Human Activity Recognition and Movement Visualization using Smartwatches

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Md Abu Sayeed Mondol

May 2020

APPROVAL SHEET

This Dissertation
is submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Author Signature: ______________________

This Dissertation has been read and approved by the examining committee:

Advisor: John A. Stankovic

Committee Member: Alfred C. Weaver

Committee Member: Jack W. Davidson

Committee Member: Laura E. Barnes

Committee Member: John C. Lach

Committee Member: ______________________

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

May 2020

Abstract

Automatic recognition of human activities is attributed with great importance for its widespread and potential applications in different domains, including healthcare, safety, behavior monitoring, energy management, manufacturing, and elderly care. Human Activity Recognition (HAR) is the cornerstone for most of these applications, and so the performance of such an application largely depends on the accuracy and robustness of the underlying activity recognition models. However, activity recognition is challenging, particularly in natural settings, due to issues like confounding gestures present in different activities, large diversity in performing the same activity, a wide range of possible human activities, and usability of the solutions. Also, on-device processing, required by many real-time applications, is challenging due to limited resources available in the wearable devices. In contrast to the state-of-the-art methods that mostly emphasize feature engineering and classification techniques for HAR, our works focus on leveraging the orientation of the device and the distribution of the data in developing more efficient, robust, and accurate solutions for activity recognition. We developed a novel and efficient algorithm for detecting eating events from wrist-worn accelerometers. The algorithm improves eating gesture detection f1-score by 0.19 with less than 20% computation compared to a baseline method. We developed and deployed a comprehensive system for monitoring family eating dynamics that uses the solution for eating event detection. We present a solution for hand washing detection that reduces the false positives from unseen activities by about 77% and improves the overall F1-score by 0.17 compared to a baseline method. Data visualization is useful in understanding the data and their characteristics, and it is a fundamental step toward developing data-driven solutions. A novel visualization method that provides additional utility to the existing methods is of utmost desire. We developed a novel method for visualizing movement and orientation using inertial

sensors. We analyzed a dataset related to smoking activity recognition and provided several insights through representing the data using our method. We developed an efficient solution for smoking puff detection using insights from the data. Additionally, we illustrated how our method can be used to monitor movement patterns related to several rehabilitation exercises. Reminders are often used in activity recognition systems. Though the focus of the thesis is activity recognition and movement visualization, we developed a voice-based interactive reminder system to close the loop. Finally, this dissertation describes an easy-to-use tool that we have developed to collect sensor data from smartwatches.

Acknowledgements

First and foremost, I thank my advisor Professor John A. Stankovic, who guided me all through my Ph.D. studies. He taught me thinking critically, conducting impactful research, and expressing myself more clearly. He inspired me to solve problems for the greater benefits of the society and the research community. He has been my mentor, guardian, and inspiration, not only for research but also in different aspects of my life. I am fortunate to have the opportunity to work with this extraordinary person.

I would like to take this opportunity to express my eternal gratitude to my family members, who have always been there to provide unconditional love and support for me. They encourage and help me to overcome the ups and downs in my life and to move forward. I also thank all my friends who are always there for me. I especially thank Anindya Tahsin Prodhan, who has been consistently helping and supporting me from the very beginning of my life in the United States.

I am thankful to the respected members of my dissertation committee. They have supported me consistently, and their feedback has helped me make this dissertation better. I would like to thank the instructors for the different courses I took that laid the foundation of my research, development, and problem-solving skills.

I am grateful to my lab mates, who supported me all the way. It was a great pleasure working with them. I have learned a lot from them, and their feedback was valuable to my work. I would especially like to mention Ifat, Sarah, and Meiyi, with whom I have collaborated on several projects. I thank all the office staff of the Computer Science Department and the Link Lab of the University of Virginia who were there to help when I needed it.

I thank all my collaborators, including but not limited to Professor John Lach and Ridwan Alam from the University of Virginia, Professor Donna Spruijt-Metz, Professor Kayla de la Haye, and Brooke Bell from the University of Southern California, and Dr. Laura Lee from the University of Virginia hospital. I am grateful to all the participants of the research studies who contributed to developing the datasets used in this dissertation.

Finally, I am grateful to the University of Virginia for giving me the opportunity and supporting me throughout the study.

To my parents, my wife, and my son.

Contents

Chapter 1

Introduction

Automatic recognition of human activities is attributed with great importance for its widespread and potential applications in different domains, including healthcare, safety, behavior monitoring, energy management, manufacturing, and elderly care [1–7]. Human Activity Recognition (HAR) is the cornerstone for most of these applications, and so the performance of such an application largely depends on the accuracy and robustness of the underlying activity recognition models. For example, the performance of a hand wash monitoring system in a hospital is mainly determined by the accuracy and the robustness of the system in detecting hand wash gestures of the caregivers in the hospital settings.

The sensing approaches for activity recognition can be grouped into two main categories: in-situ and wearable. In-situ sensors are embedded in the environment, and such systems usually require no effort from the users to be used in daily life. On the other hand, wearable devices are attached to the body, and some effort is required in putting on/off and recharging the devices. However, in-situ systems suffer from several limitations that the wearable systems are almost free from. Sensing capability of in-situ sensors is limited in certain areas, and so they cannot be used in detecting activities of a subject when the subject is beyond those areas. In contrast, wearable sensors can be used to detect the activities of a subject anywhere. Additionally, installation and maintenance of the in-situ sensors usually require significant effort and cost, and the building infrastructure often needs to be invaded for these purposes. Most of the in-situ systems cannot identify the subject performing the activities, and so they are not suitable where subject identification is important. Some in-situ systems for HAR that use cameras can detect the subject to some extent. One of the major issues of camera-based systems is privacy, and many people do not feel comfortable to be recorded by a camera. In addition to the fact that the camera-based solutions inherit the limitations of the in-situ systems as mentioned earlier, their performance also depends on several factors including distance and orientation of the subject with respect to the camera, and environmental attributes like lighting. Camera-based systems are suitable for domains like security (e.g. intrusion monitoring) and gaming, but not for most activity recognition tasks.

Many people use electronic reminder systems in daily life. Smartphone apps are widely used as reminders. Smartwatches available today are also capable of providing reminders. The smartwatches are often more convenient and useful in providing reminders compared to smartphones. Most of the time users hold the phone in one hand and use it with the other. Occupation of the hands for using the phones and the attention required to use the apps justify that smartphone-based systems are more intrusive and less convenient than smartwatches. Also, smartphones provide limited effectiveness in different contexts [8]. A user is very likely to miss a reminder at home when the phone is far enough from his/her location at the time the reminder is given. For instance, a user may miss a reminder when he/she is busy in the kitchen, but the phone is in the bedroom far away from the user. Moreover, remainders may be missed while listening to songs, TVs or videos even if the phone is located near the user. In many situations like in meetings and classrooms, smartphones typically need to be kept silent, and users often forget to return the devices back to the non-silent mode when silence is not required anymore. It is very likely that a user misses some reminders in such scenarios.

One of the major advantages of wearable devices is that these devices can be used for other useful purposes beyond activity recognition and reminders. For example, smartwatches are used as a timepiece as well as for notifications, controlling functionalities for other devices (e.g., smartphones), and measuring vital signs. Biosensors can be embedded in wearable devices for real-time measurement of several vital signs such as heart rate, body temperature, and skin conductivity. The biosensor data provide useful insights about the user's health condition and well being, and so they are very important toward providing effective treatments and prescriptions. Many people use smartwatches now, and the use of this device is increasing rapidly [9]. So, smartwatch apps can be readily delivered to a huge group of population. Activity recognition and reminders can be implemented on these devices as an additional service with no/very low additional cost. The competitiveness of the wearable devices motivates their use in recognizing human activities and providing reminders.

Activity recognition is challenging, particularly in natural settings, due to issues like confounding gestures present in different activities, diversity in performing the same activity, and a wide range of possible human activities [10]. Also, on-device processing, required by many real-time applications, is challenging due to limited resources available in the wearable devices. In contrast to the state-of-the-art methods that mostly emphasize on feature engineering and classification techniques for HAR, our works focus on leveraging the orientation of the device and the distribution of the data in developing more efficient, robust and accurate solutions for activity recognition. Most of the existing solutions for activity recognition are data-driven where machine learning models are trained and evaluated using a set of data. Understanding the data and their characteristics is fundamental toward developing data-driven solutions. Researchers use different visualization techniques (e.g., histogram, bar chart, scatter plot, and line plot) for analyzing data, presenting results, and communicating insights. Different visualization techniques are useful for different types of data, and they often complement each other. A novel visualization method that would provide additional utility to the existing methods is of utmost desire. However, such novel methods for visualization are rarely invented, particularly in the area of wearable and mobile sensing. We developed a novel method for visualizing movement and orientation using inertial sensors. Our visualization method is not a replacement to the existing methods; rather it provides additional utility toward better understanding the movements associated with different activities.

In terms of providing reminders, a smartwatch is almost free from the problems of smartphones, as mentioned earlier. However, a major challenge in developing interactive systems for smartwatches is their form factor. A watch can provide/take very limited information to/from the user via the display. Voice can be used to address this limitation, but it is not suitable for on-device voice processing and inference because the computational resources available in the smartwatches are very limited. Voice data transmission to a server drains significant energy from the watch, and it does not work well when network connectivity is poor. We have designed and developed a keyword-based interactive reminder system. Keyword detection requires low computational resources, and so can be implemented on devices with limited resources like the smartwatches.

Data is essential to develop and evaluate any data-driven solution. Researchers, especially in the area of ubiquitous and wearable systems, often spend a significant amount of effort and time in developing devices and/or apps for data collection. Most of the apps and devices are customized with limited options, and they are often not available publicly. We developed an easy-to-use app for sensor data collection using commercial off-the-shelf Android smartwatches. The app is publicly available, and it facilitates prompt data collection without requiring expertise and effort for custom device/app development. The app is being extended by others for Apple watches.

1.1 Challenges

Inter-class similarity and intra-class variability are the major challenges for any classification task. These challenges are severe for activity recognition, particularly in natural settings where confounding gestures are present in different activities, and the same activity can be performed in different ways. Wearable based systems also come with additional challenges like limited resources and usability. We discuss the challenges below in more detail.

1.1.1 Confounding Gestures

Similar motions are often present in different activities performed in daily life. For example, similar wrist motions are usually present in both hand washing and dish washing activities. People perform a wide range of activities in daily life where confounding gestures among different activities are prevalent. As similarity in the gestures results in similarity in the underlying sensor data, confounding gestures among different activities reduce the performance of the recognition tasks. For example, detecting eating bites has been shown to be accurate when monitoring in controlled settings such as eating specific foods at a table with fork and knife and few, if any, confounding gestures. Detecting eating bites, in general, is a challenging task [11].

1.1.2 Diversity in the Same Activity

The ways an activity is performed by different individuals often differ. Even the same person often performs the same activity differently. The contexts of the user as well as the tools used to perform an activity also make differences in the gestures. For example, the hand to mouth gestures for eating differ from one food type to another. Even the gestures for eating the same food differ based on whether the food is eaten with a spoon, fork, or bare hand. The posture and ambulation of the person also affect gesture patterns and corresponding sensor data.

1.1.3 Wide Range of Possible Activities

Activity recognition tasks are generally focused on detecting a limited number of activities. We refer any activity other than the activities of interests a NULL activity. It is extremely difficult, if not impossible, to enumerate all possible human activities and thus the set of NULL activities. State-of-the-art solutions are mostly data-driven, and so the performance of the solutions largely depends on the data used to develop the models. The activity recognition models are usually developed and evaluated using data from a limited number of NULL activities, often in lab settings. Consequently, the models might perform poorly in the free-living context where many other NULL activities could be present.

1.1.4 Limited Resources of Wearable Devices

The computational resources (e.g., processor, memory) and the battery life of wearable devices are very limited. Any technique that requires significant computation, memory, or energy is not suitable to be implemented on these devices. However, many applications require real-time processing. It is challenging to develop solutions that are efficient enough to be implemented in wearable devices while providing desired performance.

1.1.5 Usability

Usability is perhaps the most important issue for a wearable system when used in daily life. However, this issue is often overlooked, and solutions are developed focusing on performance without considering user convenience. Many solutions place sensors at multiple positions on the body [12–14], and they are not convenient to be used in daily life. Also, sensor placement at a specific position might provide better accuracy for some activities. For example, it is possible to detect eating with more accuracy using acoustic sensors placed on the throat [15]. However, such systems are not convenient to be used in daily life. The usability of wrist devices is well established, but it very challenging to detect different activities using sensor data from a single smartwatch.

1.2 Thesis Statement

"Human activity can be recognized with more efficiency, accuracy, and robustness when using wrist wearable sensors by leveraging the orientation of the device, the distribution of the data, and aided by new visualization techniques."

To be more specific, we investigated the following suppositions:

- The orientation of a wrist provides useful information about an activity performed by the corresponding hand. We can leverage the orientation of the device, that actually presents the orientation of the wrist, to better separate an activity from others.

- We can use orientation to filter out most of the NULL activities very efficiently before using classification techniques like neural networks that require significant resources. Thus, the overall resource required to run the solution is reduced.

- Most of the state-of-the-art solutions for activity recognition are neural network based. It is not possible to collect all possible NULL activities to train the networks, and consequently, the networks perform worse in the presence of unseen NULL activities. Each layer of a neural network transforms its input features to another feature space, and the output of the penultimate layer represents the final embedding of the input. We can use the distribution of the output of the penultimate layer to build solutions that are more robust and accurate against unseen NULL activities.

- The orientation of the device, as represented by a quaternion, can be decomposed into unit vectors. We can develop useful solutions for visualizing the movement and orientation of the device by placing the unit vectors on a unit sphere and dividing the area of the sphere into some cells.

1.3 Contributions

We have developed a set of solutions addressing the challenges mentioned earlier. Neural networks are usually more effective in recognizing activities from wearable sensor data than other machine learning techniques like Random Forest and Support Vector Machines [12]. Most of the state-of-the-art solutions for activity recognition use neural networks that address the challenge of confounding gestures and diversity in the same activity to some extent [12–14, 16, 17]. We also use neural networks in our solutions, but our works are not focused on the architecture or hyper-parameter tuning of the network. Our solutions improve performance or reduce the overall resource consumption by leveraging the orientation of the device and the distribution of the data. All of our solutions use only a single wrist device, making them very convenient and easy to use.

The major contributions of this dissertation are:

- We present a novel and efficient algorithm for detecting eating events from wrist-worn accelerometers. The algorithm improves eating gesture detection f1-score by 0.19 with less than 20% computation compared to the state-of-the-art methods.

- We designed, developed and deployed MFED, a comprehensive system for monitoring family eating dynamics. To the best of our knowledge, this is the first system for monitoring FED. In addition to monitoring eating events, the system collects data on theoretically relevant features of family eating dynamics including: individual states (mood, stress, hunger/satiety) and characteristics of eating events (the type of eating occasion, who they are eating with). We deployed the system for approximately two weeks in 20 real homes with a total of 74 participants.

- We have developed a robust and effective solution for hand washing detection by recognizing out-of-distribution instances. Our method reduces the false positives from unseen activities by about 77% and improves overall F1-score by 0.17. We have collected a dataset consisting of hand washing and several other activities and additionally used a public dataset to demonstrate the effectiveness and robustness of our solution.

- We have developed a novel method (consisting of multiple primitives) for movement visualization using wearable inertial sensors. Our visualization method is based on quaternions and helps to analyze and understand data that can further help in developing effective and efficient solutions. We analyzed a dataset related to smoking activity recognition and provided several insights through representing the data using our method. We developed an efficient solution for smoking puff detection using insights from the data. Additionally, we illustrated how our method can be used to monitor movement patterns related to several rehabilitation exercises.

- We have developed a keyword-based voice interactive reminder system for smartwatches that reduces errors in detecting user command by 20.39%.

- We have developed an easy-to-use app for sensor data collection from commercial off-the-shelf Android smartwatches. The app facilitates prompt data collection without requiring expertise and effort for custom device/app development.

- We have developed three datasets that would be valuable to move the research forward. The datasets are on hand washing activity, eating activity, and EMA responses related to family eating dynamics.

1.4 Dissertation Outline

The organization of the rest of this dissertation is given below.

- Chapter 2 discusses the state-of-the-art works related to activity recognition, movement visualization, reminders, and data collection tools using wearable sensors.

- Chapter 3 describes the MFED system and the method for detecting eating gestures. It discusses the performance of the eating gesture detection method as well as the results from the deployments of the system.

- Chapter 4 presents the solution for hand washing detection and demonstrates its robustness and effectiveness in addressing NULL activities.

- Chapter 5 introduces the method for movement visualization and explains the utility of the method using examples of several activities of daily living. It demonstrates the use of our method in data analysis, solution development, and movement monitoring.

- Chapter 6 describes the keyword-based interactive reminder system as well as experimental results involving both native and non-native speakers.
- Chapter 7 presents the tool for data collection. It describes different features of the tool and the format of the collected data. The tool is available in a public repository [18].
- Chapter 8 summarizes the contributions presented in this dissertation. It concludes the dissertation by providing directions for future research.

Chapter 2

Related Work

2.1 Monitoring Family Eating Dynamics

Obesity, diabetes, cardiovascular diseases, and many other human health applications motivate the research in the area of automated dietary monitoring and eating detection. Consequently, most of the related works in this area attempt to estimate energy or calorie intake, characterize food, and quantify mass intake by monitoring the eating gestures or sips as well as the eating events such as meals or snacks [19, 20]. State-of-the-art approaches have been using various wearable sensing modalities to detect either eating gestures or eating events. These approaches vary not only on the employed sensing modalities, but also on the placement of such sensors on the body as well as the signal processing, classification, and learning approaches [21–23].

Advances in imaging technologies have provided the opportunity to continuously carry a portable or wearable camera which can capture dietary moments. With image processing techniques such as normalized cut based segmentation and SVM based classification [24], scale invariant feature transform based 3D reconstruction [25], or cloud-based human computing platforms [26, 27], images captured from a hand-held, or on-body camera have been used to assess users' dietary behavior. These approaches show the potential in addressing multiple aspects of dietary applications, such as ingested mass quantification, food intake characterization, as well as energy intake estimation. Yet, privacy concerns have restricted the ability of these approaches to be extended to real-world deployments [20, 22].

Acoustic signals of chewing and swallowing have also been investigated as indicators of dietary activity. In early attempts, miniature microphones were placed inside the ear canal as earplugs, which acquired acoustic signal related to eating episodes, and those signals were filtered and classified using CART, NB, KNN, and HMM classifiers to detect mastication and swallowing in controlled settings [28, 29]. These approaches showed high recognition rate (upto 99%), but only on a handful of foods (e.g. potato chips, apple) with a small population (4-5 subjects) in controlled settings. A throat microphone was used to capture better swallowing sound in addition to a microphone in the ear canal for chewing sounds to collect a total of 64.5 hours of data [30–32]. These data were used for eating episode detection by using frequency and wavelet features, and clustering and affinity propagation were used to estimate counts of food items in a certain episode. In real-world settings, the acoustic signal becomes corrupted by environmental noise and surrounding sounds, and thus hampers the performance of those audio-based methods. To avoid the impacts of environmental noise, methods have been proposed attempting various noise subtraction algorithms based on energy and spectral properties of the eating related sounds and the environmental noise collected using two independent microphones packaged as a hearing aid [33–35]. With similar motivation, a uni-directional microphone based neckpiece was designed to capture acoustic signals from the throat while reducing external noise, and was able to recognize twelve activities at 79.5% F-score in lab setting and four activities at 71.5% F-score in realistic setting [36]. With assumptions that a body-worn piezoelectric microphone would be less affected by environmental noise, a neckpiece sensor was designed using such microphone, yet didn't take off for real-world evaluation of eating episode detection [37, 38]. A wrist-worn microphone was used in a semi-controlled study to acquire data around eating episodes and used random forest classification on clustered audio frames to achieve 79.8% F-score [39]. Other modalities such as electrogluttography (EGG) and electromyography (EMG) have also been proposed to capture jaw motion and swallowing events in a similar manner [40–42]. Also, combining audio with images from an ear-worn device were proposed for both eating gesture detection and food identification based energy intake estimation [43]. These above mentioned works demonstrate that audio based systems can acquire good performance in detecting eating gestures such as eating gesture and ingestion in controlled environments, but fail to translate those performance in real-world scenarios [44, 45].

Human motion sensing has shown notable potential in inferring various activities of daily living including eating [46]. Specially with the recent proliferation of such sensing modalities in pervasive and ubiquitous computing devices, such sensors require much less effort to deploy in real-world while providing signals with reliable quality. Consequently, many researchers have attempted to use motion as a supplementary modality for detecting eating activity. Motion signal from lower and upper arm, audio from ear canal and neck collar, and EMG from throat were combined to perform feature distance based classification of eating events [47]. Wrist motion was used in combination with chewing and swallowing sound to acquire better performance in eating gesture detection and food characterization [19, 48]. While applying pattern recognition with these sensor streams provide around 80% accuracy in controlled settings, such methods were not evaluated in long-term free-living settings. [49] proposed a dual location motion based system by using both smart watch and Google Glass accelerometers, but evaluated on a controlled protocol of activities. Similarly, motion data from head movement was captured using a glass-mounted accelerometer and KNN classification was performed on small amount of lab collected data [50].

Considering the importance of data collection from the real-world, recent approaches are focusing more on wrist-worn motion sensing due to its unobtrusiveness and higher user conformance. Focusing on energy intake estimation, a rule-based eating gesture counting algorithm and a wrist worn sensing device named eating gesture counter were proposed [51–53]. They were evaluated in two controlled settings - a lab setup and an instrumented cafeteria; performance evaluations in general scenarios require further attention. Also, the participants had to manually press a button to start the data collection and eating gesture counting. Among other approaches for eating gesture detection, motif based template matching based pattern segmentation followed by Random Forest classification were evaluated in semi-controlled environments [54]. Similarly, a micro-movement segmentation based HMM classifier was designed to detect eating gestures in a known eating episode [55]. These approaches highlight the need for an automated meal detection method for those to be applicable in real-world scenarios.

With the motivation for free-living data collection and application, Dong et al. attempted to develop an eating period detection algorithm [56, 57]. They strapped a smart phone on the wrist of the user's dominant hand to utilize the accelerometer and the gyroscope of the phone and collected data for one-day per participant from 43 participants in free-living settings. The proposed algorithm was based on an assumption that eating periods are preceded and succeeded by vigorous motion patterns, which is too simplistic against real-world confounding gestures. Though this work found correlation between energy intake and eating gesture counts, but was not evaluated against long-term inter- and intra-person variations and uncertainty.

Recent approaches are emerging to translate the in-lab high performance systems to out-of-lab realistic long-term settings. For example, lab data collected from 20 subjects wrist-worn motion sensor were used to train models which were evaluated in free-living setting on 7 participants each for 24 hour and on 1 participant for a month [58]. To acquire ground truth on food intake, a first-person point-of-view camera was worn by the participants. Even though the data lacks inter- and intra-person variation in eating behavior; yet this was a major step toward implementing realistic solution that can be used by various dietary monitoring applications. Following that trend, Mirtchouk et al. [45, 59] have conducted an investigative study on eating recognition using head- and wrist-worn motion sensors and an ear-canal audio sensor for 12 participants in a lab and out-of-lab settings. Their contributed ACE (accelerometer and audio-based calorie estimation) dataset contains 6 participants total 12 hour data in a lab setting, ACE-free-living dataset contains data for 5 of those ACE participants one-day each data in real-world settings, and ACE-external dataset contains data for 6 new participants, 5 of which for 2 days and the 6th one for 5 days in free-living contexts [60]. DrinkWatch [61] is a smartwatch based application for logging drinking activity only. The data were collected in a lab setting where the number of non-drinking activities are limited, and so the performance of the system is not validated in the real world context. Keum et al. [62] instrumented a necklace with proximity sensors that detect eating through sensing head and jaw-bone movements. They evaluated the performance of the system both in a controlled setting and in free-living contexts. Though performance drops significantly in the free-living context compared to the lab study, the results show the promise of using such devices for real-world deployments. Earbit [63] is a head-mounted wearable system for detecting eating episodes. Though EarBit can collect data from multiple sensors (inertial, acoustic, and optical), the study shows that two inertial sensors, one behind the ear and the other behind the neck are more effective than other sensing modalities. Results from the study show that EarBit is effective in detecting chewing and eating episodes in an unconstrained environment. However, such solutions are not convenient to be used in the wild.

Ecological Momentary Assessment (EMA) is widely used to understand eating behavior and the influence of different factors on eating. Julia et al. [64] used EMA to find the effect of stress, negative and positive emotions on eating behavior, particularly on taste- and hunger-based eating. The study shows that stress and emotions influence eating behavior significantly. For example, higher stress reduces, but positive emotion increases taste-eating. Genevieve et al. [65] used EMA to find associations between stress and eating in mother-child dyads. EMA was used to collect information on perceived stress, and healthy/unhealthy food consumption. Results show that healthy and unhealthy eating by the children are coupled with those by the mothers' at the day level. It depicts the effect of family members on each other in terms of eating habits. Andrea et al. [66] used EMA to collect data from adults with obesity and studied the association between contextual factors and eating in the absence of hunger (EAH). The study reports that there is a lack of hunger in 21% of eating events, and the participants perceived overeating for these events. Khouloud et al. [67] developed a smartphone EMA app to record data related to wanting and liking of food. Results show less food wanting and lower intensity of food liking among the adults with more body fat.

2.2 Hand Washing Detection

Due to the enormous importance of monitoring and ensuring hand hygiene compliances, the issue of developing efficient and effective systems to serve this purpose has received significant attention from the research community. A smartphone based application called iScrub helps human observers to collect and manage data easily [68]. The application is customizable, and an observer can export data in comma-separated-value format to e-mail addresses of his/her choice. In [69], a closed surveillance camera is positioned at the entrance to a surgical ward, and video data are collected. A pre-installed alcohol gel dispenser is available at the entrance of the ward. Entrants to the ward are supposed to use the alcohol gel before they enter into the ward. The footage is reviewed by human observers to monitor compliance by the entrants. All these systems involve one or more human observers and require lots of human effort and related costs. They often result in biased data and uncomfortable working environments.

An automated dispenser monitoring system is presented in [70] to count the hand wash episodes in hospitals. Pressure sensors are used to detect depression in wall-mounted soap and alcohol gel dispensers. A single press on the dispenser is then associated with a single hand-hygiene episode. The system generates false positives when a person dispenses multiple times during a single episode. In [71], a system to detect the use of alcohol-based sanitizer is presented. An alcohol sensor is used to detect the vaporization that comes from the hand after use of the sanitizer. The system does not work when hands are washed using soap and water.

