FinDroidHR: Smartwatch Gesture Input with Optical Heartrate Monitor

YU ZHANG, RMIT University and CSIRO's Data61
TAO GU, RMIT University
CHU LUO, The University of Melbourne
VASSILIS KOSTAKOS, The University of Melbourne
ARUNA SENEVIRATNE, The University of New South Wales and CSIRO's Data61

We present FinDroidHR, a novel gesture input technique for off-the-shelf smartwatches. Our technique is designed to detect 10 hand gestures on the hand wearing a smartwatch. The technique is enabled by analysing features of the Photoplethysmography (PPG) signal that optical heart-rate sensors capture. In a study with 20 participants, we show that FinDroidHR achieves 90.55% accuracy and 90.73% recall. Our work is the first study to explore the feasibility of using optical sensors on the off-the-shelf wearable devices to recognise gestures. Without requiring bespoke hardware, FinDroidHR can be readily used on existing smartwatches.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; **Interaction design**; *Mobile computing*; • **Computing methodologies → Machine learning**; • **Mathematics of computing** → *Mathematical analysis*;

Additional Key Words and Phrases: Wearable Computing, Gesture Interaction, Machine Learning, Mobile Sensors

ACM Reference Format:
Yu Zhang, Tao Gu, Chu Luo, Vassilis Kostakos, and Aruna Seneviratne. 2018. FinDroidHR: Smartwatch Gesture Input with Optical Heartrate Monitor. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 56 (March 2018), 42 pages.
https://doi.org/10.1145/3191788

1 INTRODUCTION

Smart wearable devices are becoming increasingly popular and leading to new ways of interaction in our daily life. Recently, smartwatches and smart fitness bands have proliferated and received wide adoption, such as the Apple Watch, Android Wear and Fitbit Band. However, interaction with these wearable devices is still limited by physical buttons and screen-touching technologies. Since the screens of smartwatches are significantly smaller than smartphones, it remains inconvenient and inefficient to interact with smartwatches through touch alone.

This work is supported by the Australian Research Council (ARC) Discovery Grant, DP180103932.

Author's address: Y. Zhang; T. Gu, School of Science, RMIT University, VIC 3000 Melbourne, Australia, E-mail: zac.lhjzyzzoo@gmail.com; tao.gu@rmit.edu.au; C. Luo; V. Kostakos, School of Computing and Information Systems, University of Melbourne, VIC 3010 Parkville, Australia, E-mail: CHUL3@student.unimelb.edu.au; vassilis.kostakos@unimelb.edu.au; A. Seneviratne, CSIRO's Data61, Locked Bag 9013 Alexandria NSW 1435 Australia, E-mail: Aruna.Seneviratne@data61.csiro.au.
Authors' addresses: Yu Zhang, RMIT University and CSIRO's Data61, zac.lhjzyzzoo@gmail.com; Tao Gu, RMIT University, tao.gu@rmit.edu.au; Chu Luo, The University of Melbourne, CHUL3@student.unimelb.edu.au; Vassilis Kostakos, The University of Melbourne, vassilis.kostakos@unimelb.edu.au; Aruna Seneviratne, The University of New South Wales and CSIRO's Data61, Aruna.Seneviratne@data61.csiro.au.

2474-9567/2018/3-ART56 $15.00
https://doi.org/10.1145/3191788

Moreover, smartwatch touch interaction requires using both hands, and by definition obscures the already small screen. Therefore, hand-free or other novel interaction solutions have became a key requirement in daily life.

While a large body of previous work [12] [10] [36] have focused on novel ways of interacting with smartwatches, they require bespoke or additional hardware which may not be readily available. Therefore, it can be quite beneficial to develop smartwatch interaction techniques that solely rely on the off-the-shelf built-in hardware/sensors, including the rich set of biosensors aimed at healthcare purposes [9]. On the other hand, a few of prior works [45] use the inertial measurement unit (IMU) (accelerometers and gyroscopes) built-in wearable devices to recognise basic hand gestures as input with a fairly appropriate accuracy. However, as the IMU sensors are usually sensitive to motions and environmental effects [29], such as ambient motion, altitude, and even Earth's gravity, they may be limited to perform in some certain cases. Hence, is there any other proper way to enable hand-free interactions on the off-the-shelf wearable devices for more use cases?

In this paper, we present FinDroidHR, a novel way of interacting with smartwatches by investigating biosensors on the off-the-shelf smartwatches. FinDroidHR relies on optical heart rate monitors that are largely and increasingly embedded in off-the-shelf smartwatches, enabling users to perform various gestures with the hand wearing smartwatch. Our work extends a large body of previous work that aims to develop a hands-free user gesture input technique on smartwatches.

Conceptually, our technique relies on what Abowd has termed "your noise is my signal" [1]. Specifically, the heart rate monitors on the back side of smartwatches (i.e. facing the users' skin) use Photoplethysmography, a process that measures blood volume changes in the microvascular bed of users' tissue. Previous work has shown that Photoplethysmography is susceptible to noise caused by "motion artefacts" or hand gestures in general [28]. This has meant that hand and finger motions affect the volume of blood in the users' tissue, introducing noise to the heart rate measurements.

In summary, our work makes the following contributions:

- We demonstrate how the noise captured by Photoplethysmography (PPG-based) heart rate monitors can in fact be mapped to specific hand gestures that users perform. As such, we have developed a novel mechanism called the Gradient-based Movement Detection (GMD) Unit for detecting hand gestures using the heart rate monitor on commodity smartwatches.
- To quantify all practically potential hand gestures for PPG-based wearable devices, we conduct a sufficient gesture study based on all participants' feedback. A proposed set of 26 gestures is identified after the analysis of the study. From these, we have finally identified a conflict-free set of 10 gestures for our recognition purpose.
- We propose an effective set of PPG features for our interaction recognition purpose. We first use brute-force strategy to explore PPG signal features from time-frequency domain. We then perform 3 well-developed feature selection algorithms combined with 2 widely used classifiers to cross validate the selected feature set.
- We evaluate our system in both stationary study and motion study with 20 participants. Besides, we implement an IMU-based alternative system to compare our system side by side. The results highlight the strengths of our PPG-based technique over the IMU-based solution in power consumption and stability aspects.

2 RELATED WORK

2.1 Enhanced Smartwatch Interaction

Substantial previous work has aimed to extend or augment the standard touchscreen-based interaction capability of smartwatches. WatchIt [33] enhanced watchbands with input gestures including slider controls and touch input. This mechanism relies on cheap, low-power, 1-D position sensors as additional hardware under the watchband,

and enables interaction in eyes-free contexts. Similarly, Baudisch et al. [8] proposed embedding a touch pad onto the rear of very small devices, such as the bottom surface of a watchband. When pressure is applied, the ring of elastic silicone inside the touch pad can create a conductive path to send the activation signal. With this touch pad, back-of-device interaction can be implemented independent of the device size, thus overcoming the inherently limited potential for touch-based interaction on small screens.

For smartwatches with round touchscreen, Ashbrook et al. [4] designed wedge-shaped touchable buttons around the touchscreen perimeter. Driven by electronics in the bezel, the extra buttons provide user shortcuts. They found that the error rate of input is affected by the angular and radial width of buttons. Similarly, Xiao et al. [46] transformed the watch face to a mechanical interface which allows coarse physical manipulations, including twisting and tilting. Mimicking the flexibility of various possible physical controls, this transformation introduced a new modality for smartwatch interaction. Exploiting finger areas on the hand wearing a smartwatch, DigitSpace [18] created a thumb-to-fingers interface as a touch pad. This interface allows keyboard-like text input and 2-D gestures.

In our own work, FinDroidHR aims to enhance smartwatches interaction beyond the touch screen, but also provides a novel and practical modality for one-handed and eyes-free interaction.

2.2 Using the Human Body as an Input Surface

Increasingly, research has focused on new interaction techniques that use the human body as the medium. For instance, SkinTrack [51] allows wearable devices to continuously track touch on human skin. Based on a ring emitting AC signals and a wristband with multiple electrodes, this system can compute 2D finger touch traces by detecting phase differences of the signals. With sufficient training data, this system is able to identifytouch events at 99% accuracy and a target rework of 7.6mm on average.

The iSkin [43] system introduced a new type of skin-worn sensors to track touch events on human body. The sensors are bio-compatible, stretchable "stickers" impregnated with conductive traces, which can capacitively sense finger touches. The evaluation results show that the sensors are able to bend (within radii of 5mm) and stretch (within 30%) with moving the skin. Similarly, SkinMarks [44] also enable on-body interactions using "stickers" made of sub-millimetre electrodes. SkinMarks can precisely localise input on fine body landmarks from gestures such as touch, squeeze and bend. In addition, it can output visual lights on these body landmarks.

A limitation of these methods is that users must wear extra input/output modules beyond the intended device on body. Comparatively, FinDroidHR only uses the PPG sensor embedded in smartwatches, measuring the blood volume changes inside skin tissue to support hand and finger gesture recognition.

2.3 Mobile Sensor Gesture Recognition

Substantial work has focused on sensor-based gesture recognitions. Saponas et al. [37] presented a system for directly sensing muscle activity by EMG sensors to detect delicate gestures. Lu et al. [27] used an accelerometer sensor for detecting larger-scale gestures and a SEMG for small-scale gestures to improve recognition accuracy. Alternatively, WristFlex [14] relied on on force sensitive resistors (FSR) to detect tendon movement. Beyond EMG-based approaches, an advantage of this approach is low power consumption, whereas the building prototype time and cost are high for integrating with existing wearable devices.

In short, existing approaches require extra hardware support, which increase the cost and reduce the wearability of detecting hand and finger-based gestures. Furthermore, many such sensors are not intended to be worn routinely and longitudinally. Comparatively, FinDroidHR works with off-the-shelf smartwatches or fitness bands, and can be comfortably worn longitudinally, even in one's sleep. Similarly, Serendipity [45] employed IMU sensors on smartwatches to classify hand gestures for interaction purpose. However, it is lack of a sufficient gesture study and only 5 gestures are proposed for the system. By contrast, we tailored a hand gesture set for our system based on a full gesture study.

2.4 Near Distance Gesture Interaction

An important body of work relevant to our own has considered methods to enable interaction between humans and devices at a short distance. For instance, Fingerpad [10] used a nail-mounted set of Hall sensor and reader to capture 2D gestures from an index finger and thumb on the same hand. Since each posture has a different magnetic field strength, this system is able to achieve 92-93% accuracy for a moving or stationary user, enough to support general input tasks on mobile devices. Similarly, PocketTouch [36] is able to recognise finger-strokes through fabric, such as a pocket of pants. This approach relies on a customised multi-touch capacitive sensing pad, and allows users to interact with a mobile device using stroke-based gesture input and text entry.

Beyond 2D gestures, Chen et al. [12] proposed uTrack which instruments the back of the thumb as well as the back of fingers with magnetometers for enabling 3D input. To localise the 3D position of the thumb-mounted magnet, they designed a searching algorithm with a positioning accuracy of 4.82 mm and 303 dpci (dots per cube inch). The results of a user study showed that users can effectively complete several pointing tasks using uTrack.

Conceptually, FinDroidHR also enables such near-distance interaction: finger and hand gestures can be recognised as input on a smartwatch without extra sensors.

2.5 Acoustic-driven Gesture Interaction

Several approaches have enhanced mobile interaction using acoustic-based localisation. For example, FingerIO [32] configures a device to become an active sonar that emits inaudible sound signals and recognises the echoes of fingers near the microphones. SoundWave [17] leverages Doppler shifts from acoustic transmissions to recognise gestures like moving the hand away from or towards the device. Jess et al. [30] propose a ultrasound-based system for imaging changes of inside mid-forearm when performing 10 discrete hand gestures, which achieves an accuracy above 98%. However, the system relies on a heavy ultrasound imaging device and can not be applied for daily use.

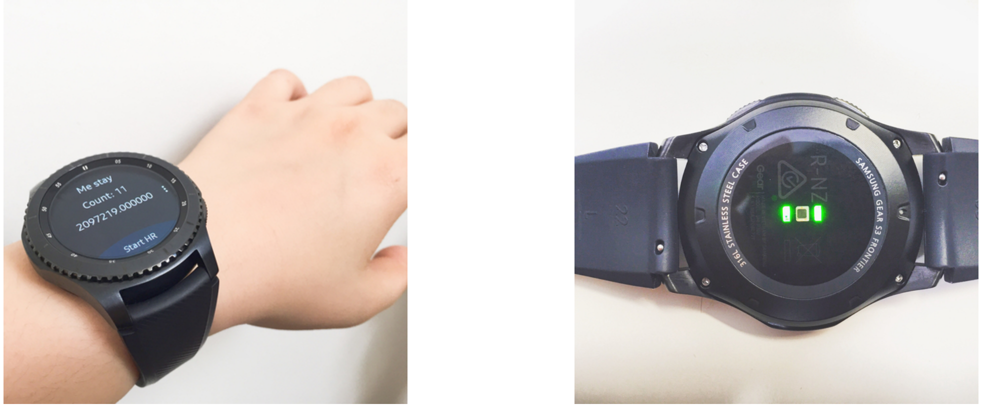
In contrast, FinDroidHR relies on light effects through human skin, rather than sound waves, for the recognition of centimetre-level finger gestures. Besides, it is mainly developed for actual daily use.

2.6 RF-based Gesture Recognition

Radio-frequency techniques have also been used to enable diverse gesture recognition systems. For instance, AllSee [19] is a low-power gesture-recognition system that can fetch the information of a gesture from current wireless signals (e.g., TV transmissions). It solely relies on RFID tags and power-harvesting sensors. In through-the-pocket circumstances, evaluation results indicate that AllSee can classify a set of eight gestures with a mean accuracy of 97%.

Beyond home scenarios, WiDraw [39] makes use of WiFi interference to enable gesture recognition without any wearable devices. To track the user's hand trajectory, WiDraw extracts the Angle-of-Arrival values of wireless signals received by a nearby mobile device. Based on commodity wireless cards, a prototype can track the user's hand with a median accuracy of 5 cm. Also, a prototype of a mid-air handwriting application using WiDraw can achieve a mean accuracy of 91% for word recognition tasks involving letters, words, and sentences.

However, only a limited number of off-the-shelf smartwatches have embedded Wi-Fi modules. Hence, FinDroidHR uses the heart rate monitoring sensor only, which is already available on most of existing smartwatches and fitness bands.



(a) PPG Signal Collection by FinDroidHR

(b) Heart rate monitor embedded in off-the-shelf smartwatch

Fig. 1. FinDroidHR uses optical heart rate monitors embedded in off-the-shelf smartwatches to allow the hand and fingers to be appropriated for gesture recognition via a heart rate monitor sensor.

2.7 Photoplethysmography's Potential Advantages

The traditional optical approach for interaction with wearable device mainly replies on Infrared (IR) proximity sensors to recognise gestures, coupled with computer vision. For example, HoverFlow [24] and Gesture Watch [21] develops their prototypes with multiple IR proximity sensors to recognise coarse hand gestures for around-device interaction. Using a customized wrist-worn device, Kei et al. [31] implement the idea of IR reflection to track finger position using the back-of-hand as an input surface. Furthermore, Skin Buttons [25] integrate tiny IR proximity sensors as projectors into the smartwatch to represent click icons on users' skin. However, they all require extra sensors and instrumentation.

From the set of biosensors available on commodity smartwatches, most promising for gesture input is Photoplethysmography (PPG), which is an optical technique for monitoring the volumetric changes in body [9]. PPG sensors on smartwatches use a low-cost LED light to illuminate the tissue, and Photodetectors to measure the variations of absorbed and reflected light by the blood flow [9]. A number of previous studies [34] [2] [41] have successfully employed the PPG techniques in the context of biosensing. Therefore, we hypothesise that the Photoplethysmography-based sensors on smartwatches can potentially be an appropriate way for sensing hand gestures.

3 FINDROIDHR METHODOLOGY AND SYSTEM DESIGN

We use an off-the-shelf Samsung Gear S3 frontier smartwatch to build our gesture recognition system, as shown in Figure 1. In a pilot study we choose to develop on the Android Platform as the accessibility and functionality of Android Wear OS is more comprehensive than the Tizen Wearable OS. However, we could not access the heart rate sample data by most of Android smartwatches, since both of the Andoird API and the driver of the kernel do not support the access of raw signal data.

FinDroidHR achieves centimetre-level finger-gesture recognition by using a green-light-based heart rate monitor on the smartwatch. Figure 2 demonstrates the system work flow of FinDroidHR. There are 4 major components: an off-line PPG data collection application, a Signal Processing unit, a Gradient-based Movement Detection (GMD) unit, and a Classification unit. Since PPG signals contains noise from the wrist, the challenge is

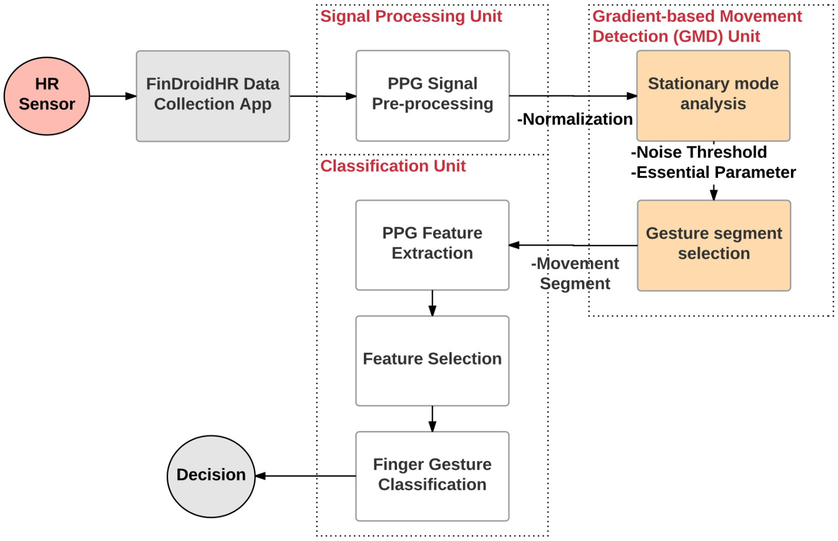


Fig. 2. FinDroidHR gesture recognition workflow.

to accurately identify the gesture movements for usable gesture recognition. We adopt a noise removal technique called GMD to overcome this challenge.

