Raja, Muneeba; Ghaderi, Viviane; Sigg, Stephan

WiBot! In-vehicle behaviour and gesture recognition using wireless network edge

Published in:
Proceedings - 2018 IEEE 38th International Conference on Distributed Computing Systems, ICDCS 2018

DOI:
10.1109/ICDCS.2018.00045

Published: 19/07/2018

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Raja, M., Ghaderi, V., & Sigg, S. (2018). WiBot! In-vehicle behaviour and gesture recognition using wireless network edge. In *Proceedings - 2018 IEEE 38th International Conference on Distributed Computing Systems, ICDCS 2018* (Vol. 2018-July, pp. 376-387). (International Conference on Distributed Computing Systems). IEEE. https://doi.org/10.1109/ICDCS.2018.00045

WiBot! In-Vehicle Behaviour and Gesture Recognition Using Wireless Network Edge

Muneeba Raja
muneeba.raja@aalto.fi

Viviane Ghaderi
viviane.ghaderi@bmwgroup.com

Stephan Sigg
stephan.sigg@aalto.fi

***Abstract*—Recent advancements in vehicular technology have meant that integrated wireless devices such as Wi-Fi access points or bluetooth are deployed in vehicles at an increasingly dense scale. These vehicular network edge devices, while enabling in car wireless connectivity and infotainment services, can also be exploited as sensors to improve environmental and behavioural awareness that in turn can provide better and more personalised driver feedback and improve road safety.**

We present WiBot! a network-edge based behaviour recognition and gesture based personal assistant system for cars. WiBot leverages the vehicular network edge to detect distracted behaviour based on unusual head turns and arm movements during driving situations by monitoring radio frequency fluctuation patterns in real-time. Additionally, WiBot can recognise known gestures from natural arm movements while driving and use such gestures for passenger-car interaction. A key element of WiBot design is its impulsive windowing approach that allows start and end of gestures to be accurately identified in a continuous stream of data.

We validate the system in a realistic driving environment by conducting a non-choreographed continuous recognition study with 40 participants at BMW Group Research, New Technologies and Innovation centre. By combining impulsive windowing with a unique selection of features from peaks and subcarrier analysis of RF CSI phase information, the system is able to achieve 94.5% accuracy for head- vs. arm movement separation. We can further confidently differentiate relevant gestures from random arm and head movements, head turns and idle movement with 90.5% accuracy.

I. Introduction

Driver-car interaction is subject to intensive theoretical research [1], human-studies [2] and long-term investigation [3]. Intelligent personal assistants like Amazon Alexa [1] and Siri [2] enable the natural human like interaction utilising speech as the medium. Additionally, camera-based, touchless-gesture control systems are being introduced in new cars, like BMW 7-series, with confined detection area, allowing users to perform simple operations, like adjusting volume.

In car, since any interaction could potentially distract the driver and thus increase the risk of accidents, a distraction-free interaction that can be conducted without taking the attention off the road is essential. Common solutions cover the inclusion of haptic feedback on physical control elements (e.g. designing their shapes and surface characteristically), or buttons with characteristic haptic design [1]. For instance, touch screens

[1]https://developer.amazon.com/docs/alexa-voice-service/api-overview.html
[2]https://www.apple.com/ios/siri/

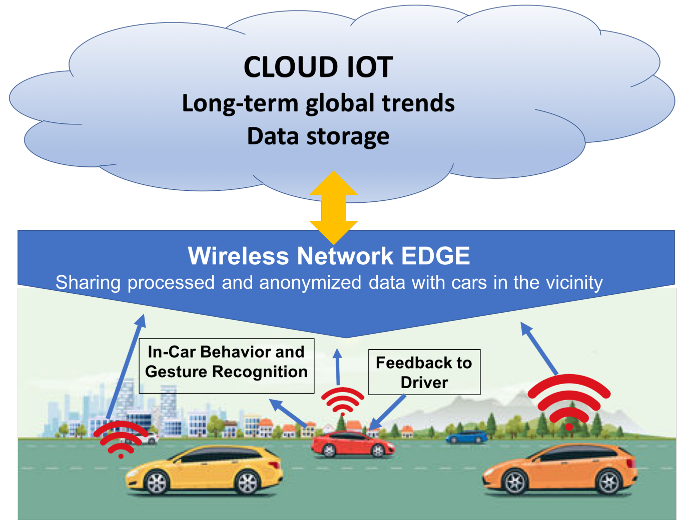


Fig. 1: WiBot! network-edge based behaviour and gesture recognition system.

that replace traditional interfaces might be problematic with respect to learnability. They can also be distracting, because the driver has to rely on visual feedback and cannot form a muscle memory or map of the controls over time [4].

For this reason, speech-based interaction is an active field of research and integrated in first in-car interaction systems [5]. However, speech interaction requires discipline from the driver and can be perceived as a disturbance to the social interaction when driving in company. Especially for simple interaction and commands, speech is not optimal and can also be distracting [6]. Moreover, people with speech disabilities can be deprived of such facilities. We propose to supplement these existing driver-car interfaces with network-edge based gesture recognition interface.

The purpose of our research is to enhance the capability of intelligent personal assistants, by enabling them understand the behaviour of the driver, which leads to appropriate assistance and feedback. In this paper, we focus on the distracted behaviour detection. WiBot recognises the behaviour based on human body movements, such as frequent head turns, random arm movements and posture changes. In addition, WiBot also recognises two simple non-distractive gestures, 'push' and 'swipe' from all other random actions being performed during a drive. The purpose of these two gestures is to enable the driver to interact with the personal assistant to command 'yes' and no' respectively. WiBot is limited to two easy to remember gestures in order to keep the human computer interface simple, reduce cognitive overhead and broaden the spatial range of gesture detection in real time processing. WiBot performs its data computations locally in-car and is designed as an edge based system to guarantee fast and timely user feedback (cf. Figure 1). The integration of edge based RF recognition into the car is natural as most contemporary cars already feature RF interfaces such as Wi-Fi or Bluetooth.

We focus on Wi-Fi and, in particular, channel state information (CSI)-based solutions, as these will soon replace Bluetooth for in-car interaction and entertainment [7]. Exploiting in-car edge devices enable efficient data processing and real-time response generation to best facilitate the drivers. WiBot is unique because it captures behaviour from body movements. It can otherwise be done using video cameras, which not only come with heavy image computational challenges, but also is a great privacy concern for people using cars in their everyday lives. Inattentiveness detection has been exploited based on gaze tracking techniques using video cameras [8]. The limitation of this approach is low visibility at night, inappropriate for people wearing sunglasses in daylight and limited camera focus which looses information if person moves out of camera frame.

Our contributions are (1) accurate recognition of unscripted free/natural movements from continuous CSI phase information by impulsive windowing technique, (2) Distinction between arm-movement and head-movement from a single-receiver CSI system in a vehicular setting, (3) implementation of a Wi-Fi-based human behaviour detection system for vehicular settings that also enables gesture-based interaction patterns, and (4) a case study with 40 subjects in realistic driving environments.

We propose a Wi-Fi based human behaviour detection system for a driver in autonomous and non-autonomous vehicles. The behaviour detection is based on unusual head and arm movements. We conduct a distraction based human study with 40 participants at BMW Group Research, New Technologies and Innovations, Germany. Additionally, we propose a gesture recognition system to communicate with the car bot/communication system. The gestures are 'push' and 'swipe' which translate to 'yes' and 'no'. Contrary to other researches, we do not pre-define a set of activities to distinguish from our interaction gestures. Rather, we conduct a totally non-choreographed study, without instructing users to behave in a particular way. We induce triggers to distract the study participants, which lead to various body movements. We propose our impulsive windowing method for identifying start and end point of movements/activities that a subject does during his drive. We propose a feature selection method based on peak analysis and subcarrier analysis to distinguish between random activities.

