet management, insurance premium adjustment, fuel consumption optimization or CO_2 emission reduction. In this paper, we analyze how smartphone sensors can be used to identify driving maneuvers and propose SenseFleet, a driver profile platform that is able to detect risky driving events independently from the mobile device and vehicle. A fuzzy system is used to compute a score for the different drivers using real-time context information like route topology or weather conditions. To validate our platform, we present an evaluation study considering multiple drivers along a predefined path. The results show that our platform is able to accurately detect risky driving events and provide a representative score for each individual driver.

Digital Object Identifier 10.1109/MITS.2014.2328673
Date of publication: 20 January 2015

IMAGE PUBLISHED BY INGRAM PUBLISHING

I. Introduction

Driving behavior profiling has an increasing relevance in different application contexts. For instance, in the fleet management domain, fleet administrators are interested in fine-grained information about fleet usage, which is influenced by different driver usage patterns. In the car insurance market, Usage-Based Insurance (UBI) or Pay-As-You-Drive (PAYD) schemes aim to adapt the insurance premium to individual driver behavior. In order to track driver behavior, dedicated telematics boxes have been introduced (e.g., Ingenie [1], Fairpay [2]) to log different sensing variables and driving events. The information logged by these boxes can then be manually retrieved or sent over the Internet through a wireless connection. However, the main drawbacks of such systems are their high initial cost and low customer acceptance, which limit wide and rapid platform deployment.

Due to increasing sensing capacities and the proliferation of mobile devices like tablets and smartphones (e.g., accelerometers, magnetometers, GPS), smartphone-based telematics systems are gaining increasing attention. In the car-insurance market, Aviva RateMyDrive [3], StateFarm DriverFeedback [4] and AXA Drive (in Belgium) [5] appear to be the most popular mobile

Driver monitoring and profiling with mobile devices is an emerging trend that suits the needs of multiple markets.

applications for iOS and Android. In the case of RateMyDrive and DriverFeedback, the provided score is used as an input to adjust the insurance premium, providing up to 20% discount. In contrast, Greenroad [6] is an online platform for fleet management. In this platform, drivers use the sensing application and regularly send driving traces to the system, which aggregates metrics from different drivers to provide fleet administrators with a description of individual riskiness, eco-driving and fleet usage information (e.g., fuel consumption, CO_2 emissions).

In our previous work [7] [8], we analyzed the capabilities of smartphones to profile drivers. We studied the output of smartphone sensors and GPS under risky and normal driving conditions in order to provide the driver with a score. In the platform proposed in this paper, we focus on detecting risky driving events rather than analyzing driving traces as a whole and calculating a score at the end of the trip. By detecting events, we are able to provide the driver with immediate feedback so as to allow him to adapt his driving. However, the heterogeneity of existing smartphone sensing hardware and vehicle characteristics prevents the definition of fixed rules to profile drivers. To address this issue, we propose a fuzzy logic mechanism to detect risky driving events, including over-speed, acceleration, braking and steering. In order for the system to provide meaningful results, we have implemented an adaptive profiling mechanism that works independently of the type of mobile phone and car. For each driver we collect an initial dataset and perform a statistical analysis to identify event thresholds. We also propose a scoring process that assigns different levels of riskiness to driving events depending on the road topology and weather information. The fuzzy logic event detection mechanism is implemented in an Android application. In this paper we present an evaluation study that analyzes the driving profiles of multiple participants and computes a representative score reflecting the *risk factors* for each driver.

SenseFleet is a new smartphone-based driver profiling platform. Compared to existing tools, SenseFleet adaptive profiling can correctly detect risky driving events, independently of the mobile device and vehicle used, by performing a statistical analysis on the data collected by each driver. This allows the identification of dynamic event thresholds that are unique for each driver. Moreover, because of the way sensing data is considered in SenseFleet, there is no restriction on the initial positioning and orientation of the device. By fusing a variety of sensing data, the device can be manipulated and its orientation changed when the vehicle is stopped without introducing any bias in the measurement which would negatively impact event detection.

The remainder of this paper is organized as follows. In Section II we discuss potential uses of such a platform. In Section III we present the related work on smartphone-based driver profiling tools. Then, in Section IV, we provide the details on our platform, including event detection and driver scoring mechanisms. In Section V, we present an evaluation study of the SenseFleet platform. For this study we consider different drivers following a single path using an electric vehicle in order to analyze the event detection and scoring performance and also the effect of using different phones and specific parameters for the application. Finally in Section VI we present a discussion of open issues and in Section VII we conclude the paper.

II. Driver Monitoring: Potential Usages

Driver monitoring and profiling with mobile devices is an emerging trend that suits the needs of multiple markets. A potential market is car insurance, which has been interested in monitoring driving activities in order to provide fair insurance premiums to its customers. This concept is referred to as Pay As You Drive (PAYD) or Usage Based Insurance (UBI) [9]. While there has been an interest in UBI in the past, most solutions have focused on telematics systems (e.g., black boxes), that must be fixed into the car to gather a number of parameters that classify drivers. Such systems have not succeed for several reasons. First, telematics boxes imply an investment for insurance companies, not only to provide the boxes and communication links to their clients but also to maintain them. Experience shows that drivers have been reluctant to accept such equipment because they disliked being observed. A study carried out by Delloite [10] confirms that 58% of young drivers in the UK do not want to use telematics boxes. In the case of smartphone-based driver monitoring, costs are greatly reduced since all it requires is the installation of a mobile application on the driver's personal device. Also, since there is a trust relation between drivers and their personal mobile devices, users may be less averse to smartphone-based monitoring, since they have an increased level of control over the monitoring service.

A second potential application of smartphone-based driver monitoring is the fleet management market. Logistics fleet administrators need to know how their vehicles are being used and how their drivers behave in order to mitigate potential risks and reduce operational costs. Nowadays, there are numerous solutions that are based on telematics boxes, mostly to record the distance driven and the speed distribution. Corporate smartphones can serve an additional purpose by replacing legacy telematics systems. Further, the trend to closely integrate mobile devices with the on-board systems will allow additional information to be retrieved from the vehicle. One possibility is to use On-Board Diagnosis (OBD-II) adapters that can be plugged into the vehicle's Controller Area Network (CAN) and wirelessly transmit relevant vehicle information to the smartphone (e.g., speed, fuel consumption, engine load, fault codes). Then, using the Internet connection of the smartphone, fleet administrators may access real-time vehicle and driver information in order to optimize logistics and reduce overall fuel consumption.

