Leveraging Smartphone Sensors to Detect Distracted Driving Activities

Kaoutar Ben Ahmed, Bharti Goel, Pratool Bharti, Sriram Chellappan and Mohammed Bouhorma

***Abstract*—In this paper, we explore the feasibility of leveraging the accelerometer and gyroscope sensors in modern smartphones to detect instances of distracting driving activities (e.g., calling, texting and reading while driving). To do so, we conducted an experiment with 16 subjects on a realistic driving simulator. As discussed later, the simulator is equipped with a realistic steering wheel, acceleration/ braking pedals, and a wide screen to visualize background vehicular traffic. It is also programmed to simulate multiple environmental conditions like day time, night time, fog and rain/ snow. Subjects were instructed to drive the simulator while performing a randomized sequence of activities that included texting, calling and reading from a phone while they were driving, during which the accelerometer and gyroscope in the phone were logging sensory data. By extracting features from this sensory data, we then implemented a machine learning technique based on Random Forests to detect distracted driving. Our technique achieves very good Precision, Recall and F-Measure across all environmental conditions we tested. We believe that our contributions in this paper can have significant impact for enhancing road safety.**

***Keywords*—smart sensing, intelligent transportation systems, distracted driving, machine learning, smartphones.**

I. INTRODUCTION

As of today, driving while simultaneously using a smartphone is one of the most significant dangers to road safety. It is reported that in 2014 alone, $3,129$ people were killed in distracted driving crashes in the US [1], and in many of them, usage of smartphones was identified as a major contributing factor. The impact of phone usage on reaction time is explained with reference to a phenomenon referred to as "inattentional blindness" or "perceptual blindness", well documented in the psychological literature, wherein a person who is focusing his or her attention on one particular task will fail to notice an unexpected stimulus (even if he or she looks at it) [2]. Five seconds is the average time a person's eyes are off the road while texting. When traveling at 55mph, that is enough time to cover the length of a football field blindfolded, which explains the severity of the problem. Naturally, there is renewed urgency in the US and across the globe to combat this problem, with several laws being enforced that either restrict, or totally ban the use of smartphones while driving. Unfortunately, even with restrictions/ bans, nothing prevents a person from reaching out to a phone while driving, which makes this issue complicated.

K. B. Ahmed, B. Goel, P. Bharti, S. Chellappan are with Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA. (email: kbenahmed@mail.usf.edu; bharti@mail.usf.edu; pratool@mail.usf.edu; sriramc@usf.edu). M. Bouhorma is with the Faculty of Sciences and Technologies, University Abdelmalek Essaadi, Tangier, Morocco (email:bouhorma@gmail.com). P. Bharti was a student at University of South Florida while working on this paper.

A. *A case for technology based intervention*

Recently, there are some earnest attempts at leveraging technology based solutions to combat the problem of smartphone use during driving. These are partly due to people accepting such solutions, as evidenced by a recent finding in a study sponsored by the American National Safety Council in 2016 [3]. In that study, more than $2,400$ drivers across the country were surveyed, and 55% of drivers said that they would accept any simple to use technological solution that will prevent distractions while driving.

Efforts in Industry: There are some innovative products developed by industry to cater to this need. AT&T and other service providers offer apps that when executed by a driver before driving will prevent notifications sent to the driver while driving [4]. Unfortunately, users have to proactively turn on the app before starting to drive that can be cumbersome. "Snapshot" from Progressive Insurance is a small device that plugs to a car, typically located beneath the steering column [5]. The device collects and sends driving information like miles driven, driving times, and braking patterns to a server for approximately one policy term. In exchange for this data, prudent drivers can receive discounts off their premiums. The "Metromile Pulse" is another device that collects driving stats and engine health among others, which can be used to save on insurance premiums [6]. While these products encourage safe driving, and could be used to detect more fine-grained driver activities, they are not designed to enable the detection of distracted driving activities as and when they happen. Furthermore, such devices are external, and are additional investments, hence limiting their permeability for a wide audience.

Efforts in Academia: In the academia also, there are interesting solutions in the space of enhancing driver safety and comfort with external devices (like smartphones). In [7], images from smartphone cameras attached on car windshields are processed for predicting signal times, while in [8], smartphone images of drivers are used to detect tired or distracted drivers for road safety. Other papers like [9]–[11], use inertial sensory data from smartphones or wearables placed inside cars to profile driver routes, that could have potential applications in path planning. Other works in the academia more closely related to the one in this paper are [12] and [13], where inertial sensors in smartphones are used to detect driving related activities like acceleration, steering, braking etc. In [14], a technique is proposed to only detect texting while driving by assessing the timing differences between key pad entries on

Fig. 1: Four Participants in the Driving Simulator in Different Environmental Conditions (Day time, Night time, Fog and Rain/Snow)

the phone. More sophisticated techniques like a) using external antennas [15]; b) using acoustic ranging techniques between a phone and car speakers [16]; and c) computing centripetal acceleration from smart-phones sensors to assess vehicular dynamics [17] are also attempts to detect phone use while driving. However, none of these techniques aim to detect the type of distracted driving activity among texting, calling and reading on the phone, which we accomplish in this paper.

B. Our Contributions

In this paper, our problem is to leverage the in-built inertial sensing functionalities of modern smartphones to detect instances of distracted driving. Unfortunately, getting real data for such a purpose is dangerous as it puts subjects at risk. To overcome this, we utilize a state of the art driving simulator, called the CAREN platform [18] which is equipped with a realistic steering wheel, driver seat with belts, acceleration/ braking pedals, and a wide screen to visualize background vehicular traffic. The system also enables subjects feel accelerations/ decelerations of the platform when they simulate driving. It is also programmed to simulate multiple environmental conditions like day time, night time, fog and rain/ snow.