Challenges we take: As the focus of our research is to analyse behaviour from naturally occurring movements, we encountered several challenges during the course of experiments, and we present solutions to these challenges in the following sections of this paper.

- C1: How accurately can we actually perform gesture recognition when the person is in driving situation, and is not instructed to behave or move in a certain way? Can we do this without defining an overhead of preamble which demands performing additional gestures that are not the optimum solution in real driving?
- C2: Can we distinguish between arm movements and head movements with a single transmitter and receiver?
- C3: Can we identify simultaneously occurring arm movements and head turns?
- C4: Can we distinguish between any random arm/ head movements from defined gestures?

The rest of the paper is structured as follows. Firstly, we acknowledge the research work done in the domain of RF based activity sensing in Section II. Then we illustrate the necessary theoretical knowledge about CSI and phase correction in Section III. Section IV is about the hardware prototype setup and configuration for data collection. Section V demonstrates the in-depth human study carried out during the course of this research. Section VI covers the system architecture, overview, method details and results. Section VII highlights the challenges that are yet to overcome, Section VIII tells about the real world applications of WiBot. We conclude our paper in Section IX.

II. Evolution of Wi-Fi as activity sensors

Human movements and activities have been widely studied with computer vision, wearable sensor-based and ambient device-based sensors [9], [10], [11]. Recent advances in infrared LED and depth camera like Microsoft Kinect [12], [13] have overcome limitations such as dependence on light illumination and darkness, however, there are still open issues that need to be addressed in the future such as privacy intrusion, need for installing dedicate devices, inherent requirement for line of sight and intensive computation for real-time processing. These limitations in the existing technologies means that newer and better methods have to be sought for movement detection. One such potential area that is catching traction recently is to use WiFi receivers as sensors. One of the obvious benefits of using WiFi receivers as sensors is their existing installed base within the edge network infrastructure. The initial research in this domain was focussed on Received Signal Strength Indicator (RSSI) fluctuations as the primary indicator for sensing and localisation [14], [15], [16], [17], [18], [19]. The focus has now shifted to Channel State Information (CSI) in thirst for fine grained activity detection and higher accuracy. Furthermore, researchers have taken the challenge of phase correction problem to even capture the direction of motions [20]. Another improvement in this research area is a shift from dedicated or specialised hardware to commercially available Wi-Fi devices. Most commonly used hardware in this field is Software Defined Radios (USRP) which provide high accuracy for fine-grained activities [21],

[22], [18]. Human activity recognition with Radars is also a very tempting solution due to its high frequency, higher distance resolution and ability to detect the micro-doppler variations [23], [24], [25]. But these system require dedicated hardware which is high in cost as compared to commercially available Wi-Fi devices, such as WLAN cards.

Most recent research on human activity recognition with CSI, amplitude and phase information include [26], [27], [28]. Fall detection [29], indoor localisation [30], crowd sensing [31], smoke detection [32] and direction based exergames [20] are among the prominent ones. The effect of human movements on channel state information (CSI) leveraged by these studies proves that CSI has an advantage over visible light, infrared, or thermal energy for detecting human movements. Widance [20] present a novel approach for finding direction of movement from doppler effect. The limitation of Widance [20] is that the method of performing leg movements is highly choreographed and other body movements are controlled. In Smoky [32], the smoking detection system uses subcarrier level information and image processing techniques to obtain the fine grain movement patterns. They distinguish between similar activities and predefine a set of actions that makes smoking a composite activity, like holding, putting up, sucking, putting down etc. Although the focus is more towards fine grained activities, but the trained classes of activities are limited and do not take into account all the possible random movements. We take inspiration from the recent studies and extend our work towards detection of natural, non-choreographed movements and behaviour.

All the above mentioned RF based gesture or activity recognition techniques do not exploit body movements to detect human behaviour. To the best of our knowledge this is the first work, which develops RF-based behaviour recognition system for a complex and natural scenario like car driving where the gestures or movements are not known before hand, by utilising CSI data collected from commercially off-the shelf WLAN card. Furthermore, WiBot distinguishes two simple gestures, push and swipe, from all other movements (upper body) happening during a drive. The theoretical knowledge about CSI data applied in our system is explained in the following section.

III. Channel State Information

Current Wi-Fi standards (IEEE 802.11 a/g/n) widely use orthogonal frequency division multiplexing (OFDM). With OFDM, wireless data is transmitted over multiple orthogonal subcarriers. These subcarriers are a result of spectrum partitioning with OFDM. The frequency selective fading is mitigated using the same modulation and coding scheme (MCS) [33].

CSI in comparison to RSSI, provides subcarrier-level amplitude as well as phase information of the OFDM channel. Therefore, it tends to be more informative and stable representation of channel characteristics than RSSI. Wireless Network interface cards (WLAN NICs) capture the channel state information for every frame for decoding the payload. Equation 1 is the fundamental equation depicting the traditional transmitted-received signal in a multi-path environment. $x(t)$ is the input radio signal, $H(f,t)$ is the complex valued channel response/channel transfer function at a frequency f and time t which models the channel and y(t) is the output signal. $H(f,t)$ is called as channel frequency response (CFR) which is defined on the basis of channel noise.

Wi-Fi NICs report CFR values in the form of CSI matrices [28]. If N_{T_x} is number of transmitter antennas and N_{R_x} is number of receiver antennas and S is number of OFDM subcarriers, then for one Wi-Fi frame, every single CSI measurement contains S matrices of dimension N_{T_x} x N_{R_x}. In our case, the $N_{T_x} = 2$ and $N_{R_x} = 3$, S = 30 (from Linux CSI tool [34]).

$$y(t) = x(t).H(f,t) \tag{1}$$

A. Relative Phase Utilization

The output signal in Equation 2 depends on N multi-paths due to line-of-sight (LOS) path and non-line-of-sight (NLOS) paths from surrounding objects reflection. $H(f,t)$ [35] can be written as follow:

$$\begin{aligned} H(f,t) &= \sum_{k=1}^{K} \alpha_{k(t)} e^{-j2\pi f \tau_k(t)} \\ \hat{H}(f,t) &= H(f,t).e^{-j2\pi(\Delta t f + \Delta f t)} \end{aligned} \tag{2}$$

where K is total number of paths, $\alpha_k(t)$ and $\tau_k(t)$ are the complex attenuation factor and time of flight for k-th path respectively. In theory [28], when the clocks between transmitter and receiver are perfectly synchronised, the accurate phase information can be retrieved, say in RFID systems. However, the commercial Wi-Fi devices, specifically WLAN cards can have unknown frequency shifts and timing offsets between transmitter and receiver, leading to erroneous phase measurements. In Equation 2, $2\pi(\Delta t f + \Delta f t)$ is phase shift caused by carrier frequency and timing offset. IEEE 802.11 standards accept the carrier frequency drift of 100KHz. This leads to random phase shifts in channel state information. Inspired from [28], [20] we utilise the relative phase ideology derived from static and dynamic components for our analysis. In short, we take relative angle between two antennas. We take two signals $y_1(t)$, and $y_2(t)$ and we model them as complex numbers (vectors) containing real and imaginary parts. We want to find the angel between them. The usual mathematical way to find phase between two vectors (represented as complex numbers), is to find the dot product of the vectors and then divide by the multiplication of the vector magnitudes.