A system consisting of small credit card sized devices is described in [72]. The devices are programmed to perform three different roles called badges, beacons, and triggers. The badges are worn by the health workers, the beacons are placed in the patient rooms, and the triggers are attached to the dispensers. The devices can communicate using a protocol for personal area network (PAN). Another automated hand hygiene documentation and reminder system is described in [73]. Pressure or vibrator sensors are attached to the dispensers to detect dispensing of soap or alcohol gels. Upon detection of a dispensing event, it enables a passive infrared sensor to detect human hands, and then an alcohol sensor is used to identify the use of sanitizer. Ultrasonic hotspots are created in wash zones and patient bed zones, and each healthcare worker wears a wireless tag as a badge that detects the ultrasonic hotspots. These systems cannot detect the worker who performed a hand wash when multiple persons are present in a wash zone at a time.

A vision based technique to measure the quality of hand washes is presented in [74]. Each sink is equipped with a camera, and gestures of the hands are analyzed to measure the quality. To discriminate between different types of hand gestures, a support vector machine (SVM) classification technique is applied to the features extracted from each frame. It does not identify the person performing the hand wash. Also it requires one camera for each of the sink or dispenser area where hand hygiene is performed. Any camera based approach comes with a number of limitations and problems. Deploying cameras in washroom areas is very uncomfortable for the users as well as raises privacy issues. Also for easy access, sanitizer dispensers are often kept in the areas where patients are located. Placing cameras in these areas is threatening to the privacy of the patients [75]. The required number of cameras for settings like hospitals is generally very large; and massive computing and communication resources are required to collect and process the data. Such systems are very expensive to install and maintain. The performance of vision based system also depends on environmental factors like lighting and decoration of the surroundings.

Human activity recognition using wearable sensors is an active research area with significant involvement of researchers from different domains, including computational science, healthcare, and engineering. Recent developments in wearable technology, particularly availability and widespread use of sensor-enabled tiny devices like fitness trackers and smartwatches, have thrust research in this direction. Farhad et al. [76] presents an active learning method for smartwatches that recognize 5 activities of daily living; namely running, walking, standing, sitting, and lying down. This work does not address the problem of NULL activities. Most of the state-of-the-art solutions for activity recognition use neural networks. Hammerla et al. [13] explore different neural network architectures including convolutional and recurrent neural networks for activity recognition using wearable sensors. They propose a regularization technique to improve the performance of the networks. DeepConvLSTM [12] segments the time series sensor data from the wearables and applies both convolution and recurrent neural networks on each of the segments independently. Guan et al. [14] ensembles a set of Long Short Term Memory (LSTM) networks to improve the performance of activity recognition tasks. They save the LSTM model after each of the epochs where the data used to train the network during an epoch is randomly selected from the training data and ensemble the top performing LSTM learners for activity recognition. These works do not address the issue of NULL activities, particularly unseen NULL activities.

Galuzzi et al. [77] use sensors from wrist devices to detect hand washing. In addition to hand washing, they collect a limited number of NULL activities that include opening a jar, opening and eating the candy form the jar, tying shoes and applying bandages. They evaluated the performance of different machine learning techniques, namely K-Nearest Neighbors, Decision Tree, Neural Network, and Naive Bayes, for hand washing detection where the neural network outperforms the other methods. WristWash [78] uses Hidden Markov Model to detect different hand rubbing approaches suggested by World Health Organization (WHO) [79]. These works does not address the problem of separating hand-washing activities from unseen NULL activities.

Detecting out-of-distribution is an active area of research. Hendrycks et al. [80] uses probabilities from softmax distributions to detect out-of-distribution samples. They evaluated their method using datasets from computer vision, natural language processing, and speech recognition. Lee et al. [81] propose a method that can be used with any pre-trained softmax neural classifier to detect abnormal samples. They use class conditional Gaussian distributions of the outputs of different layers of a neural network along with Mahalanobis distance to find confidence. The method works for both out-of-distribution and adversarial samples. They also demonstrated the use of their method in learning new classes. This work is focused on image data. They use pre-trained convolutional neural networks on some vision datasets including CIFAR [82], ImageNet [83], LSUN [84] and SVHN [85]. These works on out-of-distribution detection are focused on computer vision, natural language processing, or speech recognition. We developed a solution for hand washing detection using wearable sensors.

2.3 Movement Visualization

Our visualization method places unit vectors from orientation on a unit sphere which is divided into some nearly uniform and numbered cells. The earth's shape resembles a sphere, and so spheres are widely used for visualization of geographical data. The sphere is often divided into uniform cells using a regular icosahedron and 2 of the 12 vertices of the icosahedron are aligned with the North and the South Poles [86] of the earth. A global weather prediction model, named GME, has been proposed based on the icosahedral–hexagonal grid [87]. These works focus on representing geographical data like weather for different regions of the earth.

State-of-the-art works on wearable and mobile sensing, including those for activity and gesture recognition, are generally focused on data transformation, feature engineering or classification. Existing techniques like line plot, scatter plot, pie chart, histogram, heat map, and bar chart are used for visualization of the sensor data. Some works transform data from one representation to another and visualize the transformed data using traditional plots mentioned above. Maaten et al. developed t-SNE [88] to visualize high-dimensional data in a two or three-dimensional space. For that purpose, the high-dimensional data are embedded into two or three dimensions using stochastic neighbor embedding. The embedded representation depends on different parameters used in the method including perplexity and number of iterations. The final embedding is usually visualized using scatter plots.

Though accelerations and rotation rates are the most widely used sensing modalities, orientation has also been used for activity recognition in several studies [89–91]. These works are focused on recognizing some specific activities, not visualization. RisQ [92] use inertial sensors on smart wristbands for recognizing smoking gestures. It uses quaternion to extract the orientation of the wrist. It computes the trajectory of the wrist relative to the elbow assuming that the elbow is stationary and in a fixed distance away from the wrist. This assumption is not often practical in free-living contexts. We focus more on visualization and the solution for smoking we present here does not have the above mentioned limitation.

2.4 Interactive Reminders

To avoid missing important tasks, people use different kinds of reminders, ranging from traditional methods like notes to technology-enabled systems like text messages and smartphone apps. With the ubiquity of mobile phones, the use of these devices for medication alerts and tracking has received significant attention from different stakeholders including patients, caregivers, developers, and researchers. Text messages are used for health intervention in several studies [93][94][95]. The text message based systems are not convenient for user interactions and therefore inflexible in re-scheduling reminders and tracking medication. Most of the limitations of the smartphone-based systems, as mentioned earlier, are also applicable for the text message-based systems.

A number of smartphone applications with different features are available in app stores for providing medication reminders and tracking intakes [96] [97]. A functionality review of 229 of the apps, as reported in [98], shows that many of the apps lack important features like re-scheduling, medication pictures, and data export. For example, only 17% of the apps offered an option to re-schedule or postpone a reminder. Researchers have also designed, developed, and evaluated smartphone-based reminder and tracking systems. Wedjat [99] is such a system that provides potential drug-drug/drug-food interaction information to the users in addition to medication reminder and tracking features. UbiMed [100] presents a solution that incorporates smart phone apps to provide reminders, and to support the tracking of prescribed medication for the aging and disabled population. As described earlier, smartphone-based systems for medication reminders and tracking come with a number of limitations. A feasibility study [8] reveals some of the limitations.

Smart wearable devices like smartwatches and wrist bands are usually enriched with many features like touch screens, microphones, sensors, BlueTooth and Wi/Fi. These devices are being used widely in healthcare applications including activity tracking, wellbeing monitoring, and reminders. A diary-like system for diabetes patients is presented in [101] that uses both smartphones and smartwatches to log information from and provide reminders to diabetes patients. SPARK [102] is a framework that combines smartphones and smartwatches together in monitoring symptoms of patients with Parkinson's Disease. It also supports physicians in providing tele-interventions to the patients. Fabian et al. [103] proposes to show pictures of the drugs on the display of the wrist device to reduce confusion of the patients when multiple drugs need to be taken. Some smartphone apps also synchronize reminders with smartwatches [104]. However, these systems use only the small display of the wrist device, and therefore can not provide detailed information related to a reminder using the wrist device only. Also, wrist devices used in the existing systems do not support rescheduling the reminders. Most of these reminders and tracking systems are aimed for some specific groups of patients or users. In contrast, our system is a general-purpose reminder and tracking system that can be customized according to the needs of the users. It combines speech recognition and text-to-speech technologies with intelligent interface design in providing reminders and logging input from users.

2.5 Data Collection Tools

Several research works collect sensor data from smartphones and smartwatches using custom made apps. Mirtchouk et al. [59] collected inertial sensor data from LG-G watches to develop a dataset related to eating and drinking activities. Shoaib et al. [105] developed a data logger to collect sensor data from Android smartphones and smartwatches. Weiss et al. [106] collected sensor data from Android smartphones and smartwatches for different activities of daily living. HAD-AW [107] is a dataset for human activity recognition collected using Apple watches. Though these articles describe the datasets, there is no detailed information available about the apps and their availability.

There are several apps available for collecting data from smartphones, and some of them support data collection from a paired smartwatch. G-sensor Logger [108] is a smartphone app that can collect accelerometer data from the phone. It also calculates and shows some features like magnitude, minimum and maximum of the data. Sensor Log [109] is an app that can be used to log, label and export data from different sensors available in the smartphone. Sensor Sense [110] also supports data collection from different sensors of a smartphone. It displays sensor data on the screen of the phone in real-time. These apps can collect data from smartphones only. They do not have features to collect data from smartwatches.

There are several apps available for smartphones that also support data collection from a paired smartwatch. Sensor Data Logger [111] can stream data from paired smartwatches to the smartphone. It has features to show the data on the screen of the phone in real-time. However, this app does not have features to label the collected data. ExtraSensory [112] is an app for a smartphone that can collect sensor data from the phone as well as from a paired smartwatch. It can prompt users to collect detailed labels about their context and activities. It also supports self-reported labels by the user. The app has been used to develop a dataset called ExtraSensory Dataset, which has data from 60 participants from free-living contexts. UniMiB AAL [113]] is a suite of apps that can be used to collect and label data from smartphones and paired smartwatches. It mainly consists of two apps where one app is used to collect sensor data and the other is for labeling the data. All these tools collect data only from paired smartwatches, which means the watch needs to be connected to a phone. For example, UniMiB AAL needs a smartphone app to label the data from the paired watch. In contrast, our app is standalone and it does not need any smartphone or other devices to collect the data.

Swear [114] is an app for Android smartwatches that collects data from different sensors of the watch and then uploads the data to an Amazon S3 bucket. The sensors and the sampling rate are pre-defined. For example, the accelerometer is sampled at 10 Hz, and the light and heart rate sensor at 1 Hz. This app is not configurable and flexible. In contrast, our app can collect data from any sensor available in the watch with different sampling rates. Users can define the sensors, sampling rate for each of the sensors, and labels for the data. The app along with a companion desktop app provides easy to use mechanisms and interfaces for providing the inputs. It also comes with several handy features like time synchronization, data navigation, and data download.

Chapter 3

Monitoring Family Eating Dynamics

Obesity increases one's risk for several health issues, including Type 2 diabetes, cardiovascular diseases, sleep apnea, osteoarthritis, kidney diseases, and certain cancers [115, 116]. While unhealthy dietary intake is one of the primary contributors to obesity risk, dietary intake has proven very difficult to measure and track. Research has shown that certain features of eating behaviors, contexts and events, such as when, where, with whom, and particular states of mind surrounding those eating behaviors, might be powerful determinants of food consumption and ultimately obesity [117–119]. With the rise of mobile technologies, possibilities have opened up to understand these eating behaviors more fully, in context and in real time. While the field has historically relied on self-report tools (such as 24-hour recalls, journals, and food frequency questionnaires [120]) and more recently on image analysis of plated meals [121], all of these measurement modalities have substantial limitations in their accuracy to assess food intake. These measurement modalities can incur substantial participant burden, and/or research costs, for instance hours of coding, and cannot provide information in real time. Real time measurement will be important for future efforts that involve intervening just-in-time. Furthermore, these assessment tools often focus on the measurement of food intake independent of eating behaviors and the contexts in which eating occurs. Understanding these complex determinants of eating is an important, but understudied topic. The development of systems that can monitor and measure eating behaviors and their complex determinants in real time and in context is now possible with advances in wearable activity monitors.

Family systems and family social networks (i.e., characteristics of the relationships and interactions among family members and emergent patterns of these interactions) are important dynamic milieus that impact eating. Empirical evidence shows that family members engage in similar food choices [122] and eating behaviors [123], even across generations. Many dimensions of family systems can be barriers or promoters of healthy eating: family members are models for healthy or unhealthy habits [124, 125] and family relationships can provide (or lack) information and support that influence eating behaviors and dietary intake [122]. Family relationships also influence, and are influenced by, personal states such as stress and mood; which may indirectly impact family members' eating. These strong links between family dynamics and eating highlight the potential to harness family influence to promote healthy eating and reduce disease risk, and mobile and wireless technologies offer new opportunities for measuring, tracking, and ultimately intervening upon these family eating dynamics (FED).

Real-time monitoring of FED requires detecting eating activities and related individual states and family features in the home, because this is a key environment in which family members interact, as well as prepare and consume food. We developed MFED, a system for monitoring family eating dynamics in the home and in real-time. MFED uses smart wearables and Bluetooth beacons to monitor theoretically important features of family eating events and family dynamics while the users are at home. The main components of the systems are smartwatches, Bluetooth beacons, smartphones, a base station, and a cloud serever. Smartwatches are used to detect when eating events are occurring in the home for each user. Bluetooth beacons are placed at different locations of the home to determine user location in the home. Smartphones are used to measure individual states and situations (e.g., hunger, stress), via self-report on brief surveys, which are not captured by the sensor system. A base-station, placed at the home, collects and processes data from the watches and phones as well as manages the EMA (Ecological Momentary Assessment) surveys.

In contrast to traditional approaches that focus on individual's dietary intake (e.g., what and how much is being eaten), our FED-based approach focuses on temporally dense and highly contextualized monitoring of family members' eating events while simultaneously capturing other theoretically-relevant states such as hunger, satiety, mood and stress, and family members' presence and interaction in the home. In the MFED system, eating events are monitored by detecting hand gestures for moving food or drink to mouth. An "eating event" is a set of such gestures and represents phenomenon like consuming a meal, a snack, a drink or a combination of these consumption behaviors where eating gestures are clustered temporally. The MFED system is built for real-time detection of eating events at home so that EMA questions can be asked immediately after eating. It should be noted that our system monitors eating events, but it does not measure what people are eating.

There are several challenges associated with developing a system like MFED. Some of the major challenges are discussed below. Though we explained some of these challenges before in Chapter 1, we discuss them here in the perspective of developing a system for monitoring family eating dynamics.

- *Limited resources available in the smartwatch*: Detecting activities in real time using smartwatches is challenging as the resources (e.g., energy, computation, and memory) available in such devices are very limited. Continuous streaming of data to another device or to the base-station from a smartwatch consumes significant power from the watch. On the other hand, high performing methods like Convolutional Neural Networks (CNN) require not only significant computation and memory, but also more energy to run the models.

- *Detecting eating events in free-living context*: Detecting eating activities using wrist worn sensors, particularly in free-living context, is challenging [11]. The eating gestures and so the corresponding signals from the sensors differ widely for different foods, contexts, and utensils used as well as speed of moving hand to mouth. Confounding signals generated from wide range of non-eating activities make it difficult to detect eating gestures from a stream of sensor data.

- *Ground truth collection*: To evaluate the performance of the system in monitoring FED, and further improve it, ground truth is needed. However, in contrast to lab settings, it is challenging to collect ground truth in the wild, particularly for systems like FED.

- *System installation*: It is important to reduce the burden of installing a system and avoid intruding into the infrastructures (e.g., wiring, drilling etc.) of participants' homes.

We developed innovative and effective approaches to address the challenges. Instead of streaming the sensor data continuously from the smartwatch to the server, we store the data in the smartwatch temporarily and detect potential eating events in the watch using an efficient approach. The data are uploaded to the base-station using Wi-Fi only when such an event is detected. The eating events are detected conservatively to avoid false negatives. The data are further processed in the base-station with a more complex and effective method to finally detect the eating events. This approach reduces energy consumption from the smartwatch due to event-based data transfer instead of continuous transmission.

We developed a two step solution to detect eating events in the base-station. At first, we detect a set of potential eating gestures using a threshold based technique, and then we use a Convolutional Neural Network (CNN) to detect eating gestures from the set of potential gestures. The detected eating gestures are clustered together to detect an eating event. The algorithm is very efficient as most of the non-eating data are discarded using a simple threshold based method and only a small portion of the data is processed by CNN. Efficiency of our eating detection method allows MFED to use low-cost devices for base-stations. When an eating event is detected, the corresponding participant is asked to confirm the eating event immediately via a brief EMA survey, providing ground truth that is free from memory bias. In cases of correctly detected eating events, the participant is asked who they were eating with, among other questions, and response to this query provides ground truth for other participants who do not respond to the EMA or for whom the system fails to detect the eating event. This "collaborative ground truth" approach is novel for acquiring ground truth in free-living context.

MFED is very easy to install, and it is not intruding to the home infrastructures. Beacons are battery powered and they are attached in the walls using a removable mounting tape that does not damage paint or wallpaper. A beacon usually runs several years without battery replacement [126], and so the burden of maintenance is low even if the system is deployed for longer time. Apps for MFED can be easily installed in the smartwatches and the smartphones. Such a system is also very suitable for short and medium term deployment because no device needs to be installed permanently in the home.

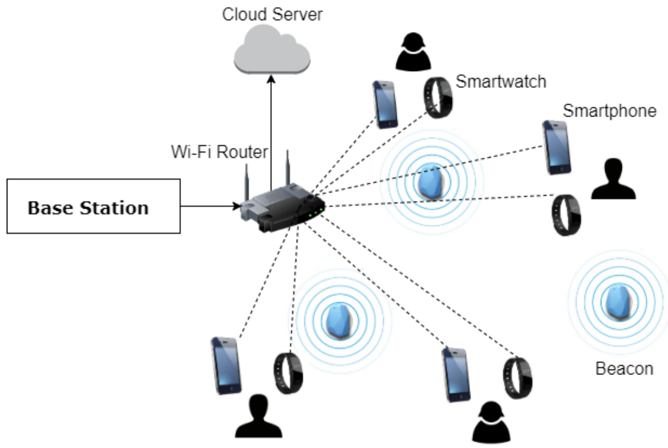


Figure 3.1: The MFED system mainly consists of smartphones, smartwatches, Bluetooth beacons, a base-station, and a cloud server. The smartphones, smartwaches, and the base-station are connected through a Wi-Fi router. The beacons broadcast Bluetooth packets that are scanned by the smartwatches.

3.1 System Description

The MFED system consists of smartwatches, smartphones, Bluetooth beacons, a base station, and a cloud server. Figure 3.1 illustrates the connectivity among the devices in the system. The smartphones, the smartwatches, and the base station in the system are connected through a Wi-Fi router. Each of the family members uses a smartphone and wears a smartwatch. The beacons are placed at different locations in the home. Eating detection in MFED consists of two parts, one in the watch and the other in the base station. After an eating event is detected, an EMA is sent to the smartphone of the corresponding participant. To better understand the mood of the family members throughout the day (and not solely following detected eating events), the system deploys additional EMAs that include brief validated mood survey items several times over the day. The data flow in the system is depicted in Figure 3.2, and the following sections give the details of the different components of the system.

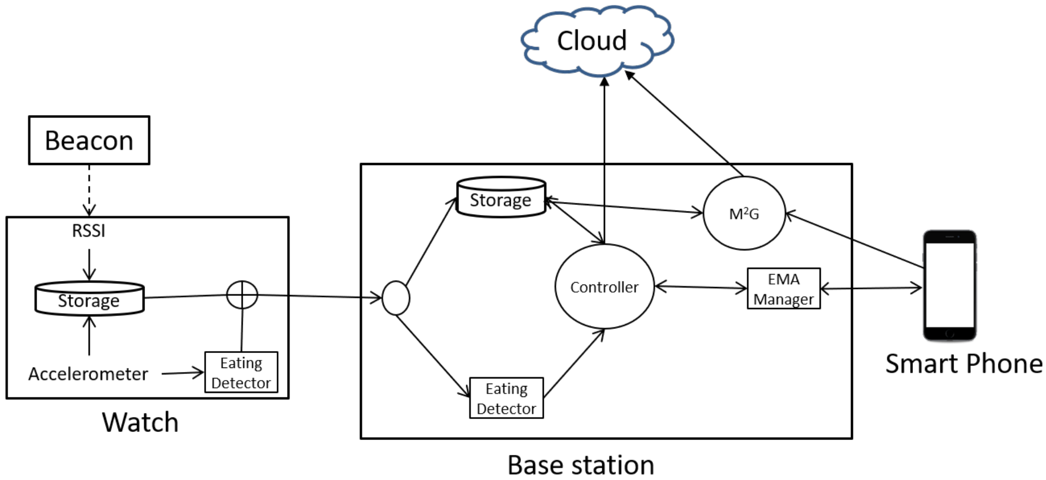


Figure 3.2: Dataflow in the MFED system

3.1.1 Sensors

Beacon:

Bluetooth Low Energy (BLE) beacons are placed at different locations in the home. A beacon broadcasts packets that include the unique MAC address of the Bluetooth interface. The smartwatch scans for the Bluetooth packets and records the Received Signal Strength Indicator (RSSI) values of the signals that indicate the proximity of the watch to the beacons. There is no connection established between the beacons and the watches, rather it is a one-way communication where the beacons broadcast the packets independently, and any Bluetooth enabled device can receive the packet. The beacons broadcast continuously with several packets per second (a configurable parameter). We use Estimote [126] beacons in our system.

Smartwatch:

In our system, we use the Sony Smartwatch 3 [127], an Android-powered watch that has Wi-Fi capability. There were several reasons for choosing to use a wrist-worn sensor for eating detection. Though homes and utensils can be instrumented with sensors for detecting eating activity, this approach requires significant cost and effort for installation and maintenance. This approach is also limited in detecting the person who is eating, a major requirement for MFED. Video-based systems, often used for activity detection and person identification, are not suitable due to privacy issues. Also, such a system fails to detect when a person eats outside the camera view. Wrist devices such as smartwatches are ubiquitous and require very little effort for installation and maintenance. Because a wrist-worn device is a personal gadget, person identification using such a device is intuitive. Therefore, the use of smartwatches for eating detection in the MFED system appeared most advantageous. The Sony Smartwach 3 has several sensors including accelerometer and gyroscopes. The watch also has Blueotooth Low Energy (BLE) and Wi-Fi capability.

3.1.2 Smartwatch App

We have developed an app for the smartwatch that collects data, detects potential eating events, and uploads the data to the base station in the background. The acceleromter data are collected and processed continuously when the watch is not being charged. The data are stored temporarily in the watch and uploaded to the base-station for further processing and actions whenever a sequence of potential eating gestures is detected. The details of detecting potential eating gestures are described in Section 3.2. Since scanning RSSI signals from Bluetooth beacons consumes significant energy, the watch scans the beacon signals for 5 seconds with 2-minute interval to reduce energy consumption. The app also records the battery percentage of the watch once in the 2-minute interval. The battery and beacon RSSI data are uploaded opportunistically with the accelerometer data.

The watch app has Graphical User Interfaces (GUIs) which are used for configuration, testing and debugging. Once started, the app runs in the background without requiring any user interaction. When the watch is restarted or turned on, the app starts automatically without any intervention. This is very critical for real-world deployments where users might forget to turn on the app when the watch is turned on. In fact, the user does not need to interact with the watch app at all. The GUIs are used by the research and deployment teams only.

3.1.3 Base Station

The base station has four major sub-systems: the Eating Event Detector, the EMA Manager, the MFED Controller, and the System Monitor. They are are described below.

Eating Event Detector:

The base-station collects, processes and stores data from all the smartwatches used for MFED in the home. The eating event detector processed the accelerometer data and detects eating events in real-time. More detail about detecting eating events in the base-station is discussed in Section 3.2. It should be noted that eating events are detected in the smartwatches just for uploading data to the base-station. The events detected in the base-station are ultimately used to send EMA surveys to the participants.

Ecological Momentary Assessment (EMA) Manager:

There are two components of the EMA sub-system: The EMA Manager that runs in the base station, and the EMA App that runs in the smartphone. The EMA Manager is responsible for sending EMAs to the smartphones, gathering the EMA responses from the smartphones, and storing the response data in the base station. The response data are eventually uploaded to the cloud by the MFED Controller.

There are two types of EMAs: eating EMA and mood EMA. An eating EMA is pushed out when an eating event is detected and a mood EMA that is pushed out according to a set time. To manage participant burden during the two week deployment, the system was set to send an EMA to participants no more than once per hour. This biases the system to the first eating event detected in the hour. So, if an eating event is detected at 12:15 and an eating EMA is sent out, the system will ignore an eating event detected at 12:45. If no eating event is detected in an hour, the mood EMA is sent out. The flow of the queries for the eating EMA is shown in Figure 3.3 (1). At first, the participant is asked if he/she was eating. A negative response indicates that the eating event detected by the system is inaccurate. In such a case, the system asks the participant what he/she was doing and then a set of questions related to mood. If the detected eating event is accurate, the participant is asked to confirm whether he/she has finished eating. If eating is not finished, the participant is requested to press a DONE button available in the EMA app when finishing eating. When the eating event is finished, an EMA survey with validated measures of hunger, satiety, mindful eating, mood and stress is sent. The participant also provides information about who he/she was eating with, based on a multiple-response item. An eating EMA survey also includes questions related to mood.

As explained above, the mood EMAs are sent to the participants hourly unless an eating event is detected. The optimal frequency with which to assess stress and mood via ecological momentary assessment, with regard to both accuracy and compliance, is still an open question in the field. We chose 1-hour intervals to measure within-subject changes in mood and stress throughout the day. To account for the variation in daily routines and sleeping patterns of participants, we collect data about typical awake time from each participant before deployment. EMAs are sent only during this "personalized participation window" period. For example, if a participant's window is from 6 am to 10 pm, EMAs are sent to that participant only during that time period.