III. Related Work

In this section, we introduce some existing driver profiling systems based on smartphone sensing data. Eren et al. [11] designed a driver classification algorithm that distinguishes between risky and safe drivers. They considered smoothed acceleration, gyroscope and magnetometer data from smartphones to detect start and end times for driving events (e.g., sudden maneuvers, aggressive steering, braking or acceleration) using a moving average algorithm and empirical thresholds. The authors computed the similarity of each event to template data (i.e., for risky and safe event patterns that had been previously collected) using Dynamic Time Warping (DTW) and used Bayesian classification to decide whether the driver was risky or safe. They present an evaluation study for fifteen drivers using iPhone devices and fixed departure and arrival points, showing a successful classification rate of 93.3%. Johnson et al. [12] also proposed a DTW-based driver profile algorithm, MIROAD, using smartphone sensors, GPS and camera. Their work evaluated the performance of different sensor fusion sets to detect lateral and longitudinal movements. After evaluating over 200 driving events, the authors showed that the sensor fusion set composed of the x-axis (i.e., gravity axis) rotation rate, y-axis (i.e., lateral movements) acceleration and pitch provide the best classification performance using DTW.

Paefgen et al. [13] focus on the precision of smartphone sensing data for an analysis of driver behavior mainly oriented towards insurance market. After a calibration process, in which the user manually sets the main direction of the vehicle, the mobile application starts collecting acceleration, braking and steering events. These events are triggered if the sensing data surpasses some predefined thresholds (e.g., $0.1g$ for acceleration and braking and $0.2g$ for steering). The authors presented a measurement study to compare event detection using smartphone sensors against a fixed telematics box based on an internal Inertial Measurement Unit (IMU). They observed that the obtained event count distribution matched different statistical distributions, which was mainly due to variations in smartphone-to-car fixing and positioning inside the vehicle. However, the authors found some correlations between smartphones and IMU-based events and described some possible sources of error.

You et al. [14] describe CarSafe, a smartphone application which fuses information from front and rear cameras, sensors and GPS to detect dangerous driving events. In particular, the authors showed that drowsiness (one of the main causes of car accidents [15]) can be detected using the front camera and image processing algorithms with an accuracy of 85%.

With the aim of providing drivers with useful hints to reduce energy consumption, Araujo et al. [16] developed a smartphone application that combines GPS and CAN-bus information (using an OBD-II device). Some of the possible hints are to switch off the engine, to shift gears earlier or to decelerate. As input data, they considered average, minimum and maximum values for speed, acceleration and fuel consumption, which they combined using a fuzzy system. They evaluated and validated their algorithms using a mobile platform and several experiments on a single car.

Also based on smartphones, Mohan et al. describe Nericell [17]. In this work, they focus not on driver behavior analysis and profiling but rather on using acceleration, microphone and GPS data to detect the road's quality (e.g., presence of potholes, bumps) and traffic condition (e.g., stop-and-go, fluid). However the techniques for acceleration and braking event detection presented in their paper are also suitable for the driver profiling problem.

Fazeen et al. [18] highlight the concept of *feedback* when monitoring drivers in order to effectively correct bad driving habits and behaviors. They collected a set of experimental data to analyze the detection of acceleration, braking and lane-changing events. In their experiments, they carefully fixed the phone in predefined positions inside the car by always keeping the device parallel to the floor, with the top of the device pointing forward. This facilitates the identification of driving events, since longitudinal and lateral acceleration samples exactly match the x and y axes of the device's coordinate system, this way no coordinates transformation is needed. In order to classify driving events, they considered the acceleration variation (jerk) and a maximum acceleration threshold. However, the need to fix the device's position limits the usability of the proposed platform, since a slight modification of the phone orientation (e.g., user manipulation, device vibration) considerably impacts the event classification performance.

As it has been presented, existing solutions for event detection are commonly based on fixed thresholds for the different input variables used to detect acceleration, braking or steering events. To overcome this limitation, we propose an adaptive profiling mechanism that consist in a calibration phase that will dynamically set up input variable

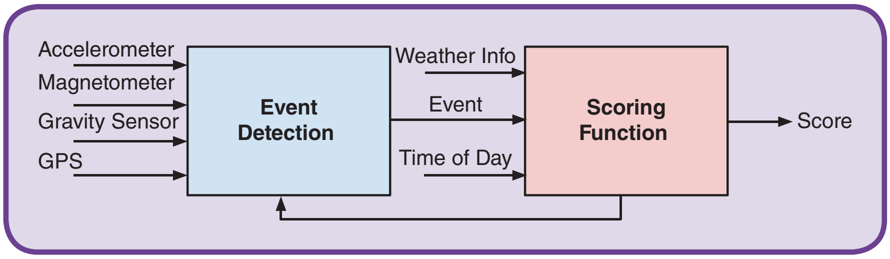

FIG 1 Event detection and scoring.

thresholds by performing statistical analysis. In the following we introduce our solution and evaluate its performance through experimentation.

IV. Risky Event Detection and Scoring

In this section, we describe *SenseFleet*, the proposed event detection and scoring platform. As illustrated in Fig. 1, the event detection algorithm considers the output of motion sensors and GPS. Acceleration, braking and steering events are detected using fuzzy logic. Moreover, we consider overspeed events by considering real speed limits for each particular road. During a single trip, these events are combined with weather information and time-of-day to better determine the riskiness of the events and score the driver.

A. Application Platform: SenseFleet

As shown in Fig. 1, this application includes the event detector and the scoring mechanisms. It stores event detection and scoring data in an internal SQLite database. The traces generated for single or multiple trips can be remotely pulled to a central server, which aggregates data from different trips and drivers for further analysis and reporting. SenseFleet's user interface shows the overall score for all the trips and the relative distribution of event types. The driver can also see the instantaneous score and event rate for the current trip. Each time an event is detected, a sound and text notification is triggered by the application. Moreover, the user has the possibility to analyze his driver performance offline, through the mobile application or a web-based dashboard.

B. Fuzzy Logic Based Event Detection

Existing driver profiling mechanisms are generally based on multiple input data and fixed-threshold based event detection. For example, over-speed events are triggered if the vehicle's instantaneous speed is greater than 120 km/h [6], which is unrealistic for example in motorway scenarios, where speed limits can vary due to different type of roads. Commercial applications like Greenroad [6] also rely on GPS and smartphone sensor data to detect events. In this application, the score is then simply calculated as an event rate, i.e., the number of events per unit of distance that the application has counted. In this case, all types of events have the same relevance for scoring and are simply merged in a global event counter.

In SenseFleet we consider GPS and motion sensor input data simultaneously. As illustrated in Fig. 2, the internal linear accelerometer is used to compute the jerk (i.e., the rate of change of the acceleration with respect to time). We consider the output of the device's accelerometer, $a(t) = [a_x(t), a_y(t), a_z(t)]$ in m/s^2, and the magnitude of the acceleration vector as described in Eq. 1.

$$|a(t)| = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \quad (1)$$

Initially, we tried to infer longitudinal and lateral movements of the car by considering each acceleration axis independently. For this purpose, we translated the acceleration vector to the Earth coordinate system in order to be coherent with the vehicle's trajectory. However, even when those signals were filtered (using Kalman filters) it was not possible to clearly decompose vehicle's longitudinal and lateral movements from this output. We then decided to compute the magnitude of the acceleration vector to mitigate this problem. Note that the magnitude is invariant with regards to the coordinate system (e.g., device, Earth), which allows device rotation or manipulation when the driver is using the application and the vehicle stopped.

