Application for the Detection of Dangerous Driving and an Associated Gamification Framework

Keshav Bahadoor and Patrick Hosein
Department of Computer Science
The University of the West Indies
St. Augustine, Trinidad
Email: {keshav.bahadoor@gmail.com, patrick.hosein@sta.uwi.edu}

***Abstract*—In today's highly technology-based world, wireless devices such as smartphones, are being utilized for solving daily problems and in making our daily lives more efficient. Although smartphones are sometimes to be blamed for vehicular accidents, they can also be used for helping to avoid accidents. In this paper we consider one such possible solution (called Project Drive), in which a smartphone based application is used to bridge the gap between negative driving detection and user motivation for safer driving behavior. Project Drive leverages a hosted web service to support the Android based application. Weather data is integrated with a modular negative driving detection component to create a point based system where positive driving is rewarded. Project Drive employs user motivation and retention strategies, such as gamification and social networking, as a means to promote safe driving. Users gain badges based on positive driving events that are seen by their contacts via the in-application social feed. Points of interest, including positive driving event areas, are visible via an interactive map. A small scale prototype was developed and used for testing of the associated algorithms as well as the social aspects of the framework.**

***Index Terms*—APP, sensor networks, gamification, collision avoidance, driving monitor**

I. Introduction

Vehicular accidents and accident prevention continue to be a major issue in many countries today especially with the increasing affordability of cars in developing countries. To address the problem of unsafe or negative driving behavior, countries typically have laws, penalties and some form of a vehicle driving education system. In spite of this, drivers today are not well motivated to express safe or positive driving behaviors.

For example, on the island of Trinidad and Tobago, there have been over 200,000 collisions for the period 2007 to 2012, with 55% being attributed to speeding, and 30% to driving under the influence of alcohol [1]. It is currently estimated that 16.7 vehicular accidents occur per 100,000 people in Trinidad and Tobago [2]. This, in conjunction with the increase in the number of registered vehicles, has highlighted road and driving safety as a major concern. We believe that mobile or smartphone technology can be utilized in addressing negative driving behavior.

Mobile smartphone technology is constantly advancing, with new devices being released each year to a growing consumer market [3]. The rising penetration rate of mobile devices, coupled with the day to day integration with our daily lives has changed the way mobile applications are created, ensuring better compatibility with users [4], [5]. Mobile applications, or apps, are being designed in innovative ways that advocate the engagement, motivation and retention of users [6].

The majority of smartphones on the market today are equipped with Global Positioning System (GPS) and sensor technologies. Mobile applications have access to data produced from on-board accelerometers, gyroscopes, and other sensor types which can differ by manufacturer or model [7]. As previous researchers explored, the use of mobile sensors as well as GPS data can be used to identify negative driving patterns [8]–[10]. Currently there are no apps that can be found on vendor app stores, such as Google Play [11] and Apple App Store [12], which implement such solutions using the mobile device as the main component. However, there exists some mobile applications that gauges the driving behavior of users by utilizing on-board diagnostics devices [13], [14]. Our objective is a simple-to-use application that does not require any add-on devices and so can be used by anyone at any time and in any vehicle as long as they have a supported smartphone.

On-board diagnostic devices, commonly referred to as OBD-II devices, are available for consumer purchase, and can be connected from the vehicle to the mobile device via Bluetooth. They are not popular as mainstream social based applications and they are somewhat inconvenient to use. This inconvenience, through extra tasks required, during the use of a product can be referred to as user intrusion. To our knowledge, no mobile application exists on popular mobile app stores that provides a non-intrusive approach to detecting negative driving behavior while at the same time engaging users to actively develop better driving habits.

In this paper, we present a mobile application called Project Drive that bridges the gap between negative driving behavior detection and user engagement and motivation. The application has a modular design with an interchangeable driving detection component. We focus on usability and gamification elements with an aim to provide a unique social experience to the user. With an application designed to measure driving behavior, we also take into account the potential to use the application for entertainment value by competing for negative driving behaviors (e.g., competing for the highest speeds).