We conducted an experiment with 16 adult subjects driving in the CAREN simulator for multiple environmental conditions. Our data (as we show later) demonstrates that the accelerometer and gyroscope sensors in smartphones show subtle changes when subjects attempt to text, call or read from a phone while holding it and driving, that are also distinct from safe driving patterns (i.e., driving without holding the phone). Utilizing this insight, we design a machine learning technique to detect instances of distracted driving from processing the accelerometer and gyroscope data in the phone. Specifically, we first derive a large number of intuitive features from the sensor readings, from which we employ Filter-based [19] techniques to identify a subset of six features that provide a high degree of discriminatory power. Then, we design a Random Forest based machine learning classification algorithm (details in Section IV-C) to detect instances of distracted driving in real-time. Our technique achieves an overall accuracy of more than 87.94%, 80.09% and 70% respectively for same-user cross-validation, cross-user cross-validation and cross-user leave-one-out evaluation strategies in determining the specific type of distracted driving activity (i.e., texting, calling and reading) averaged across four background environmental conditions. When we reduce the problem to merely a binary classification one in terms of detecting whether or not a subject is using the phone while driving, the accuracy reaches 96% even in the more stringent cross validation with leave one out evaluation strategy, hence demonstrating the effectiveness and practicality of our proposed technique.

II. BACKGROUND ON CAREN PLATFORM

We now present details on the driving simulator used to collect data to address our problem of detecting distracting driver activities. The driving simulator used for our study is the Computer Assisted Rehabilitation Environment (CAREN) system [18], which was developed by the Center for Assistive, Rehabilitation and Robotics Technologies (CARRT) at University of South Florida, in collaboration with Motek Medical [20]. This driving simulator was originally developed to help train individuals with spinal cord injuries to learn how to drive in a safe and controlled environment. The simulator includes a six degree of freedom motion base, an optical motion capture system, a sound system, and a 180-degree projection screen.

The two drive by wire (DBW) controls a lever device to control the gas and brake components, and a small wheel device to control steering. In doing so, an electrical signal is sent to a Phidget board, which interfaces with the CAREN system. The motion platform enables the driver to feel accelerations and decelerations during driving while being in the platform's work-space. It also displays the driver's speed, surrounding traffic, and number of collisions if any. Evaluation of the system in [18] identified that subjects felt comfortable, and they also stated that the environment felt real. The sensitivities of accelerometer and braking were also identified as realistic. Note that driving scenes were created using Google SketchUp 3D modeling software. After a scene was exported, it was imported into D-Flow to be used in the driving applications. For our experiments, multiple environmental conditions were simulated including day time, night time, fog and rain/ snow in a highway like environment that comprised of a four-lane highway with speed limits of up to 70mph, while also simulating moving background traffic. Fig. 1 shows four participants in our experiments driving in each of the four environments.

III. EXPERIMENTAL DESIGN

In this section, we describe in detail the procedures of our experiment using the CAREN system for data collection. The experimental procedures were approved by the Institutional Review Board (IRB) of the University of South Florida. The authors performed experimental procedures in accordance with the approved guidelines.

A. Smartphone for the Experiments

We used in our experiment one Samsung Galaxy S4 smartphone, of dimensions $136.6 \times 69.8 \times 7.9$mm. The smartphone has Android 4.4 (KitKat) OS, and is embedded with multiple sensors including an accelerometer and a gyroscope. The LIS344ALH accelerometer sensor in the Samsung S4 phone [21] can consistently sample up to around 250Hz, and the sampling rate is programmable. It records the smartphone movement state based on three axes x, y and z in m/sec^2. The gyroscope measures rate of rotation of the smartphone with respect to x, y and z axes in rad/sec, with similar sampling rates. We designed and implemented an app on the phone that allowed the recording and storing of accelerometer and gyroscope readings at a sampling rate of 200Hz for experiments in this paper, along with time.

B. Subjects

Sixteen adults were recruited for this study. They ranged from ages 20 to 28. Ten participants were male and six were female. All subjects had a valid US driver's license, and at least two years driving experience, and reported that they drive an average of at-least ten hours per week. All subjects claimed that they were comfortable to text, make phone calls, and read from the phone while driving. After signing the IRB forms, and having all their questions answered, participants were familiarized with the driving simulator using a standardized 12 minute adaptation sequence. All participants indicated that they were comfortable to participate in the experiments, and also indicated they were alert and ready to drive (i.e. they were not tired or stressed or sleep-deprived).

C. Procedure

After the practice session, participants were asked to sit in the driver seat of the car simulator, and were given the smartphone. The participant placed the phone in a horizontal platform next to the steering wheel from where it could be easily picked up. Our app to record accelerometer readings, gyroscope readings, and time stamps was installed on the phone and turned on. Subjects were instructed to drive the simulator normally and respect the speed limits displayed on screen, perform proper lane changes, and perform proper overtaking of other vehicles on the road. Each subject drove for a total of around 11 minutes in the simulator. In the first two minutes the subject did not use the phone at all, so that he/ she can familiarize with the simulator. Around the start of the second minute, and for most of the next 9 minutes, subjects were either called on the phone to engage in a voice conversation, or were made to engage in a texting session, or were sent an article to read while driving. Each session was around two minutes long, and there was a one minute gap between each event to let the subject put the phone down and stabilize their driving in the simulator.

One of the co-authors was the other party in the communication with each subject during these sessions. Note that the sequence of texting, calling and reading sessions were randomized for every subject. All of the sixteen subjects participated in the above experiment for all four environmental conditions, namely, day time, night time, fog and rain/ snow. At the conclusion of each experiment (around the end of the 11^{th} minute), the subject gradually decelerated and exited the simulator. We point out that throughout the entire duration, each subject drove the simulator constantly and never stopped. The text messages, voice conversations, and reading content were the same for all participants, and the smartphone recorded sensor data continuously for all subjects while driving. Recall that Figs. 1 (a) to (d) shows four participants in our experiments driving in each of the four environments.

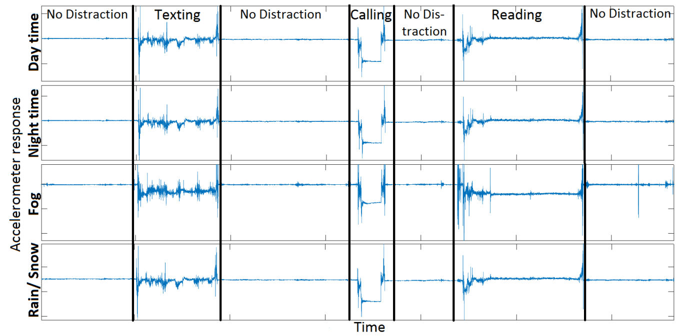


Fig. 2: Accelerometer (x-axis) Readings for a Single Subject for Multiple Events in Four Environmental Conditions