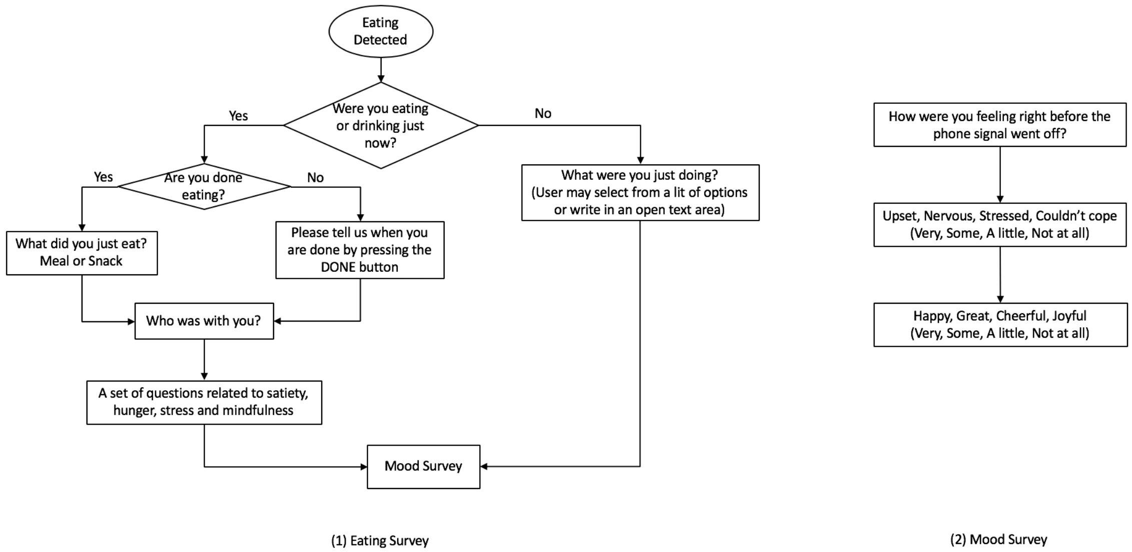


Figure 3.3: EMA flowchart

The participation windows are not same for all the participants of a home, rather they are set based on individual preference. The flowchart of mood EMA survey is shown in Figure 3.3 (2). The survey includes validated items to assess key affective states: happy, joyful, upset, nervous, etc. More detail about the EMA survey is given in Section 3.3.

MFED Controller:

The MFED Controller is the integration module that runs continuously in the base station. The main purpose of the module is to start/stop the sub-systems, aggregate data from the sub-systems, and upload data to the cloud periodically. For each new deployment, it initializes all the parameters and starts the sub-systems. When a deployment is terminated, it stops the sub-systems and uploads remaining data from the base station to the cloud. The MFED Controller maintains communication among different modules. For example, when an eating event is detected, the Eating Event Detector sends the information to the MFED controller that triggers the EMA

Manager to send EMA to the corresponding participant.

System Monitor:

MFED is a system of systems, and it is necessary to monitor the status of the subsystems in real time. For this purpose, we use M^2G [128], a monitor for research-oriented residential systems. M^2G monitors the processes of MFED running on the base-station, the responses to the EMAs from the participants, and the battery status and network connection of the smartphones and smartwatches using the data uploaded to the base-station from these devices. In case of any discrepancy, it sends an alert via email to the deployment team responsible for addressing the issues.

3.1.4 Data Storage and dashboard

The MFED controller uploads the data to a cloud server that stores all the data collected from the deployments. The continuous stream of data collected from an ongoing deployment is used to monitor the system and identify potential discrepancies or system failure. We have developed a web-based dashboard for visualization and interaction with the data from both ongoing deployments and past deployments. It displays the start time, end time, resident information, and devices for each resident of the deployment. For an ongoing deployment, the dashboard presents the critical information of the deployment including status of all the required processes (i.e., whether they are running properly or stopped at some point of time), the status of different data files that are supposed to be uploaded by the base station, and the network connectivity from the base station to the cloud. In addition, the dashboard also displays the latest battery status of the base station, the phones, and the watches used in that deployment. All this information is displayed with the corresponding timestamp. For example, if a process is not running, the dashboard shows the time

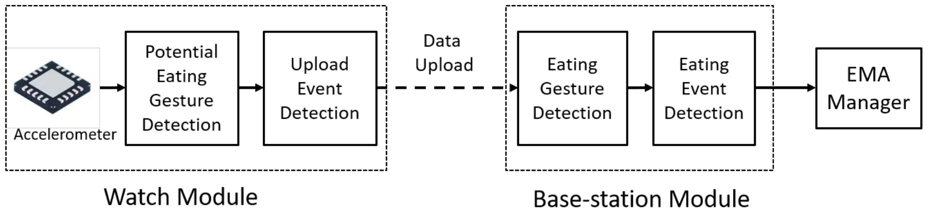


Figure 3.4: The pipeline for eating event detection. This pipeline involves two modules: the watch module and the base-station module

when it stopped. The dashboard also provides an interface to visualize different data streams that correspond to different events like meals, EMA responses, mood, etc. For example, the list of EMAs sent to the participants as well the responses from them are accessible through the dashboard. An interface is also available to easily formulate a query to retrieve such data.

3.2 Eating Detection

MFED detects eating in real-time at the base-station using the accelerometer data from the smartwatches. The pipeline for real-time eating detection consists of two modules: the watch module and the base-station module. The watch module collects data from accelerometer and uploads data to the base-station where eating events are detected. Instead of continuously streaming data to the base-station, the watch module stores data temporally in its memory, and then uploads the data to the base-station when a sequence of potential eating gestures are detected. The potential eating gestures are detected in the watch using a threshold based method, and it is used only to decide when the watch should upload data to the base-station so that the base-station can process the data further with more sophisticated algorithm, and can detect an eating event that triggers an EMA survey. The pipeline for eating event detection is depicted in Figure 3.4. We describe these two modules in more depth below.

3.2.1 Watch Module

Though people may use one or both hands to eat food or drink, using watches on both wrists is not convenient, particularly in the wild. We place a watch only on the wrist of the dominant hand because this hand is generally used more for eating than the other hand [11, 59]. To develop the model for eating gesture detection, we collected data from 29 participants in lab settings where eating and other activities were not controlled.

Potential Eating Gesture Detection:

Our potential eating gesture detection method is based on the fact that the wrist is usually inclined upward to some degree to move food to the mouth, and it goes down after that. We use the accelerometer from the wrist device to determine that the wrist is inclined upward. Accelerometers and other 3D inertial sensors are generally embedded in a wrist wearable device in such a way that one of the axes of the sensors is aligned with the arm length. For example, Figure 3.5(a) shows the axes of an Android smartwatch worn on the right wrist, and the X axis of the device is aligned with arm length. The alignment of the sensor axes with respect to the arm length might differ from device to device. Our method is also applicable for other arrangements. We use the arrangement of the Android smartwatch without loss of generality.

A value from an accelerometer is the total of the acceleration due to the gravity and the acceleration due to movement, called the gravitational acceleration and the linear acceleration, respectively. When the device doesn't move, the linear acceleration is zero, and so the accelerometer values represent the gravitational acceleration. The gravitational acceleration along the X axis is negative while the wrist (i.e. the device) is inclined upward, and vice versa. Since the arm movements (and so the linear accelerations) are not usually intense during an eating gesture, the X acceleration

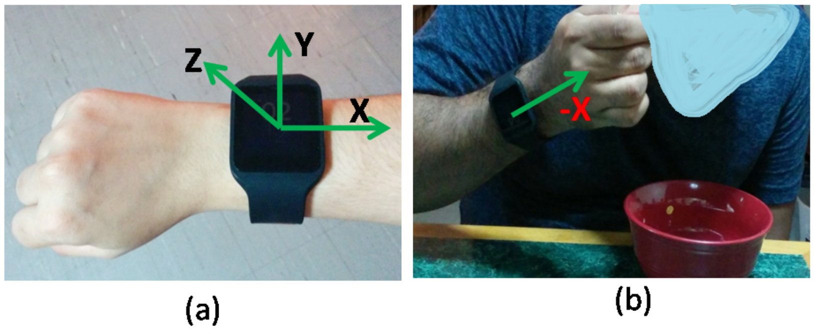


Figure 3.5: (a) Axes of the sensors for an Android smart watches, (b) Hand orientation during an eating gesture

(the total of gravitational and linear acceleration) is usually negative, particularly when the time series acceleration data is smoothed. The acceleration value generally decreases and then increases before and after an eating gesture, respectively. So, there is a negative peak in the acceleration data along the X axis during an eating gesture. A point P_i is defined as a negative peak if $AX_i < AX_{i-1}$ and $AX_i < AX_{i+1}$ where AX_i denotes the acceleration value along the X axis of the i-th sample. This results in many peaks that are very close to each other. So, we use only the peak with the more negative value when two peaks are very close. From the data we collected, we found that most of the consecutive eating gestures are separated by more than 2 seconds, therefore we take the more negative peak when two consecutive peaks are within 2 seconds. Since the wrist is inclined upward to some degree during most of the eating eating gestures, the peaks with the X acceleration value below a predefined threshold, called the AX_{th}, are selected. Finally, the peaks with very low or almost no movement are discarded, and the remaining peaks are considered as potential eating gestures. The degree of movement around a peak is measured by the variance of the acceleration around it. A window of 6 seconds is extracted around the potential eating gesture point, and the sum of the variances along each of the axes of the accelerometer is used for measuring the degree of movement. The peaks with variance greater than a threshold (V_{th}) remain as the potential eating gestures. We denote the peaks as

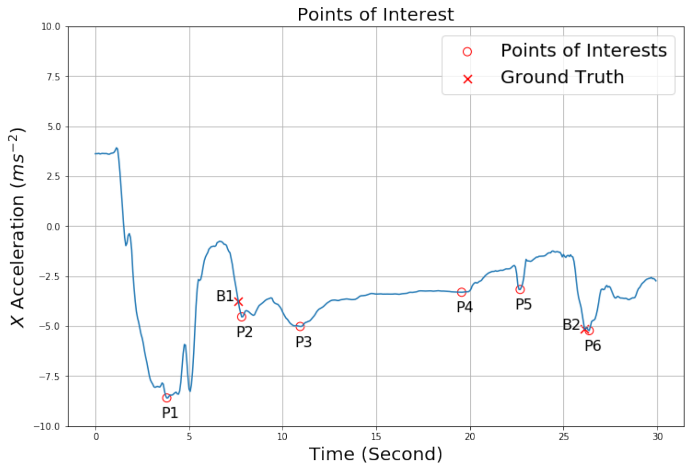


Figure 3.6: Points of Interests

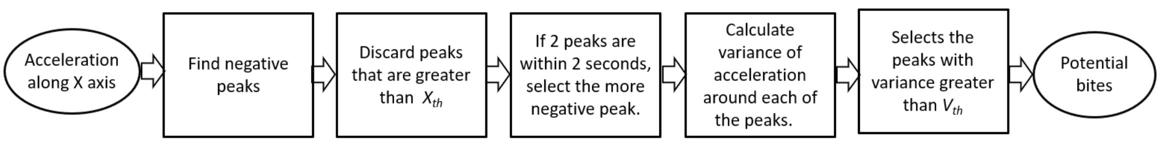


Figure 3.7: Flowchart for potential eating gesture detection

Point of Interest (PoI). Figure 3.6 shows an example of the X acceleration with the PoIs for $X_{th} = -3$, and $V_{th} = 1$. The number of potential eating gestures depends on the thresholds (X_{th} and V_{th}), and more eating gestures are discarded from the potential eating gesture set when the threshold is lower. Importantly, the threshold values are determined empirically so that a significant amount of non-eating gestures is discarded while most of the true eating gestures are retained. Figure 3.7 shows the steps for potential eating gesture detection.

Upload Event Detection:

A sequence of the potential eating gestures indicates the possibility of an eating event. If four such potential eating gestures are detected in a 2-minute time window, data is uploaded from the watch to the base station for further processing and actions. The watch does not wait for 2-minutes, rather it uploads data as soon as four potential eating gestures are detected. The minimum interval between two data uploads from the watch to the base station is one minute. This reduces battery consumption from frequent data upload. However, it ensures that eating events are detected at the base station in real time which is required to trigger the eating EMAs.

3.2.2 Base-station Module

Eating Gesture Detection:

Similar to the watch module, potential eating gestures are detected first from the accelerometer data. However, the threshold based method results many false positives. We use a Convolutional Neural Network (CNN) to detect eating gestures from the set of potential eating gestures. The length of the hand to mouth gestures for eating are not the same. Many of the windows we extracted for eating gestures includes non-eating gestures at the two ends. We can compare this with an image where the target object is somewhere in the image and does not necessarily cover the whole image. CNN has proven to be worked for image detection, and so we use CNN for eating gesture detection. CNNs have also shown to be effective in classifying time series data like those from wearable and smartphone sensors [129, 130].

The length of an eating gestures are usually less than 6 seconds [58, 59], and we extract a window of this length around the PoIs. An accelerometer provides data along X, Y and Z axes. So, the size of the window is $N \times 3$, where $N = 6 \times \textit{sampling rate of acceleration}$. We reshape each window into $N \times 3 \times 1$ that resembles a 3-dimensional image. We use a filter of size 2×2 that results feature maps of width 2 and 1 in the first and the second convolution layer, respectively. A 2×2 filter convolutes two axes together in the first convolution layer, and thus captures the correlation between the axes, a feature widely used for accelerometer based activity recognition [59, 131]. The feature maps of width 1 in the second

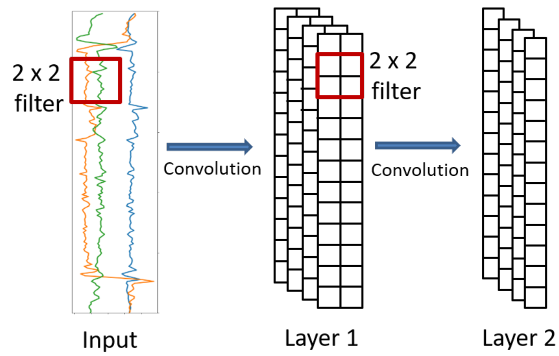


Figure 3.8: Applying convolution to the acceleration along X, Y and Z axes. The width of the first and second convolution layers are 2 and 1, respectively.

convolution layer limits further use of the filter, so we use two convolution layers in our solution. The smaller network also reduces the problem of over-fitting since the window of accelerometer data is much smaller in size than that of a typical image, and the number of instances in our dataset is much smaller than a typical image dataset. Figure 3.8 shows applying convolution to the acceleration along X, Y and Z axes.

Figure 3.9 illustrates the network architecture for eating gesture detection. The input is processed by a Conv-Pool-Conv-Pool-Flatten-Dense-Dense network as depicted in Figure 3.9. Each of the convolution layers is followed by a max-pooling layer. The output of the second pooling layer is flattened, and then two dense layers are applied. The final output is a single node that gives the probability of the input to be an eating gesture. ReLU (Rectified Linear Units) is used as activation functions for all the layers except the output layer where a sigmoid activation function is used. The output of the network is the probability of the input window being an eating gesture. Since it is a binary classification problem, we use a sigmoid activation function at the output layer. More details about the network with different parameter values and experimental results are given in 3.5.1.

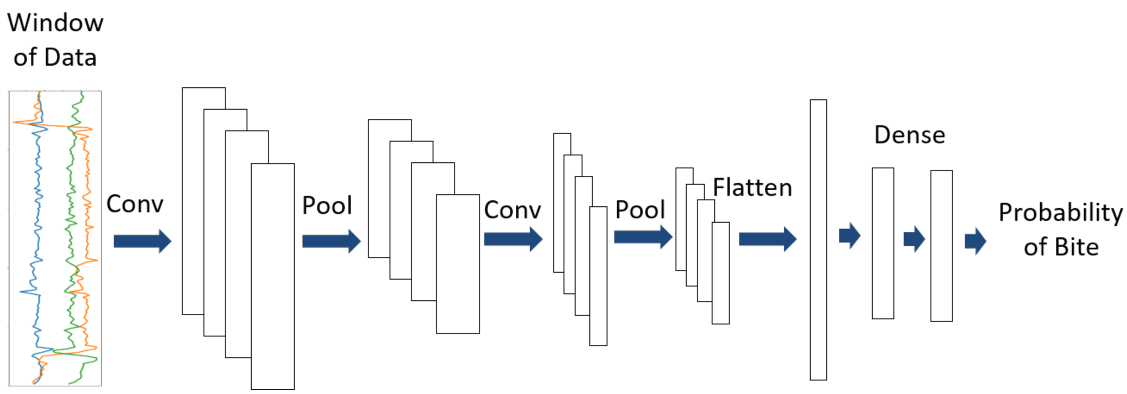


Figure 3.9: The Convolutional Neural Network for eating gesture classification.

Eating Event Detection:

An eating event is a sequence of eating gestures taken over some time and usually with irregular intervals. Like Mirtchouk et al. [59], we construct an eating event by clustering eating gestures where two eating gestures within one minute-distance belong to the same cluster. However, if the interval between two detected eating gestures from an eating event is greater than one minute, more than one cluster might be generated for that single event. Therefore, clusters that are within four minutes distance (the waiting time before sending an EMA after an event is detected) are combined together. So, a single cluster of eating gestures or a cluster of the clusters define an eating event. Figure 3.10 illustrates eating event formation from eating gestures where Event 1 is constructed from a single cluster and Event 2 is from 2 clusters. There might be some outlier eating gestures that are not part of any cluster or eating event. Such outliers are often generated by confounding gestures that are misclassified by the eating gesture detection method. It should be noted that an eating event can be detected even if some eating gestures are not detected by the system. On the other hand, some false eating gestures close to each other might form a cluster, resulting in a false eating event. Any cluster having at least 3 eating gestures is used for eating event detection. It reduces false positives while detecting

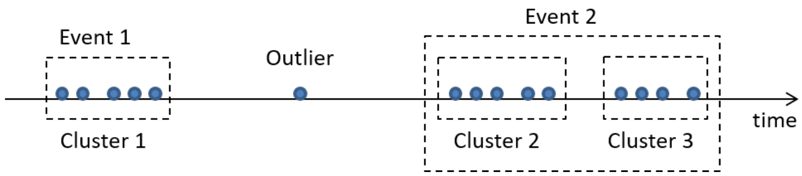


Figure 3.10: Eating event formation from eating gestures.

events with fewer eating gestures.

3.3 Individual State Detection

In addition to information regarding the detected eating events, MFED collected self-reported individual states including hunger, satiety, negative and positive affect, stress, and eating in the absence of hunger. There is no current solution available to detect these states automatically, and developing such a solution is beyond the scope of this work. We utilized Ecological Momentary Assessment (EMA) [132], a method of sampling users' behaviors and states in real time, to collect information from the user. The mood and eating EMAs included a set of queries assessing positive and negative affect and stress, while the eating EMAs additionally assessed hunger, satiety, and eating in the absence of hunger. The EMA survey questions are selected based on previous studies that have validated their usefulness in capturing the corresponding states.

- **Positive/Negative Affect and Stress**: An adapted [133, 134] 8-item survey was used to assess users' momentary positive and negative affect/stress right before the phone survey was received. Users were asked to rate their level of being happy, great, cheerful, joyful (positive affect), upset, nervous, stressed, and couldn't cope (negative affect/stress) on a 4-point Likert scale ranging from "not at all" (1) to "very" (4).

- **Hunger and Satiety:** Users were asked to rate how hungry they were right before they ate (hunger), and how full they were right after they ate (satiety), on a scale from 0 to 100 [135].

- **Eating in the Absence of Hunger:** A 16-item survey [134] was used to assess the level at which users were eating in the absence of hunger. They were asked to consider reasons why they started eating, and reasons why they kept eating, such as "food looked, tasted or smelled so good" and "my family or parents wanted me to eat" and "feeling sad or depressed". Item responses ranged from "not true" (1) to "very true" (4).

3.4 Collaborative Ground Truth

In contrast to lab settings, where ground truth of the activities of the participants can be acquired from recorded video or real-time observation, ground truth in the wild is generally collected from self-report or by asking the participants. Though some studies [11, 58] use on-body cameras for ground truth in the wild, they are substantially more problematic to use in the home due to privacy concerns. In MFED, we ask a participant for confirmation when an eating event is detected for that participant. The confirmation not only leads to following EMA questionnaires for the participant, but also serves as ground truth for the corresponding eating events. This first-person approach of ground truth collection does not work if the system fails to detect the eating event or the participant does not respond to the EMA survey. Considering that people often eat at home with other family members, we can collect ground truth for the eating event of one person from another family member when they eat together and the later person response to his/her eating EMA survey. We denote this

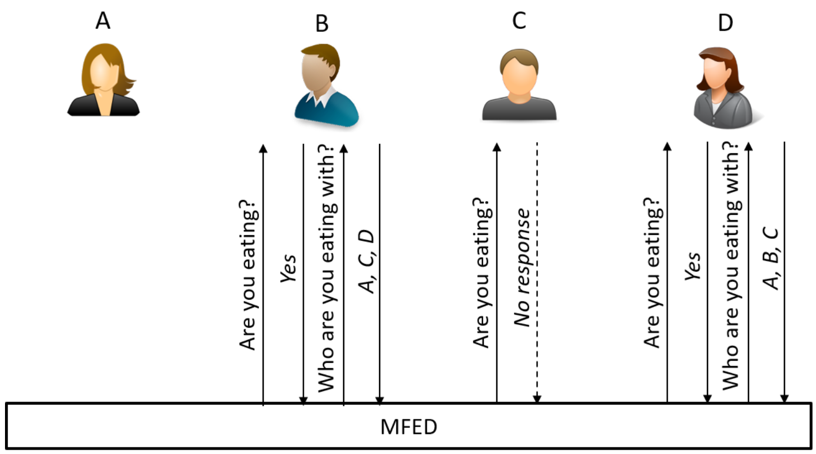


Figure 3.11: A scenario when four family members (A, B, C and D) eat together. MFED detects eating events for B, C and D. C doesn't respond to the eating EMA survey. The responses from B or D can be used to collect ground truth for A and C.

approach as "Collaborative Ground Truth". It is a novel approach for ground truth collection from free-living contexts.

When a participant confirms a detected eating event, the participant is asked about who he/she was eating with. The response to this query provides ground truth for other participants. Figure 3.11 presents a scenario where four family members (A, B, C and D) eat together. MFED detects eating events for B, C and D, and sends an EMA survey to them. However, C doesn't respond to the eating EMA survey. However, B and D confirms that they are eating, and thus provides ground truth for their corresponding eating events. Following their confirmation, B and D are asked who they are eating with. The responses from them can be used to collect ground truth for A and C. Such ground truth is also useful for localization as the two participants are co-located during the eating event.

3.5 Experiments and Results

3.5.1 Eating Gesture Detection (Lab)

Prior to real deployments, we collected data from 29 participants for a total of 42 sessions in lab settings to develop a model for eating gesture detection. The average duration of the sessions was about 24 minutes. The participants ate freely while being video recorded for ground truth purposes. The videos were annotated using the ChronoViz [136] tool. Almost all the participants in our study are right-handed, and so we use data from the right hand only. The participants ate freely during the sessions and engaged with different non-eating activities including reading books, moving around, using phones, and using computers. To capture more activities, we collected data for non-eating activities from free living context involving 4 persons. The total duration of the free-living data is about 18 hours which includes different household activities except eating. We did not video record during these free living sessions since there was no eating activity present there.

The moment when the food or drink reach to mouth is annotated, and an window around that moment is considered as an eating gesture. There are usually some time difference between a moment annotated and the PoI detected from the data for that eating gesture. In order to address this issue and reduce ambiguous instances, we label the potential eating gestures as positive, negative and ambiguous for training purpose. The potential eating gestures within 2 seconds of an annotation are labeled as positive (eating gestures). Those in the range of 2-4 seconds are labelled as ambiguous, and others as negative (non-eating gestures). The ambiguous potential eating gestures are not used for training, but all the potential eating gestures are used for testing the performance of the classification model. In order to determine the True Positives (TP), the False Positives (FP), and the False Negatives (FN), we use the method presented by Yujie et al. [56] which is suitable for eating gesture detection, particularly when there are time differences between the annotated moments and the corresponding detected eating gesture moments. We take a window of length 6 seconds around the potential eating gestures because most eating gesture gestures are captured well by a 6 second long window [58, 59]. We use 32 and 64 filters in the first and second convolution layer, respectively and 100 nodes in each of the dense layer. The experimental results used to select the filter numbers are explained later in this section. We evaluated our method using a leave one person out approach where data from each person is tested by the model developed using data from other persons.

Figures 3.13(a), 3.13(b) and 3.14(a) show the precision, recall and F1-scores of eating gesture detection for different threshold values for X_{th} and V_{th} that represent acceleration along the X axis and the total variance of acceleration, respectively. The average numbers of potential eating gestures per minute are shown in Figure 3.12. The number of potential eating gestures is much smaller for $V_{th} = 1$ than $V_{th} = 0$. However, the number decreases less significantly for further increments of V_{th}. The figure shows that the number of potential eating gestures decreases almost linearly with X_{th}. The precision values do not differ significantly for different values of V_{th} and X_{th}. However, the recall value decreases significantly particularly for $X_{th} < -4$. This is because the more X_{th} decreases, the more true eating gestures are discarded along with non-eating gestures. Figure 3.14(a) shows that F1-scores are less or similar when $X_{th} > -3$. However, it decreases significantly for $X_{th} < -4$. We found that the F1-score reduces significantly for some participants for $X_{th} = -4$. So, we selected 1 and -3 for X_{th} and V_{th}, respectively. The F1-score of our method at these thresholds is 0.74, 12% and 21% better when compared to the methods proposed by Thomaz et al. [58] and Mirtchouk et al. [59] that give F1-scores of 0.62 and 0.53, respectively, for our data.

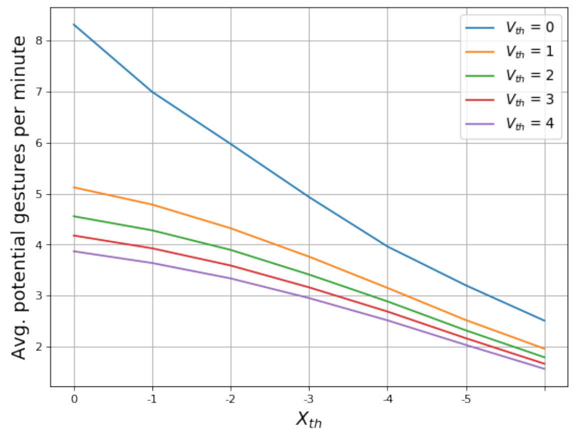
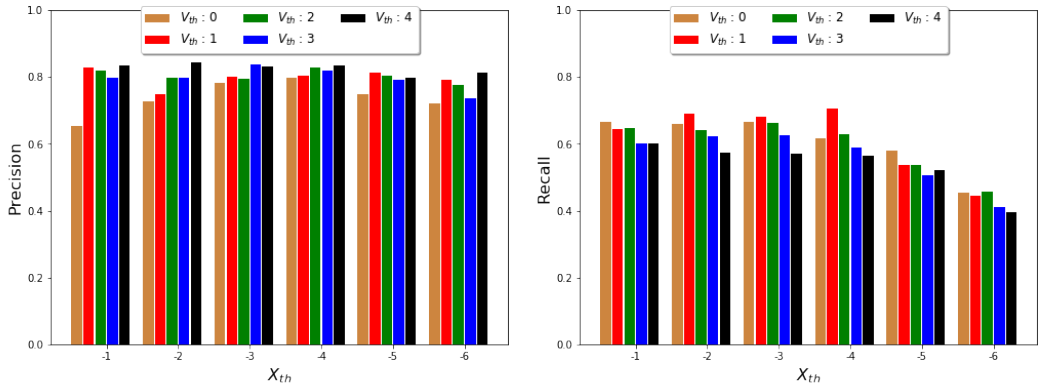


Figure 3.12: Average potential eating gestures per minute for different values of V_{th} and X_{th}

The results for using different number of filters in the convolution layers are shown in Figure 3.15 where we have used same number of filters in both the layers. There is no significant difference in the results for the different filter counts. To reduce the problem of over-fitting on the free-living data, we use 32 and 64 filters in the first and second convolution layers, following the approach of AlexNet[137] that uses fewer filters in the earlier layers than the following layers. In addition to the accelerometer, we also collected data from a gyroscope of the watch in the lab study. Figure 3.14(b) shows the F1-scores of eating gesture detection when a gyroscope is used in addition to the accelerometer. There is no significant difference between using an acceleromter only and using both an accelerometer and a gyroscope. However, a gyroscope consumes significant energy, usually more than an accelerometer [138, 139]. So, we considered not to use the gyroscope for the in-home deployments.