While driving, the current weather situation can impact on the ability, performance and visibility of the driver [15]. Correspondingly, weather conditions will have an impact on any device that measures driving behavior. To address this, we integrate weather data, including rain level, wind level and weather rating, into the process of negative driving identification. We also focus on user motivation and retention via the use of social network integration and features together with gamification techniques. While it is important to identify negative driving among users, it is also important to motivate and promote safe driving behaviors.

In the following section, we present a review of previous work done in the field of detecting negative driving using smart-phone technology. Next, we provide an overview of our mobile application and system, detailing the methods used to detect negative driving along with social and gamification techniques used. We then present results of a prototype implementation. Finally we provide future work and possible directions for improvement.

II. Related Work

Although research in the area of applying user motivation and retention to prevent negative driving behavior using smartphone technology is limited, there has recently been increasing interest in the area of detecting negative driving behavior itself. Detecting negative driving behavior can be accomplished using a smartphone device in singularity, a smartphone device coupled with a OBD-II interface device, or a smartphone device coupled with a server for data processing activities. Apart from coupling with OBD-II interface devices, most prior work measured driving behavior by using on-board sensor and GPS systems on smartphone devices.

In their paper, Dai et al. [8] introduced the idea of detecting drunk driving using accelerometer sensor readings from a non-intrusive smartphone device. While GPS information was not utilized due to its unavailability at the time, Dai et al. succeeded in showcasing the concept. Building on this approach, future work integrated mobile sensors such as the gyroscope and the magnetic compass. Currently, there exist different types of mobile smartphone devices on the market that provide onboard sensors with varying sensitivity and range. With this consideration, work done by Pholprasit et al. [9] included a system with an easily customized algorithm based on available device sensors. Combing data from multiple sensors (data fusion) is also quite common, [9], [10]. Pholprasit et al. [9] observed that data generated from the accelerometer sensor produced the highest accuracy.

When collecting sensor data from devices, one must be aware of the fact that the orientation, or movement of the device can affect readings. To eliminate unwanted variables, Bergasa et al. [16] mounted the mobile device to the back of the rear-view mirror of the vehicle, while Johnson and Trivedi [10] positioned the device on the dashboard. A simpler approach was taken by Koh and Kang [17] and Chen et al. [18] which involved placing the device in some predetermined alignment inside the vehicle, such as the console box between the seats. In contrast, Dai et al. [8] and Pholprasit et al. [9] allowed the arbitrary placement of the device in the car (as we assume in our work). This approach can be seen as less intrusive to the user, as the user need not worry about device placement.

Apart from smartphone based sensors, data can also be collected using the on-board diagnostics systems that are available on most vehicles today. This involves the purchase of an external device, that connects to the OBD-II port of the vehicle, that then acts as an interface between the mobile application on the smartphone and the computer system of the vehicle. Using Bluetooth and an appropriate OBD-II device, Meseguer et al. [19] collected data to train a neural network for negative driving detection. While shown to be accurate, this approach introduces a level of intrusion to the user due to the purchase requirement of a OBD-II device.

After data is collected, the system processes data to detect negative driving behavior. This can be implemented on the mobile device itself, or processed on the server side. Negative driving behaviors can be identified using algorithmic based detection, statistical anomaly detection, or data pattern matching. To support their neural network, Meseguer et al. [19] used the mean and standard deviation of speed, acceleration and RPM of the vehicle. As a form of pattern matching, Johnson and Trivedi [10] used Dynamic Time Warping against template data, which was classified into template events using the K-nearest means statistical algorithm. Chaovalit et al. [20] used an alternative approach by using Symbolic Aggregate Approximations. Machine learning was used by Chen et al. [18] which involved training a multi-class classifier model using support vector machines. This system was trained for over six months and performed quite well. Note that the aforementioned work did not include certain variables that impact driving, such as weather and road conditions.

III. System Design

In this section we present an overview and then describe each module of our design.

A. System Overview

Project Drive is comprised of a mobile application supported by server side storage and processing. The mobile app detects when the user is in a moving vehicle. After detection, data is captured and stored. Data is then synced to the server at appropriate times. At the server, all obtained data is processed by a driving detection component. The user is then notified of a positive driving score as well as any reward based content.