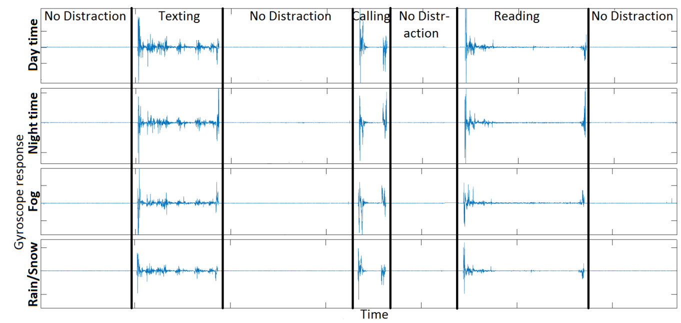


Fig. 3: Gyroscope (x-axis) Readings for a Single Subject for Multiple Events in Four Environmental Conditions

D. Data Tagging

In order to tag the data, we compared the times when an activity was initiated and terminated on the phone, with the times when the sensor readings were recorded. From this, we were able to tag the various activities with the sensor readings for subsequent model development.

IV. Our Methodology for Classifying Distracted Driving

In this section, we present in detail our methodology to detect instances of distracted driving. First, we present the rationale for our technique, followed by the process of feature extraction, and finally algorithm design.

A. Rationale of our Technique

Figs. 2 and 3 highlight the feasibility of classifying distracted driving from smartphone sensory data technique. We show for a single subject, how the readings of the accelerometer and gyroscope (both in x-axis only) collected from the smartphone are different when the subject drives without distraction, and when the subject texts, calls or reads from the phone while driving. We can also see that there are subtle differences in sensor readings among texting, calling and reading. We also notice that the corresponding sensor reading for the same event does not appear too different across the four environmental conditions (possibly indicating that the consequences of distracted driving are relatively uniform across all environmental conditions). These trends also exhibited repeatability across all three axes for both sensors, and across subjects (which we do not show due to space limitations). As such, our rationale is to leverage these insights to process the accelerometer and gyroscope sensor readings in the phone to automatically detect distracted driving activities as and when they happen.

It is important to mention that during the times of undistracted driving, the smartphone was placed on a horizontal platform next to the steering wheel. At these times, the phone was relatively stable, since the participants in our study were experienced drivers, and there was little to no abrupt driving that would cause the phone to move suddenly, which could create arbitrary spikes in sensory readings in the phone. This is why the sensor readings are relatively stable during the case of driving with no distraction [1].

TABLE I:
Initial Features Computed from Sensor Data

Feature Description	Notation
Maximum, minimum and standard deviation of accelerometer in X axis	maxX, minX, stdX
Maximum, minimum and standard deviation of accelerometer in Y axis	maxY, minY, stdY
Maximum, minimum and standard deviation of accelerometer in Z axis	maxZ, minZ, stdZ
Resultant of Accelerometer	Mean, Standard Deviation, Variance, Energy, Median, Range, Interquartile Range
Square Sum Mean of Accelerometer	$\mu_{(x^2+y^2)}$, $\mu_{(x^2+z^2)}$, $\mu_{(y^2+z^2)}$
Square Sum Variance of Gyroscope	$\rho_{(x^2+y^2)}$, $\rho_{(x^2+z^2)}$, $\rho_{(y^2+z^2)}$
Number of Peaks of Gyroscope	NumPeaks
Squared sum of data below certain percentile (25, 75) of Accelerometer	sumsq25, sumsq75
Other features of Accelerometer	$\mu(\frac{(x+y)}{2} - z)$, $\mu(\frac{(x+z)}{2} - y)$ $\mu(\frac{(y+z)}{2} - y)$

[1]Even in the case of any abrupt sensory patterns that should manifest without the driver using the smartphone (like sudden braking or encountering a pothole for example), we hypothesize that the corresponding sensor spikes will be larger in amplitude and shorter in duration, and we could filter them as noise, but is something that needs more rigorous testing.

B. Data Pre-processing and Features Extraction

Once the tagged accelerometer and gyroscope sensor data from the smartphone is available, the first step is pre-processing the raw sensor data. The readings were sampled using a 2 seconds sliding window. We tried varying the window size from 1 second to 10 seconds with different amounts of overlap. After observations and analysis, we chose a sliding window size of 2 seconds with 10% overlap for our problem. In prior related work [22], it is found that 2 to 5 seconds window works best for human activity recognition using inertial sensors, and this explains the rationale for our choice.

Once the data is pre-processed, the next step is extracting features from the sensor datasets. Feature extraction and feature selection from input data are critical for accuracy of any supervised learning algorithm. To start with, we extracted 28 features from the accelerometer and gyroscope readings, which are easy to calculate, energy efficient and intuitive for our problem. Table I presents these. However, there are trade-offs here. Too few features may not be representative, and too many features incur processing overhead and sometimes can even decrease accuracy by introducing noise. As such, it is critical that we identify a limited set of features from accelerometer and gyroscope data that provide good discriminatory power among various activities of interest, while also keeping processing delay and energy low.

TABLE II:
Six Highest Ranked Features Finally Selected

Description	Notation
Square Sum Mean of Accelerometer	$\mu_{(x^2+y^2)}$, $\mu_{(y^2+z^2)}$
Squared sum of data below 75 percentile of Accelerometer	sumsq75
Other features of Accelerometer	$\mu(\frac{(x+z)}{2} - y)$
Square Sum Variance of Gyroscope	$\rho_{(x^2+y^2)}$, $\rho_{(y^2+z^2)}$

To address this issue, we use feature selection algorithms. Broadly, feature selection algorithms fall into two categories, Filter-based approach and Wrapper based approach. The Filter-based approach relies on general characteristics of training data to select some features without involving any learning algorithm. The Wrapper-based approach requires one predetermined learning algorithm during feature selection and uses its performance to evaluate and determine which features are best. It tends to find features better suited to the predetermined learning algorithm, at times resulting in superior learning performance, but it also tends to be more computationally expensive than the Filter-based approach. When the number of features becomes large, the Filter-based approach is usually chosen due to its computational efficiency.

Filter-based approaches require some methods to find the correlation between input feature and output classes. There exists broadly two approaches to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information gain. Linear correlation is easy to measure but may not be able to capture correlations that are not linear in nature. It also requires that all features contain numerical values. On the other hand, Information gain ranks features based on their informatory power which is the direct goal of feature selection. It measures how much additional information a feature provides after adding it into the set of existing features. Therefore, it works for both linear as well as nonlinear correlations.