We first introduce our design principles of the gestures that our system can recognise. After given the set of recognised gestures, we describe the mechanisms of raw data collection. Next, we give the details of using GMD. Finally, we explore the selected feature set of PPG signals for improvements in gesture recognition.

3.1 Gesture Design

Our gesture design for smartwatch interaction relies on the wearable interaction design principle described in [38]. Since our technique is based on using the PPG biosensor embedded inside smartwatches to detect the blood volume changes of tissue, our design principle is an application of the fact that the hand gestures instantly make a change to the blood volume of hands. Besides, all proposed gestures should be designed as easy-to-use in daily life, because we make an assumption that people could interact with smartwatches by FinDroidHR naturally and friendly without any hesitations and confusions. Based on these 2 design ideas, we next select a suitable gesture set for our system.

In the prior work of wearable design [28] [48] [45] [20], 3 portions of arms on body are mainly involved: the forearm portion, the wrist portion and the finger portion, as shown in Figure 3a. For our practical purposes, the forearm portion should be stationary to provide a fixed height when users are looking at the watch. Therefore, the forearm portion is not considered in our watch gesture design. Thus, both the wrist portion and finger portion are adopted in our design.

There are 3 sub-portion areas which can generate a distinct hand movement leading to a fast change in the blood volume of hands. According to the construction of hands shown in Figure 3b, the wrist, phalanxes and fingers are able to create effective gestures. However, some of them may be difficult to perform in reality. For instance, when people perform a clicking movement by the middle finger, the ring finger will spontaneously move. Thus, we mark out those segments which represent as the hard-to-use areas in Figure 3b to contribute to our final gesture set selection.

Table 1. Designed Gesture Set for FinDroidHR Recognition.

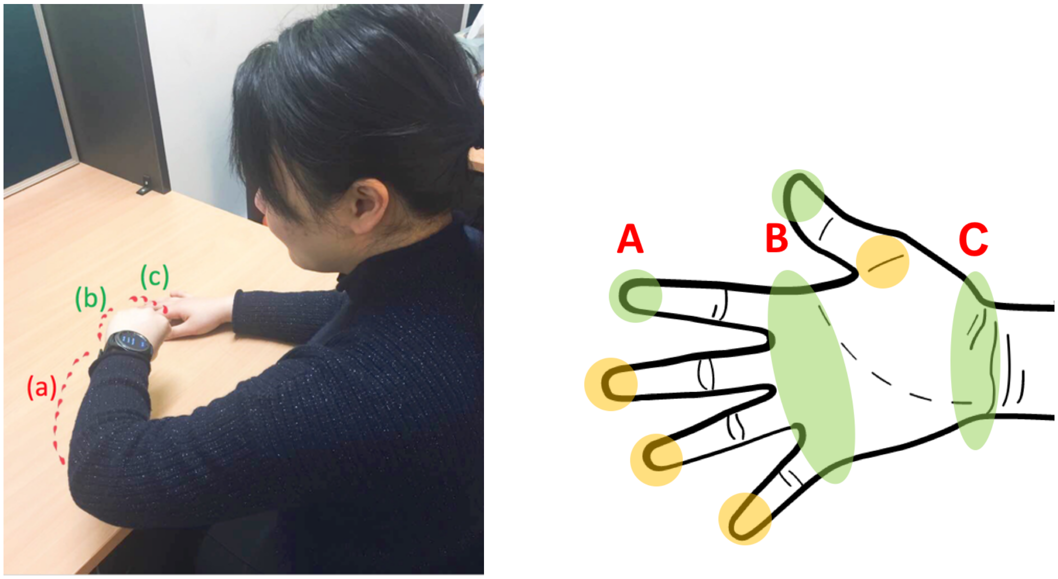
Proposed Hand Gesture Set			
Involved portion areas	Atomic gestures	Unique gestures	Easy-to-use gestures
Wrist (W) only	2	-	2
Phalanx (P) only	5	-	1
2 Fingers (F)	5	-	2
P and 2 F	4	3	4
P and 3 F	-	2	-
P and 4 F	2	-	2
P and 5 F	2	1	2
In total:	26		13

Table 2. FinDroidHR conflict-free gesture set.

Core Finger Gesture List			
No.	Short Name	Full Name	Description
1	clhand	Close Hand	Fully clench hand from relaxed mode ($P + 5F$)
2	downhand	Down Hand	Turn down hand from horizontal mode (W)
3	downrob	Down Rub Fingers	Using thumb rubs index from top to bottom ($P + 2F$)
4	index	Index Finger Flicking	Flicking index by thumb ($2F$)
5	midringclk	Middle and Ring Fingers Clicking	Using both of middle and ring clicking ($P + 2F$)
6	ophand	Open Hand	Fully open hand from relaxed mode ($P + 5F$)
7	pinch	Pinch Fingers	A squeeze with thumb and index ($P + 2F$)
8	pullhand	Pull 4 Fingers	Pull 4 Fingers back from open palm mode ($P + 4F$)
9	pushhand	Push 4 Fingers	Push out 4 Fingers from close palm mode ($P + 4F$)
10	uphand	Up Hand	Turn up hand from horizontal mode (W)

Theoretically, the potential number of hand gestures is close to infinite and tons of gestures can be combined by different fingers and arm angles. Thus, to reduce the complexity of hand gestures is a key process for actual gesture design. Prior work [20] explore a reduced conflict-free set of 43 gestures extracted from a video-taped study by participants, and all mentioned gestures are developed based on real daily use. Similarly, we also conduct a related study to collect participants' feedbacks for our gesture set design following our design principles. Table 1 indicates all gestures that we collect from participants' feedbacks. Overall, according to our design, there are 26 proposed gestures in total based on the actual experience of participants. After the gesture selection process, 13 out of 26 gestures are identified as easy-to-use gestures from participants' feedback.

Finally, we identify 10 out of 13 gestures as conflict-free, and the reason for conflict recognition will be discussed at the limitation section. Thus, we propose a set of 10 conflict-free gestures for smartwatch interaction based on the PPG techniques. These gestures are also related to the 4 proposed easy-to-use portion areas: wrist, proximal phalanxes, thumb and index finger in Figure 3b. Furthermore, Figure 4 shows all our gestures, with the detailed description for each gesture in Table 2.



(a) Up to 3 segments of users' arms that will be involved in the action of watching watches: (a) the forearm; (b) the wrist; (c) fingers.

(b) The deconstruction of whole palm for gesture identifications: there are 3 portion areas that can be performed as hand gestures, A - fingers, B - phalanxes, C - wrist (the green segments are represented as easy-to-use areas and the yellow segments are represented as hard-to-use areas).

Fig. 3. FinDroidHR gesture design.

3.2 On-line VS Off-line Training Data Collection

Since the heart rate monitor is embedded in the smartwatch, developing an application as a bridge between the sensor and users is essential for our work. We first attempt to develop the application via TCP protocol in on-line mode to directly send the PPG data to the host computer, as previous studies have done [48] [49] [34], and shown in Figure 5a.

However, after a few practical tests, we conclude that this approach is too cumbersome due to the need for data cables. Thus, we adopt an off-line data collection strategy whereby local-data access is used. This makes it more practical to collect data from many users. To transfer data from watch to our database, we use the Smart Development Bridge (SDB) powered by the Tizen via commands. Figure 5 shows our off-line data collection application.

According the Tizen Wearable API [15], the maximum frequency that the heart rate sensor offers is 100Hz, and the green light amplitude is in the range of [0, 1081216]. Finally, we use 100Hz PPG signal data with 200Hz sample rate according to the Nyquist Theorem, and these are also used for the frequency domain feature analysis.

3.2.1 Near object detection. During the actual data collection procedure, we find a couple of issues regarding the heart rate sensor of the watch. First, as shown in Figure 6a, once we activate the sensor, the data of the first 2 seconds exhibit strong fluctuation. We assume that the sensor may perform some initial bootstraping before actually returning actual readings. Thus, we discard the first 2 seconds of data.

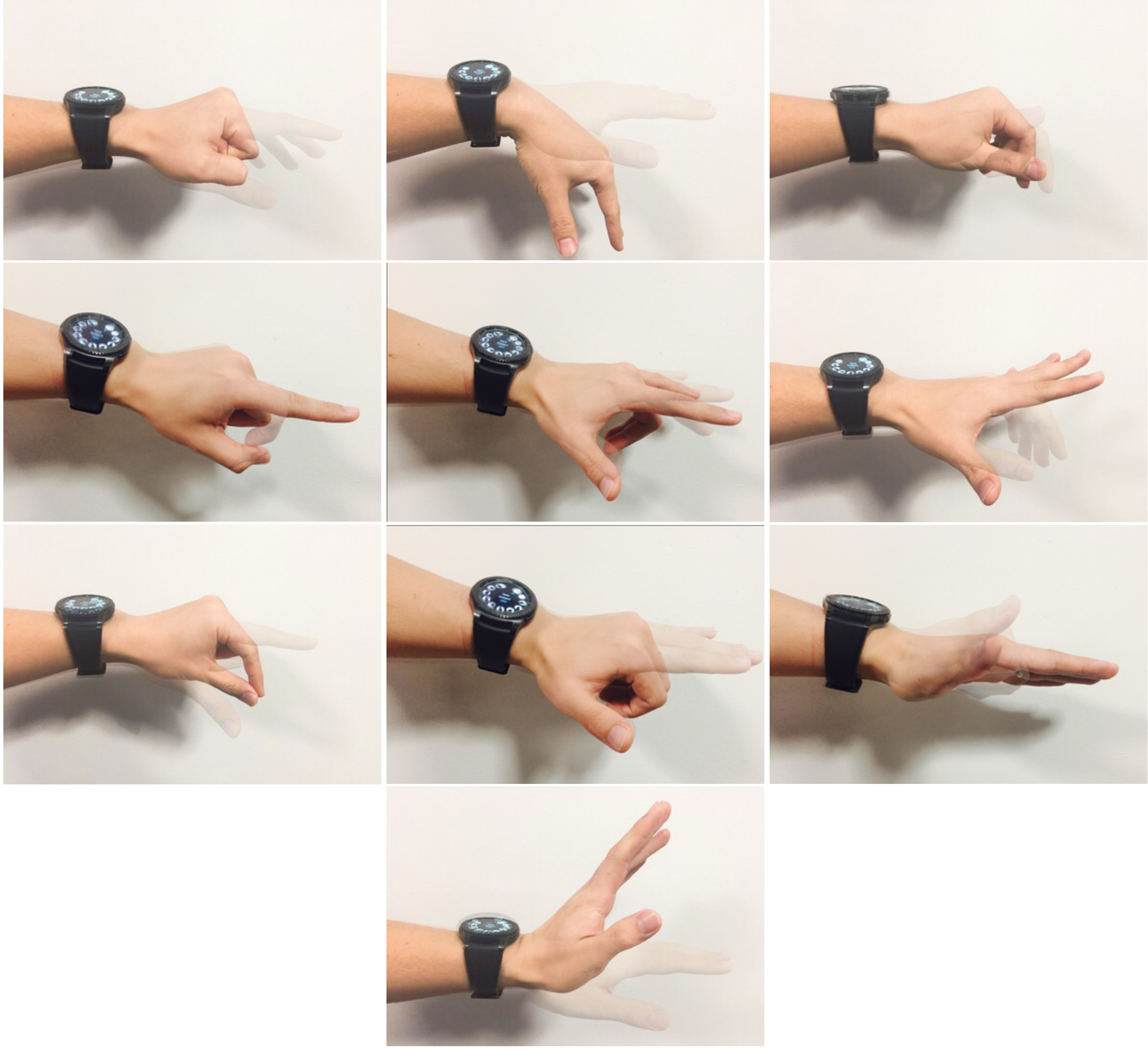
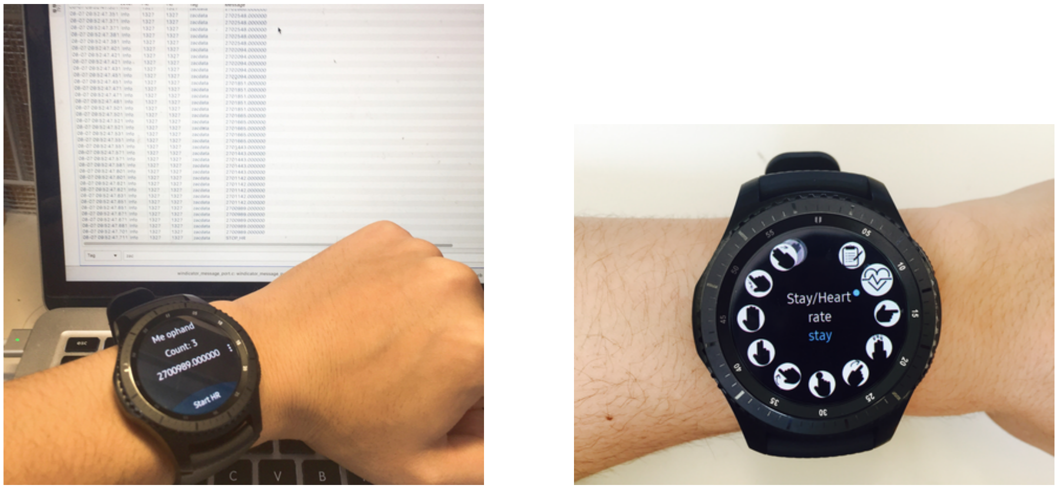


Fig. 4. The conflict-free gestures detected by our system (top-left to bottom-right): Close Hand, Down Hand, Down Rub Finger, Index Flicking, Middle and Ring Clicking, Open Hand, Pinch Fingers, Pull 4 Fingers, Push 4 Fingers, and Up Hand.

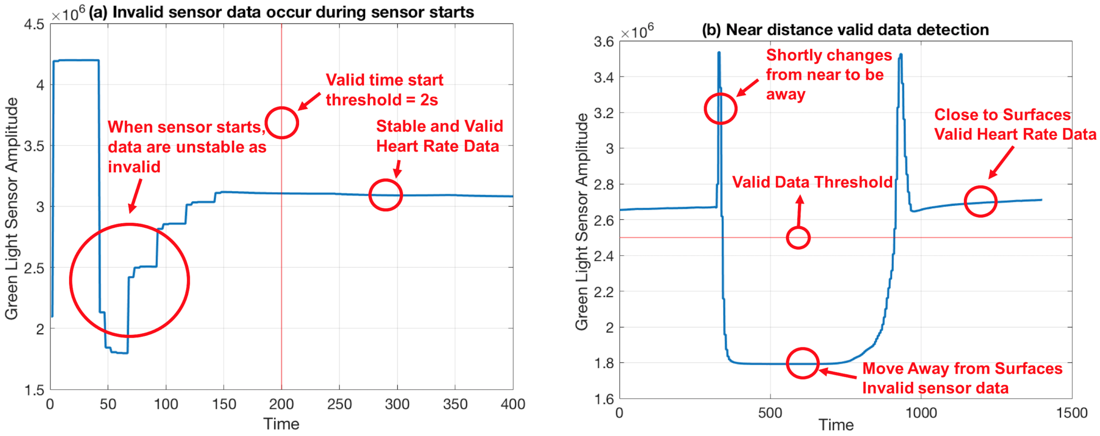
Second, the data validity of the heart rate sensor depends on how the green light is reflected from the user's skin. Figure 6b shows that the readings are affected by the distance between the sensor and skin. Once the sensor is close to the user's wrist skin, the data is valid without perceivable fluctuation. However, if the sensor is moved away from surface, the readings will effectively become invalid. Therefore, to ensure that our training data is valid, we incorporate a near object detection mechanism in our application. When the watch is moved away from the user's skin, the application will automatically ignore the readings.



(a) An on-line data application implementation, which requires connection to a host computer for receiving data in real-time via TCP.

(b) Implemented for off-line data collection, which offers a local data storage for efficiently collecting data.

Fig. 5. On-line vs. off-line FinDroidHR data collection.



(a) The initial 2s of data fluctuates on the Samsung Gear S3.

(b) Data changes depending on distance to the surface of skin.

Fig. 6. Two data collection issues that affect reliability.

3.2.2 Collection by fixed time window. We initially use a 10s time window for data collection, which means that we ask participants to perform each gestures that lasted 10s, because we need to sufficiently analyse data to extract features. However, the total time for the experiments becomes too long in this way. In addition, feedbacks from our first participants indicate that they are not willing to use gesture that take so much time. After a set of data validation, we finally use 3s as a fixed time window for data collection and gesture detection. Thus, we apply this time window in our final implementation which offers a 3s window for each gesture.

3.3 PPG Signal Preprocessing Unit

We use signal smoothing and normalization techniques to reduce noise and errors of the raw sensor readings. As a number of sophisticated denoising and normalization techniques have been proposed for signal preprocessing [9], our preprocessing unit employs some of them to improve the quality of the readings for the GMD unit process.

3.3.1 Signal denoising. Since prior work [40] shows that using the wavelet transform techniques to process PPG signal achieves a good filtered output, we employ the Discrete Wavelet Transform (DWT) [22] to smooth the signal. Figure 7 (A) shows a raw PPG signal collected by the heart rate sensor, and the shape of the raw PPG signal incorporates a small fluctuation during the whole time window. Comparatively, Figure 7 (B) shows the changes of the signal after using DWT to smooth, and the output of the denoising is significantly improved.