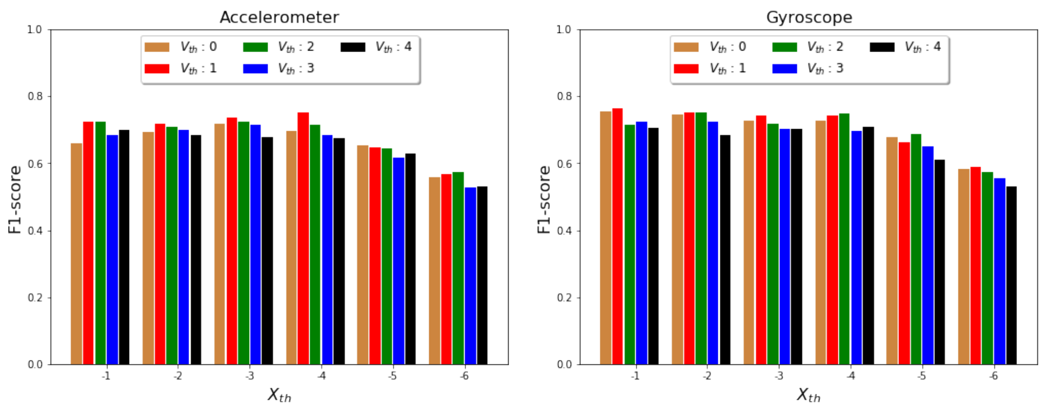
The computation required for the method used for potential eating gesture detection is negligible compared to the methods used for further classification. The sliding window based approach, used widely in state of the art solutions [58, 59], segments the data usually with an overlap between consecutive segments. The number of segments depends on the sliding length. For example, Thomaz et al. [58] segments



(a) (b)

Figure 3.13: (a) Precision and (b) Recall of eating gesture detection for different values of V_{th} and X_{th}

the data into 6 second long windows with 3 second overlap resulting 20 segments per minute. Mirtchouk et al. [59] segments the data into 5 second window with 100 millisecond step size resulting 600 segments per minute. Compared to the state of the art solutions, we use a very small number of segments for classification. For example, with $Xth = -3$ and $V_{th} = 1$, the average number of potential eating gestures (segments) per minute is 3.76, which is less than 1% and 20% compared to the methods of Mirtchouk et al. and Thomaz et al., respectively. This is a significant reduction of computation requirement, and it makes our method more suitable for on-device and real-time processing, particularly for devices that have limited resources. Computation can be reduced further by decreasing X_{th} or increasing V_{th}, but compromising performance to some extent (e.g. using $X_{th} = -5$). Figure 3.14 shows that the thresholds we selected ($X_{th} = -3$ and $V_{th} = 1$) gives similar or even better results than greater values of X_{th} or smaller values of V_{th} that result more potential eating gestures. Our method reduces computation requirement without any compromise of performance.



(a) (b)

Figure 3.14: F1-score of eating gesture detection (a) using an accelerometer (b) using both an acceleromter and a gyroscope.

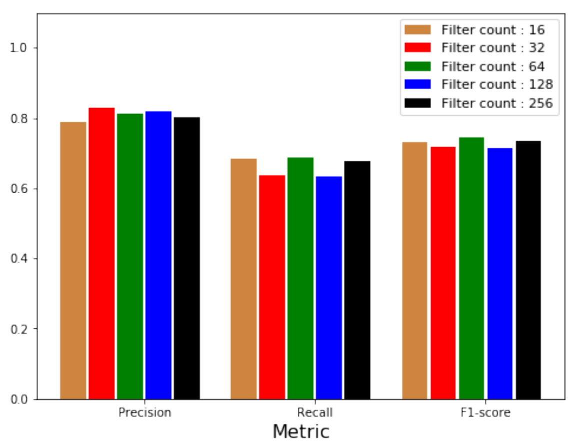


Figure 3.15: Precision, recall and F1-score for different number of filters. Same number of filters are used in both layers.

3.5.2 In-Home Deployments

We recruited families that have at least one adult parent and one child between the ages of 11 and 18 years old living in Los Angeles, California. Children under the age of 11 were not eligible to participate in the study. We deployed the system in 20 homes with a total of 74 participants. Thirty-nine children (Average age = 15.3), fourteen adult males (Average age = 44.7), and twenty-one adult females (Average age = 45.0) were included in the sample. The majority of participants identified as Hispanic or Latino (64.9%); 14.9% identified as White or Caucasian, 10.8% as mixed race, and 9.4% as other. Table 3.1 shows the number of participants from each of the families. For some families, all members were not interested in participating in the study. However, mothers from all the families participated in the study. The number of participants from a family ranges from 2 to 5.

We provided each of the participants with an Android smartwatch and an Android smartphone during the deployment period. The participants were instructed to primarily use the phone for study purpose. Because the scope of the study focused on in-home family eating dynamics, the participants were told to wear the watch only at home and during active daytime. Beacons were not placed in bedrooms or bathrooms to preserve the privacy of the participants. The system is deployed in each home for about 2 weeks. We have two protocols approved by Institutional Review Board (IRB) to collect the lab data and to deploy the system at homes, respectively.

EMA Responses:

Overall, the system sent a total of 14413 EMAs to the participants with 13776 and 637 EMAs for mood and eating, respectively. The participants responded to 4750 of the EMAs (4224 for mood and 526 for eating). Currently, MFED does not localize the participants or the study phones in real time. The EMAs are sent regardless of

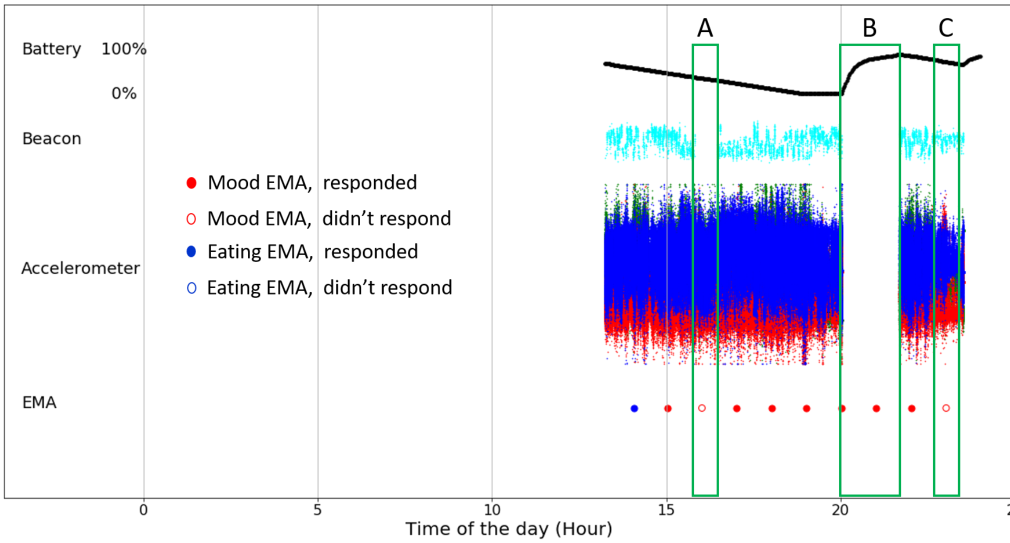
Family ID	**Adult Female**	**Adult Male**	**Children**
1	Y		YY
2	Y	Y	Y
3	Y	Y	Y
4	Y	Y	YY
5	Y	Y	Y
6	Y	Y	YY
7	Y	Y	YY
8	Y	Y	YY
9	Y		YY
10	Y		YYYY
11	Y	Y	YY
12	Y		YY
13	Y	Y	YYY
14	Y		YY
15	Y	Y	YY
16	Y		Y
17	Y	Y	YYY
18	Y	Y	YY
19	YY	Y	Y
20	Y	Y	YY

Table 3.1: Number of participants with different age groups in the families. The count of 'Y' in a cell indicates the number of participants in the corresponding age group from the corresponding family.

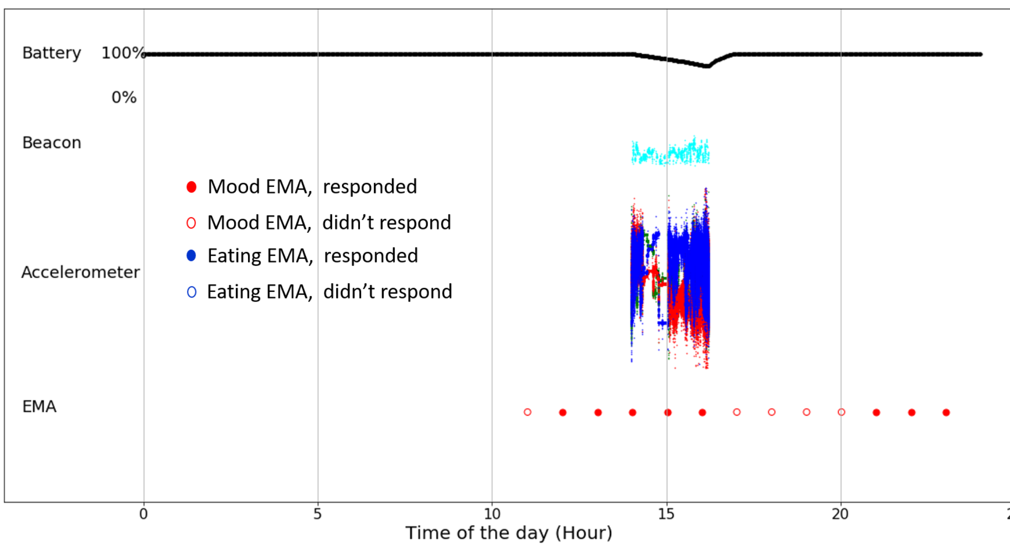
the location of the participants or the study phones. So, there are instances where a user might be out of home or away from his/her study phone when an EMA is sent.

To better explain the EMA response rate and as a clarifying example, the daylong data for two days from two different participants are shown in Figure 3.16. We see that the participants wore the watch for a part of the days, which is typical as they spend time outside of home for many purposes including work, school, sports, and shopping. Also, the watches might run out of energy while they are being used. In Figure 3.16(a), we see that the participant started wearing the watch after the noon, and there is a gap when the watch was being charged. The red and blue circles represent mood and eating EMAs, respectively, and the filled circles represent the EMAs that the participants answered. We see that the participant (Figure 3.16(a)) responded to all the EMAs that day except two.

Three scenarios are marked in the figure. For scenario A, accelerometer data is available, but there is no beacon data. We see that the participant did not answer to the EMA sent during that time. We do not know whether the participant was at home or not because the participant was either outside of home or somewhere in the home where the beacon signals were not available. During scenario B, the watch was being charged, and there was no accelerometer and beacon data. However, the participant answered to the EMAs. So, the participant was at home during that time. For scenario C, we know that the participant is at home because both beacon and accelerometer data are available, but he/she did not answer to the EMA sent during that time. In Figure 3.16(b), the participant wore the watch for about 2.5 hours, and then began charging it. The participant answered all the EMAs while wearing the watch, but then did not answer several other EMAs. In this case, we don't know whether he/she was at home when the EMAs are not answered. At night, the participant started answering the EMAs, and it indicates he/she was at home during that time but forgot to wear the watch.



(a)



(b)

Figure 3.16: Daylong data of battery percentage, beacon RSSI, accelerometer and EMA responses for two days from two different participants. (a) Three different scenarios (A, B and C) showing availability of accelerometer and beacon data as well as whether the user responded to the EMAs. (b) A scenario where the participant wore the watch for about 2.5 hours, and then began charging it. The participant answered all the EMAs while wearing the watch, but then did not answer several other EMAs.

The participants replied to about 31% of the hourly EMAs. Though the response rate is low, it is reasonable and expected. People spend a significant amount of time outside their home. The participants of our study do not carry the study phone outside home, and thus cannot respond to the EMAs during those times. Our system does not track whether a participant is at home or not. Using the smartwatches for such tracking is not practical as users do not wear watches all the time for different reasons including for charging and forgetting to wear the watch, as illustrated above. Also, users do not usually wear the watch when they are out of home. Since we do not know the location of the user, we send the mood EMAs every hour regardless of the users location. So, the response rate to the hourly EMAs is expected to be low.

The response rate for the EMAs differs from person to person. Figure 3.17 shows the response rates for each of the participants that are grouped with same color for the same family. We see that the response rates differ significantly from family to family. However, the correlation coefficient between the response rates of the participants and the mean response rate of the corresponding homes is 0.87. This indicates that the response rates among family members are highly correlated. The rates of EMA responses differs at different times of the day as depicted in Figure 3.18(a). There are more responses in the evening compared to morning and noon. This might be because people are less likely to be at home during the morning and noon than evening for reasons including work and school. Figure 3.18(b) shows the EMA response rates for different days of the week. It shows that participants responded to EMAs more during the weekend than the weekdays. This may be because people are more likely to be at home or less likely to be busy during the weekends.

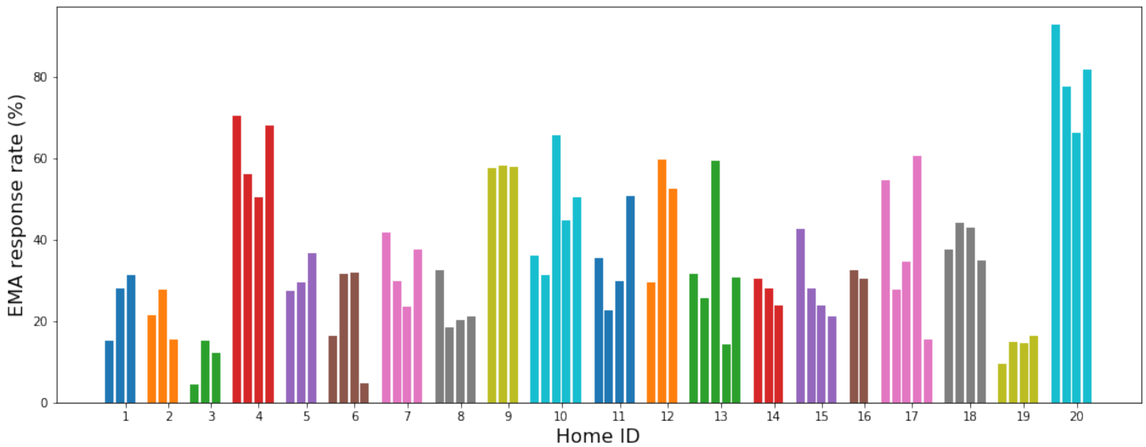


Figure 3.17: EMA response rate for each of the participants that are grouped with same color for the same family.

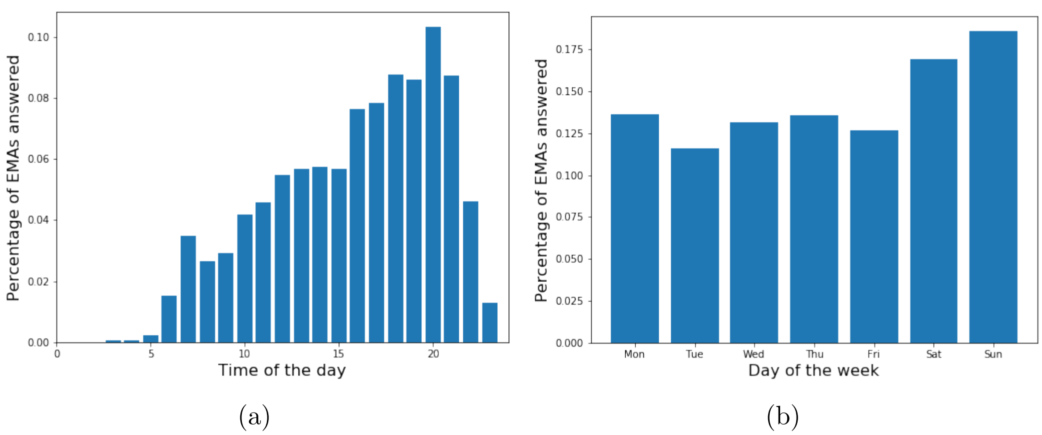


Figure 3.18: EMA response rates at different (a) time of the day and (b) days of the week.

Activities	**Count**
Using personal phone	67
Smoking	2
Fixing own hair	4
Putting on sunscreen or lotion	4
Other	71

Table 3.2: Activities during falsely detected eating events

Eating Events:

The participants answered to 526 of the eating EMAs out of the 637 sent. They responded that they were eating for 383 of the EMAs (256 meals, 87 snacks, 28 drinks, and 12 undefined). In cases where the participants were not eating, they were asked about what they were doing. There were five options available to the users: 1) Using a personal phone, 2) Smoking, 3) Fixing hair, 4) Putting Sunscreen or lotion, and 5) Other (Open text area). The participants could select one or more of the options and could provide text inputs in the open text area. The participants selected only one option for all the EMAs except for two EMAs where they selected 2 (*Using my phone* and *Other*) and all the 5 options. The response count for different activities are listed in Table 3.2. It shows that using the phone is one of the most confounding gestures for eating. The most frequent activities mentioned in the open text area include using a computer/laptop and Watching TV/movies. There are wide variety of activities that were mentioned in the open text area. It indicates that many activities found in the wild confounds with eating. The responses provided by the users can be used as ground truth for activity recognition tasks, particularly for eating activity detection.

The participants were asked about other persons who were eating with them. The options available for this query are Nobody, Spouse/Partner, Child(ren), Mother, Father, Sister(s), Brother(s), Grandparent, Other family, Friend(s) and Other people. Figure 3.19 lists the frequency of the family members or others present during eating. It shows that both eating alone and eating with family members are common at homes. Figure 3.20 shows the breakdown of meals, snacks and drinks during eating alone and eating with others. It depicts a phenomenon of eating at home - people eat meals together more than alone and eat snacks alone more than with others.

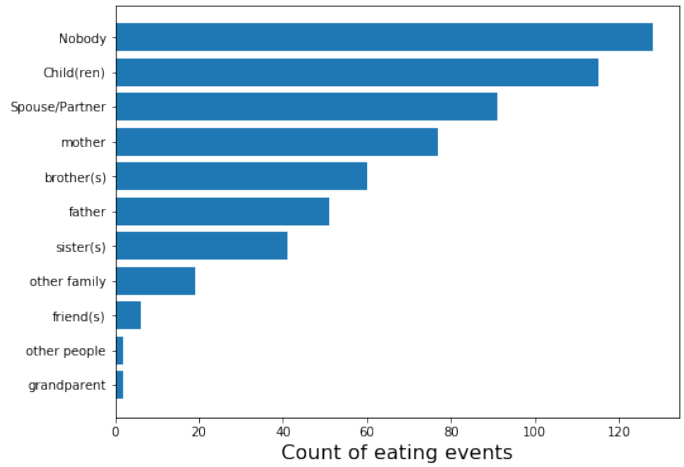


Figure 3.19: Frequency of other family members or other persons eating with the participant who responded to the eating EMA.

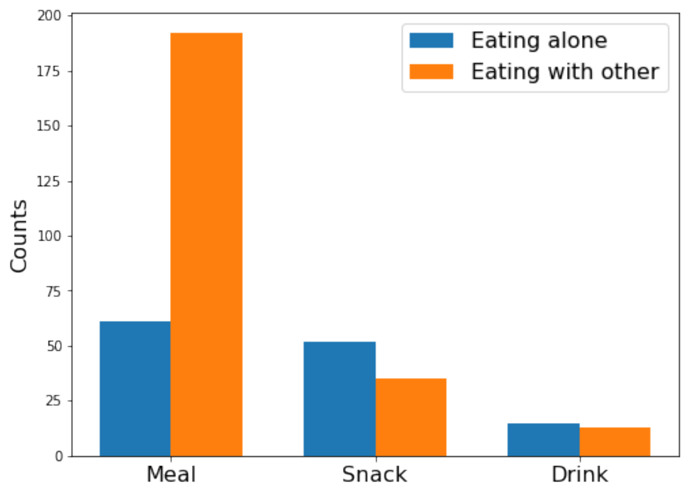


Figure 3.20: Counts of meals, snacks and drinks while eating alone and eating with others.

Collaborative Ground Truth:

We use the family role of each of the participants and the relationship between the participants for collaborative ground truth. For example, when a child in a family answers that she is eating with her mother, it provides ground truth for the mother. If the system fails to detect that eating event for the mother or she does not respond to her eating EMA, the response from the child can be used as ground truth for the mother.

There are 240 EMA responses where the participants mentioned that they were eating with others. As mentioned earlier, all members were not interested in participating in the study for some families. There are 68 EMA responses where the participants mentioned they ate with some family members (spouse/partner, child, mother, father, brother, sister) who didn't participate in the study. There are 101 EMA responses where the participants mentioned the relationship with other participants she was eating with, but the other participant cannot be unambiguously identified because multiple similar participants are available at that home. For example, when a mother said that she is eating with her children, and there are more than one child in the home, we cannot unambiguously determine which child she was eating with. In 165 of the EMA responses the participants mentioned about a total of 257 other participants who can be detected unambiguously. However, multiple participants may mention about one participant. For example, two children may mention that they were eating with their mother. In such case, we get the ground truth for the mother from two sources. There are 35 such instances, and so we get ground truth for 222 instances through the collaborative approach. We considered a time window of 15 minutes for this purpose. 56 instances also have ground truth through the first-person approach where the other participants responded to their EMA survey. So, the collaborative approach gives ground truth for 170 instances for which first-person ground truth is not available. There are several reasons for the absence of first-person ground truth for eating events. Firstly, the participant might not wearing the watch during the meal, and so no eating event was detected for that participant. Such scenarios are common when the watch is being charged or the participants forget to wear the watch. Secondly, the system might have detected the eating event, but the participant did not respond to the EMA. It might happen when the phone is not near the participant and he/she does not see the notification. Thirdly, our system might fail to detect that eating event for the participant.

3.6 Discussion

MFED is designed and developed to capture a wide range of information related to family eating dynamics. The broader purpose is to build dynamic and networked models of FED, and use these models to drive personalized, adaptive and just-in-time interventions that have potential to be effective in the long term modification of eating behavior and prevention of obesity. Modeling FED, designing interventions or validating the effectiveness of FED approach are beyond the scope of this work. The goal of this work is to build a foundation that would support future endeavors related to family eating dynamics. EMA is not associated with the localization of the participants, and this work does not address the problem of localization. In future, we will incorporate localization features in MFED. The beacon RSSI data collected from the deployments will be helpful to develop solutions for in-home localization of the users.

In contrast to lab-studies, real-world deployments need to address many challenges including usability, user convenience, and resource constraints of the devices. A wristwatch is a very common personal device, and there is almost no inconvenience in using it. Considering all the issues, MFED uses a single smartwatch for eating detection, though detecting eating activities in the real-world using a smartwatch has proven to be challenging [11, 58, 59]. Additionally, MFED requires detecting the eating events in real-time. We designed the system to address these challenges and incorporated features to reduce burden on the users. For example, the watch app starts automatically when the watch restarts, and so the users do not need to start the app. In fact, the users do not interact with the watch app at all. It runs seamlessly without user intervention.

Before real deployments, we collected data from lab settings to build eating gesture classification models. We also included data for non-eating activities from free-living context. The eating gestures associated sensor data differ significantly based on several factors including the type of food, utensils (e.g., spoon, fork, chopstick, bare hand) used to eat the food, body postures, context, and individual differences. These factors along with confounding gestures from non-eating activities make eating gesture detection challenging, particularly in the the wild. Though the lab data captures a wide range of gestures, they are still limited compared to the diversity of gestures in the real-world. The data we collected from the deployments provide ground truth for many eating events, most importantly from real home contexts. Though there is no ground truth available for individual eating gestures, the ground truth for eating events can be exploited with semi-supervised methods to build more robust and accurate eating gesture detection models.

In addition to eating gesture detection, it is also challenging to define an eating event from detected eating gestures. The intervals between eating gestures are usually irregular and depend on factors like context, food type, and habits. In addition to irregular intervals, the accuracy limitation of the eating gesture classification model makes it more challenging to detect the eating events. We use a heuristic as proposed by Mirtchouk et al. [59] that clusters the eating gestures within one-minute intervals, and then we detect eating events using these clusters. Future work includes using the data from the deployments to better understand the structures of eating events, and to develop better methods for eating event detection. Personalized models usually work better than general models, and in the future, we will incorporate personalized models in MFED for eating gesture and eating event detection.

We did not attempt to keep journals or ask participants to fill in missed meals. So, it is not possible to detect the false negatives as our system fails to detect the eating events. We get only the true positives and the false positives. Any model that increases true positives and decreases false positives for this data is likely to decrease false negatives. Future works include using the data to develop more accurate models that reduces false positives. Though it is not possible to verify the false negatives, if a model detects eating events with very few false positives, the newly detected eating events (that were not detected during deployment) can be used to estimate the proportion of false negatives. The data we collected will facilitate such research in the future.

The optimal frequency with which to assess stress and mood via ecological momentary assessment, with regard to both accuracy and compliance, is still an open question in the field of behavioral science. The data we collected in this study will facilitate future research in this area. However, to balance the trade-off between collecting more temporally granular data and the convenience of the users, we chose 1-hour intervals for measuring within-subject changes in mood and stress. Future works include finding out better timing and frequency for EMA surveys using the data collected from this study. Technologies for automatic detection of mood and stress, particularly using wearables [140, 141] are advancing. In order to reduce the burden on the users and to gather more temporally granular data on mood and stress, future works should focus on incorporating automatic solutions for mood and stress detection in MFED, ensuring the privacy of the users. Due to the small form factor, smartwatches are not suitable for EMA surveys. However, voice-based interactive systems like MedRem [142] can be developed in the watch for EMA purposes. In the future, we will explore the feasibility of using such a solution for MFED.