In this paper, we use the Filter-based approach for feature extraction using a two step process based on linear correlation and information gain. First, we identify those features from our initial list that correlate the most with the classes that need to be identified (i.e., texting, calling and reading while driving). Once these features are identified, we then want to narrow down to those features that show sufficient contrast with each other (i.e., high information gain), since we do not want to select features that are already correlating among themselves. To do this next step, we employ the notion of symmetrical uncertainty [23] as explained below.

Consider a Random Variable $X = (x_1, x_2, ...x_n)$, where n is size of Variable X, which in our case is the feature vector computed over multiple time windows. Let $H(X)$ be the entropy of variable X. It is given by:

$$H(X) = -\sum P(x_i)log_2(P(x_i)), \quad (1)$$

where, $P(x_i)$ is the probability that variable x is in state x_i (i.e., $x_i \in X$).

Physically, the entropy here is the unpredictability of a value of the feature vector (X) as it is computed over multiple windows from sensor data for each class of activity assessed. Let $H(X|Y)$ be the entropy of vector X after observing values of another vector $Y = (y_1, y_2, ...y_m)$, where m is size of vector Y, which is given by:

$$H(X|Y) = -\sum{}_i P(y_j)\sum{}_i P(x_i|y_j)log_2 P(x_i|y_j))). \quad (2)$$

Let, $IG(X|Y)$ be the information gain about vector X provided by vector Y. A feature vector Y is regarded more correlated to feature vector X than to feature vector Z, if $IG(X|Y) > IG(Z|Y)$. Formally,

$$IG(X|Y) = H(X)–H(X|Y). \quad (3)$$

Finally the symmetrical uncertainty is given by:

$$SU(X, Y) = 2 * (\frac{(IG(X|Y))}{(H(X) + H(Y))}). \quad (4)$$

The symmetrical uncertainty between two feature vectors X and Y essentially reveals how much information feature vector X gives us, with knowledge of feature vector Y. Its value is one when knowledge of feature vector Y allows us to completely and correctly determine feature vector X. It is $zero$ when knowledge of feature vector Y reveals nothing about feature vector X. Ideally, we need to choose those features that have a high degree of discriminatory power (or low correlations). This will also provide good contrast for features used during classification.

As a result of this process, we selected six best features from this pool that considered both accelerometer and gyroscope sensor data. All selected features are listed in Table II. These features serve as an input vector x into Random Forest algorithm for activity classification, presented next.

C. Random Forest Classification Algorithm

Decision trees are popular in machine learning for classification. A decision tree represents a graphical tree where leaf nodes are the classes depicting the final prediction, while non-leaf/ internal nodes correspond to a decision that is made based on one of the features. Each internal node generates several branches depending on the condition placed on the corresponding feature. Random Forests (RF) [24] is a decision tree-based ensemble learning technique used for classification, regression, among others. It has the advantage of being extremely fast, efficient on big data and capable of overcoming overfitting. RF is a voting based ensemble of L decision trees (DT). Each DT works as an independent classifier and predicts one activity from processing that particular tree. The final activity selected from the algorithm is the one selected by the majority of trees. A DT is represented as $T_i(x, \theta_i)$, where x is an input feature vector extracted from raw sensor data and θ_i is a random vector that controls the structure of i^{th} tree. The random vector θ_i is generated independently but with the same distribution of the preceding θ_1, θ_2..., $\theta_{(i-1)}$ vectors. In the random subspace method, θ_i consists of K integers ($K \ll M$) randomly drawn from a uniform distribution in the interval $[1, M]$, where M is the number of available features. Given a dataset that contains N feature vectors, each consisting of M features, the RF algorithm builds the trained model using Algorithm 1.

Algorithm 1 Random Forest Algorithm

1) Draw N samples at random with replacement from the dataset with bootstrapping, to generate the training set of the tree.
2) Select any K features randomly from the set of available features M, where $K \ll M$.
3) Among the values for each of the K features drawn, choose the best binary split according to the Gini impurity index [25], which measures impurity degree in dataset. Gini index value lies between 0 and 1. It is maximum when all classes in dataset have equal probability and minimum when any one class has maximum probability. Finally select those features, which has the least impurity.
4) Grow the tree to its maximum size according to the stopping criterion chosen and let the tree unpruned.

Once the forest has been created, an unseen data sample is labeled with one of the activity classes by taking the majority vote: i.e., it is labeled with the activity, which is selected by maximum number of ensemble trees. In RF, given a decision tree T, and an input feature vector x to be classified, let us denote by $v(x)$ the leaf node where x falls when it is classified by T. The probability $P(a|x, T)$ that a sample feature vector x belongs to the activity a, where $a \in \{A_1, A_2, A_3, A_4\}$ for four activities, namely Texting, Reading, Calling and No Distraction

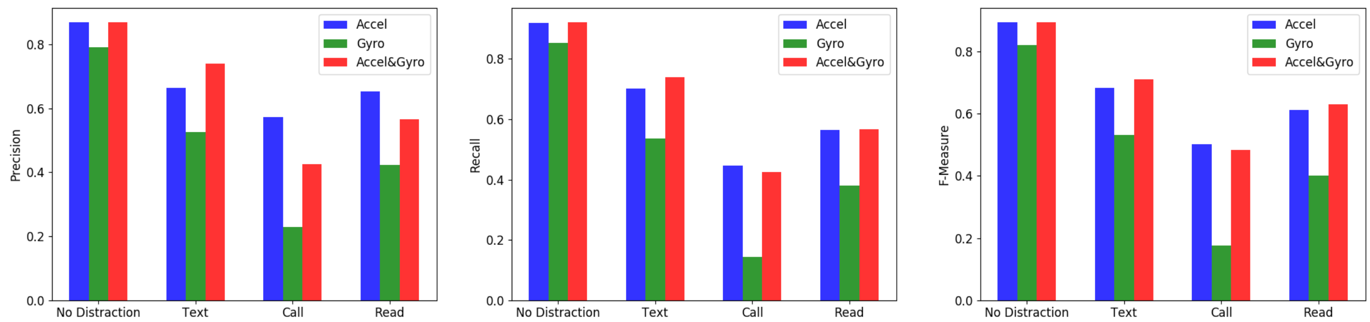


Fig. 4: Precision, Recall and F-Measure performance of our system for four activities for Same-user 10-fold cross-validation strategy for all four environmental conditions.