3.3.2 Normalization. In the data collection procedure, we find that the value range of the green light on the heart rate sensor of the Gear S3 [15] varied in an unexpected way. The gap value between 2 sets of data may be over $1e6$. We assume that the heart rate sensor of the Gear S3 may be designed to have some light power modulated mechanisms to ensure the accuracy of the PPG signal. Thus, we apply signal normalization to the PPG data to standardise each signal stream to the same unit level. Hereby, assuming that N is the length of the original signal S, the normalized signal S_n is given by [5]:

$$S_n = \frac{S}{\sqrt{\frac{\sum_{i=1}^{N}|S_i^2|}{N}}} \tag{1}$$

Figure 7 (C) shows the output of the normalization processing. The value of each signal is scaled around 0, which is helpful for the subsequent GMD Unit and Classification Unit.

3.4 Gradient-based Movement Detection (GMD) Unit

We propose a core unit using gradient-based techniques to further filter the noise from the PPG signal. In prior work, gradient-based algorithms or strategies are widely used for image denoising [50], object recognition fields [7], and hand-writing or documentation recognition [26]. The advantage of using a gradient-based technique to remove noise is that the gradient processed signal can retain some distinct features which are more sensitive to noise than other standard edge-based algorithms [7]. Additionally, using the gradient-based approach, machine learning techniques tend to attain better classification result for high-dimensional patterns [26].

Once the signal is converted to a gradient curve, the noise is usually presented to be centralized or crowded together since the noise tends to intrinsically have high frequency. In this way, the noise can be easily extracted from the original signal or detected by linear search algorithms. As the gradient-processed PPG signal has a similar performance after the meticulous observations, we finally use the GMD unit system for our gesture recognition.

Our GMD unit contains 2 major procedures: stationary mode analysis, and gesture segment selection. The first procedure of GMD mainly focuses on the generation of the Heart Rate Noising Threshold from the stationary mode analysis. It can filter the periodic heartbeat noise [13] and nonspontaneous movements [28]. Conversely, the second procedure of GMD is to detect and select the segment which represents an actual gesture movement using the generated threshold.

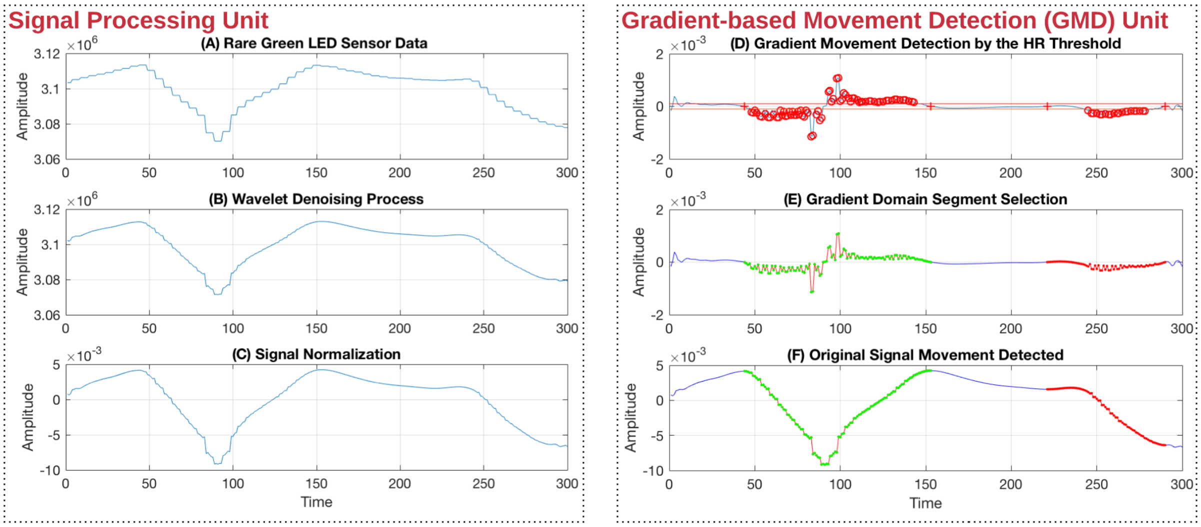


Fig. 7. FinDroidHR PPG Signal Processing Work Flow: The above data represents the Open Hand Gesture Detection within the fixed time window 3.2.2. (A) shows the original sensor data collected by a heart rate monitor; (B) is the de-noised signal by the Wavelet Denoising Algorithm; (C) represents the signal normalised on the y-axis; (D) converts the normalized signal into the gradient domain and finds multiple movement segments by the HR threshold and extra essential parameters of our Gradient processing approach; (E) identifies one of the segments as a gesture; (F) maps the selected gradient segment to the original normalised signal and passes the movement signal segment for further feature extraction.

Figure 7 shows that a normalized PPG signal is processed by the 2 procedures of the GMD unit to generate its actual segment representing a gesture movement. Table 3 illustrates 5 essential parameters of the GMD unit. The detailed use of each parameter is addressed in the next sections.

3.4.1 Stationary mode analysis. The purpose of the stationary mode analysis is to find a set of peaks by the given stationary mode data set. Then the absolute maximum peaks work as the Heart Rate (HR) Threshold in Table 3 for denoising the effect of heartbeats, and we measure the mean distance between these peaks as the Peak-to-peak Distance (PPD) in Table 3 for separating the data into different small movement segments subsequently. The benefit is the improvement of the adaptability when different participants use FinDroidHR. We attempt to use statistic techniques to calculate a universal threshold to remove the heartbeat noise and non-spontaneous movements for all participants. However, according to our empirical investigation, the signals vary across different demographic characteristics, such as age, height, weight and gender. Thus, dynamically detecting and extracting different heart rate noising threshold for each user is necessary for our work, instead of simply generating a universal noising threshold.

Regarding Heart Rate Detection, most of prior work has aimed to accurately detect the heartbeats in PPG signals. For example, the work in [13] performs Heart Rate Detection in a noisy environment, and proposed a detection algorithm with the short term autocorrelation approach to determine the heart rate against other noising including motion artefacts. In our work we are interested to remove heartbeats. The difference is that we attempt to calculate a threshold by measuring the heart rate in order to remove the heartbeat noise, which inversely purifies the finger gesture movements by using the gradient-based filtering approaches.

Due to the effect of the gradient process, the normalized PPG signal is converted into a fluctuated gradient curve from its original shape. This leads to an issue that there is more difficulty in accurately identifying the real

Table 3. An Overview of the Essential Parameters in the GMD Unit: more detailed use cases are in the related algorithm sections.

GMD Unit Essential Parameter Set			
No.	Short Name	Real Name	Description
1	SWS	Shift Window Size	Input of the proposed Find Peaks algorithm. The default value is 10, which splits the length of the given signal by a 10-point window size.
2	HR Threshold	Heart Rate Noise Threshold	Output of the proposed HR-filter algorithm. Representing a range to distinguish either valid movement data or noise, which is generated by searching an absolute maximum peak based on the stationary mode data set.
3	PPD	Peak-to-peak Distance	Output of the proposed HR-filter algorithm. Representing a mean of the peak-to-peak distance of the gradient processed data. Applied for the segment separation in the segment selection procedure.
4	MSTW	Minimal Segment Time Window	Representing a minimal actual gesture movement time length, which is observed by real test measurements. The default value is 15, which means 150 ms: if a movement is less than this, the movement will be treated as a nonspontaneous movement. In the case, it is used to filter the nonspontaneous movement noise. 3.4.2
5	SSO	Side Search Offset	Input of the proposed GMD algorithm. The default value is a half of PPD. Applied in the segment selection procedure.

peaks of the curve. Since the general peak-finding algorithms will report many invalid peaks in our case, we have to make it smooth for finding a small subset of valid peaks. Thus, we use a Signal Smoothing approach to address this issue. There is a large number of well-known algorithms aiming to smooth a signal. The Moving Average Filter algorithm [6] is widely used in such smoothing signal circumstances, but it can not properly preserve the original frequency components of data, and has transient effects which are not suitable for our work. Hence, it is important to find another proper smoothing algorithm that preserves such fluctuated frequency of the curve to accurately find an absolute maximum peak of the curve.

The Savitzky-Golay filter algorithm [6], instead, is suitable for our goals because it is designed to keep the original frequency of the data and account for the transient effects. However, after several practical tests, the results of using the Savitzky-Golay filter algorithm provided by Matlab are not sufficiently accurate for finding the peaks. The problem is that the original shape and amplitude of the data is still partially modified, so that the generated HR threshold and PPD are not effective to remove the heartbeats in many cases. To generate more accurate HR threshold and PPD, we propose an improved Signal Smoothing filter called the HR-filter for this work.

The idea of the HR-filter algorithm is inspired by the Signal Smoothing approaches. It shifts a time window linearly to compare neighbours and find peaks. However, the difference is that there is no need to smooth the signal. Therefore, the HR-filter does not include any smoothing operations by polynomial fitting, which improves the algorithm performance, beyond the Savitzky-Golay filter. Here we present the HR-filter algorithm as pseudo code 1 including the Find Peaks function 2.

In terms of the *Findpeak* algorithm 2, because it is designed as a linear search for split neighbour segment comparison, its time complexity is $O(n)$.

ALGORITHM 1: Generate HR Threshold and Mean PPD

```
input  : A List of Stationary Mode Data $List$ of size $l$
output : A $HR\_threshold$ and A $PPD$
$shift\_window \leftarrow 10$ ;                     // the default value of the SWS in Table 3
$PeakList \leftarrow \emptyset$;
$Distances \leftarrow \emptyset$;
for $i \leftarrow 1$ to $l$ do
    $rowdata \leftarrow List[i]$;
    $[peaks, indexs] \leftarrow$ Findpeak($rowdata, shift\_window$);
    $PeakList[i] \leftarrow$ Mean($peaks$) ;    // In order to balance the effect of the nonspontaneous movements
    $Distances[i] \leftarrow$ FindDistance($indexs$) ;    // Calculate the distance between each pair of peaks
end
$MaximumPeak \leftarrow$ GetMax($PeakList$);
$MinimumPeak \leftarrow$ GetMin($PeakList$);
if Absolute($MinimumPeak$) $>$ $MaximumPeak$ then
    $HR\_threshold \leftarrow$ Absolute($MinimumPeak$);
else
    $HR\_threshold \leftarrow MaximumPeak$;
end
$PPD \leftarrow$ Mean($Distances$) $+distance\_offset$ ;  // Round toward The closest integer, which can be divisible by
  the shift_window
return A pair of $HR\_threshold$ and $PPD$;
```

3.4.2 *The effect of nonspontaneous movements.* In the stationary mode analysis, all participants are asked to keep their body position stationary and look at the watch without any other movements. Ideally, the data should represent only their periodic heartbeats. However, the collected data is often aperiodic with some short-term high frequency data circled in Figure 8. According to prior work [28], Yuka et al. pointed out that the heart rate can be sensitively contaminated by any kinds of small movements which are difficult to observe, such as respiration or nervous system activity. Thus, the data involves nonspontaneous movement or some slight movement that is difficult to manually observe. We refer to these as nonspontaneous movements.

Since nonspontaneous movements are considered unexpected data, data collection in high frequency within a short time period will strongly affect the algorithm. As a result, the algorithm may generate many invalid peaks. To address this problem, we update the algorithm conditions by a mean operation on the selected peaks to

ALGORITHM 2: Search for Peak Points by the Given Gradient Signal

```
input  : A List of Signal Data $List$ of size $l$ and the SWS defined in Table 3 $shift\_window$
output : A List of $Peaks$ and A List of $Indexs$
Struct $Point$ contains
    float value;
    int index;
end
$i \leftarrow 1$;
$Peaks \leftarrow \emptyset$;
$Indexs \leftarrow \emptyset$;
$Front\_point \leftarrow Point$;
$Rear\_point \leftarrow Point$;
$Middle\_point \leftarrow$ SelectMax($i, shift\_window, List$);        // Select the maximum value within a shift_window
while $i < l$ do
    $i \leftarrow i + shift\_window$ ;                                  // Moving to the next point of the input List
    $Rear\_point \leftarrow$ SelectMax($i, shift\_window, List$);
    $Peak \leftarrow$ IsPeak($Front\_point, Middle\_point, Rear\_point$) ;   // Check whether the value of Middle point is
     greater than both of Front point and Rear point.
    if $Peak$ is not empty then
        Add $Peak.value$ into the List $Peaks$;
        Add $Peak.index$ into the List $Indexs$;
    $Front\_point \leftarrow Middle\_point$;
    $Middle\_point \leftarrow Rear\_point$;
end
return $Peaks$ and $Indexs$;
```

balance the negative effect of the nonspontaneous movements before finding the absolute maximum peak value. The pseudo code 1 presents this enhanced method.

On the other hand, nonspontaneous movements are generally identified as short-term invalid peaks, which are also treated as a kind of noise that the GMD Unit has to filter. We observe all gesture movements average time window length to provide a minimal segment time window in Table 3 as a threshold to appropriately minimise the effect of the nonspontaneous movements.

Figure 8 shows an example of running the HR-filter algorithm to process the stationary mode signal data. The bottom red line is the absolute maximum peak value, which also represents the HR threshold for this certain data. When we collect a set of data, we add a mean operation for each selected peak to balance the effect of unexpected nonspontaneous movements.

3.4.3 The Shift window size dissuasion. Regarding the HR-filter algorithm 1, we keep the value of the SWS in table 3 as default 10, since we achieve the best segment selection outcomes by the default value in this work. However, the SWS can be modified in further study since there is a negative correlation between the SWS and the HR threshold. If the SWS is reduced, the smoothing window is shorter. And the quantity of selected peaks with smaller values will increase. By then, if we perform a mean operation, the HR threshold will decrease as there are more small values in the peak set. On the other hand, if the SWS increases , the smoothing window is enlarged. Then, the quantity of selected peaks will drop and a few peaks with maximum value will remain in the peak set. In this way, the HR threshold will be increased after the mean operation. We find that the change of

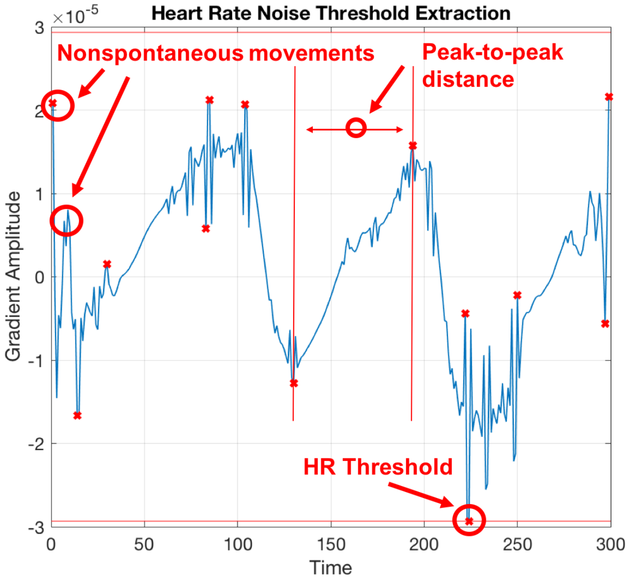


Fig. 8. The above signal is collected when participants remain in the stationary mode without performing any gestures. The HR-filter algorithm 1 calculates a threshold to remove the heart rate against the gesture movements. Additionally, the mean peak-to-peak distance is used to split the signal into multiple small segments. Furthermore, the signal contains a few instances of nonspontaneous movements 3.4.2, which typically occurs at the beginning and the end of the data sequence.

the SWS depends on the quality of the PPG data. For instance, if the collected data set includes data which is contaminated by the nonspontaneous movements, we recommend to reduce the SWS to balance the influence, as the mean of peaks is increased by the noisy data. Thus, the performance of the SWS can be validated based on the collected data set.

3.4.4 Movement segment detection. In this section, we propose the movement segment detection algorithm aiming to detect one or more possible gesture movements on the processed non-noise data set. Since we apply the HR threshold generated by the HR filter, the gradient data is separated into a noise set and a non-noise set, which is implemented by the *ExtractMove* function in code 3. Next, the non-noise data set is what we are interested.

First of all, once a non-noise data set is generated, the PPD in Table 3 is applied for segment separation. As we assume the PPD as the maximum distance between 2 peaks, if the length of the non-noise set is greater than the PPD, it will be split to multiple subsets by the PPD. Each subset represents a possible movement. This is implemented by the *SplitSegment* function in code 3.

Secondly, we believe that the shape of a complete gesture data is *S_shaped*, which includes at least one maximum peak and one minimum peak by the data observation. If this kind of typical data is processed to a gradient data, the shape will be a *U_shaped* parabola. Besides, we aim to extract an approximately complete segment to represent the gesture movement from one time window. Therefore, the proposed movement segment detection algorithm can be used. The idea of this algorithm is inspired by the greedy strategy. We design the selection function for linear neighbour searching both sides of the non-noise data set until finding the point which is closest to 0 on each side of data. The implementation is in code 3. As this algorithm is a linear searching, the complexity is $O(n)$.

Figure 9 gives an example to explain the above 2 steps. A *ExtractMove* processed non-noise data set is split by the *SplitSegment* function to 2 different segments. Using the Side Search Offset (SSO) to searching both sides of each non-noise set, the search stops at the red-cross marked position. Finally, we obtain 2 approximately complete segments to represent the possible movements within the current whole time window.

There is a challenge in this algorithm implementation: the segment overlap issue. The reason is that the default state of the SSO is not applicable for searching the end points. We attempt to use a random value, such as 30 or 40, for SSO, which is greater than the PPD. However, the output signal shows that one segment overlaps the other. Then, after several tests, we find that the maximum SSO should be the half of the PPD to ensure that the side searching range will not affect the other segment.