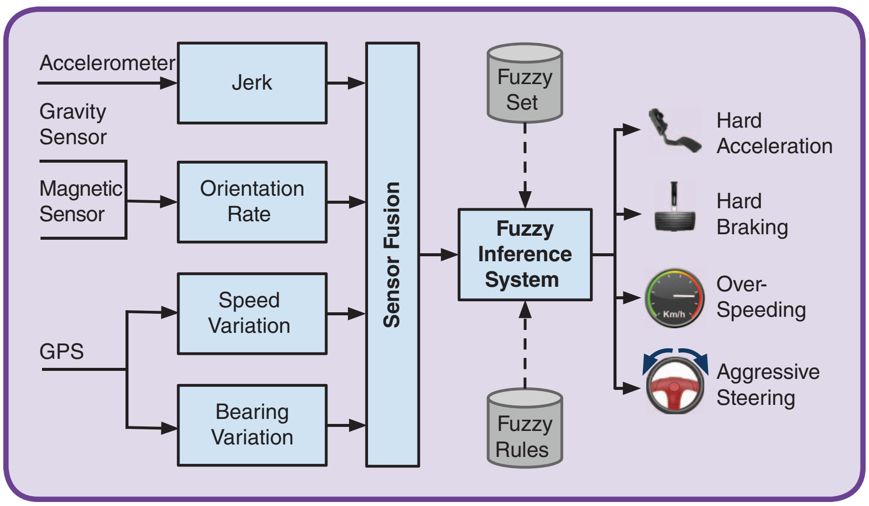

FIG 2 Event detection process.

Finally, jerk (j) is calculated as the time derivative of the acceleration magnitude (see Eq. 2).

$$j(t) = \frac{d\,|\,a(t)\,|}{dt} \tag{2}$$

Having computed jerk, in order to have a measure of the vehicle's direction variation (yaw rate) we use the device's magnetic and gravity sensors to compute the orientation vector. This vector includes the yaw, pitch and roll, which that characterize rotation around the different axes. In our application, we consider the yaw rate, y, to measure the vehicle's steering as the rotation around the axis perpendicular to the earth surface.

A major limitation of motion sensors (e.g., accelerometer, magnetic sensor and gravity sensor) is the high exposure to noise, which is mainly due to electromagnetic interference and device vibration [19]. For this reason, in the proposed mechanism we *fuse* motion sensor data with GPS data in order to accurately detect driving events. In particular for GPS-based metrics, we consider speed variation (ΔS) and bearing variation (ΔB). Note that GPS data is obtained at a sampling rate of 1 Hz and that the speed variation is computed as the difference between two consecutive samples (see Eq. 3).

$$\Delta S = S(t) - S(t-1) \tag{3}$$

Similarly, the bearing variation (i.e., the variation in the relative angle to the north, measured in $°/s$) is calculated as in Eq. 4.

$$\Delta B = B(t) - B(t-1) \tag{4}$$

The different input variables are obtained at different sampling rates. In the case of motion sensors, the sampling rate varies between 20 and 50 Hz, depending on the device hardware and operating system version. To mitigate these effects, we implemented a Sensor Fusion layer that synchronizes motion sensors and GPS samples in order to perform event detection based on different time series. Given that GPS samples are received at a fixed rate of 1 Hz, we store j and y samples over the last second. Then, for each GPS location fix, we compute ΔS and ΔB, the average yaw rate ($\mu(y)$), and the jerk standard deviation ($\sigma(j)$). We consider the jerk standard deviation instead of raw jerk or acceleration in order to further mitigate the effect of device vibration during the measurements while driving. We observed after several trials that during an acceleration, braking or steering event, the jerk standard deviation showed a clearer variation than the total acceleration or the average jerk.

As shown in Fig. 2, in order to detect driving events, we set up a fuzzy system [20] that consists of a fuzzification phase of the input data (i.e., $[\sigma(j), \mu(y), \Delta S, \Delta B]$) and the application of a set of fuzzy rules. Each rule evaluates a combination of different possible fuzzy values of the input variables and outputs a type of event (e.g., hard acceleration, hard braking, aggressive steering, over-speeding). The rules were manually derived after analyzing input variable values in a controlled scenario, considering different types of maneuvers. For the input variable fuzzification process, we consider trapezoid membership functions. For the output variable, we consider a single crisp value for each different type of event to allow the center-of-gravity defuzzification process to detect events individually. The fuzzy sets for the variables are stated in Table 1. For the fuzzy system implementation we used jFuzzyLogic [21], an open source fuzzy logic implementation for Java. However, as discussed below, the limits for those sets are dynamically established after a calibration process.

The fuzzy rules indicate the specific conditions for an event to be triggered. As an example, in order to detect *hard acceleration*, the system considers the following rule:

IF
($\sigma(j)$ IS HIGH OR $\sigma(j)$ IS VERY-HIGH) **AND**
($\mu(y)$ IS LOW) **AND**
(ΔB IS LOW) **AND**
(ΔS IS HIGH-ACC)
THEN
event IS ACCELERATION

As shown in the example, in order to trigger an acceleration event, the system evaluates $\sigma(j)$, $\mu(y)$, ΔB and ΔS. Note that the rule checks for a high speed variation and low yaw rate and bearing change.

1) Fuzzy Sets definition (Calibration phase): In order to detect events independently of the mobile device and different vehicle conditions, we carried out an initial calibration phase to establish the boundaries of the fuzzy membership functions for input variables. In fact, different vehicles have different acceleration, braking and steering patterns, e.g., the accelerometer output of a small city car is different compared to a luxury sedan. Moreover, different smartphones embed different sensor chipsets that have different sampling rates and magnitudes. As a consequence, it is necessary to calibrate the system to each particular vehicle and device. This process is performed the first time the

Table 1. Fuzzy sets.

Variable	Sets
$\sigma(j)$	LOW, MEDIUM, HIGH, VERY-HIGH
$\mu(y)$	LOW, MEDIUM, HIGH, VERY-HIGH
ΔB	LOW, MEDIUM, HIGH, VERY-HIGH
ΔS	HIGH-DEC, LOW-DEC, STABLE, LOW-ACC, HIGH-ACC