while Driving, is estimated by the following equation:

$$P(a|x,T) = \frac{n_a}{n}, \tag{5}$$

where, n_a is the number of training samples falling into the leaf node $v(x)$ after learning; n is the total number of training samples assigned to $v(x)$ by the training procedure; and T is a decision tree. Given a forest consisting of L trees and an unknown feature vector x to be classified, the probability $P(a|x)$ that x belongs to the activity a is computed as:

$$P(a|x) = \frac{1}{L}\sum_{i=1}^{L} P(a|x,T_i), \tag{6}$$

where, $P(a|x,T_i)$ is the conditional probability provided by the i^{th} tree (T_i) and is computed according to Eq. 5. As a consequence, for the sample feature vector x to be classified, the RF algorithm gives as output the vector:

$$p = P(A_1,x), P(A_2,x), P(A_3,x), P(A_4,x), \tag{7}$$

for four activities. The activity with the highest probability in the set is chosen as the classified activity for the i^{th} tree. The final activity is the one that gets the majority vote among all activities from all decision trees in the forest. For fine-tuning the performance of our RF model, we performed randomized grid search on hyper parameter space to find optimized hyper parameters. By evaluating just 1000 random combinations of different hyper parameter combinations, we achieved best results for the following hyper parameter values: 121 decision trees, information gain as splitting criteria, 6 as the maximum depth of each decision tree, 5 as minimum number of samples to split the internal node, and bag size of 100%. The entire process of classification (pre-processing, feature extraction and classification algorithm) was executed on a smart-phone.

V. RESULTS AND DISCUSSIONS

In this section we present results on the validation of our technique to detect distracted driving activities from processing accelerometer and gyroscope readings from the smartphone across four environmental conditions – day time, night time, fog and rain/snow.

A. Performance Measures

The results of our evaluation are presented in terms of three standard measures: Precision, Recall and F-Measure. Each measure is a function of the true positives (TP), false positives (FP), and false negatives (FN). The precision is the ratio of correctly classified positive instances to the total number of instances classified as positive. It is given by,

$$Precision = \frac{TP}{TP+FP}. \tag{8}$$

Recall is the ratio of correctly classified positive instances to the total number of positive instances, and is given by,

$$Recall = \frac{TP}{TP+FN}. \tag{9}$$

The F-Measure combines precision and recall into a single value. It is given by,

$$F-Measure = 2 * \frac{Precision * Recall}{Precision + Recall}. \tag{10}$$

Precision indicates how many of the testing samples classified as a particular activity actually belonged to that particular activity. Recall indicates how many of the instances of a particular activity were correctly classified as that activity. The F-Measure balances Precision and Recall.

B. Overview of Evaluation Methods

In this paper, we evaluate the performance of our system using three well-established methods that are standard for our problem scope. These testing methods are same-user K-fold cross-validation, cross-user K-fold cross-validation and cross user leave-one-out cross-validation.

In K-fold cross-validation, we divide the dataset into K subsets, and evaluates them K times. Each time, one of the K subsets is used as the test set and the other K-1 subsets are grouped together to form a training set. Then, the average error across all K trials is computed for final result. Within this method, there are two approaches to evaluate. In the Same user K-fold cross-validation method, the data evaluated belongs to only one subject. In Cross user K-fold cross-validation method, the data is aggregated from all subjects and then K-fold cross-validation is applied. In Cross user leave-one-out

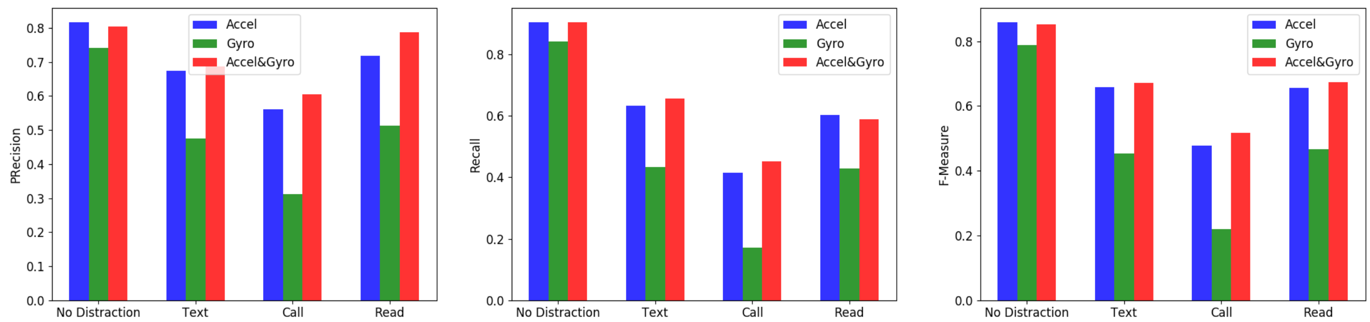


Fig. 5: Precision, Recall and F-Measure performance of our system for four activities for Cross-user 10-fold cross-validation strategy for all four environmental conditions.

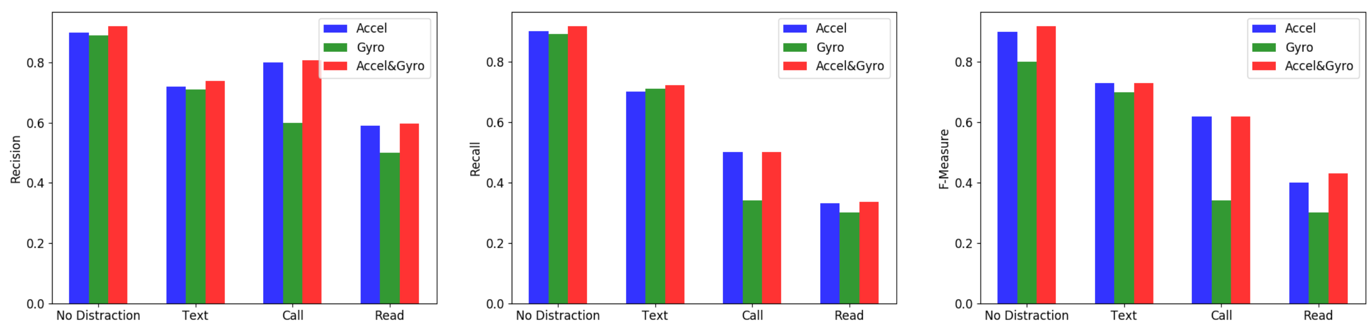


Fig. 6: Precision, Recall and F-Measure performance of our system for four activities for Cross-user leave-one-out strategy for all four environmental conditions.

method, out of n subjects, $n-1$ are chosen for training dataset and one is left for testing. The process repeats for every subject then average is computed for final result.