ALGORITHM 3: Movement Segment Detection

```
input  : A List of Signal Data $List$ of size $l$ and the $HR\_Threshold$ and the $PPD$
output : A Matrix of Movement Segments $Move$ of size $a \times b$
$search\_offset \leftarrow PPD/2$ ;    // The Side Search Offset (SSO), which is defined in the Table 3
$Points \leftarrow \emptyset$;
$Gradient\_list \leftarrow$ Gradient($List$) ;    // Calculate the gradient data by the given signal
$Points \leftarrow$ ExtractMove($HR\_Threshold$);    // Select all points which are greater or less than the
  HR_threshold as movement points. Meanwhile, remove the points in the range $[0, search\_offset]$ and
  $[l - search\_offset]$ due to the nonspontaneous movements
$Move \leftarrow$ SplitSegment($Points$);    // Segment set separation process, splitting the points into different
  movement segment lists
// Search both sides of each segment group by SSO until find the gradient value is closest to 0 in
   order to extract a complete movement period
for $i \leftarrow 1$ to $a$ do
    $segment \leftarrow Move[i]$;
    $length \leftarrow$ size of $segment$;
    $left\_end \leftarrow False$;
    $right\_end \leftarrow False$;
    $left\_index \leftarrow 1$;
    $right\_index \leftarrow length$;
    while True do
        if $left\_index - search\_offset < 1$ then
            $left\_end \leftarrow True$;
        if $right\_index + search\_offset > length$ then
            $right\_end \leftarrow True$;
        if $left\_end$ and $right\_end$ then
            Break;
        // Search both sides until the value of the neighbours is closest to 0
        $left\_min\_index \leftarrow$ FindEdgeMin($left\_index - search\_offset, left\_index$);
        $right\_min\_index \leftarrow$ FindEdgeMin($right\_index, right\_index + search\_offset$);
    end
    Add $left\_min\_index$ and $right\_min\_index$ into $Move[i]$;
end
return $Move$;
```

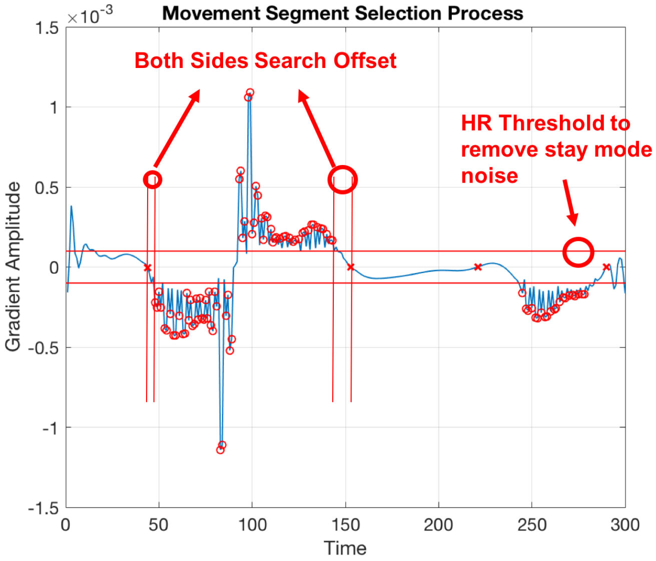


Fig. 9. Gradient-based Movement Detection Algorithm 3: this algorithm requires a couple of inputs: HR threshold and Peak-to-peak distance (generated by the stationary mode detection algorithm 1). The final processed signal is in Figure 7 (E), which shows a green marked segment marking the gesture movement.

3.4.5 Movement segment selection. In this section, we propose a segment selection algorithm with double filter conditions to target one segment from the above detected multiple segments. Algorithm 4 describes the process. Since a set of approximately complete segments is generated by the segment detection procedure, we aim to pick up the real movement segment from this set.

The target function of this algorithm is to find the segment with the $|Max_{Peak-to-peak\_Amplitude}|$. However, this is not applicable for the segments contaminated by the nonspontaneous movements, as these movements usually have the $|Max_{Peak-to-peak\_Amplitude}|$ and outperform the real candidate from the set. To tackle this issue, we design the segment selection algorithm with double filter function to remove these nonspontaneous movements. The first filter condition is based on the MSTW defined in Table 3. Since the nonspontaneous movements have high frequency but with short time window, when the time window is less than the MSTW, the segment will be removed from the set. While the MSTW is measured by our observation, and actual the minimum time window is 15, which means 150ms to perform the *pinky* gesture. By doing so, most of nonspontaneous movements can be filtered, but a few of them are still left. Then, we present the second filter condition for the one side checking. As we noticed that the shapes of nonspontaneous movements are regularly either increasing or decreasing, we can identify them by checking whether a segment is monotonic. Thus, the *OnesideCheck* function with code 4 is applied to check the monotone of each segment. Finally, the output signals of both filter conditions are highly accurate for the GMD Unit.

Thereby, we integrate all proposed techniques into FinDroidHR implementation. Figure 10 depicts a final outcome by the GMD Unit that a *downhand* gesture is successfully selected from the fixed time window.

ALGORITHM 4: Find A Real Movement from a List of Movement Segments

```
input  : The Matrix of Movement Segments $Move$ of size $a \times b$ and the $MSTW$ defined in Table 3 and the $PPD$
output: A Gesture Movement $Move\_Segment$
$Max\_peak \leftarrow 0$;
$Move\_Segment \leftarrow \emptyset$;
for $i \leftarrow 1$ to $a$ do
    $Segment \leftarrow Move[i]$;
    $isBothSide \leftarrow$ OnesideCheck($Segment, MSTW, PPD$);    // First filter the points of the nonspontaneous
     movements as many as possible
    if $isBothSide$ is $True$ then
        $Max\_point \leftarrow$ FindMax($Segment$);
        $Min\_point \leftarrow$ FindMin($Segment$);
        $Peak \leftarrow$ Absolute($Min\_point$) $+Max\_point$;
        // Second filter the rest of nonspontaneous points. The valid points should be greater than
           the defined MSTW. Then select one of segment which has the maximum amplitude as a real
           gesture segment
        if $Peak > MSTW$ and $Max\_peak < Peak$ then
            $Max\_peak \leftarrow Peak$;
            $Move\_Segment \leftarrow Segment$;
end
return $Move\_Segment$;
```

3.5 PPG Signal Feature Extraction

In this section, we fully explore all potential features for PPG signal recognition. According to prior work [9] [34] [3], we select in total 69 features across time-frequency domain from all 5 perspectives, including the Peak-to-peak analysis [13], the Power spectrum analysis [34], the Discrete wavelet transform (DWT) frequency domain analysis [22], the Maximum entropy computing analysis [11] [42], and the Fast Dynamic Time Warping (FDTW) to calculate the similarity distance [16] [9]. Detailed description for each feature is in the Appendix A. Since our work adopts the gradient-based techniques, the features of gradient data are equally important to classification. Thus, both original selected segments and gradient selected segments are used to extract the feature set with a double size (i.e., 138 features).

3.6 Selected Feature Set by Different Selection Algorithms

To explore the effectiveness and feasibility of our proposed 138-feature set, we select 3 well-developed feature selection algorithms to perform cross-validation on the reduced feature subsets. We employ the Sequential feature selection (SFS) algorithm [3], Random subset feature selection (RSFS) algorithm [49], and ReliefF feature selection algorithm [35] on the feature set. The results of the evaluation are shown in Table 4.

4 EVALUATION

We first investigate the influence of the level of wearing tightness for PPG signals. Subsequently, we report a controlled experiment with 20 participants to evaluate our system performance. Overall, our evaluation focuses on the following 7 aspects:

- The performance difference when using dominant vs. non-dominant hand.
- The performance difference across 3 body poses when performing gestures.

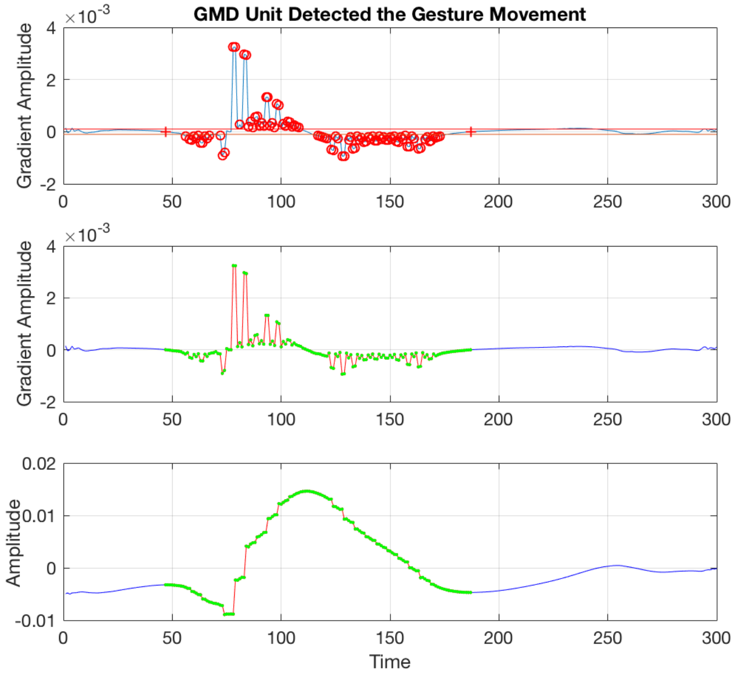


Fig. 10. The GMD unit detecting a *downhand* gesture.

- The performance difference under both user motion and ambient motion conditions compared with an implemented IMU based gesture recognition system.
- The accuracy of SVM and Random Forest classifiers using 3 selected feature sets.
- The effectiveness of our Gradient-based Movement Detection (GMD) Unit.
- The comparison between actual gesture time window and selected gesture time window by GMD.
- The added power consumption of FinDroidHR compared to the baseline use.

Because PPG biosensor measures blood, and the blood in our body is often affected by ambient temperature and emotional reaction, all tests take place in indoor conditions with a stable temperature (25 to 26 degrees) and outdoor conditions with a stable temperature (21 to 23 degrees), no emotional effects are involved through the whole session tests. All participants are in good health condition without any heart situations.

4.1 Wearing Style

In practical daily life, people may have different styles to wear smartwatches. Some people would like to wear loosely for most of time, while others prefer to wear tightly to secure smartwatches. On the other hand, as the heart rate monitor of smartwatches is an optical-based sensitive PPG biosensor, different wearing positions may affect the PPG signals, and then the extracted features from PPG signals in time-frequency domain may vary as well. To ensure our system performance, we conduct a test to explore the effects of different wearing styles to PPG signals.

Table 4. FinDroid Selected Feature Set. Detailed feature descriptions are in Appendix A.

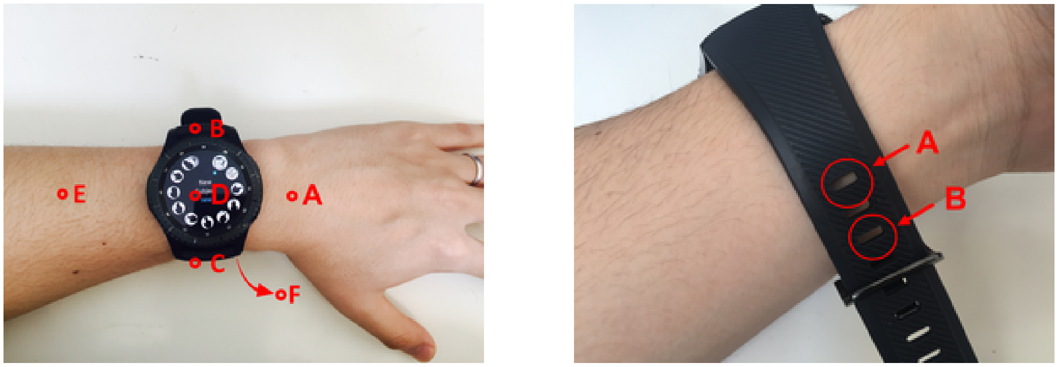
Selected Feature Set for Different Algorithms			
Algorithm	Features	Original Signal Features	Gradient Signal Features
Sequential feature selection (SFS)	16	meanS(1), meanHPSDMED(21), medianHPSDMED(23), meanLMAXE(56), maxGRAE(67)	meanS(1), stdS(2), medianS(3), stdPKA(8), disDTW(10), entroS(14), stdHPSDME(16), medianLME(38), maxLMINE(49), minLMAXE(53), medianGRAE(69)
Random forest feature importance (RFFI)	20	meanS(1), entroS(14), medianHPSDST(20), stdHPSDMED(22), maxLMAXE(50), meanLMAXE(56), medianGRAE(69)	meanS(1), medianPKT(6), meanPKA(7), medianPKA(9), medianPSD(13), entroS(14), stdHPSDME(16), medianHPSDST(20), stdMINHPSD(28), medianE(47), maxLFDTW(48), maxLMAXE(50), minLMINE(52)
ReliefF feature selection (RF)	25	stdS(2), meanHPSDME(15), medianHPSDME(17), meanHPSDMED(21), medianHPSDMED(23), meanMINHPSD(27), medianMINHPSD(29), stdLME(37), meanLMED(42), minLFDTW(51), medianLMAX(62)	meanS(1), stdS(2), disDTW(10), meanHPSDME(15), medianHPSDME(17), meanHPSDMED(21), medianHPSDMED(23), meanMAXHPSD(24), meanMINHPSD(27), medianMINHPSD(29), stdLME(37), meanLMINE(55), stdLFDTW(57), medianLMAX(62)

4.1.1 *Loose wearing.*

We first map 6 most possible positions when wearing watch loosely, and Figure 11a demonstrates these positions. For each position, we perform the *ophand* gesture to point out differences.

Figure 12a clearly indicates that the PPG signals are changed by different wearing positions in time-domain when performing the same *ophand* gesture. As can be seen from the figure, the purple line representing the position E shows a huge fluctuating when doing the gesture, while the position A also significantly affects the signal's amplitude. Besides, the curve of the position F shows a negative wave, which is different from others. In fact, the first blue line of the position D (Common Snug) is the most valid and reliable signal for our system, but the change of the position D is the smallest among these positions. Since the PPG signals are sensitively affected by different positions, the extracted features from the time-domain have also been changed to be invalid to our system. On the other hand, Figure 12b shows the signals in gradient-time domain. We can also easily see that there are a large movement when wearing different positions except the best comfort and reliable position D. These incorrect movements will leverage inappropriate features to our system so that the performance of FinDroidHR will also be influenced. Therefore, to ensure a fair performance by using our system, the position D for a snug fit is selected as the ideal wearing position, which is close to the wrist around 1cm to 2cm and the heart rate monitor has to directly face the skin.

To improve the fault-tolerant of our system, we applied an instant position check mechanism for our system to avoid invalid signal data. And the idea behind the mechanism relies on checking whether the signal has such



(a) The possible watch positions if wearing loosely: A is on the wrist (against the bone), B is outward slant, C is inward slant, D is the common snug wearing (close to wrist around 1cm), E is top of the wrist and F is facing down to hand

(b) 2 Levels for tightness test: position A is the tightest option, while position B is common comfort wearing option.

Fig. 11. Wearing Tightness Check.

huge amplitude changes in both time and gradient domains. Once users are performing gestures by inappropriate positions, our system will give users a message or notice to reminder wearing properly.

4.1.2 Tight wearing. In terms of wearing smartwatches tightly, we choose 2-level options on the watch to adjust the tightness. Figure 11b shows that option A is tighter than option B. Similarly, we manage a test whether different tightness can be able to influence PPG signals. For each option, we still use *ophand* gesture to analyse.

Figure 13a in the time domain illustrates that the signal by option A wearing tight has a slightly bigger fluctuation than by option B, while Figure 13b in the gradient domain shows that there is no major or huge difference between both options, and the values of both signals' amplitude are identified as valid and correct data to our system. However, one practical issue is emerged that some of $P + F$ and W gestures by option A are harder to perform than by option B. It is because that the wrist can not be able to flexibly move as usual after wearing tight, which affect to gesture performance. However, wearing tight benefits to recognise gestures because it increases the blood pressure on the wrist.

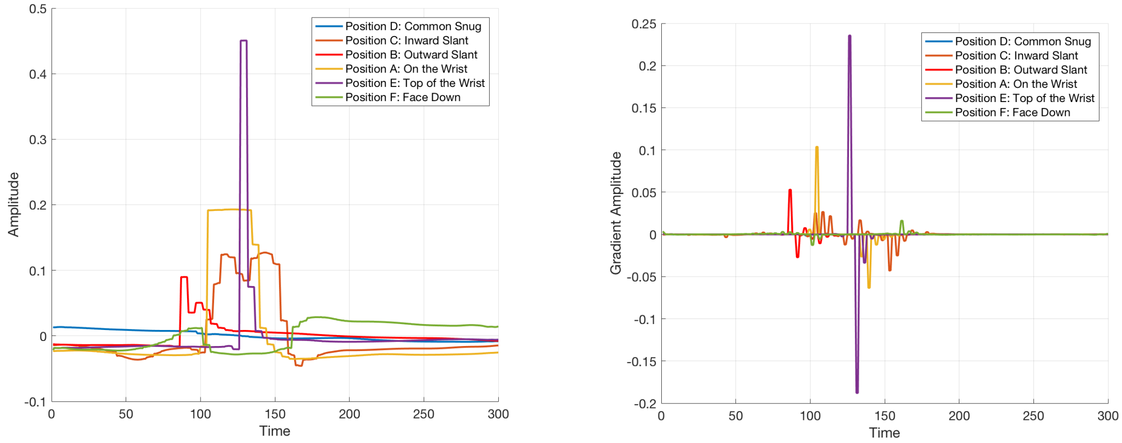
From these, we summary that both tightness options work fine to our system. Considering from wearing comfort perspective, we still recommend to wear the smartwatch for a common snug fit.

In conclusion, the recommended style is to wear watch close to the wrist with a snug fit. We also ask all participants to wear by this proposed style to take all tasks.

4.2 Experimental Methodology

We recruit 20 participants (14 male and 6 female) ranging in age from 23 to 31 years (mean 27 years), height from 158cm to 182cm and weight from 48kg to 85kg. As the tasks are time-consuming to these participants, we separated them into 4 study groups (5 people per group) with different sessions to assess them by all given tasks. Besides, there are 2 major studies involved in our project, the stationary study and the motion study.