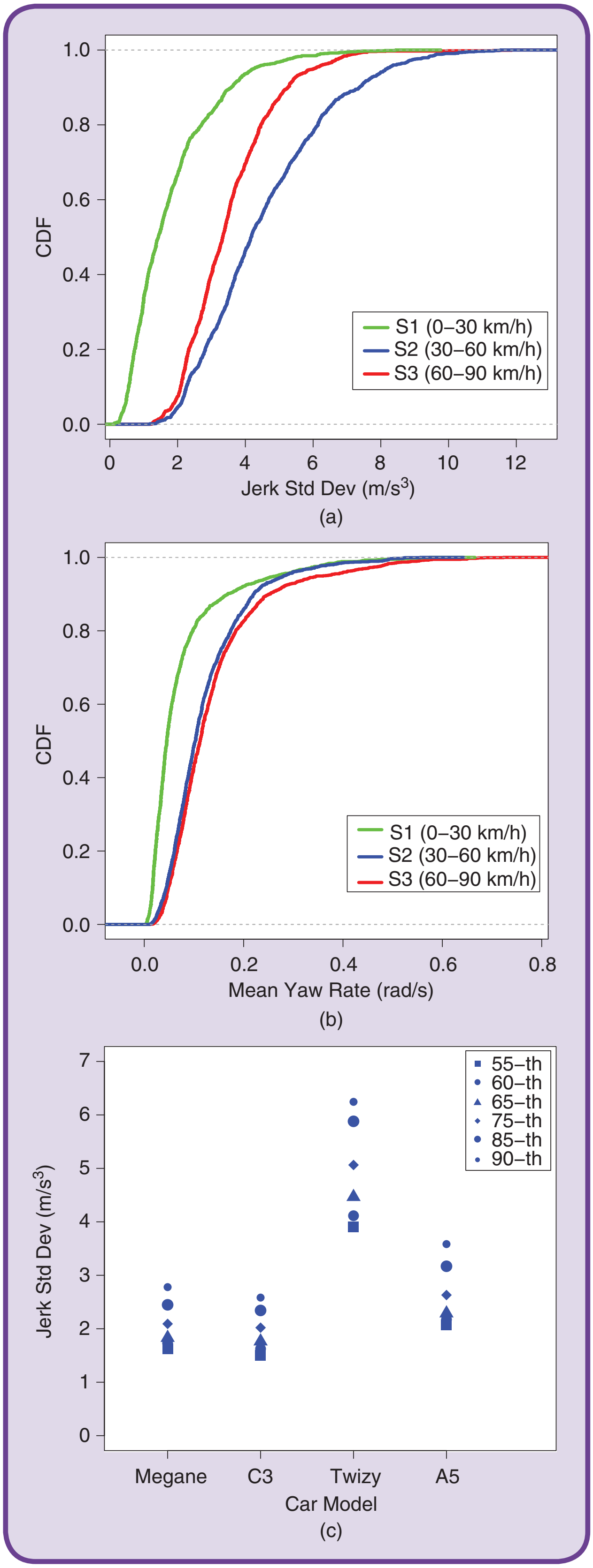

FIG 3 Calibration process. (a) $\sigma(j)$ CDF for n = 1500, (b) $\mu(y)$ CDF for n = 1500, and (c) Calibration process for different cars.

application is used by a single driver and vehicle. This calibration phase consists in the collection of a fixed number of input samples and the computation of their cumulative distribution function. Each sample represents a collection of the input variables $[\sigma(j), \mu(y), \Delta S, \Delta B]$ at a given time. After the collection of the calibration samples, the system dynamically adjusts the fuzzy sets for the variables and starts the event detection and scoring phase. To this end, we consider different percentiles of the cumulative distribution of $\sigma(j)$ and $\mu(y)$, representing the threshold values for event detection.

In more detail, the calibration process is segmented by speed ranges. This is to mitigate motion sensor's noise due to vehicle's speed. To this end, n training samples are collected for each of the following speed ranges (in km/h): $S_1:(0,30], S_2:(30,60], S_3:(60,90]$ and $S_4:(90,\infty)$. These speed ranges were manually derived after several experimental trials. Note that the dynamic adjustment of the fuzzy sets is only done for jerk $(\sigma(j))$ and yaw rate $(\mu(y))$, since speed variation and bearing rate fuzzy sets can be fixed regardless of the mobile-device-specific hardware and the vehicle characteristics. In Fig. 3a and 3b we show the cumulative distribution functions (CDF) for $\sigma(j)$ and $\mu(y)$ respectively. These distributions correspond to a calibration phase of 1500 samples using a Renault Twizy, an ultra-compact electric vehicle limited to 80 km/h maximum speed, and are then used to establish the limits of the fuzzy sets. Note that the highest jerk standard deviation is observed in the second speed set (between 30 and 60 km/h) and is lower at higher speeds, where the driver tends to have a more constant speed pattern (e.g., free-flow along an avenue or highway). In the case of the yaw rate, as expected, for an increasing speed, the angular velocity at intersections increases. In practice, the fuzzy limits for $\sigma(j)$ and $\mu(y)$ (low, medium, high and very-high) are obtained from the CDFs by considering the last percentiles of the distribution. This dynamic adaptation of the fuzzy sets in the calibration phase allows acceleration and steering events to be identified at different speeds. Also, as illustrated in Fig. 3c, this calibration process allows the fuzzy sets to be matched to the acceleration profiles of different car models, with differing power and suspension. In Fig. 3c, we used SenseFleet in a Samsung Galaxy Gio (S5660) smartphone and $n = 1500$. The results show that the Renault Twizy has a considerably different distribution for $\sigma(j)$, compared to the three internal combustion engine cars (Citroen C3, Renault Megane and Audi A5) due to stiffer suspensions in the former.

C. Driver Scoring

During a single trip, the user collects input data from motion sensors and GPS and the event detection system decides whether these samples correspond to a risky driving event or not. The device counts the different events of different types. For each event, the device gathers the current weather condition and the time of day. To this end, we make use of the OpenWeatherMap API [22], which provides very detailed weather information through a web service, including rain and snow levels, temperature and humidity. Time of day is obtained by using sunrise and sunset information compared to current time. Weather is organized in JSON format and is obtained by simply requesting a web service and providing latitude and longitude as parameters.

In SenseFleet, any single trip is scored with a value between 0 and 100 (being 100 the best possible score). When a trip starts, the driver gets 100 points. Then, when an event occurs the driver loses points depending on the type of event and its context (i.e., weather condition and time of day). Both weather condition and time of day were considered since their impact on fatal accident rates has been proven in different studies [23] [15]. For instance, as stated in [23], nighttime increases fatal accidents rate by a factor of five. In our platform, weather conditions can be *normal*, *rain*, *storm*, *snow*, *fog* or *extreme*. Possible values for daytime are *day* or *night*. Based on both variables, we defined four different severity levels for the events: *low*, *medium*, *high* and *extreme*. Each severity level corresponds to a number of points to be deducted from the score.

We assigned a greater severity level for events done during nighttime and with bad weather conditions so there is a higher impact on the score than the case of events performed in normal weather condition and daytime. Then, for each combination of event type and severity, the system reduces the score by a predefined number of points. On the other hand, if the driver style improves during the trip, points are earned if no events are detected.

V. Experimental Evaluation

We performed experimental evaluation of our platform. We focused on two main aspects, event detection and driver scoring. For each aspect we carried out a set of experiments considering two different testbeds. All the experiments have been carried out with similar traffic conditions.

A. Event Detection Accuracy

1) Testbed Setup: In order to measure the accuracy of the event detection, a single driver performed four different runs with a single car (Renault Twizy) using different numbers of samples for the calibration phase. The methodology of the experiment consisted in counting the number of detected and undetected events of each type. In our previous work [8] we have performed a set of experiments to validate the capacity of smartphone sensors to detect driving maneuvers by comparing it to OBD-II vehicle data. In the proposed experiment, we focus on the accuracy of the proposed fuzzy logic event detection depending on the number of calibration samples.