$$\theta = \arccos\left(\frac{\mathrm{Re}(y_1 \cdot y_2)}{\|y_1\|\|y_2\|}\right) \tag{3}$$

This is computationally expensive to perform for every received packet. We can use another interesting mathematical properly of complex numbers. Multiplying two complex numbers causes there magnitudes to be multiplied while their angle add. Taking the conjugate of one of the complex numbers

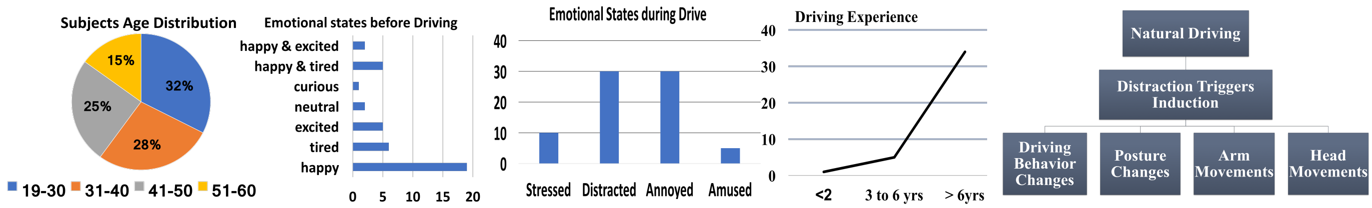


Fig. 2: Study statistics; age distribution, emotional states before and during the drive, driving experience and study design.

Fig. 3: Images captured during the human study; subjects are performing head and arm activities in response to triggers.

before multiplication would then mean that the angles of the two vectors would subtract, giving us the angle between the vectors shown in Equation 4.

$$y_1.y_2 = (|y_1||y_2|)[cos(\theta_1 + \theta_2) + isin(\theta_1 + \theta_2)] \tag{4}$$

Using Equation 4, if $y_3(t) = y_1(t)Xy_2(t)'$, and $y_1(t)$ and $y_2(t)$ both are rotating at the same speed (two signals at same frequency), $y_3(t)$ will ideally not be rotating at all since phase of $y_3(t)$ is the difference between phases of $y_2(t)$ and $y_1(t)$, and that should (under static conditions) not change much, although there may be a constant phase difference between $y_1(t)$ and $y_2(t)$ due to spacial diversity.

IV. Hardware Prototype

In order to collect the CSI data with the modified Linux driver [34], we use a Dell and an HP laptop with IWL 5300 card (having transmit power= 32mW and bandwidth up to 450Mbps), installed as a transmitter and receiver. We installed the CSI capturing tool on both transmitter and receiver. At the receiver end, where majority of the signal processing and machine learning is applied to the signal, the laptop used is HP 6930p with Intel Core 2 Duo T9400 Processor (2.53GHz), 2GB 800MHz DDR2 RAM and 7200 RPM HDD. The injection mode was setup for transmission with carrier frequency 5.32GHz, sampling frequency 1KHz and channel bandwidth of 20MHz. The 5GHz frequency ensures less wireless interferences and better distance resolution in comparison to 2.4GHz. We recorded the ground truth data using GoPro Hero Session Camera. Time information is critical in our experiment for data labelling, therefore we record the timestamps in milliseconds with each packet. All the devices are remotely controlled and the collected data is cross labelled using ELAN [3] open source labelling tool and further processed in MATLAB.

[3]http://tla.mpi.nl/tools/tla-tools/elan/

V. Non-Choreographed Human Study

With the exceptional advancement in vehicular technology in the past decade, from non-autonomous to highly autonomous, brings a lot of potential research possibilities. The ideology behind these innovations are to provide unprecedented services to humans and improve their quality of life. The effective time utilisation by performing additional tasks while taking a ride is the highlight of this technology. This means that driver can perform other activities by transferring driving control to the car. Also, the feature such as speech recognition bots in cars lets the driver give commands or interact with the car in more human way. In order to interact with the driver, bot system in car needs to understand and learn human behaviour in order to respond in a required way. As we consider that person can be in driving and non-driving mode, the gesture recognition should be applicable for both scenarios. Currently, speech recognition is used as the modality for communicating with the driver. Taking into account that about 9% people on average have speech problems [4], this modality does not cover the whole range of people using cars. Our idea is to define gestures for 'Yes' and 'No', which can be an alternate for such people. Moreover, people without any speech problems can have different emotional state and they might not always want to talk to bot by speaking. Also speech recognition isn't readily available in all languages. this also makes a good case for gestures.

We carried out an extensive human study of 40 participants at BMW Group Research, New Technologies and Innovations, Germany. As a prerequisite, the data privacy consent was signed by each participant. Before the experiment, the general information about each participant (name, age, sex, driving experience, current emotional state and any external variables present at the time of experiment) were recorded.

[4]https://www.nidcd.nih.gov/health/statistics/statistics-voice-speech-and-language

The participants ranged from 18 to 60 years, 27 males and overall driving experience of over 6 years (cf. Figure 2).

A. Study Design

The thought behind the study is to distract the subject during his otherwise smooth drive and capture body movements which happen in response to distraction triggers. Additional thing that we capture is two gestures; push and swipe. The car used for experiment is BMW Mini Cooper with Augmented Reality video used in place of simulator. The drive time is around 10 to 15 minutes. There are 3 separate phases in which driver has to drive the car. In this paper, we discuss the phase where we induce numerous triggers to distract, annoy and alarm the driver. The audio triggers used are different noises emerging from different directions, for example, crying baby from back seat, approaching ambulance siren from left side and repetitive horn sounds from right side. In response to the direction of these triggers, the driver moves his head and arms in different directions. Due to repetition of sounds, his attention is diverted and his way of driving changes. Same driving simulation is used for all the drivers, the sequence of triggers is randomly permuted for each experiment and no instructions are provided to the subject during the course of the experiment. The push and swipe gestures are recorded by using them as indicators of 'yes' or no' respectively. The data recording is done by asking questions over audio before and after the drive to mitigate the in-car bot conversation. In a feedback of this experiment, more than 80% of the people reported that they felt distracted and annoyed during the drive Figure 2.

The data captured gives natural movements, for example, small head movements during driving, random arm movements like scratching head, adjusting mirror and significant head movements like turning head backwards, left or right due to induced triggers. Hence the movements are not defined in advance, there is no fixed count of movements performed and there is no specific way advised to do anything. The push and swipe gestures, however are explained in advance and the general suggestion to behave naturally is given prior to the experiment. Despite defining push and swipe gesture, the way of doing the gesture, speed and duration is still different for each participant. Figure 3 shows the images captured during the study.

VI. System Architecture

Detection of human movements in real time vehicular scenarios requires fast processing of data and feedback to user, which if delayed, can be meaningless. Communicating directly with cloud is not an efficient solution as it increases the end to end latency of the system due to longer network path. Computing and storing the data locally at the edge device is the most suitable choice for vehicular scenario. Reason being, there is only one user per edge device and is located very close to it. Most time consuming step is learning and classification, which in this case is simplified as the patterns stored locally are for a single user, and only the most recent and repeated ones need to be used for detecting behaviour. This also ensures privacy and anonymity of the user, as the data is not continuously shared on the network. Depending on the length of activity being performed, the edge devices in car process window times ranging from msec to few seconds for computation. The real time data collection, processing and classification can be performed in this time. As the feedback is required only when certain threshold is met, based on the repetition or duration of movements, this reduces the load of generating continuous feedback. This means, we don't need to run classification once every second, but rather once per window, and window may span over multiple seconds based on the activity frequency, so we save ourselves from continuously running classification. The computation and storage can be extended to cloud to record global trends for features, patterns and feedbacks and handling computations that can not be solved by the local edge device. For example, when the data is collected for which pattern does not exist locally, the request can be forwarded to cloud to get an appropriate feedback, albeit this comes at the expense of added delay due to potentially longer round trip time to the cloud nodes. Figure 4 shows the high level architecture for WiBot.