Though we run the model in the base station, energy is still a critical issue for the watch because we detect eating in real time, and streaming data continuously to the base station would drain significant energy from the watch. Computation and memory are not critical for eating detection on the base station, but they are critical for the watch. However, the base-station detects eating from all the watches of the corresponding home in real-time. So, an efficient method allows us to use low-cost device as a base-station. Since our method requires low computation and memory, it can be used in further works that attempt to detect eating events in a watch.

The collaborative ground truth is more feasible for a home than outside the home because there are usually few people in a home, and the relationship among participants are defined. For example, when a daughter confirms that she is eating with her father, we know who her father is. Since a person can eat outside of home with many different people with different relationships, collecting such ground truth would be relatively more complex. However, the collaborative ground truth has potential applications for other home based applications beyond MFED and also for other settings where the relationship between the participants can be defined.

The dataset collected from the deployments consists of accelerometer, battery and beacon readings from smartwatches as well as EMA responses from the users. This is a unique dataset that will be made public, and it would be invaluable for future research related to family eating dynamics and other family based systems. The dataset can be used in building better FED monitoring systems as well as new, dynamic and networked models of FED that will support real-time, in the wild interventions. Approaches based on family eating dynamics have the potential to be very effective in addressing obesity. Our work lays the foundation that would support future endeavors to tackle the obesity problem using FED.

Chapter 4

Hand Washing Detection

Hand hygiene is extremely important for personal well-being as well as in healthcare settings and food businesses. It is one of the most effective tools in preventing healthcare associated infections (HAIs) in hospitals [143–145]. In the year of 2011, the number of HAIs in acute care hospitals of the USA was estimated to be 0.72 million, and deaths associated with HAIs were about 75 thousand [146]. The annual direct medical cost of HAIs to U.S. hospitals is estimated to be in the range of 28 to 45 billion US dollars [147]. The rates and risks in the low and middle income countries are significantly higher compared to the high-income or developed countries [148]. Hand hygiene of food workers is essential in preventing food contamination and food-borne illness. One study shows that food contamination by food workers is responsible for about 89% of the food-borne illness outbreaks [149].

Washing hands properly is the cornerstone of hand hygiene compliance. However, adherence with hand washing practices among health care workers is significantly low compared to the requirements and depend upon a number of factors like demographic characteristics of the health care workers, accessibility of hygiene product supplies, workload, and individual cognitive factors [150]. It is very important for the hospital/business authorities as well as public health agencies to monitor and measure hand hygiene adherence by the workers and to provide feedback to the stakeholders. Different methods exist for this purpose like surveys, self-reporting, direct observation by human observers, indirect observations based on product utilization, and automated monitoring systems. Surveys and self-reporting approaches require significant human effort. Results from these approaches are often incomplete, error-prone, and biased. Direct observations by human observers require lots of human effort and often result in biased data and an uncomfortable working environment. The indirect observation method estimates the number of hand washing events based on the utilization of handwash products. This method requires less effort than direct observation, but it is limited in providing personal and temporal information, like who uses the products, and when. Automated monitoring systems require almost no human effort and provide more accurate results. Inertial sensors available in smartwatches can capture hand movements, and so it is feasible to develop an automatic system for monitoring hand washing. Real-time monitoring of hand washing can also facilitate just-in-time alerts to the users if they forget to wash hands when necessary.

A major challenge toward developing solutions for hand washing detection is the NULL activities. We need data from hand washing activities as well as from other activities to train and validate a classification model. As mentioned earlier, it is almost impossible to enumerate, let alone collect data, for all possible human activities. The fact is that the datasets developed to train and evaluate activity recognition models contain a limited number of NULL activities. As a result, the solutions are likely to perform poorly in the presence of unseen activities from free-living context.

In this thesis, we developed a novel method to mitigate the problem of NULL activities. We leverage the distribution of the output of the penultimate layer of a neural network to detect out-of-distribution samples that mostly come from unseen activities. We collected a dataset that contains data from hand washing as well as several other activities. We trained a neural network model using our dataset and then

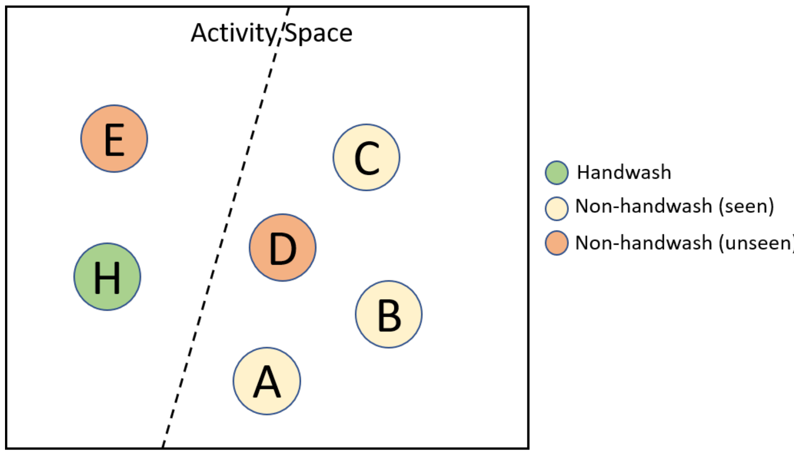


Figure 4.1: Activity space

tested the robustness and effectiveness of our solution using WISDM [106], a publicly available dataset that does not contain any data for hand washing. This dataset has data for 18 different activities, many of which are not present in our dataset. Our method reduces the false positives from the WISDM dataset by about 77% and improves overall F1-score by 0.17, that is 30% more than the baseline method.

4.1 Problem

The dataset available to train a neural network has data from hand washing and a limited number of other activities. We need to develop a solution for hand washing detection that is robust against unseen activities. We illustrate the problem in Figure 4.1 using an example where we have data for hand washing (H) and three other NULL activities (A, B and C). We do not have data for D and E that are also NULL activities. If we train a neural network using data for hand washing (H) and available NULL activities (A, B, C), the network would learn a decision boundary using these data, and consequently, activity F might be detected with very high probability as positive. A robust solution should detect F as negative without/with a very little compromise on the performance of hand washing detection.

4.2 Method

Neural networks are effective in recognizing different activities. Most of the state-of-the-art methods on activity recognition from wearable sensors focus on the architecture of the network and the parameters used to train and evaluate the models [12–14, 129]. We also use a neural network for hand washing detection but additionally leverage the distributions of the penultimate layer outputs of the network to detect NULL activities. Each layer of a neural network transforms its input features to another feature space. The output of the penultimate layer represents the final features that are usually classified by a Sigmoid or Softmax function. Figure 4.2 shows an example of a feedforward neural network with three hidden layers and one output node. The input features (F_i) are sequentially transformed to final features (F_3) where instances from the same classes come closer and from different classes moves further compared to the features from earlier layers. Figure 4.3 shows some instances from hand washing and other activities at different layers of the feedforward neural network. There are 64 nodes in each of the hidden layers, and we have used t-SNE [88] method to embed the outputs of each of layer into 2-dimension for the purpose of visualization. As the instances are better separable over the layers. The decision boundary of the network is computed on the final layer features (Figure 4.3(d)) to classify the instances. Here, we have drawn a hypothetical decision boundary just for illustration purposes. Any instances left to the boundary will be classified as hand washing and to the right as others. Now, instances in the area X and Y are from unseen activities or gestures and it is most likely that they are not from hand washing. Though the instances from Y are classified as negative by the decision boundary, instances from X are classified as hand washing.

We use class conditional Gaussian distribution of the penultimate features to detect the out-of-distribution (OOD) instances. Our method is not confined to networks

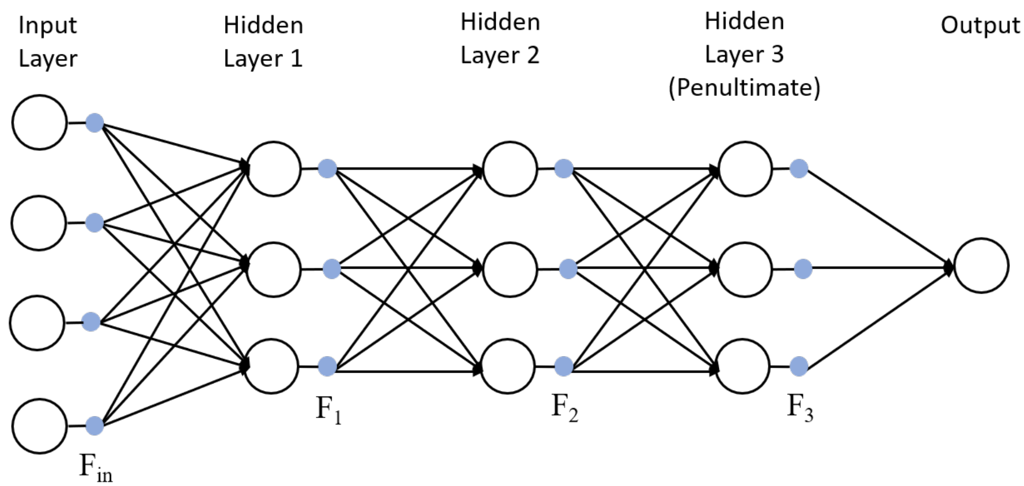


Figure 4.2: Example of a Feedforward Neural Network. Instead of connecting a node directly with the nodes in the following layer, we put a connector in between to better show the input and embedded features.

with a specific architecture. We use a pre-trained network, and get the penultimate features for a set of hand washing instances, referred to as Representative Set. For a set of representative instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, the mean and covariance are computed as:

$$\hat{\mu} = \frac{1}{N}\sum_i \mathbf{x}_i \tag{4.1}$$

$$\hat{\Sigma} = \frac{1}{N}\sum_i (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^T \tag{4.2}$$

Here, $\mathbf{x}_i$ represents the penultimate feature i.e. the output of the penultimate layer for the i^{th} instance, and so $\mathbf{x}_i$ is a N dimensional vector where N is the number of nodes in the penultimate layer. It should be noted that all the instances of the representative set are from hand washing. We compute the distance of an instance from the mean($\hat{\mu}$) using Mahanabolis distance [151] as:.

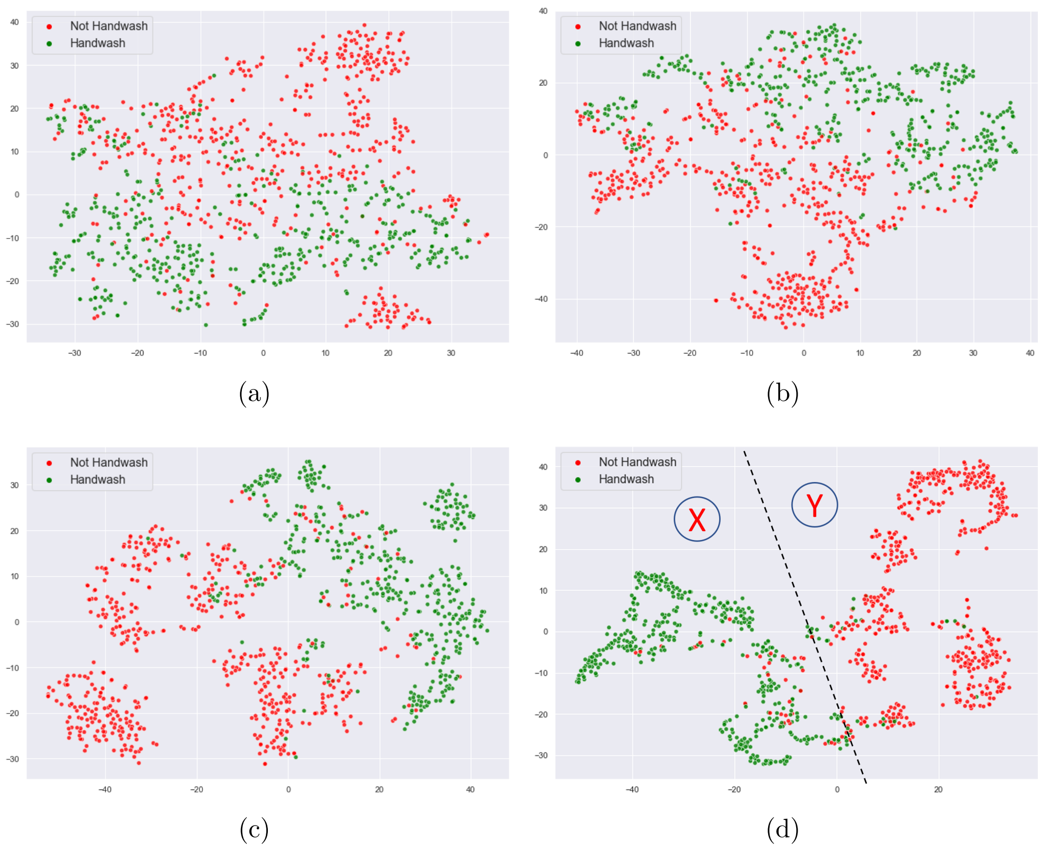


Figure 4.3: t-SNE representation of some hand washing and other instances for (a) Input features, (b) output of layer 1, (c) output of layer 2 and (d) output of layer 3 of a feedforward neural network with 3 hidden layers.

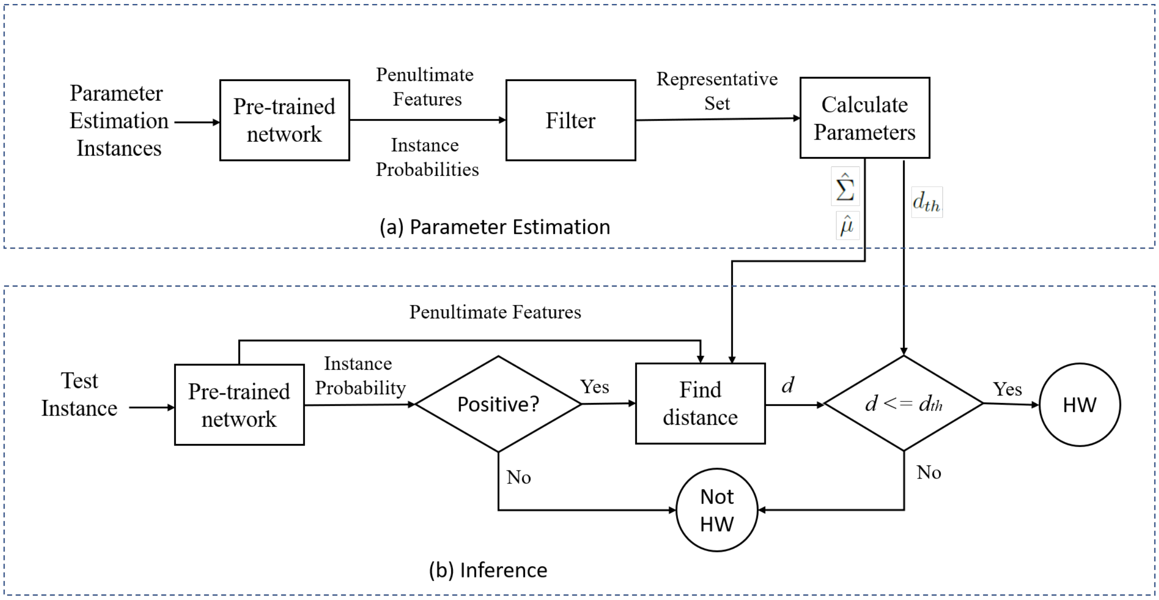


Figure 4.4: Steps in (a) Parameter Estimation (b) Inference

$$d(\mathbf{x}) = (\mathbf{x} - \hat{\mu})^T \ \hat{\Sigma}^{-1}(\mathbf{x} - \hat{\mu}) \tag{4.3}$$

The Mahalanobis distance is usually more effective than Euclidean distance for detecting OOD samples [81]. Using the representative instances, we calculate a distance threshold (d_{th}) that is used to determine the OOD samples.

As mentioned earlier, we use a pre-trained network, and so training a network is not part of our solution. We need to estimate three parameters: $\hat{\mu}$, $\hat{\Sigma}$ and d_{th}. These parameters are used to infer the class of test instances. Figure 4.4 shows the steps for parameter estimation and inference. We describe them in more detail below.

4.2.1 Parameter Estimation

To estimate the parameters, we use a set of instances from hand washing, called Parameter Estimation Instances. It can be the hand washing instances from the dataset used to train and evaluate the network or any set of similar instances. We get the penultimate features of the instances and their probability of being hand washing using the pre-trained network. The penultimate features of the instances that are detected as positive (hand washing) by the network are used to construct the representative set. Our goal is to further test only the instances that are detected as positive by the network. So, we do not include the instances detected as negative into the representative set. We use equation 4.1 and 4.2 to estimate the mean and covariance of the selected penultimate features. We calculate the Mahanabolis distance of each of the penultimate features using 4.3 and then select some percentile (P) of the distances as the distance threshold. With a P percentile distance threshold, $(100-P)\%$ of the instances truly detected as positive (True Positives) by the network are discarded as negative or OOD instance by our method. However, it would discard many negative instances that are falsely detected as positive (False Positives) by the network. We discuss this in more detail in Section 4.3.3.

4.2.2 Inference

To infer the class of a test instance, we find its probability using the pre-trained model. If the instance is detected as hand washing, we use the penultimate features of the instance to find its distance. If the distance is greater than the threshold, the instance is detected as a negative or an OOD sample; otherwise, it is considered a positive instance. We do not process any instance that is detected as negative by the network. Our solution discards many False Positives from the network with a small compromise on True Positives, resulting in significant improvement in different performance metrics.

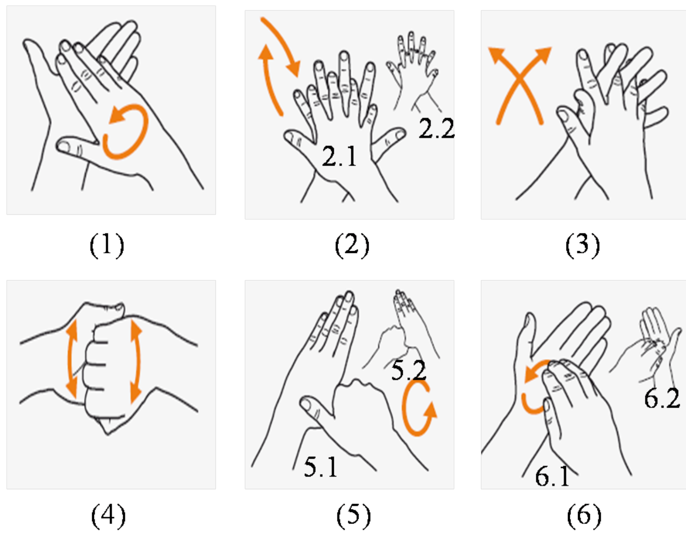


Figure 4.5: Guideline on hand washing by World Health Organization (WHO)[79]

4.3 Experiments

4.3.1 Data Description

We have developed a dataset, called HAWAD (Hand Wash Dataset), by collecting data from 16 participants (9 males, 7 females) with age range between 17 to 36 years. The participants washed hands naturally as well as following the guideline by the World Health Organization (WHO) [79], as shown in Figure 4.5. Though it is extremely important for health and food workers to follow this guideline, people in general often rub their hands in such ways to wash their hands more thoroughly. The participants also performed random gestures and different activities including wiping water from hands with a towel or napkin, walking, opening/closing doors, using computers/phones, eating, and drinking. We have collected the data using Samsung Gear Live, an Android-powered smart watch. The dataset contains about 5 hours of data from each of the hands where nearly half of the data are from hand washing. We have collected acceleration, rotation rate, linear acceleration and gravitational acceleration from the smartwatches.

We have also used a public dataset, named WISDM [106], that contains accelerometer and gyroscope data for 18 activities from 51 subjects. The activities are walking, jogging, climbing stairs, sitting, standing, typing, brushing teeth, kicking a soccer ball, playing catch with a tennis ball, dribbling a basketball, writing, clapping, folding clothes, drinking from a cup and eating soup, chips, pasta, and a sandwich. This dataset does not have any data for hand washing and so we use this dataset to detect out-of-distribution patterns. In the dataset, there are data available from both a smartphone and a smartwatch. We have used data from the smartwatch only.

4.3.2 Network Training

The WISDM dataset has both accelerometer and gyroscope data from the smartwatch. However, the data from these two sensors are not synchronized in the dataset. Also, using a gyroscope in addition to an accelerometer doesn't improve the performance significantly for hand washing detection, but consumes a significant amount of battery life from the watch [131]. So, we use data only from the accelerometer of the watch in our experiment. The accelerometer data from the watch are time series in nature. We segment the data into 1 second long frames with 0.5 second window sliding, and extract a set of features including mean, variance, root mean square, median, first quartile, third quartile, minimum, maximum, skewness, kurtosis from each of the axes of the sensors. We also use the covariance among the axes resulting a total of 33 features. There is no pre-trained neural network available for hand washing detection from accelerometer data. So, we used our HAWAD dataset to train a feedforward neural network. There are 64 nodes in each of the layers of the network and we evaluated the models for different numbers of layers. We split the data into training (80%) and testing (20%) sets with random sampling. A dropout rate of 0.25 and a validation set (10% of the training data) along with early stopping mechanism

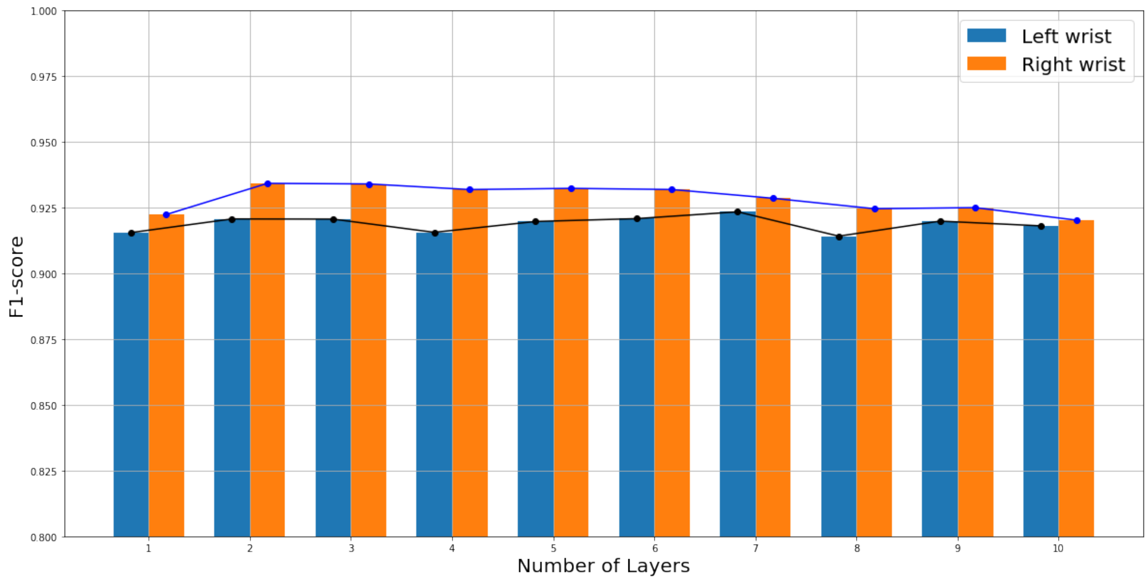
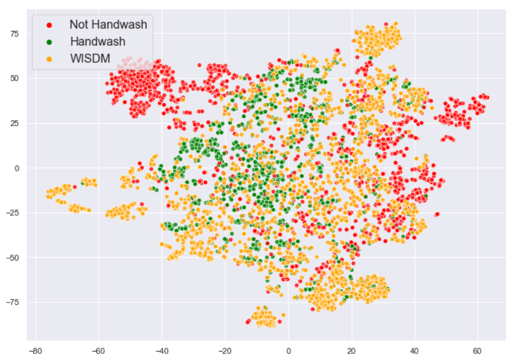


Figure 4.6: F1-score for networks with different number of layers.

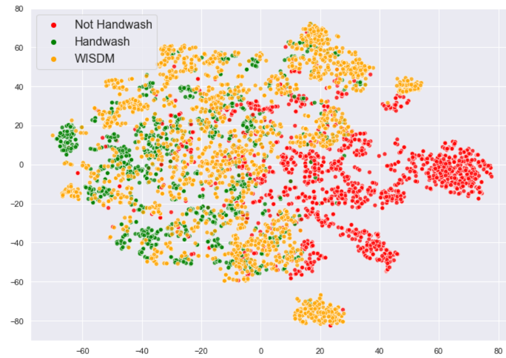
are used to reduce the problem of over-fitting. We developed and evaluated models for the left and the right hand separately. Figure 4.6 shows the F1-scores on the test data for a different numbers of layers. The performances don't differ significantly, but it reduces as the number of layer increases, particularly for the right hand. This is because the network is over-fitted as more layers are added. We have used the network with three hidden layers for the remaining experiments.

4.3.3 Out of Distribution

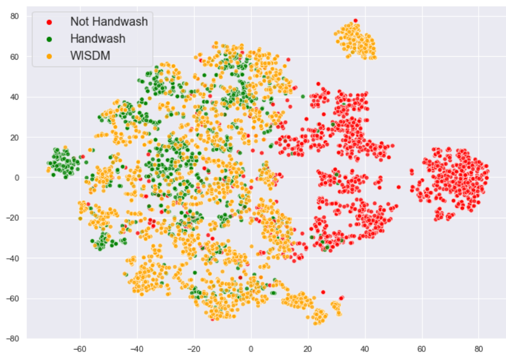
We used the pre-trained model to predict the instances from the WISDM dataset, and about 6% and 5% of the instances from the dataset are detected as false positives for the left and the right wrist, respectively. Figure 4.7 shows the output of each of the layers of the network for the hand washing and NULL instances from the validation dataset as well as some false positives from the WISDM dataset. There is some overlap in the feature space between the false positives from the WISDM dataset and the hand washing instances in the penultimate layer (Figure 4.7(d)). The neural



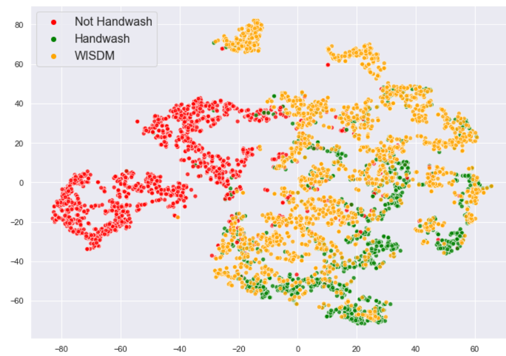
(a)



(b)



(c)



(d)

Figure 4.7: t-SNE representation of the validation instances as well as some False positives from the WISDM dataset for (a) Input features, (b) output of layer 1, (c) output of layer 2 and (d) output of layer 3 of the network.

network uses a decision boundary to distinguish between the hand washing and the NULL instances. All the false positives from the WISDM dataset are in the area of hand washing, though many of them are far away from the the hand washing instances. Our method detects such false positives using the distance threshold. It should be noted that the penultimate features are 64 dimensional for the network, and we used t-SNE to visualize it in 2-D space. Also, the t-SNE representation differs widely based on parameter settings. So, the figures here are not representative of the decision boundary, but they provide insights into the data.