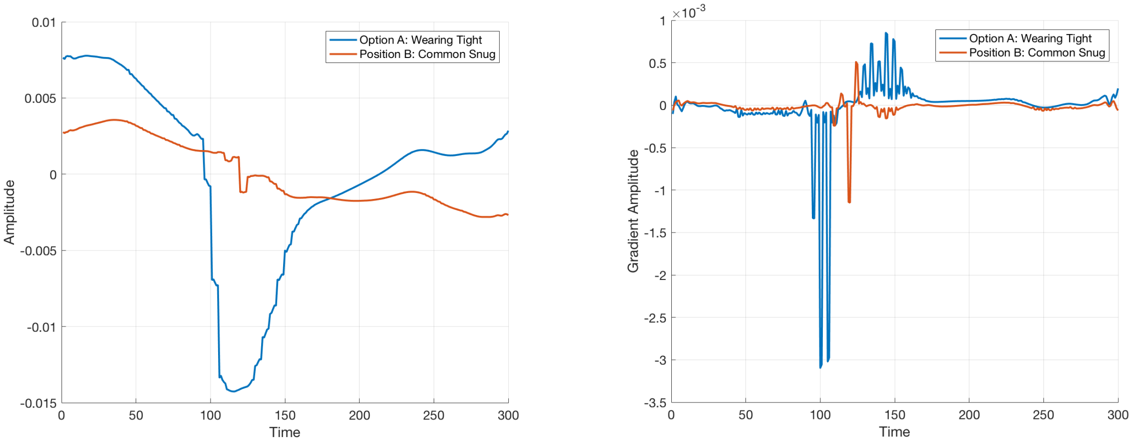
Before the tasks, we ask participants to first keep their hand steady for the stationary mode analysis 3.4.1 to generate each participant's HR threshold and PPD for both stationary study and motion study. If the task condition is changed, for example changing from sitting to standing, participants will need to take the analysis



(a) 6 wearing positions by performing Open Hand gesture showing in time domain.

(b) 6 wearing positions by performing Open Hand gesture showing in gradient-time domain.

Fig. 12. Comparison Different Wearing Position.



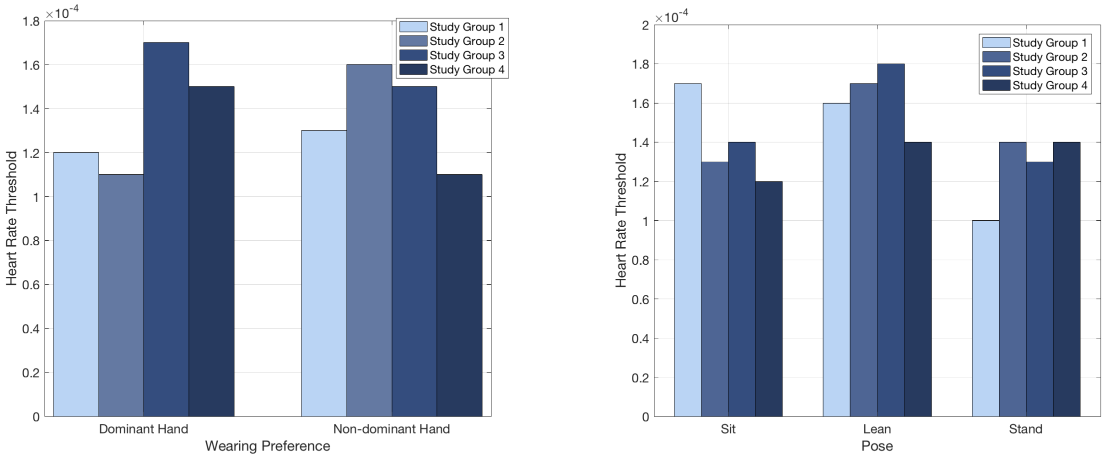
(a) 2 tightness levels by performing Open Hand gesture showing in time domain.

(b) 2 tightness levels by performing Open Hand gesture showing in gradient-time domain.

Fig. 13. Comparison Different Tightness Level.

again before performing hand gestures. Besides, we ensure that each of the task conditions can only be changed to the next when it is completed. Subsequently, participants have a short training session to get familiar with the set of 10 gestures.

For the stationary study (Study 1), we initially perform a "wearing preference test". We ask participants to perform our experimental tasks (i.e. the gestures) when sitting down, and using both their dominant and



(a) The wearing preference results by 4 groups of participants.

(b) The pose results by 4 groups of participants.

Fig. 14. Stationary Study. Overall, the averages of the HR threshold varied in a small range across each study group.

non-dominant hands. Subsequently, we ask participants to wear the watch in their non-dominant hand only, and perform the gestures under 3 different poses: sitting down, sitting down while leaning on a table, and standing still.

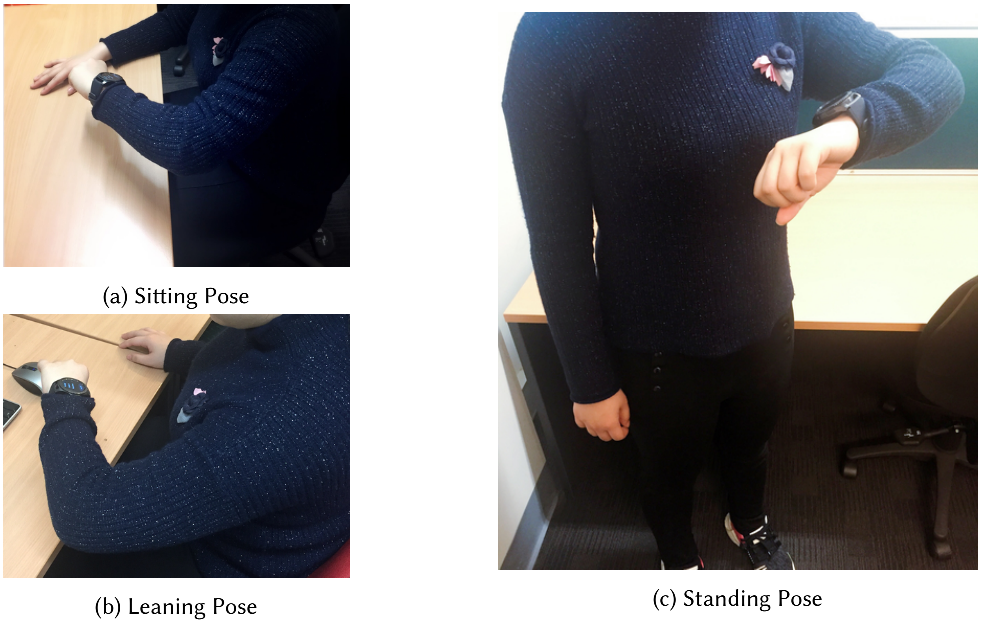
For the motion study (Study 2), all participants are asked to perform gestures in 2 conditions, walking in the garden and taking public transportations. We choose taking trams to be the task as their swaying situation is ideal to test the effects of ambient motions.

In each condition, participants perform each gesture around 20 to 25 times (3s/time fixed time window for collection 3.2.2), and the collection application on the watch records both PPG signals and IMU sensor signals simultaneously for comparison.

4.3 Stationary Study

4.3.1 Different hands wearing preference. We test whether there are differences between using dominant hand and using non-dominant hand. Figure 14a indicates the differences of the HR threshold using both hands across 4 study groups of participants. The HR threshold varies in a tolerant range. The values of PPD are in the range from 30 to 35, which also fluctuated slightly. Therefore, the classification results of using dominant and using non-dominant hand under the sitting mode show an identical average ranging in accuracy from 82.82% to 90.55%. The results show that there is no major difference regarding using different hands, when the heart rate of the participant is stable. A minor difference is that using non-dominant hand to perform the gestures is more convenient than using dominant hand, as also confirm by participants' feedback. All subsequent tests therefore use participants' non-dominant hand.

4.3.2 Poses. Our tests reflect 3 common poses that users may adopt when interacting with watches in everyday settings: sitting on a chair, sitting and leaning the arm on a table, and standing still. All participants were asked to perform the gestures in these 3 different poses. Figure 15 shows one of the participants perform gestures in each of the 3 poses.


(a) Sitting Pose

(b) Leaning Pose

(c) Standing Pose

Fig. 15. Three regular use circumstances with different poses of the smartwatch, including: Sit, Sit with the forearm leaning on a table, and Stand.

Before the experiment, we hypothesis that there is a correlation between the HR threshold and pose. Since the blood pressure of the wrist is associated with changes in the relative position of the arm, we expect that the result of the classification may be affected by the changes of the HR Threshold. However, the results indicate that there is no significant relation between the HR threshold and pose. Figure 14b shows the average HR threshold changes for three participants in different poses. The results vary: study group 1 has the highest average HR threshold in sitting, while study group 2 and 4 have the highest HR threshold when standing. Thus, we can not identify the hypothesised correlation in the experiments. However, it is still possible to propose an adapted HR-threshold approach to improve our system in the future.

Additionally, we investigate the influence of the different poses on the results of the gesture classification. Results show an identical average ranging in accuracy from 82.82% to 90.55% and recall from 82.8% to 90.73%. From the input of the GMD Unit, we find that there are two main reasons why different poses do not affect the result of the classifications. First, the HR thresholds by each participant vary in a tolerant small range, as shown in Figure 14b. Second, the value of the PPD for each participant is the same under different poses. Thus, the outcomes of the GMD Unit are close to each other. From a practical aspect, as the arm portions of participants remained horizontal in the different poses, the heart rate of each participant is relatively stable. Hence, the changes of the HR threshold are minor. In summary, results show that switching the proposed 3 poses while using FinDroidHR do not influence the performance of gesture recognition.

4.4 IMU Recognition System Implementation

As mentioned in the related work section, several hand gesture recognition systems rely on built-in IMU sensors. To evaluate our PPG-based technique for hand gesture recognition compared to IMU solutions, we alternatively implemented our IMU system inspired by one typical system [45].

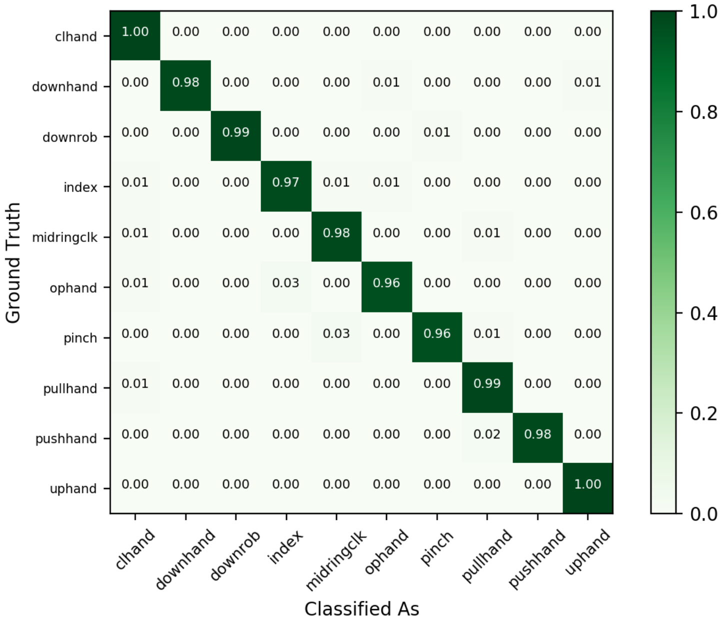


Fig. 16. The result of IMU-based system for 10 proposed gestures classification by SVM classifier.

We employed the build-in IMU sensors on the Samsung Gear S3 by an accelerometer with 3 axes and a gyroscope with 3 axes to collect gesture data. According to the work [45], we also take the magnitude of the combined axes ($\sqrt{x^2+y^2+z^2}$, $\sqrt{x^2+y^2}$, $\sqrt{x^2+z^2}$, $\sqrt{y^2+z^2}$) into account. We calculate all 7 axes of both sensors for a total of 182 features. By using the SFS feature selection method, there are 17 out of 182 features selected to improve the classification process.

To ensure the data consistency of both FinDroidHR and IMU-based system when performing gestures, we apply the same 100Hz frequency with 200Hz sample rate to both accelerometer and gyroscope signals, and collected all 3 kinds of signal data simultaneously.

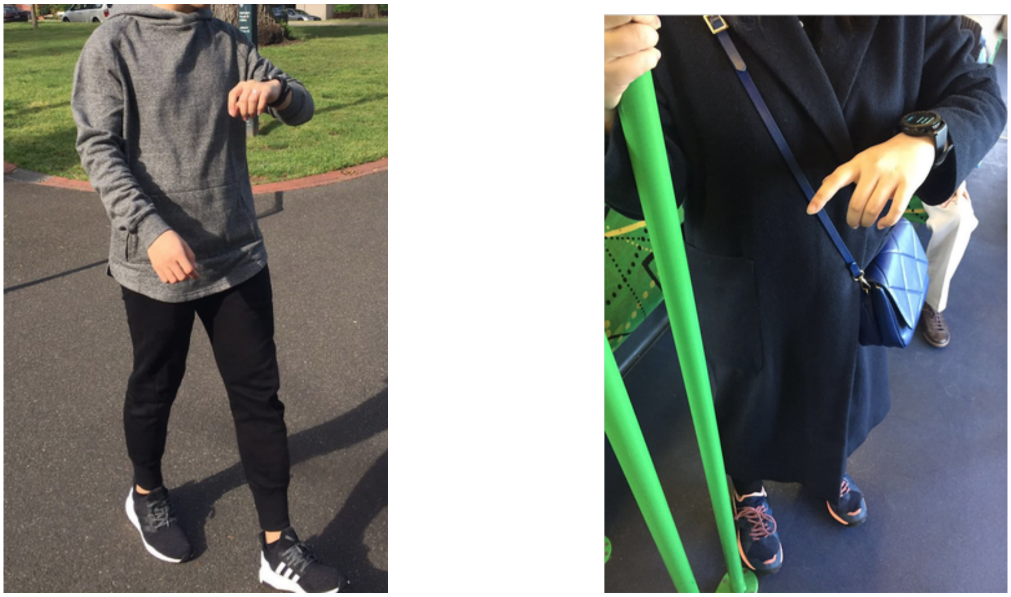
Figure 16 shows that our IMU-based system achieved an average high accuracy of 98.27% and recall 98.3% for our proposed 10-gesture recognition in Study 1. Besides, we perform the system by Random Forests classifier to evaluate the effectiveness, and the result was still an average high accuracy of 97.27% and recall 97.3%.

Overall, in the Study 1, the IMU-based system outperform FinDroidHR compared with each achieved average accuracy. Due to the purposes of sensor, it is not a surprising evidence that the IMU sensors are more precise and sensitive than the PPG biosensors.

4.5 Motion Study

In this study, we evaluate whether our technique is feasible for motion conditions, since users usually interact with smartwatches under motion noise in daily life. Meanwhile, to compare with the IMU-based solution, we conduct 2 motion tasks and 1 extra intensive task to fully evaluate both systems.

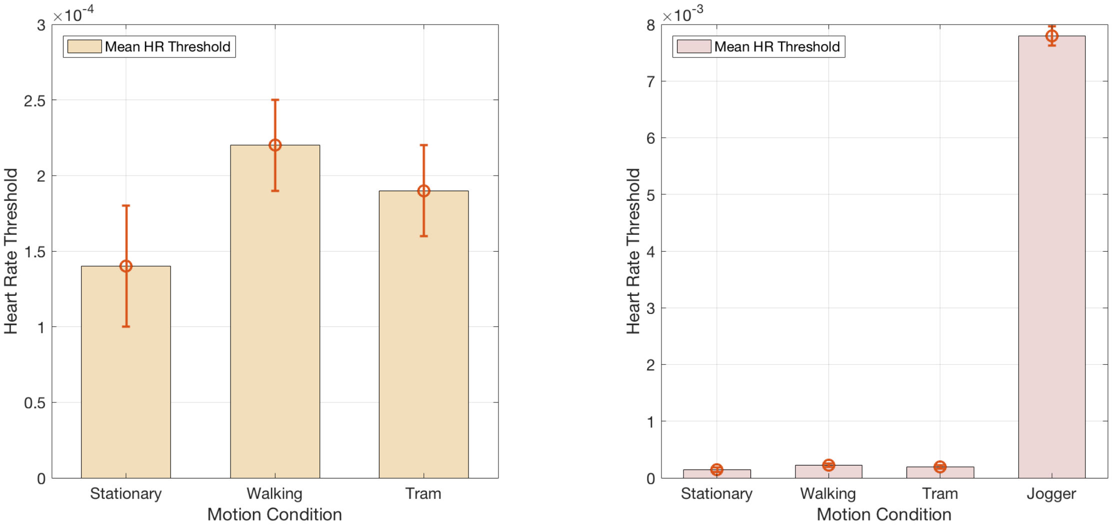
4.5.1 Ambient motion. In most common cases, the ambient motion may take place when users are driving or taking public transportations, and the motion by those vehicles may affect the performance for both FinDroidHR and IMU-based system. Hence, we design the motion task aiming to explore the effects of the ambient motion



(a) Walking in the Garden.

(b) Taking Tram.

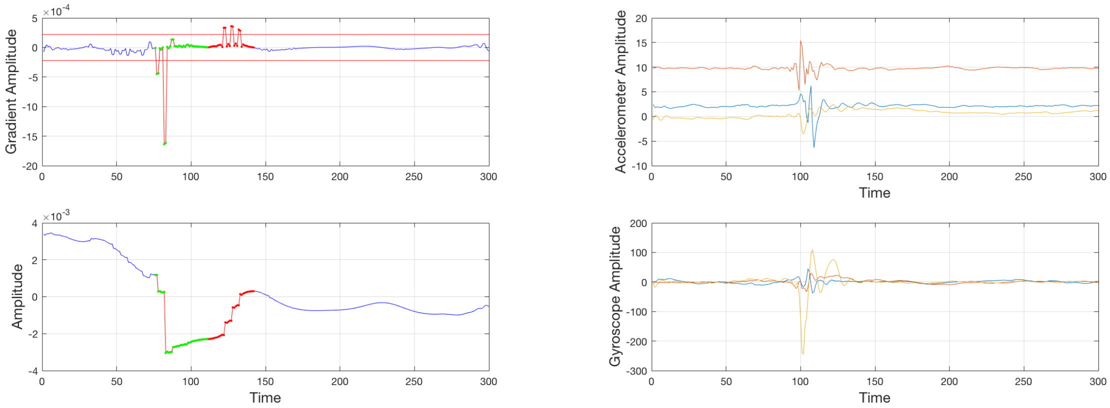
Fig. 17. Motion Study Circumstances.