Note that our datasets are relatively uniform across classes to minimize any inherent biases. Also, among the three strategies evaluated, some may show better results than others. Usually evaluations on same users show better result compared to any cross user evaluation strategy. This is intuitive, since there are subtle variations among people even when they do the same activity that are sometimes hard to detect when training and testing are done on different people. However, as we show, our algorithm still achieves high performance both within and across users, hence demonstrating the effectiveness of our technique. However, with more training and testing across more subjects, we expect improved outcomes, and this is part of our on-going work with more experiments.

C. Integrated Evaluation across all Environmental Conditions

Recall that each of the 16 subjects in our study participated in driving experiments in four environmental conditions – day time, night time, fog, and rain/snow. Instead of training and testing our model on each condition separately (that will give us better results on accuracy), we chose to train and test our technique on a mixed model, where data from all users for all four conditions are incorporated. In other words, when testing the accuracy of our system under any one environmental condition (say night time) the training data sets includes data from all of the four environmental conditions. The same is true when testing across all other environmental conditions. As such, results presented are averaged out across all environmental conditions, for all of the three evaluation strategies presented above. This is the most practical scenario, since any technique proposed to detect distracted driving activities must work for all environmental conditions, rather than be specifically tuned for a single one. This will hence provide a more realistic evaluation of our proposed technique.

D. Results

Figs. 4 5, 6 and 7 present our results. Recall that the algorithmic approach to classify activities was presented earlier in Algorithm 1. The input features used for classification were only the six features from accelerometer and gyroscope sensors in the phone identified earlier in Table II.

a. Classification Performance: For the same user 10-fold cross-validation strategy, as shown in Fig. 4, the performance of accelerometer and gyroscope sensors are high to classify activities correctly, as indicated in the Precision, Recall and F-Measures. The overall performance is 85% for Precision, 84% for Recall and 87% for F-Measure for both sensors combined. We also see that while integration of features from the gyroscope sensor and the accelerometer sensor does improve accuracy for the most part, it is not significantly more than the accuracy when only features from the accelerometer sensor are used.

For the cross user 10-fold cross-validation in Fig. 5, we can see that average performance in classifying activities in

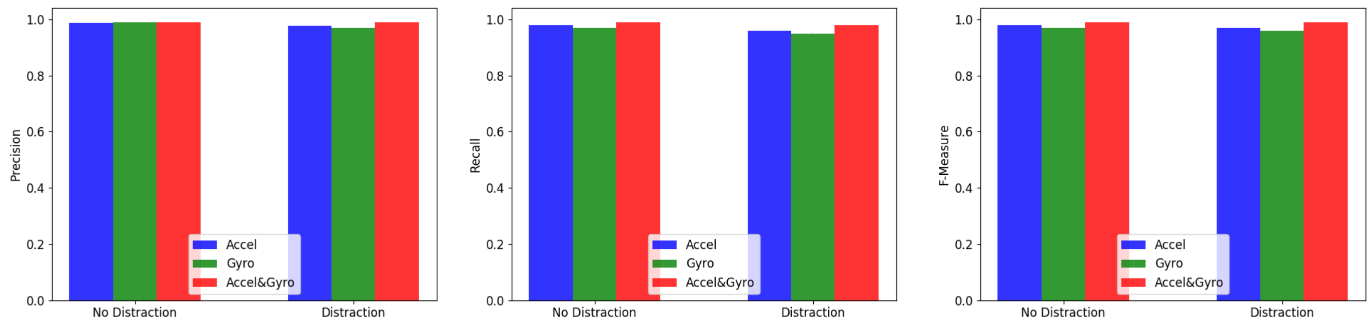


Fig. 7: Precision, Recall and F-Measure performance evaluation for Binary Classification Problem (Detecting Distracted vs. Not Distracted driving) for Cross-user leave-one-out strategy for all four environmental conditions.

terms of Precision, Recall and F-Measure is more than 80%, which again demonstrates the validity of our technique. Fig. 6 presents results for the more stringent evaluation strategy, which is cross validation with leave-one-out. We do see a drop in performance in this evaluation strategy. While the accuracy in detecting un-distracted driving is still very high (98.76%), we see that our proposed technique does not do well in detecting instances where the subject is texting, calling, or reading (where the average accuracy is only 63%). It is because, different users do have subtle differences in the way they text, call and read from a phone while driving, that can reduce classification performance with the leave one out evaluation strategy. We certainly hope that with more data from more subjects, our proposed technique will learn better for fine-grained activity classification.

However, we wanted to see how our current technique performs when we attempt to address a simpler, but nevertheless important and practical problem, which is to classify instances of distracted driving (irrespective of the type of distraction) from un-distracted driving. Fig. 7 presents the results for the cross validation with leave one out evaluation strategy. As we can see, our proposed technique achieves near perfect performance in terms of Precision, Recall and F-Measures in classifying distracted driving from un-distracted driving using the accelerometer, or a combination of the accelerometer and the gyroscope. This result further validates the practicality of our proposed technique, since it is known that irrespective of why a driver picks up a phone (for texting, or calling, or reading), they are all sources of distraction and best avoided. As such, we believe that the more complex problem of determining the fine-grained activity that causes the distraction may not be necessary after all if the motivation is to ensure safer driving by detecting any instance of distracted driving, which our system is able to determine as seen in Fig. 7.

b. Energy and Latency: We believe that our system is energy efficient and fast. For collecting accelerometer and gyroscope data continuously over 5 minutes, the energy expended on the Samsung Galaxy S4 phone was equivalent to 91mJ. The entire classification system including pre-processing, feature extraction and RF classification algorithm was implemented on a Samsung Galaxy S4 smart-phone. The total memory consumed by the app was 23MB. For 10 minutes of operation