B. Architecture Design

The prototype application was developed for Android based smartphones. The application communicates with an Application Programming Interface (API), hosted on a Linux virtual machine. The API was built using the Python Flask framework which facilitates the capture and processing of data as well as user management functions. A PostgreSQL database is used for data storage. The architecture is provided in Figure 1.

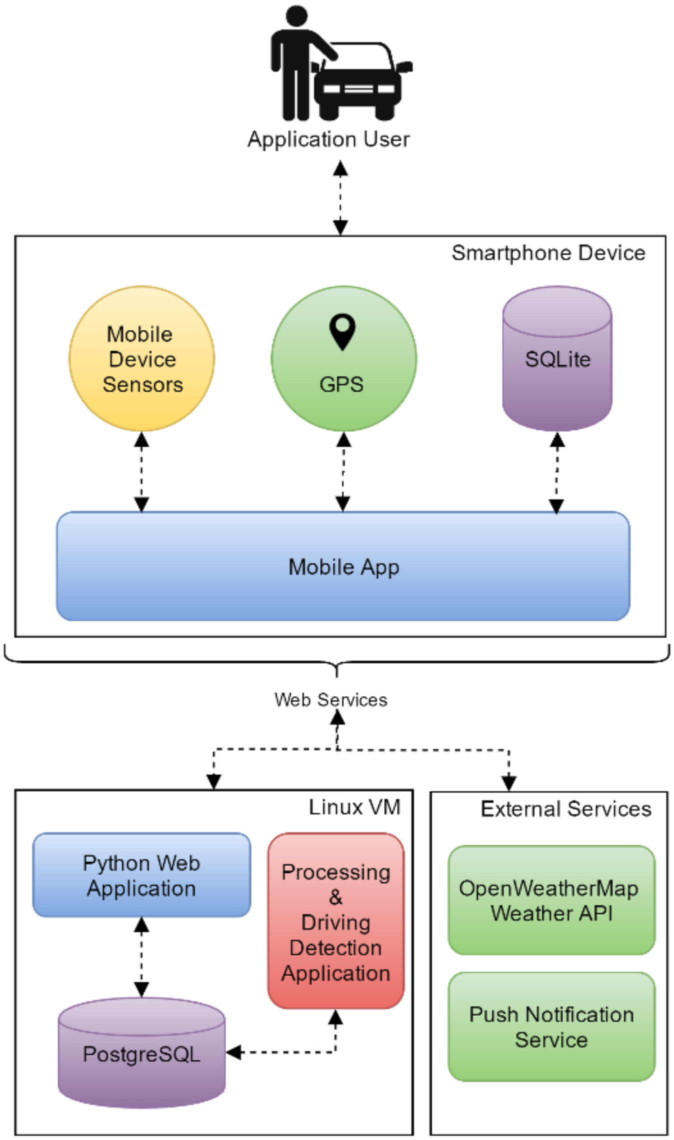

Fig. 1. System Architecture

C. Mobile Application

The mobile application can be described as two parts. The first module provides interaction with the user and can be referred to as the user experience layer. The second module is responsible for data capture, caching and decision making. This part can be referred to as the background service layer.

1) User Experience Layer: The User Experience Layer is designed with familiarity in mind. This layer is presented as several views. It consists of a news feed, a map view, and a user profile view. The news feed showcases activities and achievements of the users contacts. The map view gives the user information about his surroundings in relation to positive driving habits. The user profile view exhibits reward content for positive driving. The User Experience Layer is further described in Section III D — User Motivation and Retention.

2) Background Service Layer: The background service layer is developed as a separate Android Module Library called DrivingEventLib. This library can be used in different application contexts. DrivingEventLib consists of a continuous background Android Service, namely DriveService, that performs most of the work in a discrete system thread. By executing as a separate background process, DriveService ensures that the driving behavior of the user will be monitored even if the application is not opened.