Fig. 4: Depiction of end to end WiBot computation and latency at edge and extension to cloud.

The low level system system architecture for WiBot behaviour detection, based on human movements is shown in Figure 5. The raw CSI data in complex form is first interpolated to remove any missing information. Then we separate the amplitude and phase. We utilise both amplitude and phase in our system to capture the fine grained information in data. The phase has to be first corrected in order to be used for analysis. The de-noising is performed to preserve the critical information and remove the noise. We perform impulsive windowing for accurate detection of activity boundaries in online and offline processing. Subcarrier sanitization is performed to remove the outliers. The peak analysis and subcarrier analysis are the two major stages for feature computation. We then classify the data based on the trained model and predict the labels. Figure 5 illustrates the proposed system architecture and WiBot pseudocode with major steps listed is shown in Algorithm 1. These steps will be explained in detailed in the following sections.

A. Interpolation

Natural movements are instantaneous, the labelling data instances is a very challenging task here, as a single head movement could happen in less than one second or could take up to 3 or more seconds. This requires a very precise synchronisation between time in recorded video and the timestamps captured with each packet. Despite configuring our Wi-

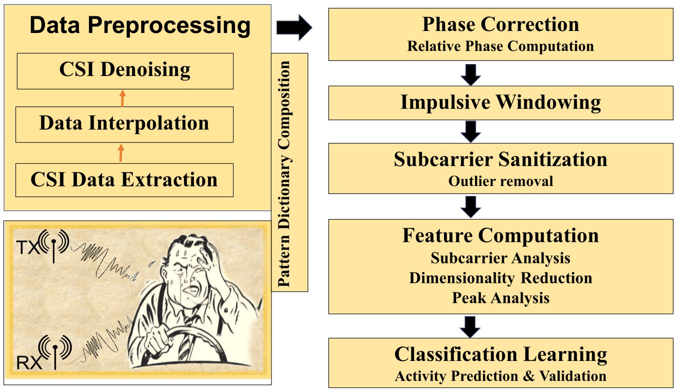


Fig. 5: WiBot proposed architecture.

Algorithm 1: WiBot Algorithm

```
function WiBotFunction (y1, y3);
Input  : Complex Signals y1(t) , y3(t)
Output: Window Label Set
initialization;
relφ(t)13=phase (y1(t), y3(t)');
denoisedPhase = lowpassFilter(relφ(t)13);
windows = computeWindowBoundaries(denoisedPhase);
for win = windows 1...n do
    unwrappedWindow = unwrap(win);
    win = removeDcComponent(unwrappedWindow);
    sanitizedWin= removeOutlyingSubc(win);
    reducedWin= dimensionReduction(santizedWin);
    peakFeatures=performWindowPeakAnalysis(reducedWin);
    subcarrierFeatures=
      performSubcarrierAnalysis(SanitizedWin) (cf.
      Algorithm 2);
    labelSet = classificationModel(peakFeatures,
      subcarrierFeatures);
    Labels[winIndex] = labelSet;
end
return Labels;
```

Fi device driver to transmit at a fixed transmission rate, we observe some non-uniformity in the recorded samples, due to packet loss and transmission delays. To overcome this, we perform interpolation for accurate time domain analysis and labelling. Linear interpolation fills the missing data points by previous data point value.

B. Phase Correction

The methodology for correcting phase information in collected CSI streams is explained in Section III. Figure 6 are the angle histogram/polar plots illustrating the distribution of phase values, grouped in accordance to their numerical range. It shows the absolute phase changes (left Figure) measured from CSI stream of a single antenna and the phase measurement after applying the relative phase correction technique (right Figure) on the same CSI stream. We can clearly see the distribution of phase changes in both figures. The absolute phase is unusable as its uniformly distributed in the range of 0° to 360°. On the contrary, the relative phase for the CSI stream tend to concentrate within the sector of 210° to 240°.

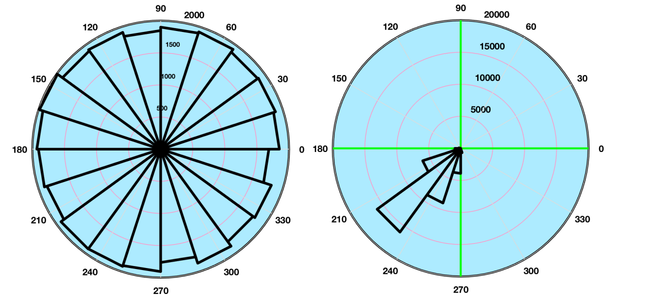


Fig. 6: Random/absolute phase shifts and corrected/relative phase shift for CSI stream.

This practical observation can also be confirmed mathematically. In our experimental setup, we are using Wi-Fi signals with a wavelength of 5cm. Our antennas are arranged as an antenna array and the distance between two antennas is 3cm. As shown in Figure 5, the reflected radio waves arrive at the receiver almost at an angle of incidence of 90 degrees. Using equation 5,

$$\Delta\phi = (2\pi d sin\theta)/\lambda \tag{5}$$

where d = distance between antennas, θ = angle of incidence of wave and λ = wavelength. Substituting $\pi/2$ in θ, 5cm in λ, and 3cm in d, we can compute the expected phase difference between the receivers and that comes out to be $1.2*\pi$ radians, which closely matches our observed relative phase difference shown in Figure 6.

C. Denoising

Upper body movements are of highest interest in this research. Normal speed of upper body movements is observed to be around $0.75m/s$. For Wi-Fi devices with carrier frequency of 5.32GHz, the frequency $f = 2V_m/\Lambda$. This makes it 30Hz, which WiDance also confirms to be the frequency of body movements [20]. The choice of cutoff frequency is critical in our case. The reason being no defined movements, the lack of periodicity and varying speed, varying intensity of both head and arm movements. In order to ensure that both arm and head movements of varying patterns are preserved in our filtered data, we use the low pass butter-worth filter with cutoff frequency of 30Hz. We filter all the subcarriers and utilise in further analysis. The raw (left) and filtered (right) phase CSI stream is shown in Figure 7. We can clearly see in this Figure that the raw phase is useless as all the activity information gets corrupted, while the denoised phase reveals very clear picture of 3 push gestures performed in between the no activity regions.

D. Pattern Dictionary Composition

Based on our data set, we observed that subjects repeatedly perform activities that can be seen as independent and disjoint

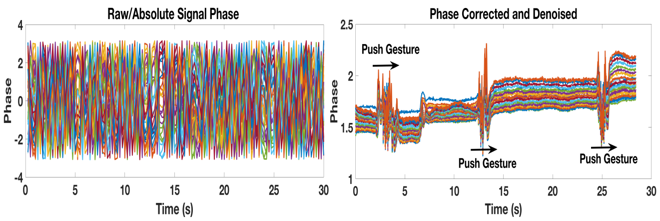


Fig. 7: Raw (left) and denoised (right) phase.