We used the hand-washing instances from our dataset to estimate the parameters ($\hat{\mu}$, $\hat{\Sigma}$, d_{th}). As mentioned earlier, a percentile is used to determine the distance threshold. Any instance detected as positive by the network is detected as negative by our method if the distance of the instance is greater than the threshold. The more we reduce the percentile the more instances are detected as negative by our solution and vice versa. Though a number of true positive instances may be detected as negative by our solution, a large number of false positives are correctly detected as negative, ultimately improving the overall performance. We define TPDNR (rate of the True Positives Detected as Negative) and FPDNR (rate of the False Positives Detected as negative) as:

$$TPDNR = \frac{Number\ of\ True\ Positives\ Detected\ as\ Negative}{Number\ of\ True\ Positives} \tag{4.4}$$

$$FPDNR = \frac{Number\ of\ False\ Positives\ Detected\ as\ Negative}{Number\ of\ False\ Positives} \tag{4.5}$$

Figure 4.8 shows the TPDNR and FPDNR for the left and the right wrist, respectively. Tthe WISDM dataset does not have any data for hand washing, and so there is no true positive instances from this dataset. The results show that our solution corrects a significant portion of the false positives from both the HAWAD and the WISDM dataset. The less the percentile (and so the distance threshold), the more false positives are corrected. However, a small portion of the true positives is also detected as negative. For example, with 95 percentile threshold, we detect 5% of the true positives as negative but reduce the false positive rate from the WISDM dataset by 48% and 45% for the left and right wrist, respectively. It also reduces the false positives for our dataset by 27% and 25%, respectively. The results show that our method is very effective in detecting instances from unseen data and activities. The more we reduce the percentile, the more false positives are corrected, but it also results in more mistakes for the true positives. The percentile should be set according to the requirements of the applications on some metrics like precision, recall or

F1-score. At the 100 percentile threshold, we select the distance using the maximum distance of all the true positives. There are some true positives that lie far away (outliers) from the mean of the Gaussian distribution. Consequently, the distance at 100 percentile is very high, and very few (nearly zero) false positives are discarded, even from the WISDM dataset. Setting the percentile to 99 corrects about 22% false positives from the WISDM dataset. This is because the distance threshold is reduced significantly compared to 100 percentile due to the removal of the outliers.

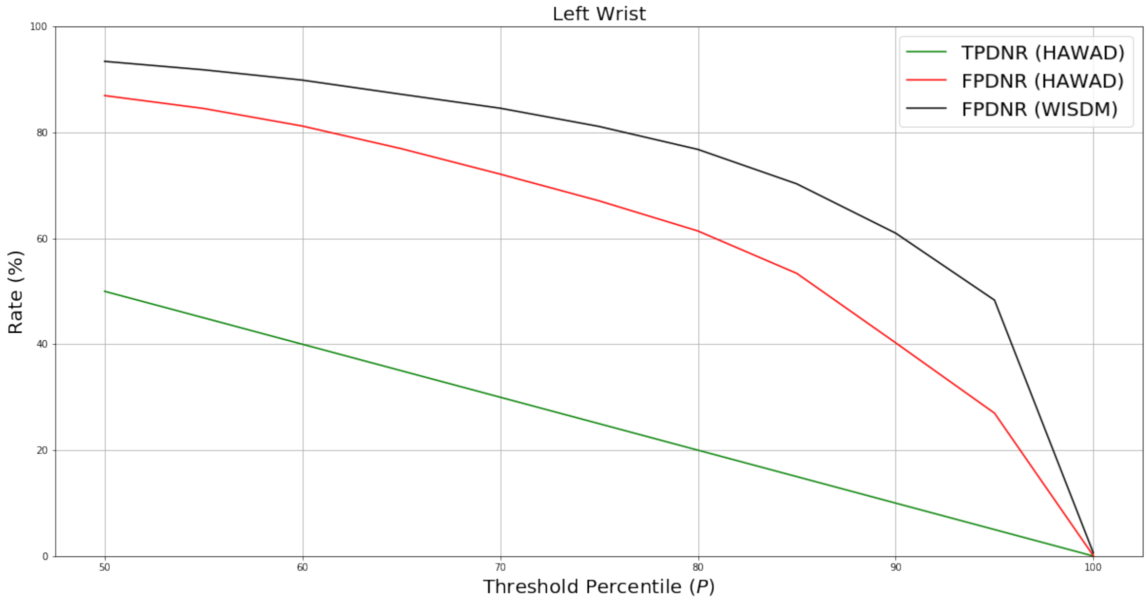
Figure 4.9 shows the precision, recall and F1-score for the left and right wrist, respectively. These metrics are defined as:

$$Precision = \frac{True\ Positive}{True\ Positive\ + False\ Positive} \tag{4.6}$$

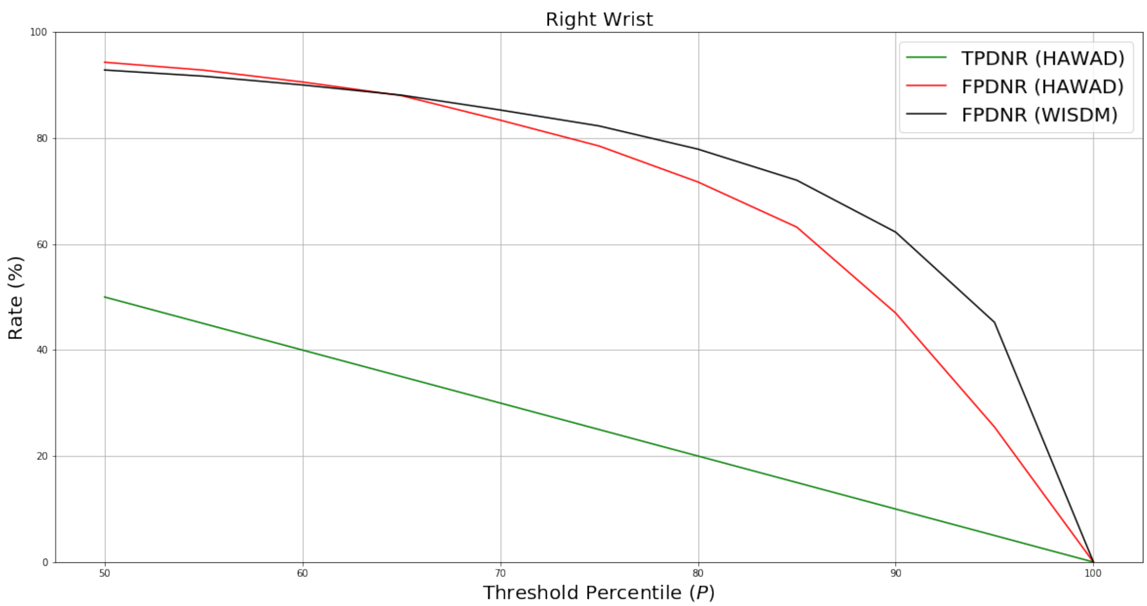
$$Recall = \frac{True\ Positive}{True\ Positive +\ False\ Negative} \tag{4.7}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision\ + Recall} \tag{4.8}$$

As expected, when the percentile is increased the precision decreases and the recall increases, and vice versa. This is because when percentile is increased, there are more true positives that increase the recall, but the number of false positives also increases, that results in reduction of the precision. The F1-score is the harmonic mean of precision and recall and widely used a metric to balance between them. We see that our solution gives the best F1-score around 80 percentile threshold. The F1-scores at this percentile are 0.72 and 0.74 for the left and the right wrist, respectively. The F1-scores of the baseline method, the pre-trained network without using out-of-distribution as proposed by Galluzzi et al.[77], are 0.55 and 0.57, respectively. So the F1-score is improved by 0.17, about 30% more compared to the baseline. At this



(a)



(b)

Figure 4.8: TPDNR (HAWAD), FPDNR (HAWAD), FPDNR(WISDM) for different percentiles for (a) left wrist, (b) right wrist.

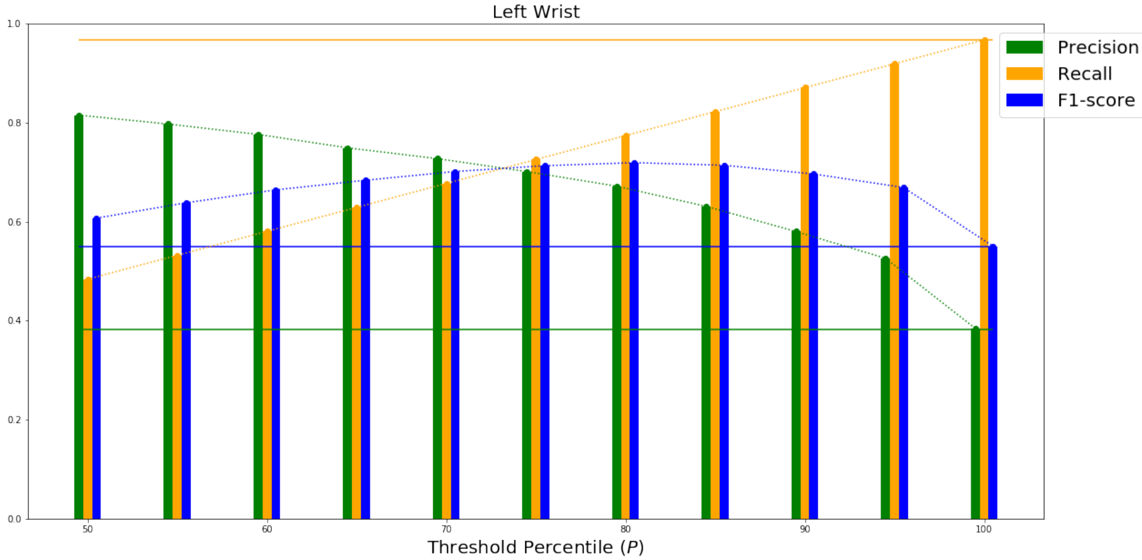
percentile, the false positive rate from the WISDM dataset is reduced by 76.8% and 77% for the left and right wrist, respectively, and the reductions for our dataset are 61.4% and 71.7%, respectively.

4.4 Discussion

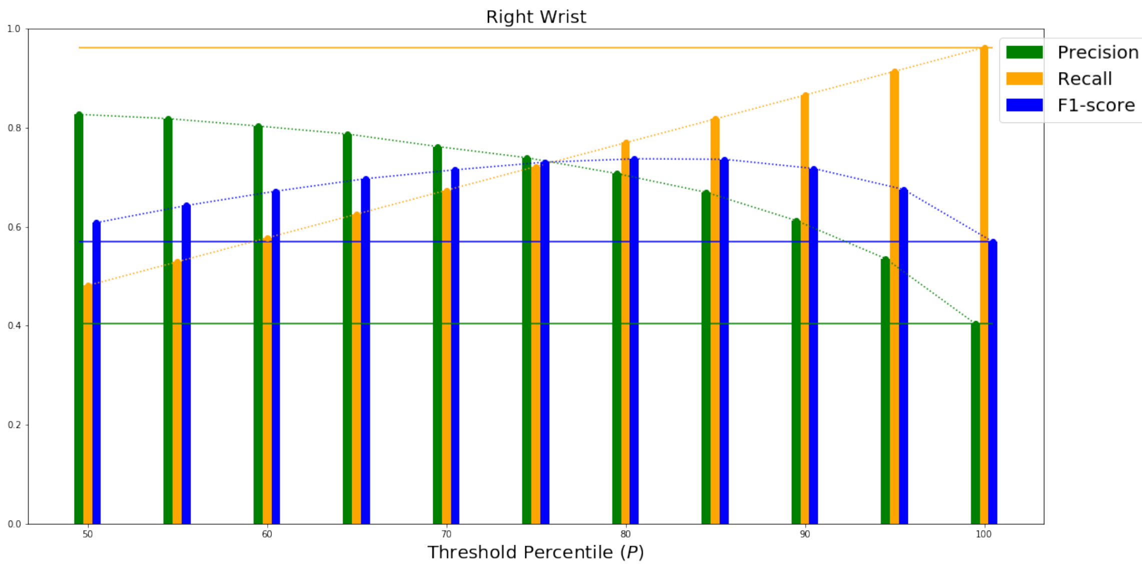
Our method is not an alternative solution to state-of-the-art neural network based methods; rather it works on top of a pre-trained neural network. So, it can be used with the existing solutions to detect hand-washing with more robustness and accuracy. We have used a feedforward network for evaluation, but our method is not network specific. We use the output of the penultimate layer only, and so it can be used for other types of neural networks like Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Though the method presented here has been evaluated for hand washing, it is generic in nature, and so it can be used for other types of activities. In the future, we will explore its effectiveness for different other activities with different types of neural networks.

As our method uses the output of a neural network model, the additional computational cost for inference is very low. We just need to compute the Mahanabolis distance and compare the distance with the pre-defined threshold. If there are N nodes in the penultimate layer, the size of the covariance matrix is $N \times N$. The larger N is, the more computation would be required. However, such computation cost is very low compared to the neural networks that require such magnitude of computation or more in each layer. Since we use a pre-trained network, there is no need to train a network for our solution. We trained a network for evaluation purpose as there is no pre-trained network available for hand-washing detection from accelerometer data.

Compared to other solutions that require a lot of parameter tuning, there is only



(a)



(b)

Figure 4.9: Precision, Recall and F1-score for different percentiles for (a) left wrist, (b) right wrist. The horizontal lines represents the metrics of the baseline.

one parameter in our method, the distance threshold (percentile) that needs to be tuned. The distance threshold should be set to balance the trade-off between different metrics like precision and recall, as required by the application where this solution will be used.

Chapter 5

Movement Visualization

Almost all the solutions for activity recognition are data-driven. So, understanding the data and their characteristics is fundamental toward developing effective solutions for activity recognition. Visualization is perhaps the most effective approach for getting insight into data as well as communicating the insights with others. A novel visualization method that would provide additional utility to the existing methods is of utmost desire. However, such novel methods for visualization are rarely invented, particularly in the area of wearable and mobile sensing. Here, we present a novel method for visualizing movement and orientation using inertial sensors. Our visualization method is not a replacement to the existing methods; rather it provides additional utility toward better understanding the movements associated with different activities.

Inertial sensors provide data for three orthogonal directions, namely along X, Y, and Z-axes. Accelerometers, gyroscopes, and magnetometers sense acceleration, rotation rate, and direction of the device, respectively. The data from these sensors are time-series in nature and usually visualized as a series of data points, often connected by straight line segments. These plots are useful in understanding the data. Our method complements the existing approaches and provides ways to visualize the data from different perspectives. The method is based on the orientation of the device where the time series orientation data are organized in 3-dimensional space, resulting in spatio-temporal representations. We extract the orientations of the device from quaternions that are constructed by fusing data from an accelerometer, a gyroscope, and a magnetometer. Our novel method consists of creating four visualization-based primitives: Orientation trace, Spatio-temporal representation, Orientation reachability, and Sphere segmentation. These primitives are novel for visualizing movements using inertial sensors. A quaternion is decomposed into three unit vectors representing the orientation of the device with respect to earth gravity, north, and east. These vectors are placed on the surface of a unit sphere to visualize orientation trace in the spatio-temporal dimension. To better visualize where the orientation reaches for some activities, the sphere is segmented into some nearly uniform cells.

The method presented here are not activity or gesture specific. It can be used to visualize movement and orientation for different activities. We presented examples of several activities of daily living to demonstrate the use of our method in understanding the underlying movement associated with the activities. Our visualization method helps to analyze and understand data that can further help in developing effective and efficient solutions. We analyzed a dataset related to smoking activity recognition and provided several insights through representing the data using our method. We developed an efficient solution for smoking puff detection using the concept of orientation reachability. Finally, we illustrated how our method to monitor movement patterns related to several rehabilitation exercises.

5.1 Quaternion Basics

As mentioned earlier, the proposed method for movement visualization is based on the orientation of the wearable devices that is expressed using quaternions. A quaternion

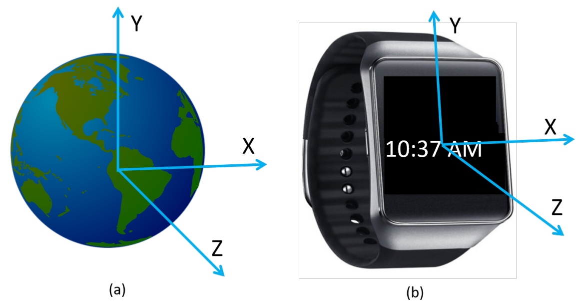


Figure 5.1: Axes of the coordinate systems for (a) the world, and (b) the device.

is a representation of the orientation of a coordinate system with respect to another coordinate system. Quaternions also provide convenient ways to manipulate the rotations of the coordinate frames. In the context of smart devices like smart phones and smart watches, a quaternion represents the orientation of the device with respect to the world coordinate system. The device coordinate system may differ based on how sensors are placed inside the device. For the Android powered devices, it is usually defined as follows: the X axis points to the right of the device, the Y axis points up, and the Z axis points towards the outside of the display, as shown in Figure 5.1(b). For the world coordinate system, the Z axis is perpendicular to the ground pointing opposite to the earth's gravity, the Y axis is tangential to the ground at the device's current location and points towards the magnetic north, and the X axis, defined as the vector product of Y and Z, is tangential to the ground at the device's current location, and roughly points to the East. The world coordinate system is shown in Figure 5.1(a).

A quaternion q is formally defined by a scalar component (q_s) and a 3D vector (q_x, q_y, q_z) as:

$$q = q_s + q_x\hat{i} + +q_y\hat{j} + q_z\hat{k} \tag{5.1}$$

where i, j, and k are the imaginary basis elements. The quaternion is called a unit quaternion if its magnitude equals to one, i.e. if $|q| = \sqrt{q_s^2 + q_x^2 + q_y^2 + q_z^2} = 1$

Both orientations and rotations can be represented with the unit quaternions. The orientation of an object with respect to a reference coordinate system can be expressed as a single rotation α, called the rotation angle, around a unit vector $(x\hat{i} + y\hat{j} + z\hat{k})$ in the reference system, called the axis of rotation. With the rotation axis and the rotation angle, a unit quaternion q can be expressed as:

$$q = cos\frac{\alpha}{2} + xsin\frac{\alpha}{2}\hat{i} + ysin\frac{\alpha}{2}\hat{j} + zsin\frac{\alpha}{2}\hat{k} \tag{5.2}$$

where $q_s = cos\frac{\alpha}{2}$, $q_x = xsin\frac{\alpha}{2}$, $q_y = ysin\frac{\alpha}{2}$, $q_z = zsin\frac{\alpha}{2}$

As mentioned before, quaternions can also be used to represent rotations. If current orientation of a device is q_1, and the device is rotated using a quaternion q_r, that means it is rotated with the rotation angle α around the unit rotation axis $(x\hat{i}+y\hat{j}+z\hat{k})$ represented by q_r, then the new orientation of the device q_2 is formulated as:

$$q_2 = q_r q_1 q_r^{-1} \tag{5.3}$$

where q_r^{-1}, the so-called conjugate quaternion of q_r, is expressed as:

$$q_r^{-1} = cos\frac{\alpha}{2} - xsin\frac{\alpha}{2}\hat{i} - ysin\frac{\alpha}{2}\hat{j} - zsin\frac{\alpha}{2}\hat{k} \tag{5.4}$$

q_1 and q_2 are often referred as orientation quaternion as these quaternions represent the orientations of the object, and q_r as rotation quaternion as it is used to rotate the object.

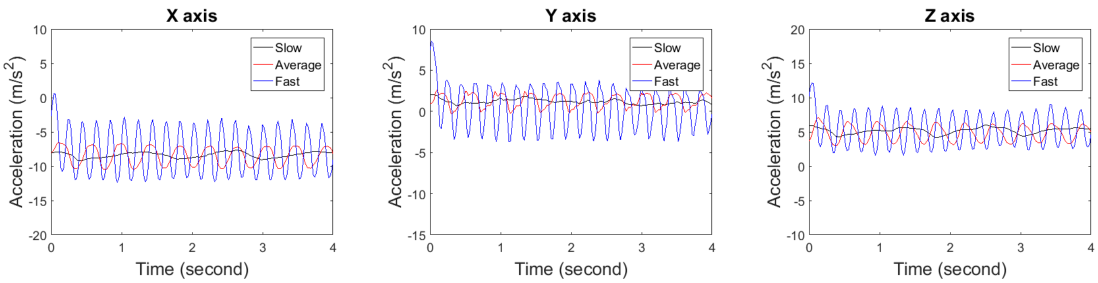
Usually the quaternions are computed through fusing the outputs of multiple sensors such as an accelerometer, a gyroscope, and a magnetometer. More details about it can be found in [152]. Many of the smart phones and smart watches available today like most Android powered phones and watches provide the unit quaternions through APIs [153].

5.2 Visualization Method

We use a quaternion to get the orientation of the device. The quaternion values depend on the orientation of the device with respect to both the direction of earth gravity and the direction of earth magnetic field (e.g. north). As a result, the quaternion values for a gesture are different from the quaternion values of another same gesture when the two gestures are performed facing in different magnetic directions. Consequently, it is difficult to interpret the movement and orientation of a device using quaternions. We decompose the orientation with respect to the earth gravity and magnetic directions using rotation matrix, an equivalent representation of the quaternion. The rotation matrix R, equivalent to the quaternion q, is defined as:

$$\begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_s q_z) & 2(q_x q_z + q_s q_y) \\ 2(q_s q_z + q_x q_2) & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_x q_z + q_s q_y) \\ 2(q_s q_z - q_s q_2) & 2(q_y q_z + q_s q_x) & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix}$$

The rotation matrix is orthogonal, and so the rows and columns of R are orthonormal i.e. orthogonal unit vectors.The first, second and third row of the matrix represent orientation of the device with respect to east (x-axis), north (y-axis) and gravity (z-axis) of the world, respectively. The first, second and third element of a unit vector represent the projection of that vector along the the X, Y and Z axes of the device, respectively. It should be noted that multiplying the unit gravity vector (the third row) by the magnitude of gravitational acceleration (g) results the actual gravitational acceleration along the different axes of the device. Though the value of g is constant at a location, it is somewhat different at different locations on the



(a) (b) (c)

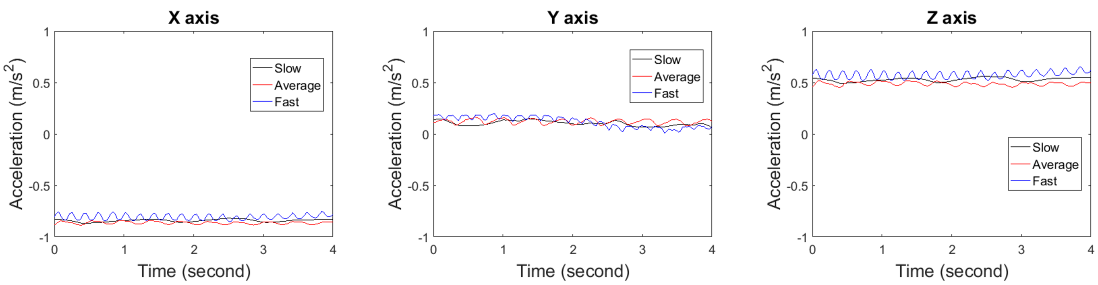
Figure 5.2: Acceleration values from a wrist worn accelerometer along (a) X-axis (b) Y-axis (c) Z-axis for brushing teeth in a specific way at different speeds (slow, average and fast).

earth. In contrast, the magnitude of the unit vector is fixed (i.e. 1) irrespective of the location on the earth.

The values of accelerations (from an accelerometer) and rotation rates (from a gyroscope) depend on the speed or intensity of the movement of the device. In contrast, unit vectors change only when the orientation of the device changes. For example, the accelerations and the unit gravity vectors from a wrist device for brushing teeth at different speed are shown in Figure 5.2 and 5.3, respectively. The gravity vectors do not change significantly as the there is no significant change of the orientation of the wrist. On the other hand, the speed or intensity of movement changes the accelerations along the different axes significantly. So, the unit vectors provide complementary insight about the movements associated with of different activities. In this section, we present a set of novel primitives for movement visualization using the unit vectors.

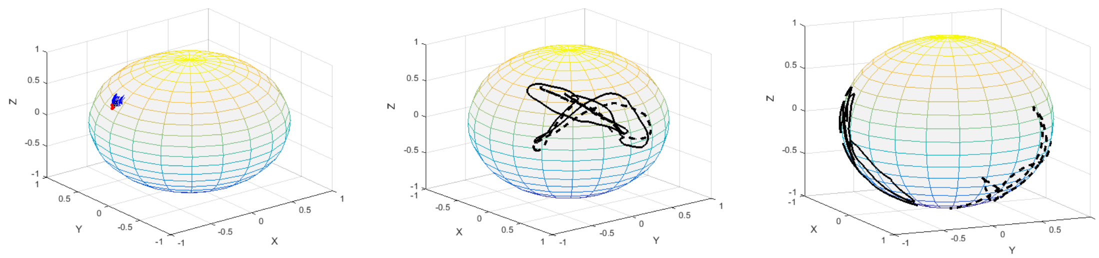
5.2.1 Orientation Trace

Since the vectors are unit in length, they can be placed on the surface of a unit sphere. The trace of the orientations on the sphere helps to understand the movement and orientation of the device, and thus of the body part where the device is placed. Figure



(a) (b) (c)

Figure 5.3: Unit gravity vector values along (a) X-axis (b) Y-axis (c) Z-axis for the tooth brushing activity corresponding to accelerometer data from Figure 5.2



(a) (b) (c)

Figure 5.4: (a) Traces of the unit gravity vectors (orientation with respect to gravity) for the tooth brushing activity from Figure 5.2, (b) Traces of the unit gravity vectors from a wrist device for opening two different doors that are similar but facing to two different directions, and (c) Traces of the unit north vectors for opening the doors.