(a) The Changes of HR Threshold in Both User Motion and Ambient Motion.

(b) Under Cycling Motion Condition.

Fig. 18. HR Threshold Comparison for Different Motion Circumstances.



(a) Processed PPG Signals by GMU under Taking Tram Condition.

(b) Collected IMU Signals under Taking Tram Condition.

Fig. 19. Performing Open Hand Gesture to Simultaneously Collect both PPG and IMU Data with Ambient Noise.

to gesture recognition. Considering that the way of taking public transportations may generate more ambient motions than other cases, we then preform the task on trams which have a free-zone in Melbourne. Figure 17b shows that one of participants is performing all gestures on tram by standing.

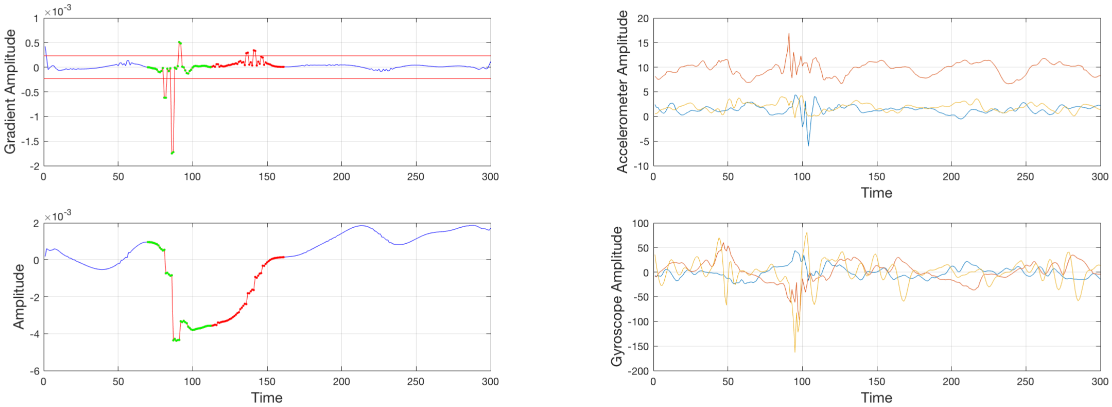
As mentioned in the experiment condition, we first ask participants to do the stationary analysis to measure their heart rates before collecting gesture data for the whole study 1 and 2, which is a kind of manual way to initialise the HR Threshold and PPD for GMU use. Figure 18a shows that the average value of HR Threshold under tram condition is slightly increased compared with the stationary mode, and the value of PPD is identical as 30. This result indicates that the ambient noise of trams has no major effect to participants' heart rate condition.

After the attempted classification, the result of FinDroidHR remains the accuracy from 82.82% to 90.55%, while the result of the IMU-based system is slightly decreased to 92.73%. Besides, Figure 19a and Figure 19b demonstrate both PPG signals and IMU signals side by side. We find that there is no impact for the *ophand* gesture recognition in both time domain and gradient domain by GMU process. Similarly, there is no major noise for IMU signals to identify the *ophand* gesture with only some small tolerant fluctuations.

4.5.2 Walking condition. We next conduct a walking task that all participants are asked to walk in the garden, which is shown in Figure 17a. In the practical perspective, we ask participants to walk at a usual speed without any rush. The average speed varies in the range from 3.5 to 4 km/h.

Figure 18a illustrates that the HR Threshold has a distinct increase among these 3 conditions, while the value of PPD is still the same. Since walking is a slightly intensive motion compared with the stationary mode and taking tram, the heart rate condition of each participant has been affected a bit.

On the other hand, the result of the attempted classification by FinDroidHR under walking condition is relatively stable, which achieves the accuracy from 82.82% to 86.36%. In contrast, the accuracy of IMU-based system is modified to around 81.82%, which is relatively decreased. From both Figure 20a and Figure 20b, we observe that the GMU can remove most of motion noise by adjusted walking HR Threshold to ensure the performance for classification process, while the IMU signals have a number of observable noise, especially for the gyroscope sensor data. That is the reason why the result of IMU-based solution becomes unstable under the effects of motion noise.



(a) Processed PPG Signals by GMU under Walking Condition.

(b) Collected IMU Signals under Walking Condition.

Fig. 20. Performing Open Hand Gesture under User Motion Noise.

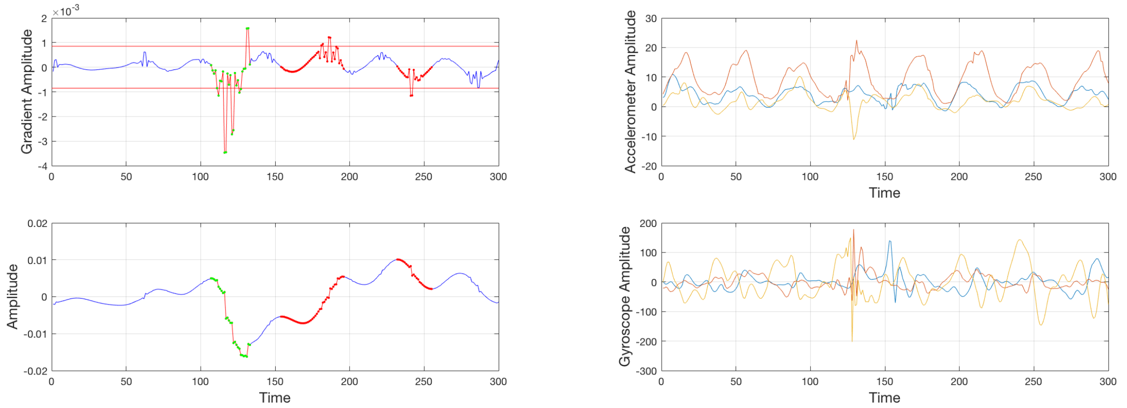
4.5.3 Circling motion. To extend the motion study, the more intensive circling motion condition is considered in our work. Practically, users may interact with the smartwatches during some workouts. We then perform an extra jogger task to discover the limits of both FinDroidHR and IMU-based system. Due to some practical issues that this task is quite difficult for most of participants, only a few of them are involved to perform this task. However, we still obtain some interesting insights based on the small-scale jogger data set.

First of all, Figure 18b shows that the value of the HR Threshold during jogger has a huge increase compared to other 3 conditions, and the value of PPD has been smaller to 20. This result indicates that the heart rate condition of each participant has a sufficient change when performing intensive motions. Besides, Figure 21a shows that our GMU technique can accurately identify the *ophand* gesture even under the jogger condition. On the other hand, Figure 21b demonstrates that it is hard to identity gestures by using IMU-based system because the motion noise has taken over the gesture. After the classification attempt, the average accuracy of FinDroidHR is going down to about 73.6%. The reason is that some of gestures, such as *midringclk* and *pinch*, are performed inappropriately as conflict gestures with others. Since participants are under running mode, they can not concentrate to perform some finger gestures well. However, the accuracy of W, $P + 4F$ and $P + 5F$ is still over 82.82%. Unlike FinDroidHR, the IMU-based system fails to recognise most of gestures because of the significant motion noise.

In conclusion, the IMU-based solution outperform FinDroidHR in Study 1, while it becomes less stable because of different motion noises in Study 2. In contrast, the overall performance of FinDroidHR is relatively stable with a reliable accuracy, and our GMU technique shows high feasibility and effectiveness for hand gesture recognition in different cases.

4.6 Results and Gesture Classification

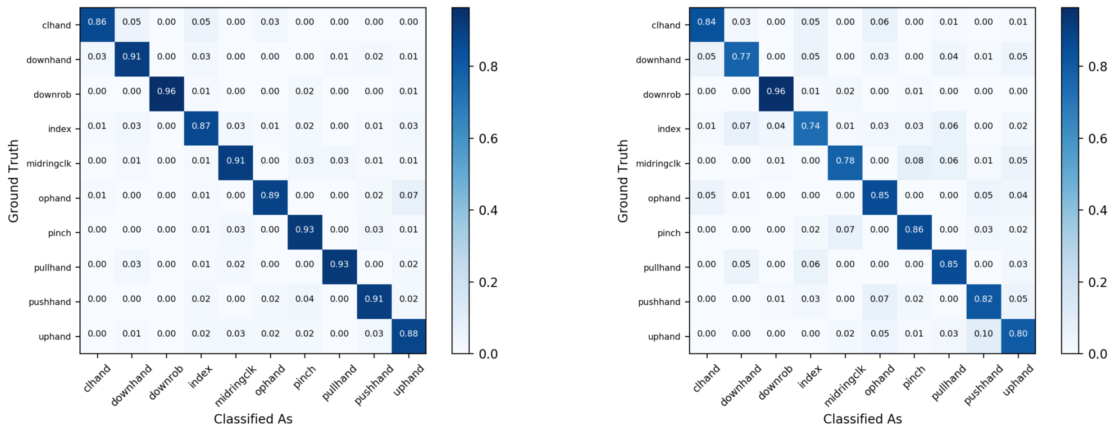
We use a linear support vector machine (SVM) classifier and a Random Forests classifier, as they are widely used and easy to implement. After feature selection (Table 4), we test both classifiers using the selected feature sets respectively. All results are processed and validated in 10-fold cross-validation. Overall, Random Forests using the SFS feature set achieves an average accuracy of 82.82%, precision 82.9% and recall 82.8%, as shown in Figure



(a) Processed PPG Signals by GMU under Jogger Condition.

(b) Collected IMU Signals under Jogger Condition.

Fig. 21. Performing Open Hand Gesture under Jogger Condition.



(a) The confusion matrix for the SVM Classifier.

(b) The confusion matrix for the Random Forest Classifier.

Fig. 22. Classifier performance.

22b. The SVM classifier with a linear kernel using the SFS feature set achieves the best results with an average of accuracy 90.55%, precision 90.55% and recall 90.73%, as shown in Figure 22a.

From the confusion matrix for the gestures, we observe that the 5 *PandF* combined gestures (*clhand*, *downrob*, *ophand*, *pinch* and *pullhand*) outperform the other *Wonly* and *Ponly* gestures. Besides, their results are more stable than the other compared by both classifiers. In particular, the classification of *downrob* in both classifiers achieves high accuracy. However, the *pinch* and the *midringclk* are confused with each other in some samples, as the changes of the blood by both gestures are occasionally similar.

Subsequently, we provide the ROC curve of all classified gestures to evaluate the specificity and sensitivity of our classifiers, as shown in Figure 23. Although ROC curve is typically used in evaluating binary classification

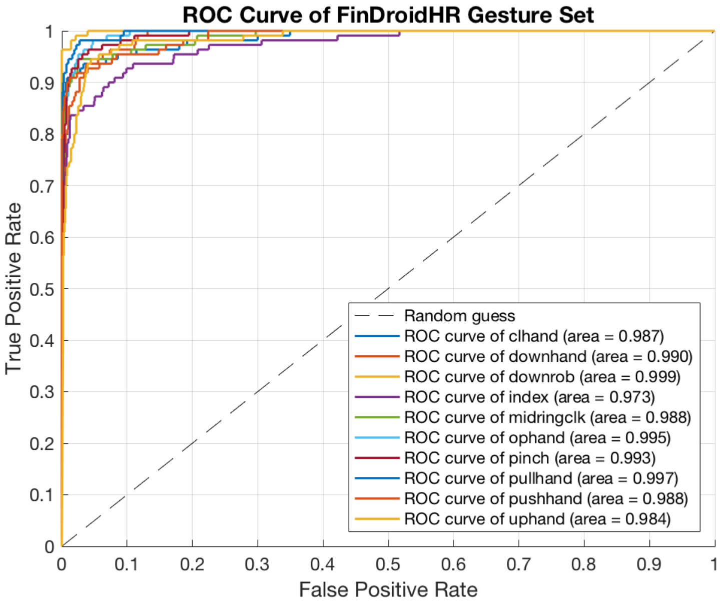


Fig. 23. ROC Curve of FinDroidHR Gestures.

cases [23], we adapt our classifiers to compute the prediction score of each class over the rest of classes as a binary case to generate the ROC curve (i.e., one class against the others). Overall, our classifiers achieve an average AUC of 98.95%, which is effective and feasible from classification perspective.

4.7 Effectiveness of the GMD Unit

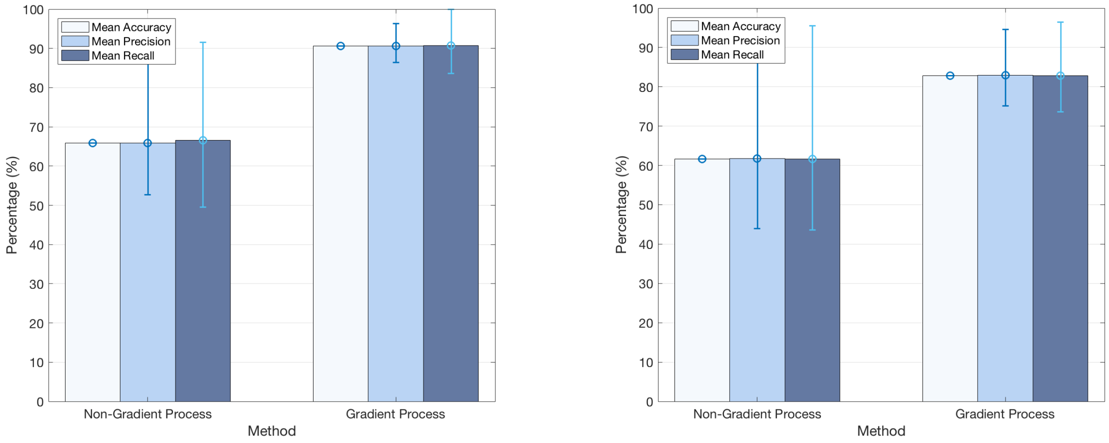
We further evaluate the effectiveness of our Gradient-based Movement Detection Unit. Figure 24 shows that the results of both classifiers had a great improvement after using the GMD Unit. We observe in Figure 24a that the average accuracy by SVM is improved by 24% by the process of the GMD Unit, from 66% to 90%. Meanwhile, the average accuracy for Random Forest is also increased from 61.6% to 82.8%. Overall, the results show that the performance of the GMD Unit is reliable and feasible for our purposes.

4.8 Evaluation of Selected Gesture Time Window by GMD

To further evaluate the reliability of our segment selection algorithm in the GMD Unit, we measured all actual time windows for gestures by observation and compare them to the outcomes of the algorithm to explore the performance difference. Since some of finger gestures are performed very quickly, such as *midringclk* and *pullhand*, we attempt to record their completion time by a stopwatch. The result of the comparison is shown in Figure 25, with the average in accuracy of our segment selection algorithm compared to the actual time window is around 85.5%, which is practical and effective for gesture recognition.

4.9 Power Consumption

On most of smartwatches, the heart rate sensor use is restricted to save power. The sensor will typically awake within a short time when users intend to measure their heart rate, and only be continuously activated during the



(a) Using SVM

(b) Using Random Forests

Fig. 24. The improvement by using the proposed Gradient-based Movement Detection Unit.