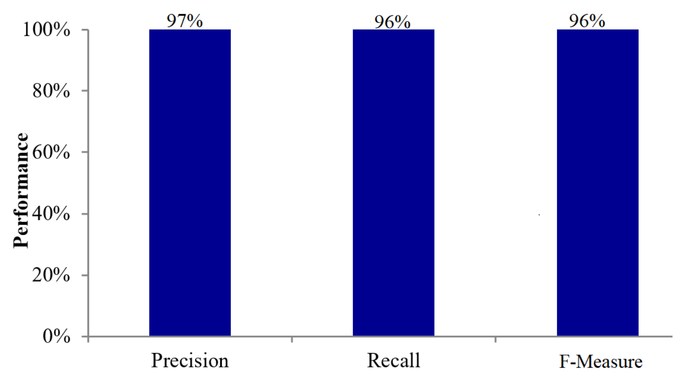


Fig. 8: Algorithm Performance in a Binary Classification Problem of Detecting Phone use during Driving as compared to Phone use during Walking, Sitting, Sitting or Being Passenger in a Car.

in an actual driving scenario, the battery consumed was about 3%, making our system energy efficient and practical. The latency of execution was less than 2 seconds.

c. A Note on Classifying Phone use while Driving from Phone Use in Other Contexts: In Fig. 8, we present results for another binary classification problem, where the goal was to detect if a subject uses a phone for texting, or reading or calling while driving, compared to when the phone is used for texting or calling or reading when performing other basic tasks which were Walking, Standing, Sitting, and Being a Passenger in a Car. All 16 subjects also participated for data collection in this experiment. The results shown are when the classification was attempted using Random Forests by leveraging the six features from accelerometer and gyroscope sensors identified earlier in Table II. The evaluation strategy was the more stringent leave one out cross user 10-fold cross validation strategy.

As we can see from Fig. 8, we have near perfect accuracy in classifying when the phone is used while driving compared to when the phone is used while performing other tasks. Now, smartphone applications can be designed to process sensory data from a phone and infer when a subject is texting, reading or calling, but initiate an alarm only when these happen while the subject is driving. This will make the detection of distracted driving activities completely pervasive to the user, which will

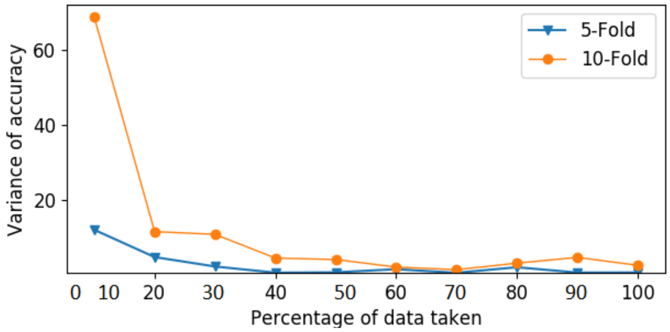


Fig. 9: Variance Trend for Different Subsets of our Dataset.

enhance practical adoption of our technologies. The downside though is the energy consumed during constant sensing and processing of sensory data on the phone. Designing strategies for optimizing this is part of our future work though.

d. A Note on Confidence of our Results: It is vital for any machine learning system to train on sufficient size datasets to avoid overfitting problems. To emphasize the sufficiency of our datasets, we point out that each of the 16 subjects gave us 11 minutes worth of data for each environmental condition, where the size of the sampling window was 2 seconds with 10% overlap. This results in an aggregate of about 1450 data points for each subject, and for each feature across across all four environmental conditions. Recall again that we have 16 subjects and six features for model development. For classifying just four activities (texting, reading, calling and no distraction), we believe that this is a good number of relatively balanced data points for our results to be representative.

Nevertheless, to validate this, we employ a variance based approach by incrementally quantifying the representativeness of our datasets for the case of 10-fold and 5-fold cross validation. This approach is well suited for our problem scope, and is also validated in the literature [26], [27]. In this approach, we randomized our entire dataset by dividing it into 10 non-overlapping subsets initially. We then classified activities using our algorithm for the first subset, that is 10% of the dataset to get 10 numbers for overall classification accuracy for the case of 10-fold cross validation evaluation. Then, we quantify the variance among classification accuracies identified. Then, we do the same for 20% of datasets, and again we compute the variance in classification accuracies among the 10 values for 10-fold cross validation. We do the same for 30%, 40%, 50%, 60%, 70%, 80%, 90% and the entire 100% of datasets. We repeat the same procedure for 5-fold cross validation.

Fig. 9 plots the resulting variances. As we see, for small size datasets, the variances are higher, but they start to decrease with more data. Beyond 60% of data, the variances are very stable tending towards zero. This we believe gives us confidence on the representativeness of our datasets for the problem we address in this paper.

VI. LIMITATIONS OF OUR STUDY

We now present important limitations of our study in this paper, and their impact on the findings and utility of our contributions. At the outset, the entire study was 'controlled' in a closed environment, but as we presented in Section II, the CAREN simulator we used has been widely tested, and evaluated by many participants to be realistic. Nevertheless, real roads are not simulated in CAREN, meaning that potholes, U-turns, speed breakers, sharp turns, excess winds etc. were not simulated. We can see this from Figs. 2 and 3 where there are no accelerometer and gyroscope readings on the phone outside of Texting, Calling and Reading. This will not be the case in real life on real roads where disturbances are common, and will be felt by the phone leading to spikes in accelerometer and gyroscope responses, and could trigger false alarms. We hypothesize that such abrupt changes could be detected and filtered as noise, but this hypothesis has not been tested, and very difficult to do as well in any driving environment (controlled or otherwise).

Secondly, our study used a subject group that was relatively younger, but homogeneous. Older people, Senior Citizens, and those with certain disabilities (but still legally allowed to drive) may have different patterns when using the phone, triggering differing sensor responses. As such, we cannot claim that our model is applicable for these groups as well. Much more diversity in testing and assessment are needed before we make generic claims on the broader impact of our study. But results in this paper provide a foundation to do so as future work.

VII. CONCLUSIONS

In this paper, we attempt to classify instances of distracted driving during phone usage via processing data collected from in-built accelerometer and gyroscope in the phones. Our contributions: a) We collected experimental data from 16 subjects that drove on a realistic car simulator, while texting, calling and reading from the phone for four environmental conditions, namely day time, night time, fog and rain/snow; b) We designed an app to collect time stamped sensory data from the accelerometer and gyroscope in the phone continuously in the background while the subject drove; c) We extracted 28 features from both sensory data, out of which only 6 features were finally selected using an information gain approach that provided good classification power; d) We then designed a Random Forests based algorithm to classify distracted driving activities in real-time, and evaluated the system extensively across multiple metrics, multiple environments, and multiple testing strategies that yielded favorable results.