2) Results: We considered four different values for the number of calibration samples and measured the performance of the event detection in terms of three parameters. First, we computed the number of *True Positive* events, i.e., the number of events that were actually been due by the driver and detected by the system. Then, we considered the *False Positive* events as the number of events that were detected by the system but that were not actually due to the driver. Finally, the *True Negative* events are those events that were due to the driver but were not detected by the system. Regarding the methodology of these experiments, a single driver has set up a Samsung Galaxy Gio (GT-S5660) smartphone in a car holder. The detected events were notified by SenseFleet with both a text and audio notification. The driver used a second mobile device also placed in a car-holder to tally the true positive, false positive and true negative events. The results of this experiment are shown in Fig. 4. We observe that for more than 1500 calibration samples, a true positive rate (i.e., events that are correctly detected) greater than 90% was obtained, requiring a calibration time of at least 17 minutes and a driven distance of 9.21 km. Note that the calibration process was automatically paused and resumed if the trips were not long enough to finish the calibration in a single run.

B. Scoring Comparison

1) Testbed Setup: In this Section, we evaluate the performance of SenseFleet in a real environment. To do this, we collected traces from 10 different drivers using the same car (Renault Twizy) over a predefined path[1]. This 9.8 km-long path encompassed different types of roads having different speed limits in the city of Luxembourg, allowing the

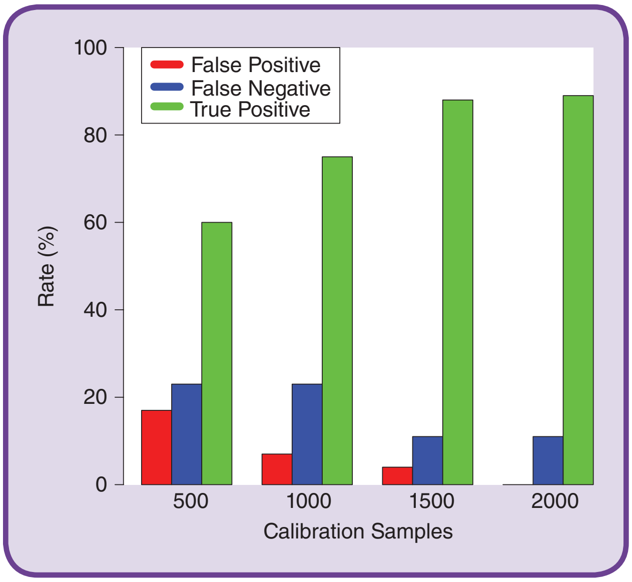

FIG 4 Event detection accuracy for different n values.

[1]49.623934N, 6.149631E.

Table 2. General results.

D	ACC		BRK		STE		OVS		Total		Score	
	C	*A*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*	*C*	*A*
01	1	11	0	12	3	6	1	12	5	41	100	35
02	2	16	0	12	4	10	1	9	7	47	87	0
03	4	9	1	4	5	3	6	9	16	25	83	18
04	4	7	3	5	0	7	1	6	8	25	77	60
05	5	16	3	6	7	9	1	7	16	38	69	31
06	2	17	0	15	12	14	3	15	17	61	69	0
07	4	8	1	8	3	5	1	16	9	37	69	0
08	9	9	2	8	8	12	5	10	24	39	68	26
09	4	13	1	10	6	6	2	9	13	38	64	0
10	5	15	4	5	6	9	6	9	21	38	58	33
Avg	16	38.9	4	12.1	1.5	8.5	5.4	8.1	2.7	10.2	74.4	20.3

driver to perform different maneuvers. All the experiments were performed during daytime and with variable weather conditions (dry, rainy and foggy). A single experiment consisted of a calibration phase ($n = 1500$) and two laps along the predefined path. During the calibration phase, the driver collected input variable samples to set up the fuzzy system. We fixed $n = 1500$ as a good compromise between detection accuracy and calibration phase delay. Once the system was calibrated, a notification told the driver to come back to the point of departure. The driver was then asked to drive two laps. In the first lap, the driver was asked to drive calm, by observing speed limits and avoiding abrupt maneuvers. During the second lap the driver was asked to drive more aggressively. The scoring algorithm reduced the driver score for each detected event: depending on the severity level of the event (i.e., low, medium, high and extreme) the score was reduced 2, 4, 6 or 8 points respectively. Additionally, the score was increased by one point when no event is detected during 0.5 km of driving.

2) General Results: Table 2 presents the general results of the experiment, including the number of events and score for both the calm (C) and aggressive (A) laps. The different drivers are labeled from D1 to D10 and ordered by decreasing score obtained during the calm lap. We observe in the results that in all the cases, the number of detected events and the scores obtained are consistent with the type of lap (calm or aggressive). Moreover, since the scoring algorithm considers not only the number of events but the current weather information, we can observe in Table 2 that drivers having different number of events may obtain the same score. This is the case for drivers 6 and 7, who both obtained 69 points during the calm lap but with different weather conditions for the experiments (i.e., driver 6 with normal and driver 7 with rainy weather).

3) Location of Events: In order to obtain a global view of the detected events, we computed the event location distribution for the different laps (calm and aggressive) and the different types of events for all the drivers. Fig. 5 shows these distributions in a set of heatmaps. In particular, Figs. 5a and 5d show the location of events for the aggressive and calm laps respectively. We can observe in these figures that for calm drivers, event hotspots are located at very precise areas on the path, that can be considered as dangerous due to specific road topology (e.g., a sudden stop point after a pronounced slope; first maneuver on leaving from the parking garage). On the other hand, the event location distribution for the aggressive laps shows a more uniform distribution of events along the path with several hotspots at intersections (where the driver tends to brake, steer and accelerate aggressively). Figs. 5b, 5c, 5e and 5f show the location distribution of the different types of events (considering only the aggressive laps). In these figures we can observe that steering events are mainly detected at intersections, indicating a minimum of false positive events. In contrast, over-speed events are mostly located in low speed-limit streets and are less frequent in main avenues, where the speed limit is 70 km/h.

4) Comparison to Driver's Subjective Score: In order to study the significance of the scores obtained using SenseFleet, we asked the drivers to provide a subjective score for their laps. To the best of our knowledge, this is the first evaluation study that has considered the relation with subjective driver scores. In particular, the drivers were asked to score both their calm and aggressive laps using a scale of 1 to 5, with 1 being the highest risk class and 5 the lowest risk (safest) class.

For evaluation purposes, we compare the platform output with the drivers' subjective scores. To do this, we categorized the results obtained with SenseFleet into five classes. We considered five features: (ACC, BRK, STE, OVS, Score), representing the number of events of each type and the score value. We then clustered them using the k-means algorithm.

Fig. 6 illustrates the five clusters that were computed. Each point in the space represents a particular calm or aggressive lap (the number indicates the driver and letters a and c indicates aggressive or calm lap respectively). The 5-dimensional space was reduced to a 2-dimensional space by performing Principal Component Analysis (PCA).