5.4(a) shows the gravity vectors of Figure 5.3 on the surface of a unit sphere. From the trace, we see that the orientations of the wrist are nearly the same even though the intensity of the wrist movement is different. The orientation traces on the sphere are limited to some specific area irrespective of the time duration of the data. The traces provide complementary insight about the orientation and movement. For example, we need to look into different axes of the 2-D plots (e.g., Figure 5.3) to understand the orientations of the wrist. On the other hand, Figure 5.4(a) presents the orientation traces compactly, and illustrates their similarity.

The magnetic direction is irrelevant for most of the human activities. For example, a person can eat facing in any direction, and so the direction might not be useful in understanding and recognizing eating pattern. However, direction provides additional information in several contexts. For example, Figure 5.4(b) and Figure 5.4(c) show the unit gravity vectors and the unit north vectors from a wrist device, respectively, for opening two different doors that are similar, but facing two different directions. Here the gestures include moving the hand to the door lock, opening the lock, pulling the door, going out through the opened door, and moving the hand down. As seen in the figures, the doors can be easily distinguished by the location of the unit north vectors on the sphere. On the other hand, the unit gravity vectors are not suitable in distinguishing the doors. Direction often provides important information regarding the context of a user, which is useful in recognizing different activities. Though a magnetometer provides direction information, its values are significantly distorted by magnetic materials (e.g. metal) and magnetic fields [11]. In contrast, quaternions are more stable as they are generated through fusion of multiple sensors.

5.2.2 Spatio-temporal Representation

The spatio-temporal representation, as defined by the location of the unit vectors on the sphere and by the order of the unit vectors over time, is robust against the speed of a gesture. It is because the orientation trace of the device is nearly the same for similar gestures even though the speeds of performing the gestures are different. Due to difference in speed, the duration of an activity or gesture differs from time to time. Figure 5.5 shows the acceleration and unit gravity data for two apple bites, one bite is slower and the other is faster. Here, bite means moving the apple to the mouth, biting the apple, and then moving the hand down. As illustrated in the figure, the gravity data has less variability than acceleration, but both data shows similar problems in terms of duration of the activity. For example, Figure 5.5(c) shows the sequences of the unit vectors on the sphere for the two apple bites corresponding to the Figure

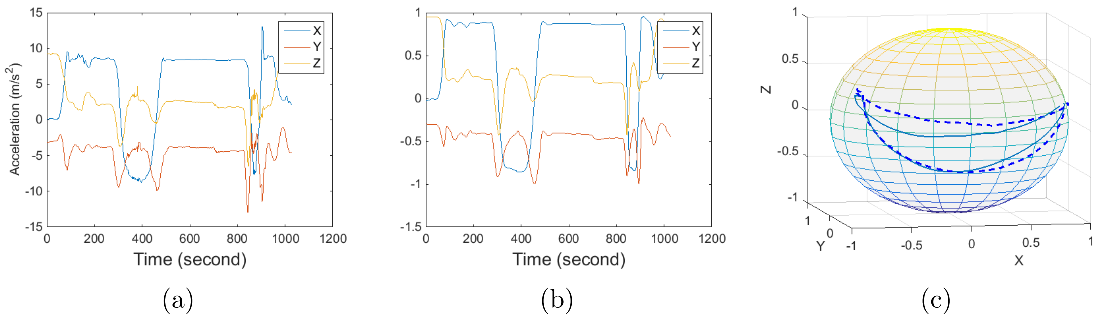


Figure 5.5: For two apple bites, one slower and another faster, (a) acceleration data in the time dimension, (b) unit gravity vectors in the time dimension, (c) unit gravity vectors in the spatio-temporal dimension

5.5(a) and 5.5(b). Here, the orientation traces are much more similar both in shape and length compared to the time series representation of 5.5(b). This representation provides more insight into the movement patterns for different activities.

5.2.3 Orientation Reachability

We define orientation reachability for an activity as the area on the sphere where the orientation trace for the activity reaches. The size and shape of the area depends on the activity. For example, Figure 5.4(a) shows the orientation reachability corresponding to the brushing teeth gestures mentioned before. Here, the trace is limited to a small area due to the fact that wrist orientation doesn't change significantly during brushing teeth in a specific way. On the other hand, Figure 5.5(c) shows the potential area on the sphere for apple bites which is quite wide. Nonetheless, the area of an activity is much less than the total area for all possible activities. The traces for these bites are longer because they were taken while standing. The hand was moved from a downward position up to the mouth, and then again to the downward position.

5.2.4 Sphere Segmentation

The orientation traces can be at any places of the sphere. For visualisation, we need to rotate the sphere to bring the trace to the front. Though we get insight into the shape of the orientation trace, it is not easy to understand location of the orientation trace on the sphere because the sphere looks uniform. For example, we have rotated the sphere of Figure 5.4(a) to bring the traces to the front. Here, the X and Y axes position are different from those of Figure 5.4(b). It is difficult to understand exactly where the traces are on the sphere and how closer or further they are. To better understand the location of the traces on the sphere, we divide the sphere into some nearly uniform cells and assign an unique number for each of the cells.

The cells on the unit sphere are generated using a regular icosahedron, a polyhedron with 20 equilateral triangular faces and 12 vertices, as shown in Figure 5.6(a). The vertices of an icosahedron lie on the surface of a sphere. The cells on the sphere are defined by the Voronoi diagram generated from the vertices of the icosahedron. Figure 5.6(b) shows the Voronoi diagram generated from the vertices of the regular icosahedron. The cells generated from the regular icosahedron are big. More fine grained cells are generated by dividing the icosahedron repeatedly. A icosahedron is divided by dividing each triangular face into four smaller triangular faces by connecting the mid-points of the edges of the face. The mid-points are then normalized so that they lie on the surface of the sphere of the icosahedron. For example, Figure 5.6(a) shows the polyhedron generated by dividing the regular icosahedron once, and Figure 5.6(b) shows the corresponding Voronoi cells. Figure 5.6 shows the cells generated by dividing the regular icosahedron 2 and 3, respectively. Icosahedron based division of sphere surface is widely used in areas like meteorology [86].

Rotating the sphere moves the cells from one place to another, and due to similarity it is very difficult to identify the individual cells. We assign unique numbers

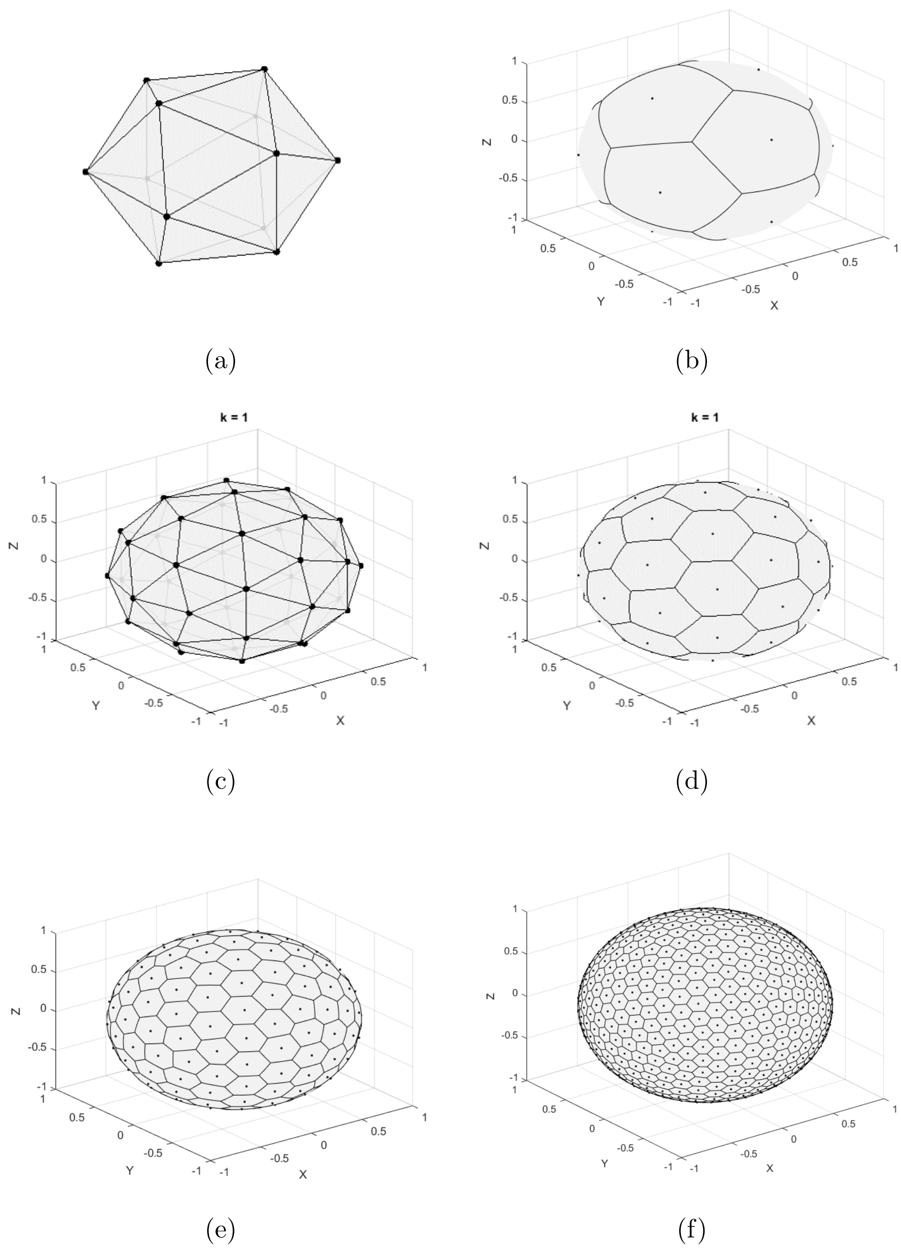


Figure 5.6: (a) A regular icosahedron (b) The Voronoi cells generated from the regular icosahedron, (c) Polyhedron generated from dividing the regular icosahedron once, (d) Voronoi cells from the vertices of the polyhedron (e)-(f) Dividing the unit sphere into cells with more granularity.

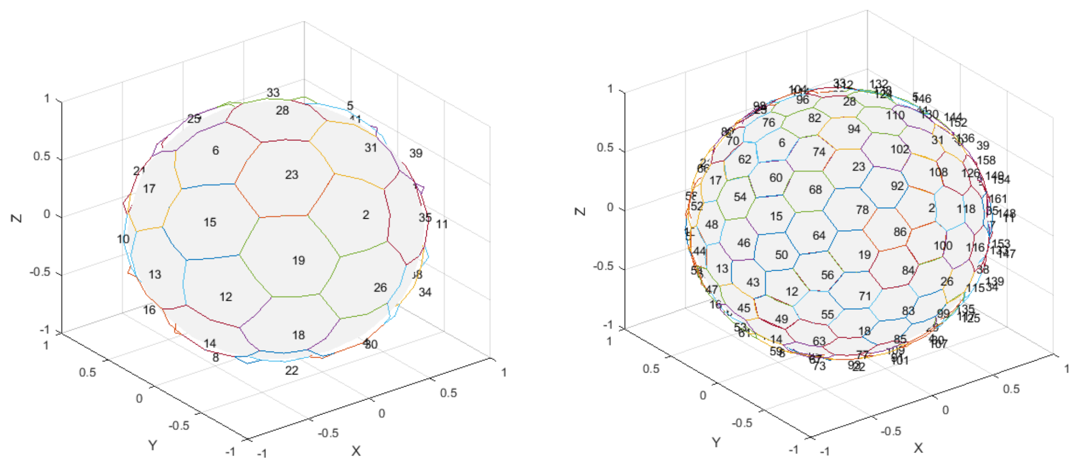
Table 5.1: Number of cells generated by dividing the regular icosahedron k times

k	Number of vertices
0	12
1	42
2	162
3	642
4	2562
5	10242

to each of the cells to identify the cells, and thus some specific areas on the sphere. The number of cells on the sphere depends on the size of the cells i.e., on how many times the regular icosahedron is divided to generate the cells. The number of cells generated from the regular icosahedron is 12, and there are $10 \times 4^k + 2$ cells when the cells are generated by dividing the regular icosahedron k times. The number of cells for different values of k are listed in Table 5.1. For identification, each of the cells is provided with a unique integer index in the range of [1, total number of cells]. When new cells are generated, the center of the old cells remain same, only size of the cells are reduced. So, the cells that have the old cell centers retain the same indices of the old cells. This ensures consistency of the cell locations with different granularity. Figure 5.7 shows assignment of the numbers to the cells for $k = 1$ and $k = 2$. Such assignment of unique numbers to the cells helps to identify a cell or location easily even if the sphere is rotated.

5.3 Smoking Activity Recognition

In this section, we use our visualization methods to analyze the data from a public dataset [105]. We used orientation reachability to develop a very efficient solution for detecting smoking puffs. The dataset contains about 45 hours of data from 11 participants for smoking, drinking, eating, sitting, standing, and walking. The participants smoked while sitting, standing, walking, and group conversation. The dataset



(a) (b)

Figure 5.7: Unique numbers of the cells for (a) k=1, (b) k=2

is weakly annotated where annotations are available for each type of activity from a participant instead of individual actions like puffs or sips. It comes with data for total acceleration, linear acceleration and rotation rate along the X, Y, and Z axis from a smartwatch worn on the right wrist and a smartphone. We derive the gravity values by subtracting linear acceleration from total acceleration along each of the axes. Then we scale the gravity values to unit vectors using L2 normalization. Here, we have used data from the wrist device only.

5.3.1 Data Analysis

The gestures for an activity differ from person to person, and even from time to time for the same person. Also, there are confounding gestures present among different activities. Figure 5.8 shows data from a participant for one minute duration for different activities. The movement patterns for smoking during standing (Figure 5.8 (c) and sitting (Figure 5.8 (f) are quite different. On the other hand, the movement patterns for drinking and smoking are similar to some extent. If we look into orien-

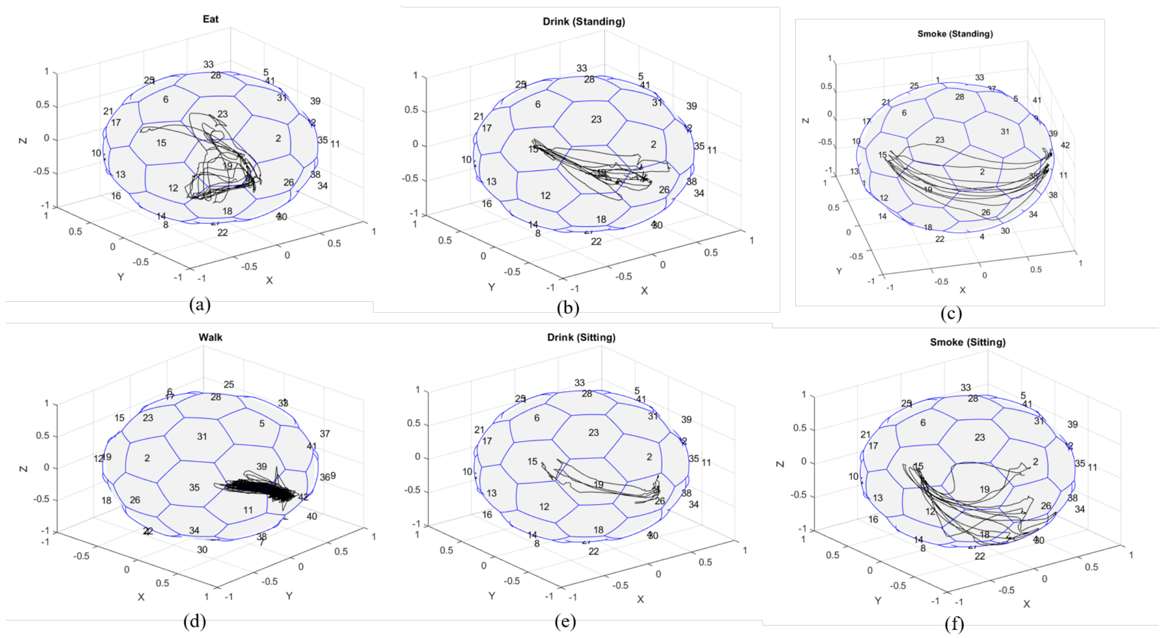


Figure 5.8: Orientation traces of wrist for different activities (a) Eating, (b) Walking, (c) Drinking while standing, (d) Drinking while sitting, (e) Smoking while standing, (f) Smoking while sitting

tation reachabilty, we see that the orientations of the wrist go in or near cell 15 when the subject took a puff or sip. In contrast, the orientation for eating seldom reaches to that cell. Here, we plotted data for just one minute for clarity, but orientation trace of longer duration of data provides additional insights. Figure 5.9 shows 10 minutes long data for the smoking (sitting), drinking (sitting) and eating using our method and traditional 2-D and 3-D plots. We see that there is significant diversity in the smoking gestures, even from the same person. It is difficult to understand this phenomenon from the 2-D plots. Though the 3-D gives some insights about this phenomenon, our method provides better insights by showing the orientation traces on a sphere with cell numbers.

Different persons might perform the same activity differently, and so it is challenging to develop robust solutions for activity recognition that works for a wide range of users. Figure 5.10 shows orientation traces for smoking activity while standing

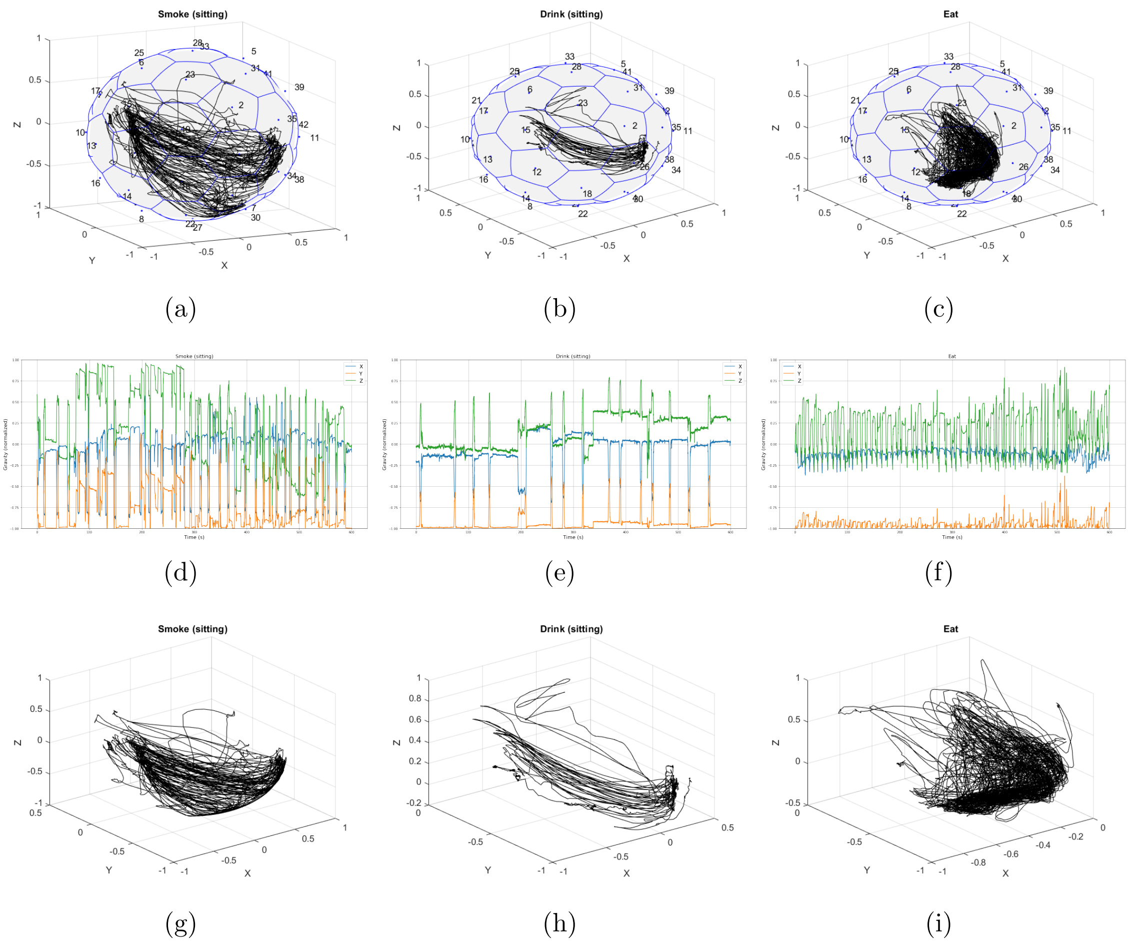
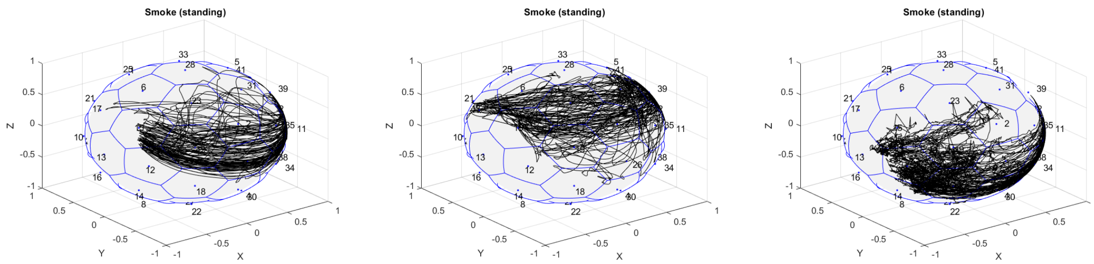


Figure 5.9: Plot of the gravity vectors for 10 minute long data using (a, b, c) our method, (d, e, f) traditional 2-D plot, (g h i) traditional 3-D for Smoke (sitting), Drink (sitting) and (c) Eat.



(a) (b) (c)

Figure 5.10: Smoking gestures from three different participants. These figures show that orientation of the wrist for smoking can differ significantly from person to person.

from three different participants. The orientation traces for the puffs are significantly different from person to person. It justifies why personalized models generally work better than person-independent models. Also, visualization helps us to understand why activity recognition is a very challenging task. Most of the works justify such facts using results (e.g., accuracy, f1-score) from classification algorithms where the results depend on the method used. Our method helps to better understand and communicate such insights through visualization of movement and orientation.

5.3.2 Solution

Similar to eating and drinking, when the right hand is moved to mouth for smoking, gravitational accelerations along the X axis go down and then go up. Figure 5.11 shows the gravity along the X axis for some puffs. We cut the X acceleration values using a threshold, and the segments that lie below the threshold contains the puffs. Such segments are also generated from other gestures like eating and drinking. The duration of a puff usually ranges between 1 to 2.4 seconds [154]. So, we only use the segments that are at least one second long and discard the others. A one-second long segment can capture puffs with duration less than one second because the puff duration is less than the corresponding segment duration.

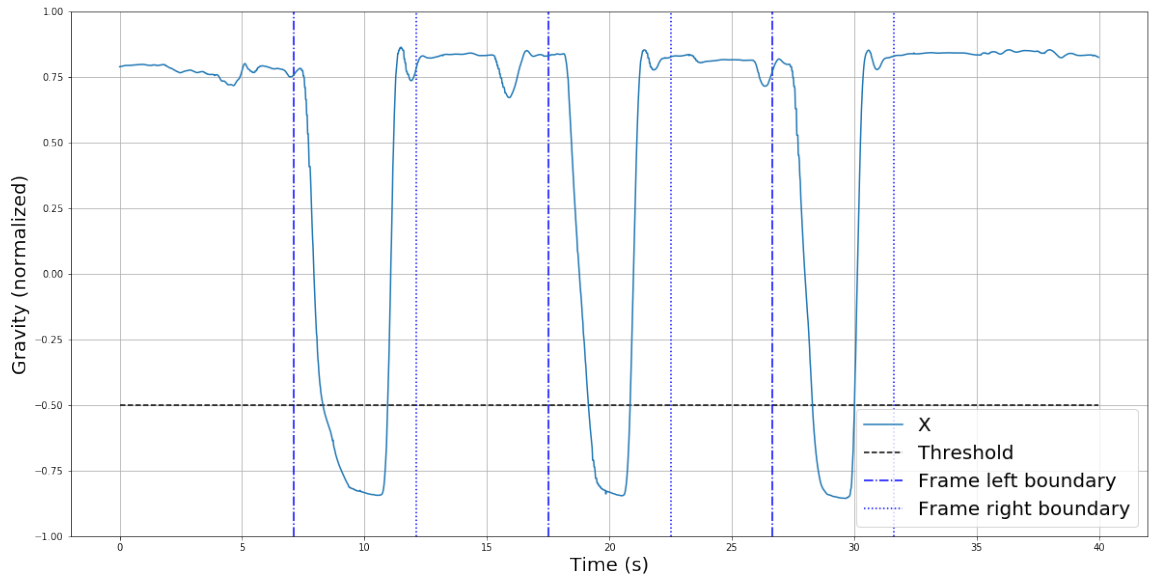
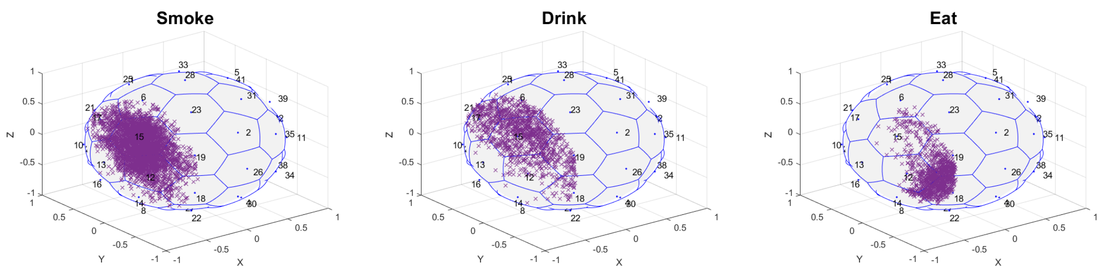


Figure 5.11: Detecting potential puffs by a threshold.

The midpoints of the segments for smoking, drinking or eating approximately represent the orientation of the wrist when the hand is moved to the mouth. Figure 5.12 shows the gravity vectors (X, Y and Z) at the midpoints of the segments for smoking, drinking and eating. We see that when the hand is moved to the mouth, the orientation of the wrist for smoking is significantly more similar to drinking than to eating. Also, we see that the orientation of the wrist is confined to a limited area for these activities. We use the orientation-reachability property of the activities to further filter-out non-smoking segments resulting reduction in overall computation. We leverage the distribution of the midpoints of the smoking segments to filter out the frames that are likely to be from non-smoking activities. The frames that remain after this filtering step are further classified using a more sophisticated classifier like a neural network. The approach used here is similar to that used for hand washing detection in Chapter 4. However, here we use the gravity vectors of the midpoints instead of the output of the penultimate layer and filter out the instances before sending it to a classifier.