We are encouraged by the fact that existing literature identifies the need for tech based solutions for better self management of potentially risky driving patterns to enhance road safety, which convinces us that our proposed contributions are impactful and practical. Limitations of our study, and the potential to overcome them were also presented at the end.

ACKNOWLEDGEMENTS

We thank Dr. S. Carey and Ms. K. Lostroscio for help with data collection. This work was supported in part by the US National Science foundations under grants CBET 1743985 and IIS 1559588. Any opinions, thoughts and findings are those of the authors and do not reflect views of the funding agency.

References

[1] "Distracted Driving: Facts And Statistics." [Online]. Available: http://www.distraction.gov/stats-research-laws/facts-and-statistics.html. [Accessed: 31-Aug-2016].

[2] "Cell Phones and Driving: Research Update." [Online]. Available: https://www.aaafoundation.org/cell-phones-and-driving-research-update. [Accessed: 04-Jul-2016].

[3] "Distracted Driving Public Opinion Poll," National Safety Council, Mar. 2016.

[4] "itcanwait." [Online]. Available: http://www.itcanwait.com/. [Accessed: 20-Sep-2017]

[5] "Snapshot." [Online]. Available: https://www.progressive.com/auto/discounts/snapshot/. [Accessed: 20-Sep-2017]

[6] "Metromile." [Online]. Available: https://www.metromile.com/ [Accessed: 20-Sep-2017]

[7] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory," In ACM MobiSys, 2011.

[8] C.-W. You, M. Montes-de Oca, T. J. Bao, N. D. Lane, H. Lu, G. Cardone, L. Torresani, and A. T. Campbell, "Carsafe: a driver safety app that detects dangerous driving behavior using dual-cameras on smartphones," In ACM UbiComp, 2012.

[9] C. M. Sarfraz Nawaz,"Mining User's Significant Driving Routes with Low-power Sensors," In ACM Conference on Embedded Networked Sensor Systems, SenSys 2014.

[10] S. Narain, T. D. Vo-Huu, K. Block, and G. Noubir, "Inferring user routes and locations using zero-permission mobile sensors," In IEEE Symposium on Security and Privacy (S&P), pp. 397–413, May 2016.

[11] B. Goel, A. K. Dey and S. Chellappan, "Detecting Routes Taken by Users on Public Vehicles From their Wearables," In IEEE 8th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, 2017.

[12] 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.

[13] P. Singh, N. Juneja, and S. Kapoor, "Using Mobile Phone Sensors to Detect Driving Behavior," In Proceedings of the 3rd ACM Symposium on Computing for Development, New York, NY, USA, pp. 53:1–53:2, 2013.

[14] C. Bo, X. Jian, X.-Y. Li, X. Mao, Y. Wang, and F. Li, "You'Re Driving and Texting: Detecting Drivers Using Personal Smart Phones by Leveraging Inertial Sensors," In Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, New York, NY, USA, pp. 199–202, 2013.

[15] H. A. Shabeer and R. S. D. Wahidabanu, "Averting mobile phone use while driving and technique to locate the mobile phone used vehicle," Procedia Eng., vol. 30, pp. 623–630, 2012.

[16] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, and R. P. Martin, "Detecting driver phone use leveraging car speakers," In Proceedings of the 17th annual international conference on Mobile computing and networking, pp. 97–108, 2011.

[17] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Sensing Vehicle Dynamics for Determining Driver Phone Use," In Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, New York, NY, USA, pp. 41–54, 2013.

[18] S. Tudor, S. Carey, and R. Dubey, "Development and Evaluation of a Dynamic Virtual Reality Driving Simulator," In Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments, New York, NY, USA, pp. 55:1–55:5, 2015.

[19] Hall, Mark A. "Correlation-based feature selection of discrete and numeric class machine learning," In Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00), Pat Langley (Ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 359-366, 2000.

[20] "Motekforce." [Online]. Available: http://www.motekforcelink.com/. [Accessed: 10-Sep-2016]

[21] "Teardown: What's inside Samsung's Galaxy S4? Electronics360." [Online]. Available: http://electronics360.globalspec.com/article/94/teardown-what-s-inside-samsung-s-galaxy-s4. [Accessed: 08-Sep-2016].

[22] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window Size Impact in Human Activity Recognition," Sensors, vol. 14, no. 4, pp. 6474–6499, Apr. 2014.

[23] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," In ICML, vol. 3, pp. 856–863, 2003.

[24] T. K. Ho, "Random decision forests," In Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282, 1995.

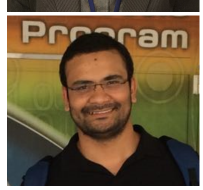
[25] J. L. Gastwirth, "The Estimation of the Lorenz Curve and Gini Index," Rev. Econ. Stat., vol. 54, no. 3, pp. 306–316, 1972.

[26] Ron Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," In Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2 (IJCAI'95), Vol. 2. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1137-1143, 1995.

[27] Y. Bengio and Y. Grandvalet, "No Unbiased Estimator of the Variance of K-Fold Cross-Validation," J. Mach. Learn. Res. 5, 1089-1105, December 2004.

Kaoutar Ben Ahmed. is a PhD student in the Department of Computer Science and Engineering at University of South Florida with research interests in big data analytics, machine learning, data mining, and smart cities.

Bharti Goel is a PhD student in the Department of Computer Science and Engineering at University of South Florida. Her current research work is in wearable computing applications.



Pratool Bharti is an R&D Manager. His interests are in Computer Vision, Deep Learning and Wearable Computing. He has a PhD in Computer Science and Engineering from the University of South Florida.

Sriram Chellappan is an associate professor in the Department of Computer Science and Engineering at University of South Florida. His current research interests include social and technological aspects in smart healthcare, public safety and cybersecurity. Chellappan has a PhD in Computer Science and Engineering from The Ohio State University.

Mohammed Bouhorma. is a professor of computer science at Abdelmalek Essaadi University, Morocco. He has received a PhD degree in telecommunications in 1995 from the National Polytechnic Institute of Toulouse, France. His principal research interests are in data analytics, serious games, wireless sensor networks, and smart cities.