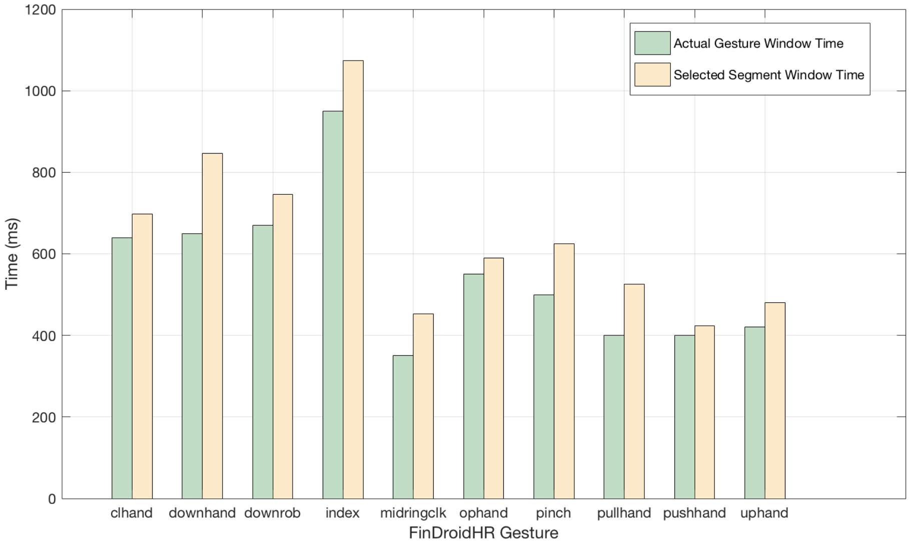


Fig. 25. Movement Time Window comparison between observed actual gesture time window and the segment time window selected by our movement detection approaches.

intensive workout mode for recording fitness data even if the screen is turned off. As FinDroidHR also requires the heart rate sensor to work continuously during the usage, the power consumption may be higher than usual usage. Hence, we apply a sleep mechanism to tackle this issue, where the heart rate sensor will sleep once the

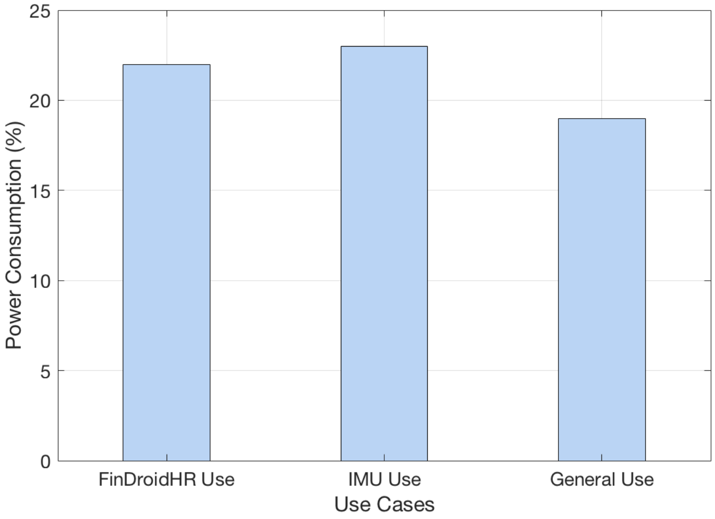


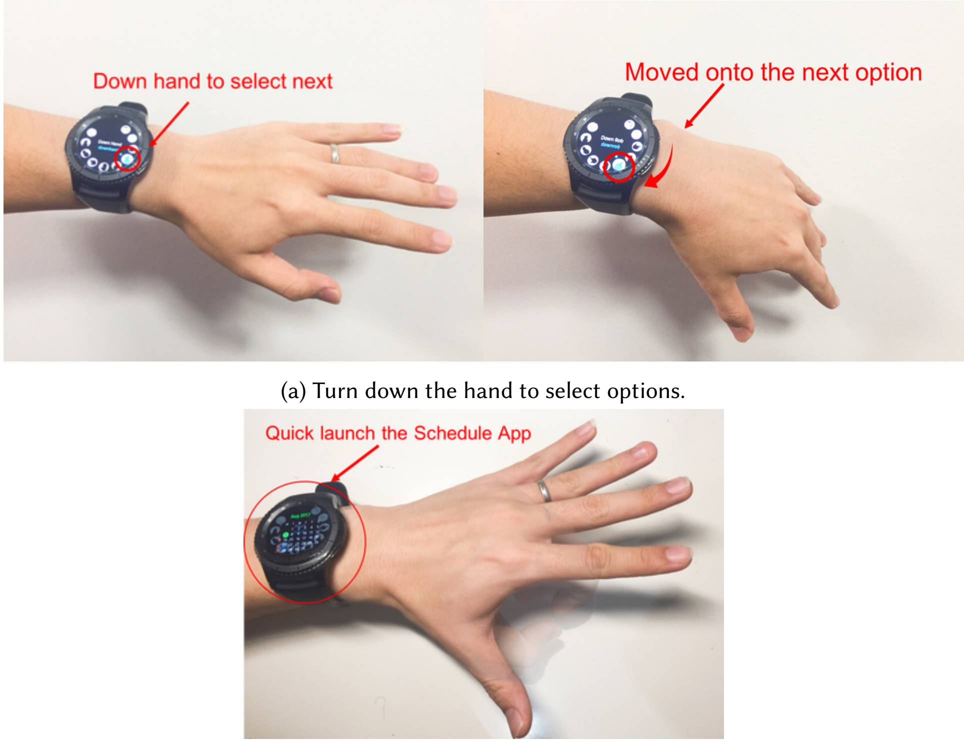
Fig. 26. Power consumption comparison among using FinDroidHR, IMU sensors and general use for half an Hour with the same interaction movements. The screen was kept awake during the test.

smartwatch screen is turned off. In this way, the power cost can be significantly reduced during the period when users do not use the devices.

To quantify the power consumption of FinDroidHR, we conduct a measurement test. During the test, the watch screen is kept as always on. We compare 3 cases: FinDroidHR, IMU and general-purpose use. For each of these 3 cases, we perform the test for 30 minutes. For general use, we preform the bezel rotation every 5s for the selection action. For FinDroidHR and IMU, we programmed the sensors and our recognition processes to automatically collect and process data every 5s to match the bezel function respectively. Figure 26 depicts the power consumption in 3 cases, showing that there is a 3% additional energy cost when using FinDroidHR system. This indicates that using the heart rate sensor has a non-negligible impact on energy consumption. Compared to IMU, the overhead of FinDroidHR is slightly lower than that of IMU. The reason for this may be that the algorithm of our IMU system takes slightly more battery, but the cost by both IMU and FinDroidHR can still be treated as identical in actual usage. Considering that half an hour of intensive usage time is an extremely long interaction on smartwatches, the energy consumption of FinDroidHR is tolerable in most use cases.

5 DISCUSSION AND FUTURE WORK

Our approach enables a hand-free interaction on smartwatches. While a number of techniques have been reported in literature, a key advantage of our approach is that it does not require additional hardware, but rather relies on an existing PPG biosensor available on smartwatches. Our analysis has provided a number of insights regarding useful PPG features for effective gesture recognition in general, which can benefit other researchers. More broadly, our system can be used in a number of scenarios, and was indeed designed as drop-in replacement for the existing input modalities of smartwatches.



(a) Turn down the hand to select options.

(b) Open hand to quick launch the Calendar application.

Fig. 27. Sample use cases: Turn down the hand to simulate the down bezel event for different option selection. Open hand to quick launch another application in the smartwatch.

5.1 Taking on/off Watch during Session

We make an assumption that people may intend to take on/off their smartwatches frequently within a short time in practical daily use. In this case, we managed several breaks in each participant's test and asked them to take off and on smartwatches for a quick recovery. Besides, the break interval is usually given as short, and participants should need to keep their current pose or motion mode until the sub-test is completed. For example, each break is planed within 1 to 3 mins and no break is over 3 mins or longer. If participants are performing in the stationary study by standing, they are required to keep standing to take breaks, and they can not change the current condition to walking as mentioned in 4.2. All processed results show that there is no effect of taking on or off smartwatch occasionally to our system. However, taking off watch for a long break or randomly changing task conditions might affect the results, as participants' heart rate could be unstable in this case. Thus, proposing a dynamic adapted HR-threshold approach should be applied to support our system in future work.

5.2 PPG-based Gesture Recognition Vs. Other Approaches

The experimental results show that our PPG-based system is able to provide reliable gesture recognition for a set of 10 gestures. The results show that both SVM and Random Forests classifier achieved an accuracy of 90.55% and 82.82% respectively. Meanwhile, the Gradient-based Movement Detection (GMD) Unit that we propose provided a substantial accuracy improvement of 24%.

A large number of existing interact techniques on smartwatches involve extra hardware components which are customised, inconvenient or expensive. Compared to previous work, an advantage of PPG-based gesture recognition is that PPG is already available on off-the-shelf smartwatches, meaning that the method can be easily deployed to these watches. Also, to adopt our technique, users do not need to purchase and install extra hardware modules. In practice, users can wear smartwatches for very long periods (e.g., working all day), so that they can start instant interaction with smartwatch in daily life without spending time to wear a interaction-specific sensors.

Furthermore, beyond RF-based gesture recognition systems, an important strength of our technique is that PPG sensors consumes relatively low power. Considering that our system has a sleep mode during screen-off states, the disturbance on the regular usage of smartwatches is non-trivial but negligible. Although RF-based gesture recognition can be achieved by Wi-Fi antennae which are embedded on a small number of smartwatches, they are highly power-hungry hardware modules [47]. RF-based gesture recognition drains the watch battery very quickly, impeding other kinds of interactions and computations required by users.

5.3 Useful Features for PPG-based Gesture Recognition

In Table 4, we show that some features are good indicators of gestures on smartwatches, such as the mean value of the given signal and fast dynamic time warping (FDTW) distance, just to name a few.Results show that both classifiers gained higher accuracy using only a subset of features, compared to using all. This finding indicates that the quality of features, rather than the quantity, plays a critical role in accurate classification models. A large number of features with redundancy may not be helpful to improve the recognition performance. Relying on the results of our feature evaluation, future work can conduct further investigation using other classifiers to improve the accuracy or minimise the computational cost without sacrificing the accuracy.

In addition, this reduction of features can decrease the computational cost of training and using classification models. To train a model, the time complexity of Random Forests is $O(m \times n \times \log(n))$ where m is the number of features and n is number of instances; SVM with a linear kernel, $O(m \times n^2)$. If the classifier uses only 16 or 20 features to train a model, it can save more than 80% of computation as compared to using all 69 features on the watch's CPU. A direct benefit is the extra speed in training a model and real-time gesture recognition. Also, it is worth nothing that CPU consumes more power than any sensor and modules, except the screen, on smartwatches. Hence, the reduction of features is necessary to build energy-efficient and accurate gesture recognition systems, leading to longer time of interaction between users and smartwatches.

5.4 Implications for Interaction Design

We summarise a number of typical impacts of our work to current wearing smartwatches circumstances in daily life.

5.4.1 One-hand interaction.

Our PPG-based gesture recognition system facilitates one-handed interaction. Common scenarios whereby one-handed interaction can be useful include standing in a crowded bus, tram or train, cycling, or driving. Similar to other one-handed gesture input approaches, smartwatch applications can map a number of gestures 2 onto their functions. To support more functions than the number of gestures, applications can use a sequence of these gestures as a complex gesture. For instance, the message reader can define the sequence $clhand \rightarrow ophand$ as turning to next page.

In addition, our work system also allows eyes-free interaction. A typical use case is multi-device interaction where users use smartwatch as an input device for another device (e.g., a large display board) which users look at. Without gaze, it is hard for users to accurately press a button on the smartwatch screen. To enable such eye-free interaction, smartwatch applications can define a communication protocol which can encode and decode gestures based on our system. When users complete a predefined gesture in 2, the smartwatch applications should also provide audio or tactile feedback to users.

5.4.2 *Gesture set availability.*

For most of current mainstream smartwatches, such as the Moto 360 2*nd*, the LG Urbane 2*nd*, and the HuaWei Watch, etc., they offer 1 or 2 buttons on one side of the watch. They also typically include a touch screen with 4 single direction touching actions and single click actions for interaction. Besides, the Samsung Gear G3 has 2 buttons, and a touch screen. It also has a watch bezel that users can turn clockwise or anti-clockwise for quick selection. Overall, our proposed gesture set is sufficient to serve these target smartwatches on marketplace as input replacements.

5.4.3 *Example deployment of FinDroidHR.*

We intend to implement a system-level touch event simulation that allows FinDroidHR to fully replace the manufacturer's way of using buttons and touch screen for watch interaction. However, due to the Tizen Wearable OS system architecture, we can not intercept or generate the system touch or button events to replace them with FinDroidHR. Instead, we have built our own application and simulated those events in this application to evaluate FinDroidHR. Figure 27 shows 2 example use cases. The first one is to do the selection action without touching or turning the bezel by using *downhand* gesture with the *pinch* gesture as the click action so that users can interact with watch in a hand-free way. The second case is that users can use the *ophand* gesture to quickly launch another application from FinDroidHR without any touching. It offers a shortcut to favourite applications.

5.5 Limitations and Future Work

5.5.1 *Conflict gestures.*

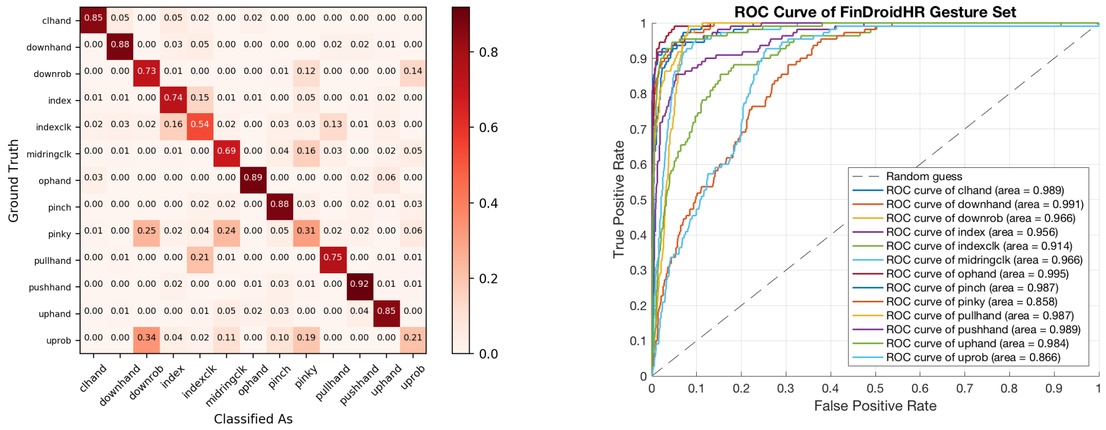
In the gesture design section 3.1, we propose a original set of 13 out of 26 gestures selected by the easy-to-do purpose shown in Table 1. However, we finally identify the set of 10 out of 13 gestures to be valid and effective. The main reason for the second selection is that we identified a few of gestures in conflict with others after initial classification attempt.

As we can see from both the confusion matrix shown in Figure 28a and ROC curve shown in Figure 28b, this classification attempt fails because the conflict situation happened. In this attempt, 3 of gestures are major false negatives, the *indexclk* of P, *pinky* of 2F and *uprob* of $P + 2F$. Meanwhile, the ROC curve of 13 gestures also indicates that the AUC values of these 3 gestures are evidently lower than others. From these, we find that the *uprob* was heavily in conflict with the *downrob*. Since these 2 gestures could be preformed reversely, the changes of blood volume behind the gestures might be similar and the classifiers can not properly distinguish them by the selected PPG features. Besides, the *pinky* is mainly in conflict with the *midringclk*, because we found that the movements of both gestures were not strong, and normally when the *midringclk* is performed, the pinky finger is affected to move with both middle and ring fingers. Moreover, the *indexclk* was partially in conflict with the *index* as using the same finger to perform different gestures might not have huge differences.

In our work, since the maximum frequency of PPG signal on the Samsung Gear S3 is offered as 100Hz, we can not increase it further to collect more samples to analyse whether there is a positive correlation between the data rate and the number of reliable gestures. Therefore, we decide to reduce the size of original gesture set to be 10 for the accuracy improvement. Thus, the idea of using PPG-based biosensor to detect hand gesture is limited for complex gesture recognition in our current work.

5.5.2 *Inherent PPG sensor limitations.*

Our work makes a number of assumptions in section 4. First, if users have health issues, such as obesity, diabetes, and arterial diseases, they may not be able to use FinDroidHR, as these conditions may result in an unstable heart rate which affects the accuracy of classification. It is unclear if blood flow artefacts can be automatically detected and removed under such conditions.



Ground Truth \ Classified As	clhand	downhand	downrob	index	indexclk	midringclk	ophand	pinch	pinky	pullhand	pushhand	uphand	uprob
clhand	0.85	0.05	0.00	0.05	0.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
downhand	0.00	0.88	0.00	0.03	0.05	0.00	0.00	0.00	0.00	0.02	0.02	0.01	0.00
downrob	0.00	0.00	0.73	0.01	0.00	0.00	0.00	0.01	0.12	0.00	0.00	0.00	0.14
index	0.01	0.01	0.00	0.74	0.15	0.01	0.01	0.00	0.05	0.00	0.01	0.02	0.00
indexclk	0.02	0.03	0.02	0.16	0.54	0.02	0.00	0.03	0.03	0.13	0.01	0.03	0.00
midringclk	0.00	0.01	0.00	0.01	0.00	0.69	0.00	0.04	0.16	0.03	0.00	0.02	0.05
ophand	0.03	0.00	0.00	0.00	0.00	0.00	0.89	0.00	0.00	0.00	0.02	0.06	0.00
pinch	0.00	0.00	0.00	0.00	0.02	0.02	0.00	0.88	0.03	0.00	0.02	0.01	0.03
pinky	0.01	0.00	0.25	0.02	0.04	0.24	0.00	0.05	0.31	0.02	0.02	0.00	0.06
pullhand	0.00	0.01	0.00	0.00	0.21	0.01	0.00	0.00	0.00	0.75	0.00	0.01	0.01
pushhand	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.03	0.01	0.00	0.92	0.01	0.01
uphand	0.00	0.01	0.00	0.00	0.01	0.05	0.02	0.03	0.00	0.00	0.04	0.85	0.00
uprob	0.00	0.00	0.34	0.04	0.02	0.11	0.00	0.10	0.19	0.00	0.00	0.00	0.21

Ground Truth
Classified As
0.8
0.6
0.4
0.2
0.0
ROC Curve of FinDroidHR Gesture Set
True Positive Rate
False Positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Random guess
ROC curve of clhand (area = 0.989)
ROC curve of downhand (area = 0.991)
ROC curve of downrob (area = 0.966)
ROC curve of index (area = 0.956)
ROC curve of indexclk (area = 0.914)
ROC curve of midringclk (area = 0.966)
ROC curve of ophand (area = 0.995)
ROC curve of pinch (area = 0.987)
ROC curve of pinky (area = 0.858)
ROC curve of pullhand (area = 0.987)
ROC curve of pushhand (area = 0.989)
ROC curve of uphand (area = 0.984)
ROC curve of uprob (area = 0.866)

(a) The Classification Matrix for original 13 gestures

(b) The ROC Curve of original 13 gestures

Fig. 28. Conflict Gestures Identification.

Second, since the PPG sensor requires the skin to be clean to absorb and reflect the green light, users who have tattoos on the wrist may also not be able to use FinDroidHR. Similarly, since the change of temperature can influence blood volume, our PPG system may not work in cold weather conditions.

Finally, the current FinDroidHR implementation offers a 3s detection time window and only one movement can be processed within the given window. At this moment, it cannot detect high-speed continuous gesture movements. For example, if users perform multiple gesture movements during the 3s detection time window, only the most significant movement will be identified by FinDroidHR as a valid gesture.