At any point in time, the application can be in one of two states: stationary or active. The state of the application is determined by a minimum threshold speed value. The speed of the device is obtained using captured GPS data. This data is gathered using Google Services Fused Location Provider API [21]. This service provides bundled speed data as determined by the GPS provider. As the minimum threshold speed is surpassed, the service switches the application into an active state.

In its active state, the application prepares and capture GPS data, accelerometer data, as well as poll the current weather data. On smartphone devices, the time in-between GPS data updates as well as the level of accuracy can be varied. In this state, both the accuracy and the rate of update of GPS data are increased. It was found that a 1-2 second interval delivered the best readings in terms of speed, accuracy and battery longevity.

Based on the work done by Pholprasit et al. [9], we decided to capture on-board accelerometer sensor values due to greater accuracy. DriveService registers the sensor listener when in its active state. Contrarily, the sensor is unregistered in the stationary state of the application. DrivingEventLib handles preprocessing tasks before accelerometer data is stored. A combination of high and low pass filters is used to remove the gravitational acceleration component from values, as well as reduce sensor noise.

One of the main aspects of this project is the integration of weather data, as this affects driving behavior. DriveService gathers current weather data by polling the Open Weather Map public web service [22]. The current latitude and longitude coordinates are used in querying the weather.

GPS data, current weather and acceleration data are stored using a local SQLite instance. SQLite is a self-contained transactional structured query language type database available on Android. DrivingEventLib contains an implementation of the Android Sync Adapter [23]. The Sync Adapter facilitates the transfer of data from the local SQLite database to the server. Data transfers are batched by the Android operating system during internet connectivity. The local storage is cleared after data is synced to the server.

D. Data Processing and Detection Algorithm

Data is captured on the server side using a Python Flask [24] application and stored to a PostgreSQL database. Data processing is performed by a separate component. Processed data is stored on the database, and applicable results are posted to the client mobile application. The processes diagram in Fig. 2. gives further details. This Driving Detector component is

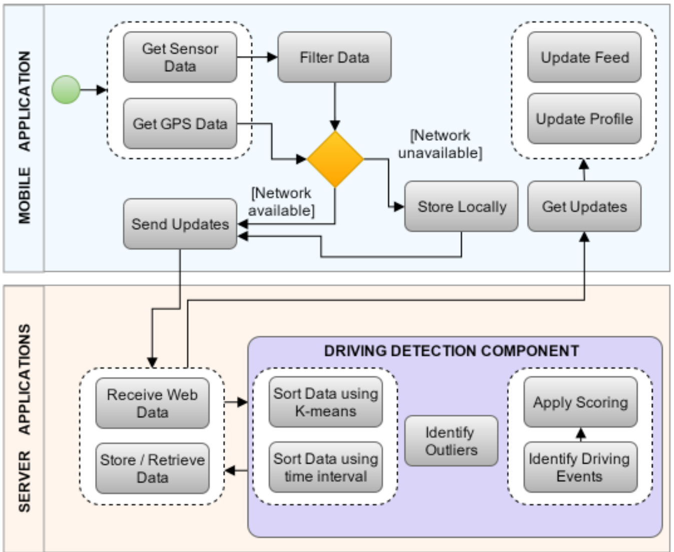

Fig. 2. Process diagram of the mobile application and server applications

designed to be easily switchable to enable further research and experimentation for Project Drive. Driving Detector is comprised of a Python application augmented with *scikit-learn* and *numpy* packages [25], [26].

Vehicle speed data is subjected to change during the day due to reasons such as peak hour traffic, low traffic periods, weekends, as well as different locations and roadways. Taking this into account we decided to partition the collected data by region and time interval.

We partitioned data in two phases. Phase one involved distributing data collected by region. This was done by using a K-means clustering approach on geographical latitude and longitude data. K-means is a simple unsupervised algorithm that is able to sort data into k groups or clusters [27]. A value of six was used for k as this represented the best distribution of regions.

Starting with this first partitioned data, phase two involves sorting the data by time period. Data was grouped into 4-hour segments. After the partition process, a single dataset contained the following for all users: GPS coordinates, vehicle speed and weather data for a particular region at a particular 4 hour time period. Acceleration data was treated separately from this process.