TABLE I: Possible labels observed during a drive.

Labels	
L_1	turning head towards right
L_2	turning head towards left
L_3	turning head partially backwards, such as looking at the rear seat
L_4	random head movement
L_5	random arm activity
L_6	performing "push" gesture with the hand
L_7	performing "swipe" gesture with the hand

from each other. We categorise such activities into the set of distinct labels as shown in Table I.

However, we also noticed that sometimes multiple activities are performed together within the same time window. This leads us to model our classification problem as multi-label classification, where each window, or data point, can have one or more labels assigned to it. Multi-label classification problem in context of high level activity recognition is explored by [36]. We use the Label Powerset (LP) method to transform our multi-label problem to single label problem, and then apply the k-nearest neighbour classifier to categorise our windows as belonging to one of the classes, where one class could be the combination of two or more base classes.

Label powerset method can be modelled as follows:

Instance $x = [x_1, ..., x_d] \in R^d$
Class labels: $L = \{1, 2, ..., L\}$
Label space: $Y = \{0, 1\}^L$
Labelset: $y = [y_1, ..., y_L] \epsilon Y$;
$y_j = 1$ if jth label relevant to x; else 0
Training set: $\{(x_i, y_i) | i = 1, ..., N\} \subset (XxY)$
Classification: $h : X \in Y$

LP transformation usually suffers from complexity issues due to dimensionality of the label space. Considering we have 7 distinct labels, the total possible label subsets will be 2^7. However we solve this dimensionality issue by taking into account the label correlations in our training data. We observe that only few of the labels occur together in the same instance or window. We observe that other combinations of labels are not possible or highly unlikely and thus ignore them from our output label subsets. For instance, head turned to right and left cant occur simultaneously, nor can push and swipe overlap. This reduces the number of label subsets and allows better performance. The label subsets with highest occurrences are shown in Table II.

TABLE II: Label subsets observed during experiments.

Label Subsets	
L_1, L_5	turning head towards right, random arm movement
L_2, L_5	turning head towards left, random arm movement
L_3, L_5	turning head partially backwards, random arm movement
L_4, L_5	random head movement, random arm activity

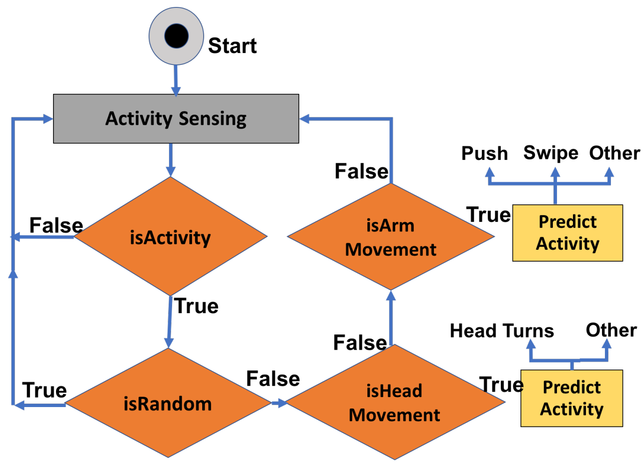


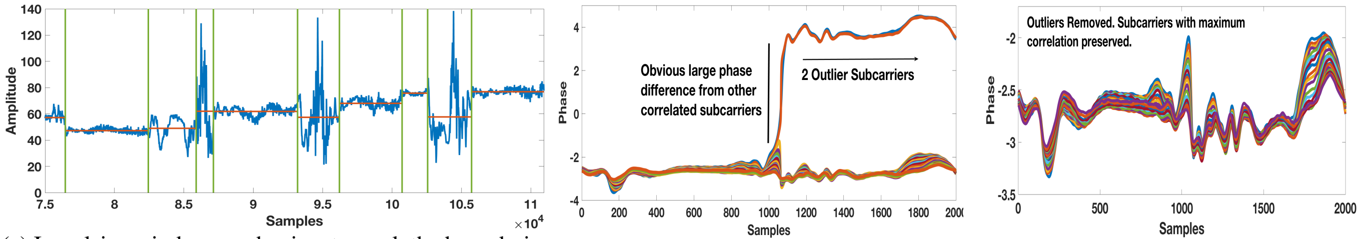
Fig. 8: Activity breakdown flowchart.

E. Impulsive Windowing

Before we can detect and identify individual activities being performed, we need to separate them from each other over time. This requires that we somehow divide the incoming RF signal into time windows. A naive windowing approach could be to choose a fixed window size, and then perform pattern recognition on each window individually. However such a fixed size temporal windowing methodology suffers from a number of drawbacks. Firstly, the activities we want to identify could vary in duration and we can not foresee how short or long an activity is going to be. Secondly, even if we have prior knowledge about the length of activity, we would need to know exactly when the activity will start in time. Since our subjects are not choreographed, we also do not have the possibility to look for some well known "preamble" that would signal to our windowing algorithm that an activity is about to follow.

Abrupt and instantaneous changes happening in time series data from natural physical environments demands for efficient detection of changes and with optimal cost. We adopt the concept of finding locations where data values are changing abruptly and utilise them for marking the boundaries of a window. We tend to identify the points in input data where statistical attributes fluctuate. This is a change points problem and we utilise the concept introduced by Killick [37] in the area of statistics.

In mathematical terms, the input data in ordered sequence can be represented as $y1 : n = (y_1, ..., y_n)$. The output model should have m number of change points, along with the locations, $\tau 1 : m = (\tau_1, ..., \tau_m)$. The change point position must be an integer. Each change point position lies in the range of 1 and $n - 1$. We define $\tau_0 = 0$ and $\tau_m + 1 = n$ and assume



(a) Impulsive window mechanism to mark the boundaries (green) of an activity.

(b) Subcarrier outlier detection.

(c) Subcarrier sanitization.

Fig. 9: Impulsive Windowing (a) and Subcarrier Sanitization (b) and (c).

that the change points are ordered such that $\tau_i < \tau_j$ if, and only if, $i < j$. Consequently, the m change points will split the data into $m+1$ segments, with the ith segment containing $y(\tau_{i_1+1}) : \tau_i$.

To solve the above mentioned issues, we make use of a maximum likelihood estimation algorithm that dynamically identifies indexes in our data, where a significant statistical change has occurred. In particular we aim to find all points in our data where the standard deviation has large and abrupt changes.

To perform this, we construct two hypothesis. The null hypothesis, $H0$, says that no change has occurred in our data and the standard deviation remains constant within the entire data set. The alternate hypothesis, $H1$, states that there is at least one point in our data set where the standard deviation has changed.

The log-likelihood ratio, given our two hypothesis, thus becomes:-

$$ln(\Lambda_1^n) = l_1^n = \sum_1^n ln(P(y_i \mid H_{1,i})) - \sum_1^n ln(P(y_i \mid H_{0,i})) \tag{6}$$

[38]

where y_i is the dataset indexed from 1 to n. Now assuming that there is one point in our dataset where its standard deviation changes, and our aim to find this point in time, $t0$.

In case of our null hypothesis, the standard deviation remains constant throughout the entire data set, we call this $\tilde{\sigma}$. In case of our alternate hypothesis, the standard deviation is σ_0 before $t0$, and after $t0$, it become σ_1.

Mathematically the log likelihood ratio then becomes:

$$l_1^n = \sum_1^{t_0-1} ln(P(y_i \mid \sigma_0)) + \sum_{t_0}^n ln(P(y_i \mid \sigma_1)) - \sum_1^n ln(P(y_i \mid \tilde{\sigma})) \tag{7}$$

Our aim then, is to maximize this log likelihood and that would give us the time instance where the standard deviation changes the most in our data set.