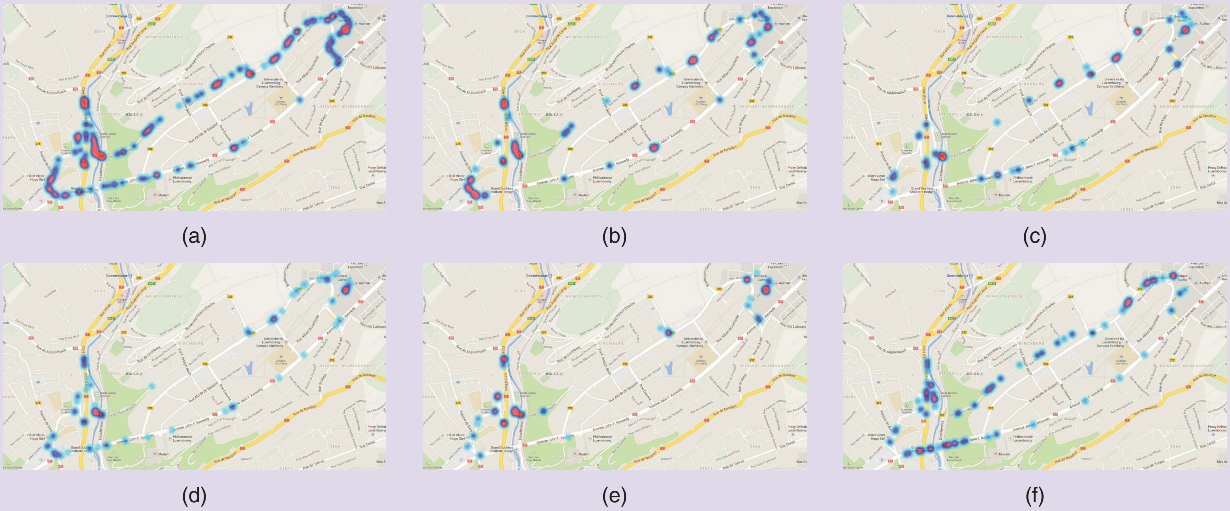

FIG 5 Event locations for the different drivers. (a) Event locations for the *aggressive* laps. (b) Acceleration event locations. (c) Braking event locations. (d) Event locations for the *calm* laps. (e) Steering event locations. (f) Overspeed event locations.

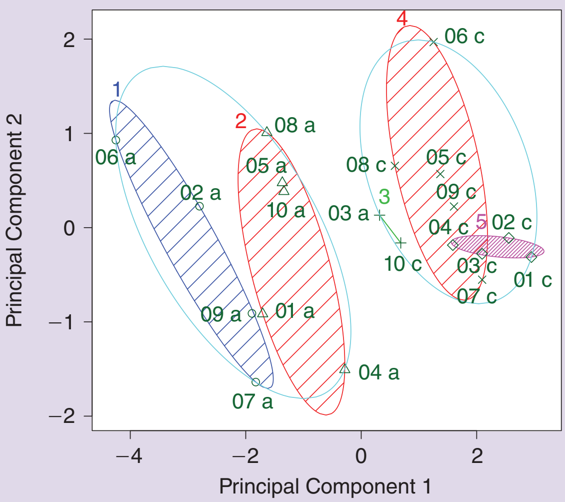

FIG 6 Lap clustering.

Table 3 indicates the two first principal components loadings. The first principal component (PC_1) is a linear combination of the number of events of different types (with loadings in the range −0.481 to −0.322) and the score with a loading of 0.481. High values of the first principal component represent a low number of events and a high score, while low values are related to a high number of events and a lower score (e.g., $PC_1 \geq 2 : n(events) \leq 12$, $PC_1 \leq -1.8 : n(events) \geq 35$).

The second principal component (PC_2) is predominantly influenced by the number of steering events (with a weight of 0.915), less importantly by over-speed and braking events with negative loadings. High values for the second principal component are related to laps with a high number of steering events. Low values of this component represent moderate driving ($PC_2 \in (-0.6, 0.6)$) or laps with a high number of over-speed events ($PC_2 \leq -1$).

Table 3. PCA loadings of the first 2 components.

Prin. Comp.	ACC	BRK	STE	OVS	Score
PC_1	−0.469	−0.481	−0.322	−0.462	0.481
PC_2	0.0	−0.130	0.915	−0.293	0.241

The aggressive and calm laps are clearly separable into two larger over-clusters (depicted in light blue in Fig. 6), with the single exception of the third driver's aggressive lap (lap 03*a*). This particular aggressive lap has been clustered as a calm lap due to a lower number of events and better weather conditions than in the rest of the aggressive laps.

The distance between the calm and the aggressive clusters shows that SenseFleet allows a clear distinction to be made between both driver behaviors. In terms of the driver score, we can observe a linear separation between calm and aggressive laps at a score value of 47.75, which is close to half the scoring range.

Table 4 indicates the centers and sizes of the clusters, ordered by increasing score (i.e., a lower cluster index corresponds to a more aggressive driving behavior). We can observe that the score values are well distributed over the scoring range. The clusters are similarly sized, except for cluster 3, which only contains two laps that are very close to the edge of the calm cluster.

In Table 5, we show a comparison between the drivers' subjective scores and the computed cluster for each

Table 4. Cluster sizes and centers.

Cluster	Size	ACC	BRK	STE	OVS	Score
1	4	13.5	11.25	8.75	12.25	0
2	5	12	7	7.8	9.4	28.6
3	2	6	4.5	6.5	6	59
4	5	4.8	1.4	7.2	2.4	67.8
5	4	2.25	1	4	2.5	88.75

Table 5. Subjective scores vs. clustering.

D	Calm		Aggr.	
	Subj.	Clust.	Subj.	Clust.
01	4	5	1	2
02	5	5	3	1
03	5	5	3	3
04	5	5	2	2
05	4	4	2	2
06	3	4	2	1
07	5	4	3	1
08	3	4	2	2
09	4	4	1	1
10	3	3	2	2

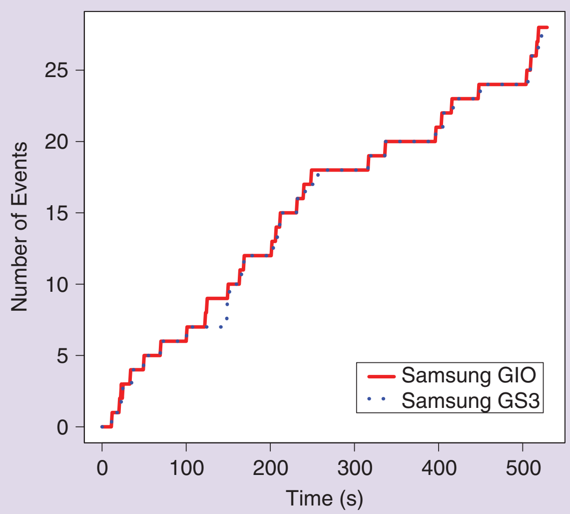

FIG 7 Event detection performance for different devices.

individual lap. Recall that each driver provided two subjective scores (one for their calm and another for the aggressive lap). For 60% of the laps, the subjective and computed cluster values are identical. This was mainly observed for the least aggressive drivers. In our experiment, there were only two laps that received a high risk subjective score (risk factor 1), whereas our clustering indicated four such laps. If a match between subjective and SenseFleet scores is assumed for a distance of ±1 between the categories, we achieve 90% matching (18 over 20 laps have equivalent subjective and SenseFleet score), which denotes a good confidence level.