After data is partitioned, Driving Detector initiates a sub component that is responsible for negative driving detection. The Interquartile Range or IQR is used to determine outliers of both speed and acceleration data for each user. The server side application calculates this using the *numpy* Python package.

IQR is defined as the difference between the 75th and 25th (q_3) percentiles. Using this we can then calculate mild outliers using

$$|x - q_3| > 1.5IQR \rightarrow x \in O_m \quad (1)$$

and extreme outliers using

$$|x - q_3| > 3IQR \rightarrow x \in O_e \quad (2)$$

TABLE I
OPEN WEATHER MAP WEATHER CONDITIONS AND SCORING

Weather ID	Description	Negative Score
501	moderate rain	20
202	thunderstorm with heavy rain	40
232	thunderstorm with heavy drizzle	40
502	heavy intensity rain	40
503	very heavy rain	80
504	extreme rain	100
960	storm	200

where x is the current value being addressed, O_m is the set of mild outliers and O_e is the set of extreme outliers.

E. User Motivation and Retention

Project Drive uses social networking and gamification techniques to retain the attention of the user as well as encourage positive driving. These features focus on rewarding the user for positive driving rather than penalizing negative driving. Users are required to sign in using their Google Plus account before the application can be used.

Project Drive includes a scoring system. For each data set processed, the server application updates a user's score. Mild and extreme outliers, as well as captured weather conditions may impact the user's score. At the same time, users are prevented from having a negative score. Open Weather Map contains weather condition codes that are used when determining the score. An excerpt of this is shown in Table I.

For each region or location cluster and for each time interval, a score is calculated per user. Depending on the outliers found, the user can be awarded a reward in the form of an in-app badge. Badges are used to reward a particular positive driving event. The server application identifies cases where badges can be applied and then assigns badges to the user accordingly. At present we have implemented four types of badges described in Table II.

The application contains a social feed where user stories are posted. This user-tagged information provides details on recent positive driving goals and rewards attained by the user and his or her contacts. The left portion of Figure 3 shows a screen capture of the social feed feature, taken from a test user. The application contains an interactive map that allows users to view their in-app rewards and the location in which they were obtained. The right portion of Figure 3 shows a screen capture of the map feature taken from a test user.

TABLE II
IN-APP REWARD BADGES

Badge	Description
Shining Star	For the given time period and region cluster, the user did not have any mild or extreme outliers for both GPS, weather and acceleration
Super 80	For the given time period and region cluster, the user remained under the 80kmph speed limit
Smooth Driver	For the given time period and region cluster, acceleration data did not have any mild or extreme outliers
Weather Feather	For the given time period and region cluster, the user did not have any mild or extreme outliers for both GPS, weather and acceleration, with the applied weather conditions

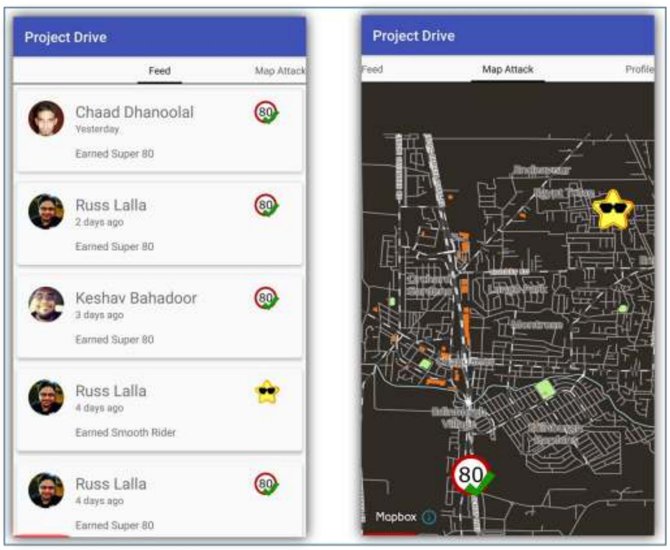

Fig. 3. GUI showcasing Social Feed and Interactive Map features

F. Application Optimization

Device battery life is a concern for any mobile application that actively uses location based services [28]. While early versions of Project Drive saw this impact, battery life has been improved by a number of optimizations. The rate of GPS updates per second is dynamically decreased when the user is not in a vehicle. Conversely it is increased when the user is driving. Similarly, the acceleration sensor is only registered by the device when in a moving vehicle. Other improvements includes using the Retrofit rest client framework [29] with a local cache for all external API calls.