Of course, in reality, we would not have a single change in our entire dataset, but there would be multiple activities being performed, spread of time, so we need to recursively keep looking for significant changes in standard deviation on both the left and the right of the index where we found the initial change.

By setting a reasonable threshold on the log likelihood ratio, we can tune the sensitivity of our algorithm to find only the most interesting points where the standard deviation changes most significantly. We call these as the window boundaries and between every two boundaries, we assume one independent activity is being performed.

1) Real-time windowing vs. offline windowing: For the purposes of this research, we first collect data from a subject and then post process the data at a later point in time. This allows us to perform windowing in an "off-line" mode where all the data is already available to us. However in a production deployment, data would be coming in as a real time stream and would need to be analysed "on-line". Our windowing algorithm explained above can be applied for on-line change detection as explained in [38]. The windowing algorithm would then behave like a Shewhart control chart where new arriving samples would be compared with previously received samples to see if the signal's log-likelihood ratio increases beyond the threshold value, then we mark a window boundary. The output of marked boundaries after applying impulsive windowing is shown in Figure 9a.

In order to detect multiple window boundaries in our sampled series, we adopt the well known binary segmentation approach [37] to apply the log-likelihood method we outlined above, recursively, to find not just one point where the standard deviation changes the most, but multiple points in the data series where standard deviation changes significantly. In the binary segmentation method, we first apply the log-likelihood method to find the first boundary. If such a boundary is not found, we deem that the whole data series is a single window and stop windowing here. If a boundary is identified, it divides the data series into two windows containing the series before and after the boundary. We then apply the boundary detection method on both these windows. This method continues dividing windows into smaller windows until no further boundaries can be found.

F. Subcarrier Sanitization

We observe in our data that all subcarriers have a fixed phase offset with each other under static conditions. However, sometimes one or a couple of subcarriers' phase offset from other subcarriers changes randomly. We call such subcarriers as outliers. In order to avoid distortion of results due to

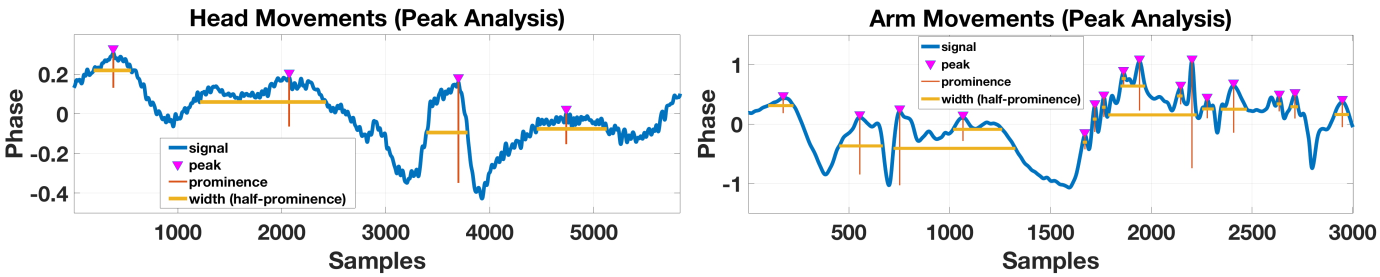


Fig. 10: Peak Analysis. Number of peaks, peak width, peak height in head (left) and Arm (right) movement.

such outliers, we filter them out during the Subcarrier Sanitization phase. This is achieved by keeping the subcarriers with maximum correlation, and discarding the ones that have less correlation with other subcarriers. Figure 9b shows the 2 outlier subcarriers with significantly higher phase difference than rest of the subcarriers, we must eliminate this in order to keep consistency in our results. After our sanitizing step, we obtain most correlated subcarriers, as shown in Figure 9c.

G. Dimensionality Reduction

In order to perform the peak analysis in our feature computation phase, we need to compress the information contained in all the subcarriers. The compressed version must be the best representation of activity information preserved in the signal. The body movements create a correlation effect among the CSI subcarriers [28]. Principal Component Analysis (PCA) is a natural choice to achieve this. We select the second principle component that contains consistent phase variations caused by human movements. The choice of second component is based on observation that 1st component preserves more noise than the other components. This is also conformed by [28].

H. Feature Computation

We identify unique features based on analysing the peak properties and subcarrier behaviour. Both of these steps are described below.

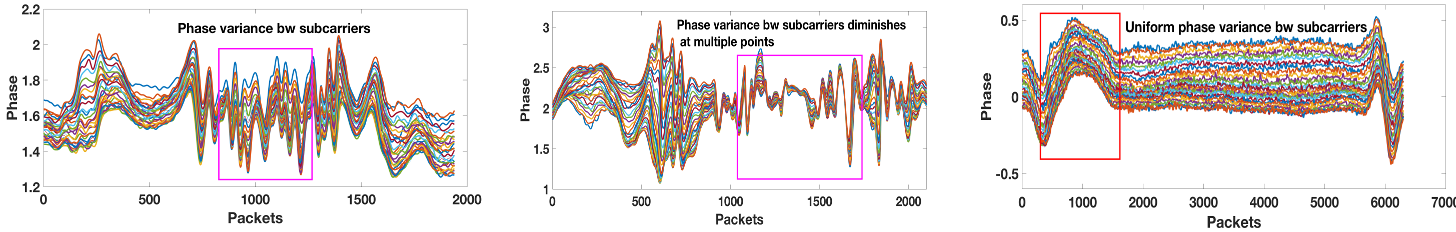
1) Peak Analysis: Peak analysis is a widely used technique in signal processing domain to identify patterns in signals. We identify peaks in our time domain signal by taking its first derivative and finding time instants where it becomes zero. These time instants are either the local maxima or minima of the signal. By taking the second derivative and observing whether it yields a positive or negative value at the points we identified earlier, we can identify maxima and minima accurately. We refer to these maxima and minima as positive and negative peaks in our signal. Setting first derivative of signal with respect to time equal to zero: $d(x(t))/d(t) = 0$ gives us values of t at which $x(t)$ has a positive or negative peak. Figure 10 shows the peak analysis of head movement (on left) and arm movement (on right). Using features such as number of peaks, peak width, number of inverted peaks and peak height, we can clearly distinguish between head and arm. Number of peaks is much greater for single arm movement than for a head movement. Similarly peak width is greater in head movement and height is smaller in comparison to arm movement. One reason is the range and impact of arm movement is greater in comparison to head movement.

2) Subcarrier Analysis: While peak analysis contributes towards distinguishing between head and arm movements, we observe another striking property among the subcarriers that help identify head and arm movements. The variance of phase difference between the subcarriers is widespread when head movement occurs, while it significantly drops down when an arm movement happens. In simple words, this can be described a separation between the subcarriers, how close the subcarriers squeeze down at closest point in a window and how many times they come very close to each other.