C. Effect of Different Smartphones

As has been previously mentioned, different devices embed different sensors and chipset brands, providing different sampling rates, ranges and resolutions for motion sensors. In order to validate the event detection accuracy of SenseFleet on different platforms, we performed another experiment. In this experiment, we installed two different smartphones simultaneously in a single car: a Samsung Galaxy Gio (S5660) and a Samsung Galaxy S3 (I9300) using two different car-holders. Note that these two devices have very different performance and capacities in terms of sensors sampling rate and resolution. The I9300 smartphone has a ST Microelectronics LSM330DLC [24] while the S5660 has a Bosch BMA220 [25] 3-axis accelerometer. The experiment consisted of a calibration process (to define the fuzzy sets) and a complete lap over the circuit used in the first experiment. Fig. 7 shows the results of this experiment in terms of event detection performance. We observe that event detection follows almost the same pattern regardless of the device.

VI. Discussion

SenseFleet enables event detection over different vehicles and mobile devices. This is provided by the collection of motion sensor traces and statistical analysis during a calibration phase. However, the way the calibration phase is performed conditions the event detection phase and consequently driver scoring. In other words, a non-representative calibration phase (e.g., the driver does not drive in the usual way), can prevent the event detector from triggering events (if the calibration phase took place during a very aggressive driving pattern) or, in contrast, it can overestimate the number of detected events (when the calibration phase was done in a very calm pattern).

To give an example, in Fig. 8 we illustrate the cumulative distribution of $\sigma(j)$ during the calibration phase and the aggressive lap for two drivers (D_A and D_B). We can observe that for D_A, both distributions are much closer than in the case of D_B. To provide a comparison metric, we computed a Kolmogorov-Smirnov test between the two distributions (calibration and aggressive lap) obtained for D_A and D_B. The resulting statistic D for D_A and D_B were 0.13 and 0.46

respectively, indicating that D_A had much more similar values of $\sigma(j)$ during the calibration phase and the aggressive lap than D_B.

In order to mitigate this effect, several solutions may be applied. First, instead of considering a fixed number of calibration samples, the system may dynamically decide when to stop the calibration phase given a certain condition. As an example, the system may observe GPS metrics during the calibration phase (like speed and bearing variation) to decide whether the calibration samples are representative or not. A second solution would be to consider a continuous calibration, i.e., instead of performing an initial phase, the system may periodically analyze the distribution of sensing input variables and adapt the event detection fuzzy system. To do this, Q-Digest [26] would be used to compute a fast quantile approximation. Finally, a potential solution could also be to consider global parameters from the event detection phase that are not individually obtained by any single car but by a remote system that can consider a much larger set of sensing data from different phones, vehicles and drivers and perform statistical analysis to compute sets of parameters for the event detection algorithm that will be then remotely enforced in every single device.

VII. Conclusion and Perspectives

In this paper, we have described SenseFleet, a new mobile device and vehicle independent driver profiling and scoring application. SenseFleet is able to detect acceleration, braking, steering and over-speeding events by fusing motion sensors and GPS data. In order to perform event detection for multiple devices and vehicles, we used a calibration phase that allows adapting the fuzzy set limits for the event detection algorithm. In particular for over-speeding events, we use a web service to obtain the speed limits for the different roads along the path. Moreover, in contrast to existing solutions, we propose a scoring algorithm that not only relies on the number of events but also considers context information such as the current weather conditions and the time of day. In order to validate our platform, we used the application under different conditions (i.e., different drivers, devices, cars) and we performed a controlled evaluation study using a single car and path and different drivers driving in both calm and aggressive patterns. The experimental results show that SenseFleet is able to accurately detect risky driving events and distinguish between aggressive and calm drivers. The scoring results were compared to a subjective risk metric provided by each individual driver for their experiments. The results show that SenseFleet scores are equivalent to individual drivers' feedback in around 90% of the cases within ±1 neighboring driver clusters. For future work, we intend to analyze the impact of calibration on the event detection. As stated in Section VI, some potential solutions for obtaining representative calibration samples have been investigated and need to be studied in a larger experimental testbed. Moreover, we aim intend to evaluate different approaches for the fuzzy sets definition, considering other types of membership functions and statistical analysis over calibration data.

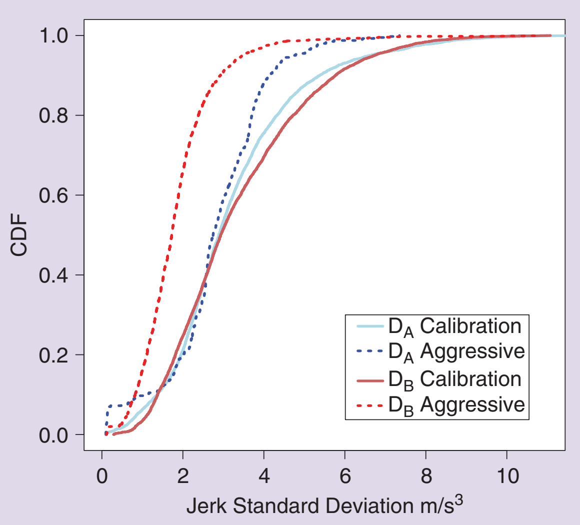

FIG 8 $\sigma(j)$ CDF for calibration and aggressive phases.

About the Authors

German Castignani is research associate at the Interdisciplinary Centre for Security, Reliability and Trust (SnT) of the University of Luxembourg (UL). He received his Computer Sciences Engineer degree at the University of Buenos Aires (FIUBA, Argentina) and a PhD in Computer Sciences at Institut Mines-Telecom, Telecom Bretagne (Rennes, France), under the direction of Prof. Xavier Lagrange and the supervision of Dr. Nicolas Montavont. He has more than 15 scientific publications in conference proceedings and international peer-reviewed journals. His research interests include Vehicular and Wireless Networks, Mobility Management, Intelligent Transportation Systems (ITS) and Advanced Driving Assistance Systems. Currently, Dr. Castignani works on several research projects at the VehicularLab, including handover management in IEEE 802.11 V2I communications, mobile sensing and crowdsourcing for driving assistance systems.