5.5.3 Future work. Future research can investigate other classification techniques to further improve the accuracy of gesture recognition. A possible direction is to use deep learning approaches, such as convolutional neural networks, to infer gestures from the PPG spectrogram without defining features. Future work can also introduce new features to help current classifiers (i.e., SVM and Random Forests) to improve accuracy. Since a significant number of smartwatches run Android OS, generalising our system to Android is a necessary step of deployment. Despite the fragmentation challenge (i.e.,different manufacturers customise Android differently), it may be possible to overwrite the OS kernel and sensor drivers to make our system compatible on Android. Ultimately, a new dynamic movement detection mechanism without using the detection time window is also possible to support the detection of continuous gesture movements.

6 CONCLUSION

In this paper we present FinDroidHR, a novel technique for hand gesture input relying on optical heart rate monitors embedded in most off-the-shelf smartwatches. We demonstrate that our system can recognise a set of 10 gestures with an an average accuracy of 90.55% and recall of 90.73%, without significant impact on battery life of a smartwatch. We also propose a technique of gradient-based noise removal and feature construction which achieves a substantial accuracy improvement for gesture recognition by 24%. Our system does not require bespoke hardware and can readily be implemented on existing smartwatches.

ACKNOWLEDGMENTS

This work is supported by the Australian Research Council (ARC) Discovery Grant, DP180103932.

A ALL EXTRACTED FEATURES

PPG All Features

No.	Feature Name	Feature Type	Description
1	meanS	Time Domain	Mean of the given signal
2	stdS	Time Domain	Std of the given signal
3	medianS	Time Domain	Median of the given signal
4	meanPKT	Time Domain	Mean of the Peak-to-peak Time Windows
5	stdPKT	Time Domain	Std of the Peak-to-peak
6	medianPKT	Time Domain	Median of the Peak-to-peak Time Windows
7	meanPKA	Time Domain	Mean of the Peak-to-peak Amplitude
8	stdPKA	Time Domain	Std of the Peak-to-peak Amplitude
9	medianPKA	Time Domain	Median of the Peak-to-peak Amplitude
10	disDTW	Time Domain	Fast Dynamic Time Warping (FDTW) Distance
11	meanPSD	Frequency Domain	Mean of the Power Spectral Density (PSD)
12	stdPSD	Frequency Domain	Std of the Power Spectral Density (PSD)
13	medianPSD	Frequency Domain	Median of the Power Spectral Density (PSD)
14	entroS	Frequency Domain	Shannon Entropy of the given signal
15	meanHPSDME	Frequency Domain	Mean of the high-pass signal PSD mean after Max Entropy process
16	stdHPSDME	Frequency Domain	Std of the high-pass signal PSD mean after Max Entropy process
17	medianHPSDME	Frequency Domain	Median of the high-pass signal PSD mean after Max Entropy process
18	meanHPSDST	Frequency Domain	Mean of high-pass signal PSD std after Max Entropy process
19	stdHPSDST	Frequency Domain	Std of high-pass signal PSD std after Max Entropy process
20	medianHPSDST	Frequency Domain	Median of high-pass signal PSD std after Max Entropy process
21	meanHPSDMED	Frequency Domain	Mean of high-pass signal PSD median after Max Entropy process
22	stdHPSDMED	Frequency Domain	Std of high-pass signal PSD median after Max Entropy process
23	medianHPSDMED	Frequency Domain	Median of high-pass signal PSD median after Max Entropy process
24	meanMAXHPSD	Frequency Domain	Mean of the maximum high-pass signal PSD
25	stdMAXHPSD	Frequency Domain	Std of the maximum high-pass signal PSD
26	medianMAXHPSD	Frequency Domain	Median of the maximum high-pass signal PSD
27	meanMINHPSD	Frequency Domain	Mean of the minimum high-pass signal PSD
28	stdMINHPSD	Frequency Domain	Std of the minimum high-pass signal PSD
29	medianMINHPSD	Frequency Domain	Median of the minimum high-pass signal PSD
30	meanMAXL	Frequency Domain	Mean of the maximum low-pass signal peak after Max Entropy process

31	stdMAXL	Frequency Domain	Std of the maximum low-pass signal peak after Max Entropy process
32	medianMAXL	Frequency Domain	Median of the maximum low-pass signal peak after Max Entropy process
33	meanMINL	Frequency Domain	Mean of the minimum low-pass signal peak after Max Entropy process
34	stdMINL	Frequency Domain	Std of the minimum low-pass signal peak after Max Entropy process
35	medianMINL	Frequency Domain	Median of the minimum low-pass signal peak after Max Entropy process
36	meanLME	Frequency Domain	Mean of the low-pass signal mean after Max Entropy process
37	stdLME	Frequency Domain	Std of the low-pass signal mean after Max Entropy process
38	medianLME	Frequency Domain	Median of the low-pass signal mean after Max Entropy process
39	meanLST	Frequency Domain	Mean of the low-pass signal std after Max Entropy process
40	stdLST	Frequency Domain	Std of the low-pass signal std after Max Entropy process
41	medianLST	Frequency Domain	Median of the low-pass signal std after Max Entropy process
42	meanLMED	Frequency Domain	Mean of the low-pass signal median after Max Entropy process
43	stdLMED	Frequency Domain	Std of the low-pass signal std after Max Entropy process
44	medianLMED	Frequency Domain	Median of the low-pass signal std after Max Entropy process
45	meanE	Frequency Domain	Mean of the high-pass signal Entropy after Max Entropy process
46	stdE	Frequency Domain	Std of the high-pass signal Entropy after Max Entropy process
47	medianE	Frequency Domain	Median of the high-pass signal Entropy after Max Entropy process
48	maxLFDTW	Frequency Domain	Maximum low-pass peak by the first Discrete Wavelet Transform (DWT) process
49	maxLMINE	Frequency Domain	Maximum low-pass peak by the minimum high-pass signal Entropy after Max Entropy process
50	maxLMAXE	Frequency Domain	Maximum low-pass peak by the maximum high-pass signal Entropy after Max Entropy process
51	minLFDTW	Frequency Domain	Minimum low-pass peak by the first Discrete Wavelet Transform (DWT) process
52	minLMINE	Frequency Domain	Minimum low-pass peak by the minimum high-pass signal Entropy after Max Entropy process
53	minLMAXE	Frequency Domain	Minimum low-pass peak by the maximum high-pass signal Entropy after Max Entropy process

54	meanLFDTW	Frequency Domain	Mean low-pass signal by the first Discrete Wavelet Transform (DWT) process
55	meanLMINE	Frequency Domain	Mean low-pass signal by the minimum high-pass signal Entropy after Max Entropy process
56	meanLMAXE	Frequency Domain	Mean low-pass signal by the maximum high-pass signal Entropy after Max Entropy process
57	stdLFDTW	Frequency Domain	Std low-pass signal by the first Discrete Wavelet Transform (DWT) process
58	stdLMINE	Frequency Domain	Std low-pass signal by the minimum high-pass signal Entropy after Max Entropy process
59	stdLMAXE	Frequency Domain	Std low-pass signal by the maximum high-pass signal Entropy after Max Entropy process
60	medianLFDTW	Frequency Domain	Median low-pass signal by the first Discrete Wavelet Transform (DWT) process
61	medianLMINE	Frequency Domain	Median low-pass signal by the minimum high-pass signal Entropy after Max Entropy process
62	medianLMAX	Frequency Domain	Median low-pass signal by the maximum high-pass signal Entropy after Max Entropy process
63	EFDTW	Frequency Domain	High-pass signal Entropy by the first Discrete Wavelet Transform (DWT) process
64	minE	Frequency Domain	Minimum high-pass signal Entropy after Max Entropy process
65	maxE	Frequency Domain	Maximum high-pass signal Entropy after Max Entropy process
66	maxGRAE	Frequency Domain	Maximum gradient of high-pass signal Entropy after Max Entropy process
67	meanGRAE	Frequency Domain	Mean gradient of high-pass signal Entropy after Max Entropy process
68	stdGRAE	Frequency Domain	Std gradient of high-pass signal Entropy after Max Entropy process
69	medianGRAE	Frequency Domain	Median gradient of high-pass signal Entropy after Max Entropy process

Table 5. FinDroidHR All Feature Set

REFERENCES

[1] Gregory D Abowd. 2012. What next, ubicomp?: celebrating an intellectual disappearing act. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 31–40.

[2] Nasimuddin Ahmed, Rohan Banerjee, Avik Ghose, and Arijit Sinharay. 2015. Feasibility Analysis for Estimation of Blood Pressure and Heart Rate using a Smart Eye Wear. In *Proc. WearSys 2015*. ACM Press, 9–14.

[3] Nazneen Akhter, Sumegh Tharewal, Vijay Kale, Ashish Bhalerao, and K.V. Kale. 2016. Heart-Based Biometrics and Possible Use of Heart Rate Variability in Biometric Recognition Systems. *Advances in Intelligent Systems and Computing* 395 (2016), 15–29.

[4] Daniel Ashbrook, Kent Lyons, and Thad Starner. 2008. An Investigation into Round Touchscreen Wristwatch Interaction. In *Proc. MobileHCI 2008*. ACM Press, 311–314.

[5] IEEE Standards Association. 2003. IEEE Standard for Transitions, Pulses, and Related Waveforms. In *IEEE Std. 181-2003*. IEEE Press.

[6] Hamed Azami, Karim Mohammadi, and Behzad Bozorgtabar. 2012. An Improved Signal Segmentation Using Moving Average and Savitzky-Golay Filter. *Journal of Signal and Information Processing* 3 (2012), 39–44. https://doi.org/10.4236/jsip.2012.31006

[7] Nick Barnes. 2005. Improved Signal To Noise Ratio And Computational Speed For Gradient-Based Detection Algorithms. In *International Conference on Robotics and Automation 2005*. IEEE Press, 4661–4667.

[8] Patrick Baudisch and Gerry Chu. 2009. Back-of-Device Interaction Allows Creating Very Small Touch Devices. In *Proc. CHI 2009*. ACM Press, 1923–1932.

[9] Jorge Blasco, Thomas M. Chen, Juan Tapiador, and Pedro Peris-Lopez. 2016. A Survey of Wearable Biometric Recognition Systems. In *Proc. CSUR 2016*. ACM Press.

[10] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. In *Proc. UIST 2013*. ACM Press, 255–260.

[11] Jikai Chen, Yanhui Dou, Yang Li, and Jiang Li. 2016. Application of Shannon Wavelet Entropy and Shannon Wavelet Packet Entropy in Analysis of Power System Transient Signals. *Entropy 2016* 18 (2016), 14. https://doi.org/10.3390/e18120437

[12] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D Input Using Two Magnetic Sensors. In *Proc. UIST 2013*. ACM Press, 237–244.

[13] Sangita Das, Saurabh Pal, and Madhuchhanda Mitra. 2016. Real Time Heart Rate Detection from PPG Signal in Noisy Environment. In *Proc. ICICPI 2016*. IEEE Press, 70–73.

[14] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex: Low-Power Gesture Input with Wrist-Worn Pressure Sensors. In *Proc. UIST 2014*. ACM Press, 161–166.

[15] Tizen Developers. 2017. Heart Rate Monitor LED Green Sensor. (2017). https://developer.tizen.org/development/guides/native-application/location-and-sensors/device-sensors#hrm_ir

[16] Florian Grutzmacher, Johann-Peter Wolff, and Christian Haubelt. 2015. Exploiting Thread-Level Parallelism in Template-Based Gesture Recognition with Dynamic Time Warping. In *Proc. WOAR 2015*. ACM Press, 6.

[17] Sidhant Gupta, Dan Morris, Shwetak N Patel, and Desney Tan. 2012. SoundWave: Using the Doppler Effect to Sense Gestures. In *Proc. CHI 2012*. ACM Press, 1911–1914.

[18] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proc. CHI 2016*. ACM Press, 1526–1537.

[19] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *Proc. NSDI 2014*. ACM Press, 303–316.

[20] Frederic Kerber, Markus Lochtefeld, Antonio Kruger, Jess McIntosh, Charlie McNeill, and Mike Fraser. 2016. Understanding Same-Side Interactions with Wrist-Worn Devices. In *Proc. NordiCHI 2016*. ACM Press.

[21] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. 2007. The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface. In *Proc. ISWC 2007*. IEEE Press, 1–8.

[22] Min Sik Kim, Taekhyun Kim, Yong-June Shin, Simon S. Lam, and Edward J. Powers. 2008. A wavelet-based approach to detect shared congestion. In *IEEE/ACM Trans. Netw 2008*. IEEE Press, 763–776.

[23] Vassilis Kostakos and Mirco Musolesi. 2017. Avoiding Pitfalls When Using Machine Learning in HCI Studies. *INTERACTIONS* (2017).

[24] Sven Kratz and Michael Rohs. MobileHCI. Hoverflow: Exploring Around-Device Interaction with IR Distance Sensors. In *Proc. MobileHCI 2009*. ACM Press, 4.

[25] Gierad Laput, Robert Xiao, Xiang 'Anthony' Chen, Scott E. Hudson, and Chris Harrison. 2014. Skin Buttons: Cheap, Small, Low-Power and Clickable Fixed-Icon Laser Projections. In *Proc. UIST 2014*. ACM Press, 389–394.

[26] YANN LECUN, LEON BOTTOU, YOSHUA BENGIO, and PATRICK HAFFNER. 1998. Gradient-Based Learning Applied to Document Recognition. In *Proc. IEEE 1998*. IEEE Press, 2278–2324.

[27] Zhiyuan Lu, Xiang Chen, Qiang Li, Xu Zhang, and Ping Zhou. 2014. A Hand Gesture Recognition Framework and Wearable Gesture-Based Interaction Prototype for Mobile Devices. In *Proc. HUMAN-MACHINE SYSTEMS 2014*. IEEE Press, 293–299.

[28] Yuka Maeda, Masaki Sekine, and Toshiyo Tamura. 2011. Relationship Between Measurement Site and Motion Artifacts in Wearable Reflected Photoplethysmography. *Journal of Medical Systems* 39, 5 (2011), 969–976.

[29] HENRY MARTIN, PAUL GROVES, and MARK NEWMAN. 2016. The Limits of In-Run Calibration of MEMS Inertial Sensors and Sensor Arrays. *NAVIGATION: Journal of The Institute of Navigation* 63 (2016).

[30] Jess McIntosh, Asier Marzo, Mike Fraser, and Carol Phillips. 2017. EchoFlex: Hand Gesture Recognition using Ultrasound Imaging. In *Proc. CHI 2017*. ACM Press.

[31] Kei Nakatsuma, Hiroyuki Shinoda, Yasutoshi Makino, Katsunari Sato, and Takashi Maeno. 2011. Touch Interface on Back of the Hand. In *Proc. SIGGRAPH 2011*. ACM Press, 1.

[32] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proc. CHI 2016*. ACM Press, 1515–1525.

[33] Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. WatchIt: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. In *Proc. CHI 2013*. ACM Press, 1451–1460.

[34] Raj Rakshit, V Ramu Reddy, and Parijat Deshpande. 2016. Emotion Detection and Recognition using HRV Features Derived from Photoplethysmogram Signals. In *Proc. ERM4CT 2016.* ACM Press.

[35] MARKO ROBNIK-SIKONJA and IGOR KONONENKO. 2003. Theoretical and Empirical Analysis of ReliefF and RReliefF. In *Machine Learning, 53, 2003.* 23–69.

[36] T. Scott Saponas, Chris Harrison, and Hrvoje Benko. 2011. PocketTouch: Through-Fabric Capacitive Touch Input. In *Proc. UIST 2011.* ACM Press, 303–308.

[37] T. Scott Saponas, Desney S. Tan, Dan Morris, Jim Turner, and James A. Landay. 2010. Making Muscle-Computer Interfaces More Practical. In *Proc. CHI 2010.* ACM Press, 851–854.

[38] Mads Soegaard and Rikke Friis Dam. 2013. *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* (2nd ed.). The Interaction Design Foundation.

[39] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. 2015. WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices. In *Proc. Mobicom 2015.* ACM Press, 77–89.

[40] S.K. Deric Tang, Y.Y. Sebastian Goh, M.L. Dennis Wong, and Y.L. Eileen Lew. 2016. PPG signal reconstruction using a combination of discrete wavelet transform and empirical mode decomposition. In *Proc. ICIAS 2016.* IEEE Press.

[41] H. Emrah Tasli, Amogh Gudi, and Marten den Uyl. 2014. Integrating Remote PPG in Facial Expression Analysis Framework. In *Proc. ICMI 2014.* ACM Press, 74–75.

[42] Ihor Vasyltsov and Seunghwan Lee. 2015. Entropy Extraction from Bio-Signals in Healthcare IoT. In *Proc. IoTPTS 2015.* ACM Press, 11–17.

[43] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jurgen Steimle. 2015. iSkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. In *Proc. CHI 2015.* ACM Press, 2991–3000.

[44] Martin Weigel, Aditya Shekhar, Nittala1 Alex Olwal, and Jurgen Steimle. 2017. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *Proc. CHI 2017.* ACM Press, 3095–3105.

[45] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. 2016. Serendipity: Finger Gesture Recognition using an Off-the-Shelf Smartwatch. In *Proc. CHI 2016.* ACM Press, 3847–3851.

[46] Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In *Proc. CHI 2014.* ACM Press, 193–196.

[47] Liu Xing and Qian Feng. 2016. Measuring and optimizing android smartwatch energy consumption: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 421–423.

[48] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2014. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition using Smartwatch. In *Proc. HotMobile 2015.* ACM Press.

[49] Yalan Ye, Wenwen He, Yunfei Cheng, Wenxia Huang, and Zhilin Zhang. 2017. A Robust Random Forest-Based Approach for Heart Rate Monitoring Using Photoplethysmography Signal Contaminated by Intense Motion Artifacts. *MDPI* 385 (2017). https://doi.org/10.3390/s17020385

[50] Xiaobo Zhang and Xiangchu Feng. 2014. A New Gradient-based Nonlinear Diffusion Method Applied to Image Denoising. In *Proc. PIC 2014.* IEEE Press, 321–315.

[51] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proc. CHI 2016.* ACM Press, 1491–1503.

Received August 2017; revised November 2017; accepted January 2018