Algorithm 2: Subcarrier Analysis

```
function performSubcarrierAnalysis(sanitizedWindow);
Input  : window with n samples W (w_1, w_2, ..., w_n)
Output: mean of variances at each point μ_{σ^2}
Each sample in wi has m subcarriers S : (s_1, .., s_m);
initialize variance list, V = [];
for W : w_1, w_2, ..., w_n do
    compute variance of all subcarriers in sample, σ^2;
    append σ^2 to V;
end
compute mean of all variances in V, μ_{σ^2} = Σ(V)/n;
return μ_{σ^2};
```

This may be attributed to the observation that quick arm movements affect some subcarriers' phase more than others. This causes the relative phase offset between the subcarriers to diminish and they appear more close to each other on the phase axis. This is shown in Figures 11a and 11b. Push gesture can be visualised as hand approaching towards receiver to the point where phase difference disappears and all subcarriers merge to 1, and then hand goes away from the receiver. In swipe, hand slides from left to right in front of receiver, so the phase difference is generally similar to push, but the number of points where it disappears is either less or not present at all. This helps distinguish between push and swipe. On the other hand, slow head movements do not have the same varying affect on subcarriers and thus, during head movements, all subcarriers' phases are affected equally and they maintain their relative phase offset with each other. This can be seen in Figure 11c.



(a) Swipe Gesture. (b) Push Gesture. (c) Head turn, pause, turn back.

Fig. 11: Subcarrier Analysis of head turns vs hand gestures.

I. *Classification Model*

The impulsive windowing and critical choice of features makes the classification model learning a simple process. After feature vector composition and data separation into test and train, we use K-Nearest neighbour, with K=6 neighbours for learning. The train and test sample data is randomly distributed from different subjects to include all possible movements for training. Performing 10-fold cross-validation and separated test samples gives us overall accuracy of 94.5% (cf. Figure 12a) for separating arm movements from head movements and no activity. Model with same characteristics gives overall accuracy of 90.5% for separating push, swipe, random simultaneous (head turns and arm movements), significant head turns and no activity. (cf. Figure 12b). The training speed of the system is approx. 1200 obs/sec while the training time is 1.5799 sec. Overall, the drop in accuracy occurs due to the head and arm activity happening simultaneously, reason being similar to both head and arm movements and dependency on the type of movement with bigger impact.

Overall, the end-to-end latency in the Wibot edge based detection includes data collection, processing, feature computation, classification and feedback. We analyzed the data in Matlab installed on MacBook Pro with 2,7 GHz Intel Core i5 Processor and 8 GB 1867 MHz DDR3 RAM. For 1000 samples, the phase correction consumes 0.65 secs, signal preprocessing takes 0.05 sec, feature computation 0.525 sec and classification takes 0.16 secs (cf. Figure 13). The most time consuming steps include phase correction and feature computation (which involves PCA). The overall latency is till acceptable since the feedback is not required for every second in case of behaviour detection. In near future, we tend to reduce the overall latency by optimising the feature computation complexity, by using an alternate method for PCA.

VII. Challenges

While the focus of this research is to explore behaviour from natural body movements, there are several aspects which are yet to be covered. The critical ones are briefly described as follows:

A. *Multiple-Passengers in Car*

This research at the moment is tested and applied for a single passenger/driver in car. The behaviour of driver is most critical in both autonomous and non-autonomous driving. However, based on the needs, it can be extended to behaviour detection of all passengers in cars. This will help, specially in car sharing environments where behaviour of one passenger can affect others. Also, it can provide feedback for improving services needed by passengers. This potentially can be done with dedicated receiver antennas for each passenger. However, the challenges of interferences due to movements of different passenger, positioning of transmitter and receiver is yet to be explored.

B. *Autonomous vs. Non-Autonomous Challenges*

Behaviour detection of a driver is important for car manufactures in both autonomous and non-autonomous driving. In former, it gives attention information about the driver. And in later, it gives valuable feedback of general behaviour of a driver to improve the comfort services in car. The major challenge that we face in non-autonomous driving is the steering movements. They have extremely similar features as head movements, and they are the reason of accuracy drop. We envision that steering movements can be detected using a steering integration accelerometer. This information abut steering movements could be feedback to our behaviour detection system. The steering movements could then be eliminated which would help improve the accuracy of the system.

C. *Hardware Dependency*

Currently our hardware prototype is based on dedicated WLAN cards with modified Linux drivers available to capture the CSI data. However, the LAN cards and Laptops compatible with these LAN cards are about a decade old and not anymore available in the market. Naturally, for implementation of this technology in cars, the Wi-Fi router installed need to provide the CSI data and configuration controls. However, the increasing popularity of Wi-Fi as sensing technology will soon create a much needed demand to W-Fi device manufacturers to reveal CSI information.

VIII. Applications

As the domain of this research is car specific, its applications for learning behaviour extend from non-autonmous to autonomous and car sharing modes. Synchronising the information regarding the road conditions and corresponding head movements can determine the state of the driver. Unusual, frequent movements can tell that driver is either lost, confused or distracted. On the other hand, significant head turning and

True Class / Predicted Class

	No Activity	Arm Movement	Head Movement	True Positive Rate	False Negative Rate
No Activity	92%		8%	92%	8%
Arm Movement		89%	11%	89%	11%
Head Movement		2%	98%	98%	2%

(a) Confusion matrix for classifying between head, arm movements and no activity. Accuracy 94.5%, Precision= 93%, Recall= 95.6% and F1 score= 0.94.

True Class / Predicted Class

	Push Gesture	No Activity	Head+Arm Movement	Swipe	Head Turns	True Positive Rate	False Negative Rate
Push	89%			11%		89%	11%
No Activity		100%				100%	
Head+Arm			76%		24%	76%	24%
Swipe			9%	91%		91%	9%
Head Turns			7%		93%	93%	7%

(b) Confusion matrix for classifying between Push, no activity, random head+arm movement, swipe and head turns. Accuracy 90.5%, Precision= 90.4%, Recall= 89.4% and F1 score= 0.91.

Fig. 12: Confusion matrices presenting the accuracy of movements, TPR and FNR.

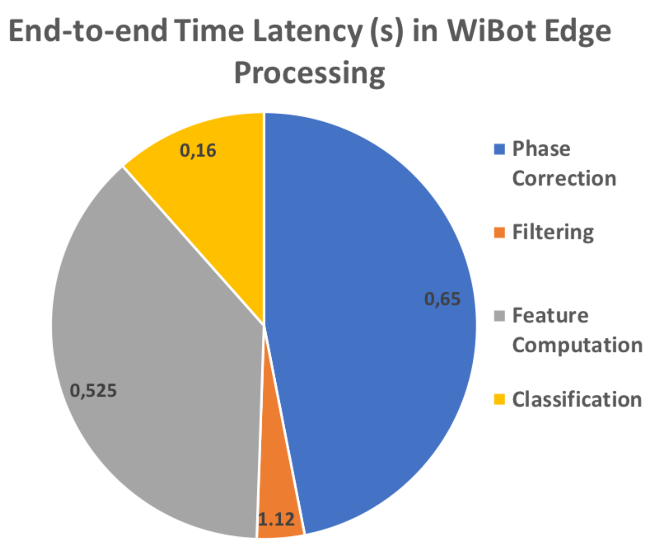


Fig. 13: WiBot end to end latency distribution over the prominent steps. Relative Phase and feature computation being major contributors.

focussing on some interest point in that location (say restaurants, scenery) can provide information about the interests of the driver. This kind of information can be helpful in providing valuable information and services to the user. Gestures on the other hand are additional feature for communicating with car. It gives flexibility to people who do not prefer talking to car assistant systems which are based on speech recognition. Furthermore, people with speech disabilities can also avail the facility of car assistant system.