G. Potential Add-Ons

The data collected from each user can potentially be used for other purposes once permission is provided by the user. For example, insurance companies may find such data useful in determining premiums for users. Parents may wish to reward their teen drivers based on their driving habits. Governments may want to address problems with certain roads.

IV. EXPERIMENTAL SETUP AND PRELIMINARY RESULTS

Project Drive was released to a small test group of 6 users for the initial prototype testing. Test users comprised of persons between the ages of 20 and 30, who own a vehicle and who drive on a regular basis. All test users owned medium to high-end Android smartphones from manufacturers such as Samsung, OnePlus and Google Nexus.

Prototype testing was carried out in two phases. The first phase lasted one week and involved users testing a basic version of Project Drive. This version excluded user motivation and retention related features such as badges, social feed, and the interactive map. The second phase also lasted a duration of one week and involved users testing the current version of Project Drive that includes all features.

Due to the small scale prototype testing, data collected was insufficient to gauge the effectiveness of integrating user motivation features after Phase Two. Instead, users were given a questionnaire to complete. This questionnaire aided the measurement of user motivation and retention features.

When asked to rate the importance of seeing the positive achievements of their contacts, 33% of users reported *somewhat important* while 67% of users reported *very important*. This metric had a scale from 1 to 5. Likewise, 100% of users answered *I was more compelled to see what badges my contacts received* when asked how did the social feed affect their usage. With respect to displaying positive driving events on the interactive map, 33% of users were neutral about the usefulness of this feature. A majority of 67%, however, regarded this feature as *somewhat useful*. To rate the importance of positive driving points, users were asked if they would be interested in trading earned points for a discount at their insurance company. All responses to this was *Yes*.

V. CONCLUSIONS AND FUTURE WORK

In this paper we presented the Project Drive system which attempts to bridge the gap between the detection of negative driving behavior and motivating the user to drive more cautiously. We showcased the importance and integration of current weather data in the process as well as ways user motivation and retention can be addressed. Project Drive is ongoing as we are currently reviewing the prototype application, assessing limitations and working on updates.

After significant updates, a larger testing group will be formed, including test users from other countries. This will be necessary to obtain sufficient samples to produce statistically significant conclusions. A user monitoring feature is also under development. This feature, called ”Driving Buddy”, allows users to monitor the driving behavior of friends and family members. Monitoring can only be done if the target user grants the requesting user permission to do so. An expanded social feed that includes features such as story sharing, commenting, up-voting, and interfacing with other social networks, including Facebook and Google Plus is also under development. Finally, we hope the statistics gathered will be used by Governmental agencies to address the growing problem of vehicular accidents.

REFERENCES

[1] A. Alive, "Statistics - arrive alive trinidad and tobago," http://www.arrivealivett.com/index.php?page=statistics, Jun 2015, (Accessed on 03/10/2016).

[2] W. H. Organization, "Gho — by category — road traffic deaths - data by country," http://apps.who.int/gho/data/node.main.A997, (Accessed on 03/10/2016).

[3] Statista, "Smartphone users worldwide 2014-2019 statistic," http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, 2016, (Accessed on 03/10/2016).

[4] L. Entis, "Life without smartphones is hard to imagine, according to nearly half of u.s. smartphone users," http://www.entrepreneur.com/article/248355, Jul 2015, (Accessed on 03/10/2016).

[5] M. Rajput, "Tracing the history and evolution of mobile apps," http://tech.co/mobile-app-history-evolution-2015-11, Nov 2015, (Accessed on 03/10/2016).

[6] G. Zichermann, "Designing gamification level 1 (basic) + certification — udemy," https://www.udemy.com/designing-gamification-level-1-certification/learn/, (Accessed on 03/10/2016).