Thierry Derrmann is PhD student at the Interdisciplinary Centre for Security, Reliability and Trust (SnT) of the University of Luxembourg (UL). He received his Computer Sciences diploma (Dipl.-Inf.) at the Karlsruhe Institute of Technology (KIT, Germany) in 2011. During his diploma thesis he worked on the design of filtering algorithms using GPUs for applications in the field of mass spectrometry. His current research interests are related to data analysis, traffic optimization and mobility applications and stochastic modeling. His PhD project on multimodal mobility models is funded by the National Research Fund of Luxembourg (FNR).

Raphaël Frank is research scientist at the Interdisciplinary Center for Security, Reliability and Trust (SnT) and head of the VehicularLab. He is currently involved in several European and national research projects. His research interests include vehicular networks and mobile computing. He is a member of the Car 2 Car Communication Consortium and the IPv6 Forum Luxembourg. He is author and co-author of more than 20 scientific papers, published in peer-reviewed international journals and conferences. Dr. Frank received his Ph.D. in Computer Science from the University of Luxembourg in 2010. During his Ph.D. studies he was a visiting scholar at the University of California in Los Angeles (UCLA) where he conducted research on data routing protocols for vehicular networks. In 2006, he received his Master Degree in Cryptography and Network Security from the University Joseph Fourier in Grenoble, France.

Thomas Engel is professor for Computer Networks and Telecommunications at the University of Luxembourg. From 1987 to 1995 he studied Physics and Computer Science at the University of Saarbruecken, Germany, where he graduated in 1992 and received the title Dr. rer. nat. in 1996. Between 1996 and 2003, as joint founder, he was a member of the board of directors and vice-director of the Fraunhofer-guided Institute for Telematics e.V. in Trier, Germany, co-responsible for the scientific orientation and development of the institute. Since 2002, he has taught and conducted research as a professor at the IST/University of Luxembourg. His SECAN-Lab team deals with performance, privacy and identity handling in Next Generation Networks. As a member of the European Security Research Advisory Board (ESRAB) of the European Commission in Brussels, he advised the Commission on the structure, content and implementation of the FP7 Security Research Programme. He is a member of the Information and Communication Security Panel ICS of NATO and Civil High-Level Expert for Electronic Communications (representing Europe) of NATO CEP/CCPC. He is speaker of the regional group Trier/Luxemburg of the German Society for Computer Science (GI). Since 2009 he has been chairman of the National IPv6 Council Luxembourg. Since 2009 he has also been the Deputy Director of SnT, the Interdisciplinary Centre for Security Reliability and Trust within the University of Luxembourg.

References

[1] Ingenie Services Limited. (2012). Car insurance for young drivers. [Online]. Available: http://www.ingenie.com

[2] Towergate Underwriting Group Limited. (2012). Fair Pay a radical new approach to motor insurance. [Online]. Available: http://www.fairpay-insurance.co.uk

[3] Aviva PLC. (2013). Aviva RateMyDrive. [Online]. Available: http: //www.aviva.co.uk/drive/

[4] State Farm Mutual Automobile Insurance Company. (2013). State Farm driver feedback. [Online]. Available: http://www.statefarm.ca/about/mobile/

[5] AXA Belgium. (2013). AXA drive. [Online]. Available: https: //www.axa.be/ab/FR/apps/Pages/Applications-mobiles.aspx

[6] Greenroad. (2013). Introducing the revolutionary GreenRoad Smartphone Edition. [Online]. Available: http://greenroad.com/uk/tour/smartphone/

[7] G. Castignani, R. Frank, and T. Engel, "Driver behavior profiling using smartphones," in *Proc. 16th Int. IEEE Conf. Intelligent Transportation Systems*, 2013, pp. 552–557.

[8] G. Castignani, R. Frank, and T. Engel, "An evaluation study of driver profiling fuzzy algorithms using smartphones," in *Proc. IEEE Workshop Vehicular Communications Applications*, 2013.

[9] *Usage Based Insurance Global Study*, PTOLEMUS Consulting Group, Brussels, Belgium, 2012.

[10] Deloitte U.K. (2013). Deloitte research highlights consumers attitudes to telematics. [Online]. Available: http://www.deloitte.com

[11] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, June 2012, pp. 234–239.

[12] D. Johnson and M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proc. 14th Int. IEEE Conf. Intelligent Transportation Systems*, 2011, pp. 1609–1615.

[13] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles. (2012). Driving behavior analysis with smartphones: Insights from a controlled field study. in *Proc. 11th Int. Conf. Mobile Ubiquitous Multimedia*. New York, pp. 36:1–36:8. [Online]. Available: http://doi.acm.org/10.1145/2406367.2406412

[14] C.-W. You, M. Montes-de Oca, T. J. Bao, N. D. Lane, H. Lu, G. Cardone, L. Torresani, and A. T. Campbell. (2012). CarSafe: A driver safety app that detects dangerous driving behavior using dual-cameras on smartphones. in *Proc. ACM Conf. Ubiquitous Computing*. New York, p. 671672. [Online]. Available: http://doi.acm.org/10.1145/2370216.2370360

[15] National Highway Traffic Safety Administration, U.S. Dept. Transport., "The 100-car naturalistic driving study—Phase II results of the 100-car field experiment," Tech. Rep. DOT HS 810 593, 2006.

[16] R. Araujo, A. Igreja, R. de Castro, and R. Araujo, "Driving coach: A smartphone application to evaluate driving efficient patterns," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, June 2012, pp. 1005–1010.

[17] P. Mohan, V. N. Padmanabhan, and R. Ramjee. (2008). Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. in *Proc. 6th ACM Conf. Embedded Network Sensor Systems*. New York, pp. 323–336. [Online]. Available: http://doi.acm.org/10.1145/1460412.1460444

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

[19] J. Almazan, L. Bergasa, J. Yebes, R. Barea, and R. Arroyo, "Full auto-calibration of a smartphone on board a vehicle using IMU and GPS embedded sensors," in *Proc. IEEE Intelligent Vehicles Symp. (IV)*, June 2013, pp. 1374–1380.

[20] J. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.

[21] P. Cingolani and J. Alcala-Fdez, "jFuzzyLogic: A robust and flexible Fuzzy-Logic inference system language implementation," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 2012, pp. 1–8.

[22] OpenWeatherMap. (2013). OpenWeatherMap API free weather data API for developers. [Online]. Available: http://openweathermap.org/API

[23] D. L. Massie, "Analysis of accident rates by age, gender, and time of day based on the 1990 Nationwide Personal Transportation Survey," Transport. Res. Inst., Univ. Michigan, Ann Arbor, MI, Tech. Rep. UMTRI-93-7, 1993.

[24] S. Microelectronics, "iNEMO inertial module: 3D accelerometer and 3D gyroscope (LSM330DLC)," Data sheet Doc ID 022162 Rev 2, 2012.

[25] Bosch Sensortec, "Digital, triaxial acceleration sensor (BMA220)," Data sheet BST-BMA220-DS003-07, 2011.

[26] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. (2004). Medians and beyond: New aggregation techniques for sensor networks. in *Proc. 2nd Int. Conf. Embedded Networked Sensor Systems*. New York, pp. 239–249. [Online]. Available: http://doi.acm.org/10.1145/1031495.1031524

ITS