IX. Conclusion

In this paper, we make the following contributions in the domain of Wi-Fi activity sensing. We propose WiBot, a Wireless network-edge based personal communication and behaviour learning system for cars. This research is done in cooperation with and in BMW Group Research, New Technologies and Innovations, Germany. WiBot enables communication with the driver by using two gestures 'push' and 'swipe' which translate to 'yes' and 'no' respectively. Secondly, it characterises human behaviour by detecting and analysing head turns and arm movements. This gesture and behaviour recognition is done by taking into account all possible natural movements (upper body) that a driver does during the drive. The data is collected in a distraction induced human study of 40 participants. The study is non-choreographed, which means that the subjects are not instructed to perform any particular activities or behave in certain way. We focus on developing a system which accounts for natural behaviour of drivers. Therefore, unlike other studies we do not pre-define a set of classes to distinguish from each other in our classification learning. In order to identify the start and end points of activities in natural environment, we introduce an impulsive windowing algorithm and also account for multiple activities happening at the same time, using label powerset method. We recognise the distracted behaviour by identifying slight head movements to significant head turns, slight arm movements to particular arm activity. In order to classify extremely similar movements performed at similar frequencies, we find a unique feature set based on peak analysis and subcarrier analysis. Our features are based on phase information collected from CSI data captured using the WLAN card with modified driver. Our K-Nearest neighbour based classification model can separate the head movements from arm movements with accuracy of 94.5%. The accuracy for elaborate label subset; push, wipe, no activity, simultaneous head a and arm movements and significant head turns can be separated with accuracy of 90.5%.

X. Future work

The human study carried out at BMW Group Research, New Technologies and Innovations centre is composed of 3 stages. It starts from normal driving to distracted driving to happy/relaxed driving. In this paper we cover only the distracted driving behaviour. We are further extending WiBot to recognise other stages of emotions and intend to distinguish between different behaviours. We will also combine body movement information with heart and breathing rate to get fine grained details. The CSI dataset will be available for the research community to further benefit from it.

ACKNOWLEDGMENT

The authors would like to acknowledge the valuable contribution by Syed Safi Ali Shah in this work.

REFERENCES

[1] M. G. Jæger, M. B. Skov, N. G. Thomassen *et al.*, "You can touch, but you can't look: interacting with in-vehicle systems," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2008, pp. 1139–1148.

[2] J. Santos, N. Merat, S. Mouta, K. Brookhuis, and D. De Waard, "The interaction between driving and in-vehicle information systems: Comparison of results from laboratory, simulator and real-world studies," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 8, no. 2, pp. 135–146, 2005.

[3] A. Várhelyi, M. Hjälmdahl, C. Hydén, and M. Draskóczy, "Effects of an active accelerator pedal on driver behaviour and traffic safety after long-term use in urban areas," *Accident Analysis & Prevention*, vol. 36, no. 5, pp. 729–737, 2004.

[4] H. Richter, R. Ecker, C. Deisler, and A. Butz, "Haptouch and the 2+ 1 state model: potentials of haptic feedback on touch based in-vehicle information systems," in *Proceedings of the 2nd international conference on automotive user interfaces and interactive vehicular applications*. ACM, 2010, pp. 72–79.

[5] D. L. Strayer, J. Turrill, J. R. Coleman, E. V. Ortiz, and J. M. Cooper, "Measuring cognitive distraction in the automobile ii: Assessing in-vehicle voice-based," *Accident Analysis & Prevention*, vol. 372, p. 379, 2014.

[6] J. He, A. Chaparro, B. Nguyen, R. J. Burge, J. Crandall, B. Chaparro, R. Ni, and S. Cao, "Texting while driving: Is speech-based text entry less risky than handheld text entry?" *Accident Analysis & Prevention*, vol. 72, pp. 287–295, 2014.

[7] J. L. Wilson, "Automotive wifi availability in dynamic urban canyon environments," *Navigation*, vol. 63, no. 2, pp. 161–172, 2016.

[8] L. Fletcher and A. Zelinsky, "Driver inattention detection based on eye gaze—road event correlation," *The International Journal of Robotics Research*, no. 6, pp. 774–801.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, *Real-Time Human Pose Recognition in Parts from Single Depth Images*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 119–135.

[10] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, I. Oikonomidis, and P. Olivier, "Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '12, 2012, pp. 167–176.

[11] G. Cohn, D. Morris, S. Patel, and D. Tan, "Humantenna: Using the body as an antenna for real-time whole-body interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12, 2012, pp. 1901–1910.

[12] Z. P. Bian, J. Hou, L. P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 2, pp. 430–439, March 2015.

[13] E. E. Stone and M. Skubic, "Fall detection in homes of older adults using the microsoft kinect," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 1, pp. 290–301, Jan 2015.

[14] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A Ubiquitous WiFi-based Gesture Recognition System," *ArXiv e-prints*, Jan. 2015.

[15] F. Adib and D. Katabi, "See through walls with wifi!" *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 75–86, aug 2013.

[16] T. Gu, Z. Wu, X. Tao, H. K. Pung, and J. Lu, "epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition," in *2009 IEEE International Conference on Pervasive Computing and Communications*, March 2009, pp. 1–9.

[17] M. Moussa and M. Youssef, "Smart cevices for smart environments: Device-free passive detection in real environments," in *2009 IEEE International Conference on Pervasive Computing and Communications*, March 2009, pp. 1–6.

[18] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '13. New York, NY, USA: ACM, 2013, pp. 27–38.

[19] S. Sigg, U. Blanke, and G. Troster, "The telepathic phone: Frictionless activity recognition from wifi-rssi," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, March 2014, pp. 148–155.

[20] K. Qian, C. Wu, Z. Zhou, Y. Zheng, Z. Yang, and Y. Liu, "Inferring motion direction using commodity wi-fi for interactive exergames," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, 2017, pp. 1961–1972.

[21] D. Huang, R. Nandakumar, and S. Gollakota, "Feasibility and limits of wi-fi imaging," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '14, 2014, pp. 266–279.

[22] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. Seattle, WA: USENIX Association, Apr. 2014, pp. 303–316.

[23] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via RF body reflections," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, Oakland, CA, 2015, pp. 279–292.

[24] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'14, 2014, pp. 317–329.

[25] P. V. Dorp and F. C. A. Groen, "Feature-based human motion parameter estimation with radar," *IET Radar, Sonar Navigation*, vol. 2, no. 2, pp. 135–145, April 2008.

[26] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15, 2015, pp. 90–102.

[27] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: ACM, 2014, pp. 617–628.

[28] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial wifi devices," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1118–1131, May 2017.

[29] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, Feb 2017.

[30] M. Kotaru, K. R. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM 2015, London, United Kingdom, August 17-21, 2015*, 2015, pp. 269–282.

[31] W. Xi, J. Zhao, X. Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using wifi," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 361–369.

[32] X. Zheng, J. Wang, L. Shangguan, Z. Zhou, and Y. Liu, "Smokey: Ubiquitous smoking detection with commercial wifi infrastructures," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[33] Y. Zhuo, H. Zhu, and H. Xue, "Identifying a new non-linear csi phase measurement error with commodity wifi devices," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, Dec 2016, pp. 72–79.

[34] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, vol. 41, no. 1, p. 53, Jan. 2011.

[35] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge University Press, 2005.

[36] J. Liono, A. K. Qin, and F. D. Salim, "Optimal time window for temporal segmentation of sensor streams in multi-activity recognition," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. MOBIQUITOUS 2016, 2016, pp. 10–19.

[37] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, no. 500, pp. 1590–1598.

[38] M. Basseville, *On -line detection of jumps in mean*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 9–26.