[7] N. T., "Did you know how many different kinds of sensors go inside a smartphone?" http://www.phonearena.com/news/Did-you-know-how-many-different-kinds-of-sensors-go-inside-a-smartphone_id57885, Jul 2014, (Accessed on 03/10/2016).

[8] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2010 4th International Conference on-NO PERMISSIONS*. IEEE, 2010, pp. 1–8.

[9] T. Pholprasit, W. Choochaiwattana, and C. Saiprasert, "A comparison of driving behaviour prediction algorithm using multi-sensory data on a smartphone," in *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2015 16th IEEE/ACIS International Conference on*. IEEE, 2015, pp. 1–6.

[10] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1609–1615.

[11] "Android apps on google play," https://play.google.com/store/apps?hl=en, (Accessed on 03/13/2016).

[12] "App store downloads on itunes," https://itunes.apple.com/en/genre/ios/id36?mt=8, (Accessed on 03/13/2016).

[13] Dash, "Dash - drive smart - android apps on google play," https://play.google.com/store/apps/details?id=com.dashlabs.dash.android, (Accessed on 03/10/2016).

[14] J. Hullinger, "Can this app actually make you a better driver?" http://www.fastcompany.com/3026237/innovation-agents/can-this-app-actually-make-you-a-better-driver, Dec 2014, (Accessed on 03/10/2016).

[15] U. D. of Transportation Federal Highway Administration Road Weather Management Program, "How do weather events impact roads? - fhwa road weather management," http://www.ops.fhwa.dot.gov/weather/q1_roadimpact.htm, Dec 2015, (Accessed on 03/10/2016).

[16] L. M. Bergasa, D. Almería, J. Almazan, J. J. Yebes, and R. Arroyo, "Drivesafe: an app for alerting inattentive drivers and scoring driving behaviors," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 240–245.

[17] D.-W. Koh and H.-B. Kang, "Smartphone-based modeling and detection of aggressiveness reactions in senior drivers," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 12–17.

[18] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li, "D 3: Abnormal driving behaviors detection and identification using smartphone sensors," in *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*. IEEE, 2015, pp. 524–532.

[19] J. E. Meseguer, C. T. Calafate, J. C. Cano, and P. Manzoni, "Drivingstyles: A smartphone application to assess driver behavior," in *Computers and Communications (ISCC), 2013 IEEE Symposium on*. IEEE, 2013, pp. 000 535–000 540.

[20] P. Chaovalit, C. Saiprasert, and T. Pholprasit, "A method for driving event detection using sax on smartphone sensors," in *ITS Telecommunications (ITST), 2013 13th International Conference on*. IEEE, 2013, pp. 450–455.

[21] Google, "Making your app location-aware — android developers," https://developer.android.com/training/location/index.html, Jun 2016, (Accessed on 06/01/2016).

[22] I. OpenWeatherMap, "Openweathermap current weather and forecast," http://openweathermap.org/, Jun 2016, (Accessed on 06/01/2016).

[23] Google, "Transferring data using sync adapters — android developers," https://developer.android.com/training/sync-adapters/index.html, Jun 2016, (Accessed on 06/01/2016).

[24] A. Ronacher, "Welcome — flask (a python microframework)," http://flask.pocoo.org/, Jun 2016, (Accessed on 06/01/2016).

[25] scikit learn, "scikit-learn: machine learning in python scikit-learn 0.17.1 documentation," http://scikit-learn.org/stable/, (Accessed on 03/14/2016).

[26] "Numpy numpy," http://www.numpy.org/, (Accessed on 03/14/2016).

[27] M. Matteucci, "Clustering - k-means," http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html, (Accessed on 03/14/2016).

[28] R. Love, "Why does gps use more battery than any other antenna or sensor in a smartphone? - forbes," http://www.forbes.com/sites/quora/2013/08/06/why-does-gps-use-more-battery-than-any-other-antenna-or-sensor-in-a-smartphone/#7fc5ade65693, Aug 2013, (Accessed on 06/01/2016).

[29] Square, "Retrofit," http://square.github.io/retrofit/, Jun 2016, (Accessed on 06/01/2016).