(a) (b) (c)

Figure 5.12: Mid points of the potential puff segments for (a) Smoking, (b) Drinking and (c) Eating.

The pre-classification filter-out process is described in Figure 5.13. We use midpoints of smoking segments to calculate the parameters ($\hat{\mu}$, $\hat{\Sigma}$, and d_{th}), and use the parameters to find the distance of a test segment. If the distance is greater than the distance threshold, we consider it as non-smoking and discard it. Otherwise, we take a 5-second long frame around the midpoint and classify the frames using a classifier to determine whether it is a puff or not. As the usual duration of a puff is 1-2.4 seconds [154], the 5-second long frame captures a puff as well as corresponding hand movements. Our solution is not dependent on the classifier, and so we can use any effective classifier to classify the frames. We have used a convolutional neural network similar to that used for eating gesture detection in Chapter 3. It should be noted that only the X values of the the gravity vectors are used to detect the potential segments. Data from all the axes (X, Y, and Z) of the gravity vectors are used for classification.

5.3.3 Result Analysis

The total duration of the data and the frames are about 45 hours and 9 hours, respectively. So, using the threshold on the gravitational acceleration along X axis, we can discard about 80% of the data in contrast to state-of-the-art methods that

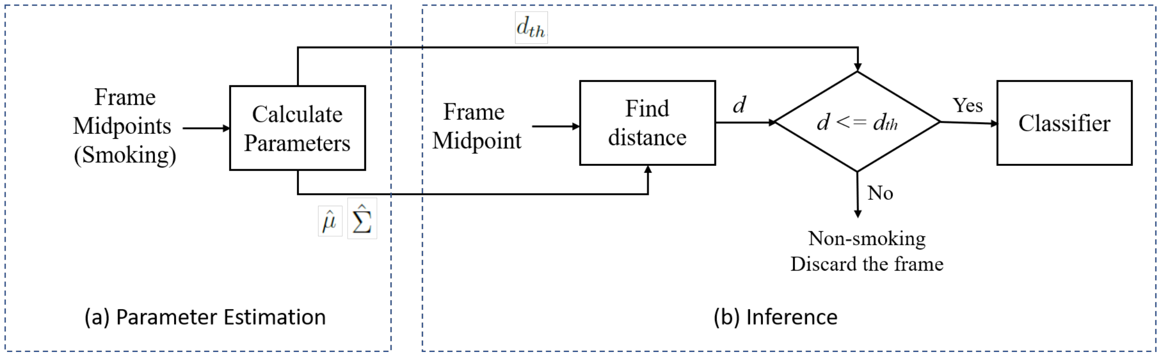


Figure 5.13: The process of filtering out frames before classification.

use computationally expensive classifiers to process all the data [105, 155]. Most of the segments come from smoking, drinking, and eating, with very few from other activities. The duration of smoking, drinking, and eating sessions in the dataset is about 72% of the total data. Usually, the ratio should be less in the real world, and so more portion of the data could be discarded.

For classification, we use Leave One Person Out (LOPO) approach. In this approach, data from one participant is tested by the model developed using the data from other participants. This process is repeated for all the participants, and results are calculated by combining the frames from all. We use the percentile of the distances of the smoking frames to determine the distance threshold. Figure 5.14 shows the rate of frames discarded at different percentiles. We see that more frames are discarded when the percentile is smaller and vice versa. For example, at the 90 percentile we discard about 22.8% of the non-smoking frames with the expense of only 5.2% of the smoking frames. It should be noted that 10 percent of the training frames are discarded at the 90 percentile threshold, but the discard rate of test frames is lower. It is because the distance threshold calculated from 10 persons (training) covers more puffs from the left-out person.

Discarding data using the distance threshold before using a computationally expensive classifier reduces overall computational overhead, but it has implications for

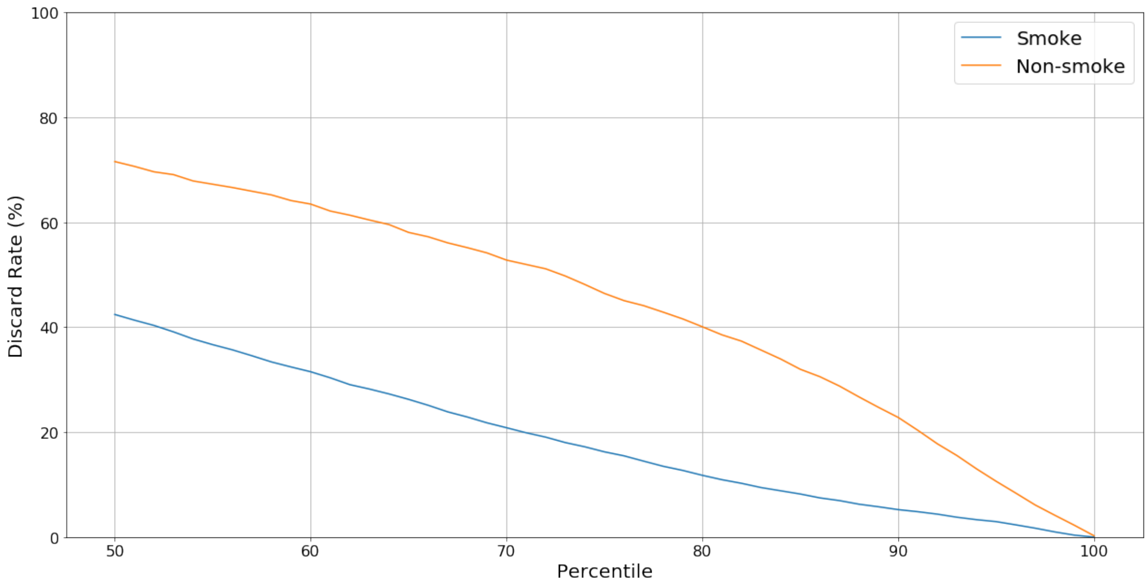


Figure 5.14: The rate of non-smoking frames discarded at different percentiles

other performance metrics. Figure 5.15 shows the precision, recall, and F1-score at different percentiles. We see that as the percentile is decreased the recall decreases significantly, particularly for the lower percentiles. However, there is no significant gain in precision. Consequently, the F1-score reduces. However, the reduction in F1-score for higher percentiles is very low. For example, at 90 percentile threshold, the recall and the f1-score are reduced by 0.04 and 0.015 only. We can use our method to balance the trade-off between resource availability and accuracy metrics. Our solution can be run on resource constraint devices without significant loss in accuracy.

5.4 Monitoring Rehabilitation Exercises

In this section, we demonstrate how our techniques can be used to monitor exercise performance for rehabilitation. Rehabilitation programs, particularly after a surgery or an injury, generally involve a set of physical exercises that helps the patients recover. The performance of the patients in performing the exercises is a good indicator of their health conditions. So, monitoring the exercise patterns of the patients

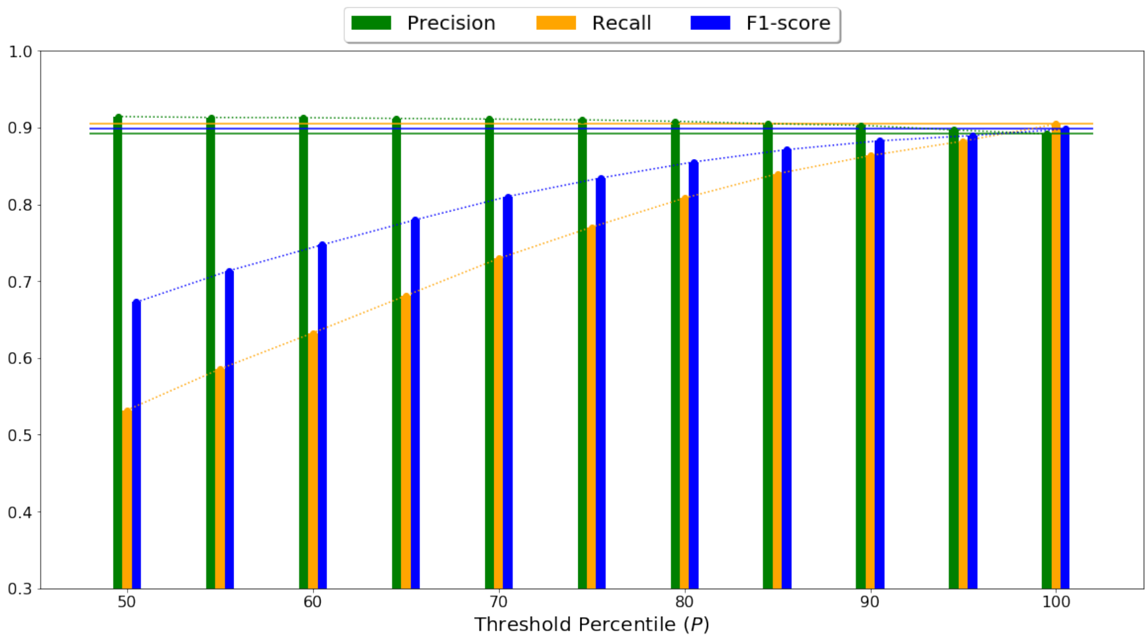
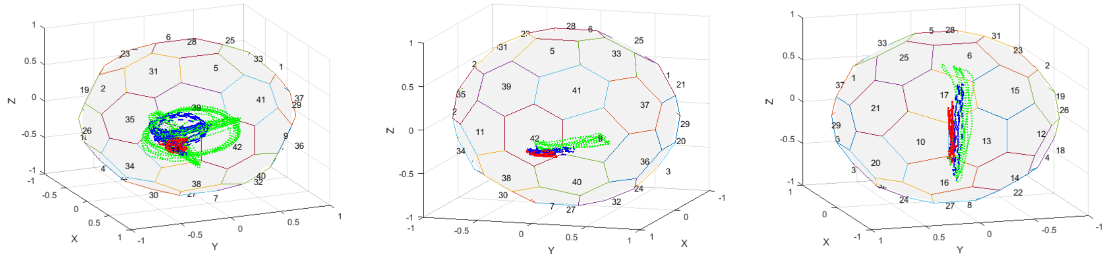


Figure 5.15: Precision, Recall and F1-score at different percentiles. The horizontal lines (dahsed) represent the metrics when no segment is discarded using the distance threshold.

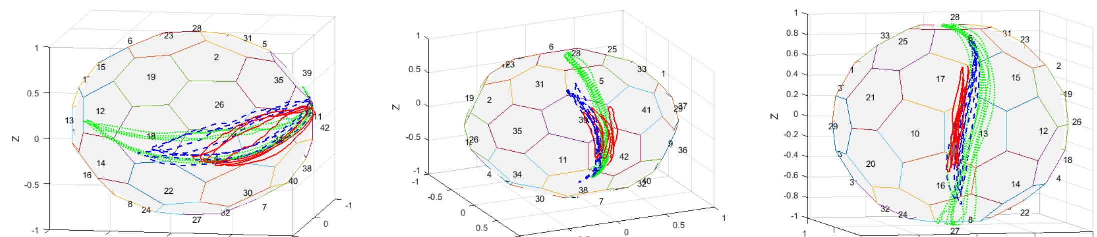
would help the caregivers and physicians in better understanding the patients' condition. We have used six exercises, namely *Pendulum*, *Passive Internal Rotation*, *Sleeper Stretch*, *Elbow Flexation*, *Bent−over Horizontal Abduction*, and *Internal−External Rotation*, that are often used for rehabilitation [156]. The first three exercises are Stretching Exercises, and the last three are Strengthening Exercises which are shown in Figure 5.16 and Figure 5.17, respectively.

For the purpose of this demonstration, the performance metric has been categorized into three categories: poor, average, and excellent. Poor performance for an exercise means very little hand movements during the exercise compared to the desired movements that is excellent. For this experiment, a participant performs all the exercises with poor, average and excellent performance following the guidelines in [156]. For each of the exercises, the participant first performs the exercise with little movement that corresponds to poor performance. Then the exercise is performed



(a) (b) (c)

Figure 5.16: Plots for Stretching Exercises (a) Pendulum, (b) Passive Internal Rotation, (c) Sleeper Stretch.



(a) (b) (c)

Figure 5.17: Plots for Strength Exercises (a) Elbow Flexion, (b) Bent-Over Horizontal Abduction, (c) Internal and External Rotation.

with more movement, and finally with desired movement that map to average and excellent performance, respectively. The participant is a graduate student, and is not a real patient. It should be noted that the purpose of here is not to monitor real patients, but to show how our methods can be used for monitoring exercises.

Figures 5.16 and 5.17 show the orientation traces from the above mentioned exercises. The red, green and blue traces represent data for poor, average and excellent performances, respectively. The figures show that the orientation traces provides useful insights about the the quality of the exercises. The red traces have much less orientation reachability compared to that of the green traces. Smaller cell sizes might be used for fine-grained assessment of the quality of the exercises. Also the cell numbering is useful for better understanding and visualization of the orientation reachabilty. For example, it might seem that the gestures of Figure 5.17(a) and 5.17(c) have overlapping reachability. But if we look into the cell numbers for these two exercises, they are different. This is due to the fact that the spheres have been rotated to bring the traces to the front. Thus, cell numbering helps in identifying the cells or location of the traces on the sphere easily.

5.5 Discussion

Visualization helps to understand data more effectively and easily. Though the scope and focus of this work is visualization, the methods presented here can help in developing better solutions for activity recognition. We presented an efficient solution for smoking puff detection using orientation reachability. It should be noted that developing a classifier is not a focus of the solution. Our solution filters out most of the non-smoking data using a threshold, and it can be used in combination with any classifier. This approach can be used to develop efficient and effective solutions for other activity recognition tasks. Filtering out most of the NULL activities not only reduces computation overhead but also it can improve accuracy by reducing class imbalance. In the future, we will explore the use of the technique for other activity recognition tasks as well as the impact on overall classification results.

The quality and quantity of data are crucial for most of the state-of-the-art solutions that are data-driven. However, developing datasets using wearable sensors, particularly with ground truth annotations, is very difficult and requires significant effort and time. The insights gained from movement visualization can be used to design data collection protocols. For example, we see here that the walking patterns are different from smoking. In order to develop a solution for smoking detection, all or most of the walking data can be discarded using orientation reachability. So, having more data on walking might not be useful in developing a more robust and accurate classification model. So, more efforts should be put for gathering data from confounding activities like drinking.

We demonstrated movement patterns of different rehabilitation exercises. It is possible to develop solutions that can measure the quality of exercise. However, our goal is to provide insights through visualization. It complements other solutions that provide some quantitative measures. We have not evaluated the effectiveness of monitoring rehabilitation exercises involving real patients and physicians/therapists. Future work includes evaluating the effectiveness of our visualization method for different types of exercises involving real patients and physicians/therapists.

Chapter 6

Interactive Reminders

Forgetfulness is a very common human trait. It usually becomes more severe as people get older. To mitigate the effect of this problem, people have long been using different techniques including alarms and reminder systems. With the advent of smartphones, reminder systems become very handy and ubiquitous. There are many reminder apps available for smartphones in app stores [96, 97]. Researchers have also designed, developed and evaluated smartphone-based reminder systems [99, 100]. However, smartphones have some limitations with regard to providing reminders. It usually requires involvement of both hands for using a smartphone. Users need to hold the phone in one hand and use it with the other. Occupation of the hands for using the phones, and the attention required to use the apps justify that smartphone-based systems are significantly intrusive. Also, smartphones provide limited effectiveness in different contexts [8]. A user is very likely to miss a reminder when the phone is far enough from his/her location at the time the reminder is given. For instance, a user may miss a reminder when he/she is busy in the kitchen, but the phone is in the bedroom far away from the user. Moreover, remainders may be missed while listening to songs, TVs or videos even if the phone is located near the user. In many situations, like in meetings and classrooms, smartphones typically need to be kept silent, and users often forget to return the devices back to the non-silent mode when silence is not required anymore. A user might misses some reminders in such scenarios.

As a smartwatch is worn on the wrist, it is free from the limitations mentioned above. However, one of the major challenges in developing interactive systems for smartwatches is their form factor. The touch screen available on a smartwatch is tiny and much smaller compared to smartphones and tablet computers. A watch can provide/take very limited information to/from the user via the display. Voice can be used to address this limitation, but it is not suitable for on-device voice processing because the computational resources available in the smartwatches are very limited. Voice data transmission to a server drains significant energy from the watch, and it does not work well when network connectivity is poor. We have developed a keyword-based interactive reminder system that requires low computational resources, and so can be implemented on devices with limited resources like the smartwatches. Our approach uses keywords in the speech of the user to understand their queries or responses. It is possible to detect keywords from speech with low computation and memory [16, 157–160]. Our system enables user interactions by incorporating speech recognition along with clever interface design. The tiny display of the device is used for minimal inputs and outputs, while a user can retrieve and provide more information from/to the system through voice commands.

The performance of a speech recognition or keyword detection solution depends on several factors including pronunciations of the speaker and background noise. The actual keywords or speech uttered by a speaker might differ from what the system recognizes. This problem is acute for non-native speakers. Though the problem is less severe for the native speakers, the errors reduce the usability of the system and result in poor user experiences. Many solutions are available for speech recognition and keyword detection [16, 157–164] that provide reasonable accuracy. Instead of developing another solution for keyword or speech recognition, we focus on correcting errors of existing solutions in the scope of reminder applications. Our system uses a personalized model that is built and updated over time to reduce errors in recognizing users' voice commands. Consequently, the system can provide a better user experience.

Medication adherence is one of the most important areas where reminders play a critical role. Proper adherence to prescribed medications is a fundamental requirement for effective health outcomes. The possible consequences of poor medication adherence include reduced effectiveness of treatments, deterioration of health conditions, longer recovery time, increased cost, irrecoverable damages to health, hospitalization, and even death. Despite the severe consequences, the medication adherence rate among patients is significantly low [165–167]. According to the World Health Organization (WHO), "Adherence to long-term therapy for chronic illnesses in developed countries averages 50%. In developing countries, The rates are even lower" [165]. Poor medication adherence is a public health problem, and WHO identifies it as a worldwide problem of striking magnitude. It causes about 33-69% of all the medication associated hospital admissions in the U.S. [168].

One of the main reasons for poor medication adherence is forgetfulness. People often forget to take medication at the appropriate times, and even sometimes take wrong dosages. Though forgetfulness is more prevalent in people with a reduced cognitive ability such as the elderly, it is also very common to healthy and young people due to factors like daily routine, habits, and changes in medication regimen. For example, irregular use of contraceptive pills increases the risk of unintended pregnancy. Nonetheless, 68.1% of the participants of a study reported missing at least one pill and 48.9% reported missing more than one during a 3-month period [169]. In the study, the average age of the participants is 20.9 years, and forgetfulness is listed as one of the main reasons for missing the pills. Several studies show that medication adherence is improved significantly with the use of automated reminder systems [170][171][172]. Considering the importance of medication, we focus on medication reminder and tracking. However, our system can be configured and used as a reminder for other purposes.

6.1 Solutions

We use both the touch screen of the watch and voice conversation for user interaction. Although limited, the interactions supported through the device screens are very useful when the users do not need or prefer voice interactions. Our app initiates a reminder by vibrating the watch and displaying some information on the screen. A user can understand the reminder by just looking at the display. However, there are cases when the touch screen is not suitable for exchanging information between the user and the system. Our system addresses the limitation of the touch screen using voice interactions. Some scenarios, when voice interactions are more suitable than the touch screen, are listed below.

- A user needs more information about a reminder beyond what is displayed on the screen.
- The user needs to reschedule the reminder at a later time of his/her preference. Providing time through the touchscreen requires attention and engagement that is not suitable for some contexts, for example, while driving.
- The user needs to record some information related to the reminder. For example, a note for the doctors related to the medication or exercise.
- The hands of the user are occupied, and the user wants to confirm medication intake.

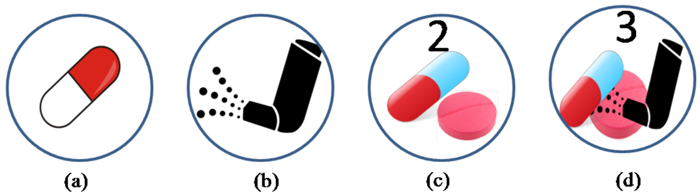


Figure 6.1: Some example symbols for a reminder (a) one pill (b) inhaler (c) two pills (d) two pills and the inhaler

6.1.1 Interactions through Touch Screen

Due to the form-factor of the wrist devices, we display very few symbols and words on the screen so that they are bigger in size and require very little attention or effort from the user to understand and interact. Different symbols and texts are used for different types of reminders. For example, a symbol of a pill is shown when the user is reminded to take a pill. Customized symbols for the reminders of different kinds of medications help the user to better comprehend why the reminder is given, particularly when the user needs to take multiple types of medications with different dosages. Examples of reminder symbols are shown in Figure 6.1. If a reminder is provided for multiple medications, the total number of medications is also displayed on the screen, as shown in Figures 6.1(c) and 6.1(d). When a user needs to be provided with critical information like changes of medicine, dosage or schedule, the display is blinked so that the user can easily understand that some important information is available there.

Our system not only provides reminders but also can record responses from the users. For example, the user can confirm medication intake just by clicking the display. It ensures medication tracking with minimum intervention. If the user hasn't taken the medication, the reminder session can be closed by double-clicking on the display, or just by leaving as it is. For the latter, the system automatically closes the reminder session as well as the display after some predefined time period. A reminder session

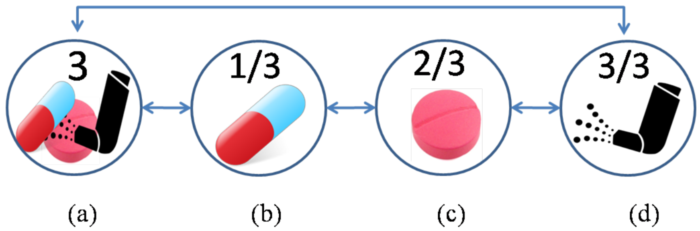


Figure 6.2: Example of navigation between pages for a multi-pill reminder.

starts when the system provides a reminder alert to the user and ends when the interaction between the user and the device is finished for that reminder. In case the medication intake is not confirmed, the reminder is rescheduled at a later time. More about rescheduling a reminder is described in the next section.

In cases where a reminder is given for multiple medications, a display page for each of the medications is available in addition to the combined display (Figure 6.2). Users can navigate between different pages by sweeping the display to the left or to the right. If the user has taken all the medications when the reminder is provided, he/she confirms it just by clicking on the combined display (Figure 6.2(a)). However, if the user has taken some of the medications, that can be confirmed by navigating to and clicking on the corresponding pages. This navigation feature enables easy tracking of partial medication intakes.

6.1.2 Interaction through Voice Conversation

To ensure that the system only talks when the user prefers, voice interaction is started when a user utters some specific word or phrase after the start of a reminder session. We denote the specific words or phrases as 'session initiators'. For example, users can use "*System*" or "*Reminder*" as a session initiator. The session initiator is configurable in our system, and so users can set it according to their preference. A user can also set multiple session initiators. If the system recognizes any of the session initiators after providing a reminder, it starts talking to the user.

Unlike a general-purpose personal assistant such as Amazon Alexa [173], Google Home [174] or Apple Siri [175], the purpose of a reminder system is very specific. We enable the specific conversations between the user and the system using regular expressions and a set of pre-defined responses. Our system doesn't require strict format for any voice command from the user except the session initiators. Users communicates with the system through natural expressions, but they need to include some required keywords or keyphrases. For example, if a user wants to reschedule a reminder after half an hour, possible expressions include, but not limited to:

"*Remind me after half an hour*"

"*Remind after thirty minutes*"

"*Please ask me after a half hour*"

Here, the command expression needs to include either "*remind*" or "*ask*", as well as it should include a specific time or interval for the reminder to be re-scheduled. For simplicity and without loss of generality, we refer the queries, commands, and responses from the users to our system as command. A set of regular expressions are defined according to the purpose of the reminders to understand the commands from the users. For each of the regular expression, a set of responses are also defined. The system cross-matches the keywords detected from a command with the set of regular expressions and responds to the user using the responses defined for the regular expression matched. For example, the above commands match with the following regular expression:

```
.* (remind|ask) .* (later | after TIME) .*
TIME → NUM hour (and)? NUM minute | NUM hour | NUM minute
NUM→ number (and half)? | half
```

If no regular expression is matched, the system informs the user that it cannot understand the command and asks the users to repeat. A list of example expressions with corresponding purpose, keywords, regular expression, and responses are listed in Table 6.1. The examples provided here are general purpose and can be used in reminders for different purposes, including medication, meeting, and scheduled events. The set of regular expressions and responses are configurable in our system, and so they can be customized according to the need of a user or user group. In cases when the user provides some information or instructions to the system like medication intake confirmation or about rescheduling a reminder, the system can be configured to repeat what it recognizes to make sure that the information is not recorded incorrectly. If it is not correct, the user can repeat.

6.1.3 Keyword Recognition

Instead of developing another solution for speech or keyword recognition, we focus on correcting errors of existing solutions in the scope of reminder applications. For that purpose, we develop a database of alternative keywords and keyphrases that are detected in place of the defined keywords by the speech or keyword recognizer. For simplicity, we use the term "keyword" in the rest of this chapter instead of distinguishing between "keyword" and "keyphrase". Each of the alternative keywords is associated to the corresponding defined keyword that is missed by the recognizer. For example, when a user tells the system "details please", the system might detect it as

Table 6.1: Examples of expressions, purposes, keywords, regular expressions and responses.

<table>
<tr><th>Example Commands</th><th>Purpose</th><th>Keyword</th><th>Regular Expression</th><th>Example Responses</th></tr>
<tr><td>System.
Hello Reminder.</td><td>Session initiation</td><td>system,
reminder.</td><td>(Hi|Hello)? (System | Reminder)</td><td>Hello.
Hi, Good Morning.</td></tr>
<tr><td>Details please.
Give me more information.
Tell me details.</td><td>To get more information related to the reminder</td><td>detail,
more.</td><td>.* (detail | more) .*</td><td>The meeting is in Room X of Building Y.
You should take the pill after eating.</td></tr>
<tr><td>Say it again please.
Please repeat it.
What?</td><td>To ask the system to repeat what it just told.</td><td>repeat,
what,
say again.</td><td>.* (repeat | what | say .* again) .*</td><td>The system repeats the last response.</td></tr>
<tr><td>I've taken the medicine.
I've done it.
Done.</td><td>To confirm that user has taken the action related to the reminder.</td><td>done,
taken.</td><td>.* (done | taken) .*</td><td>Thank you.</td></tr>
<tr><td>Yes.
Yes, I've taken.
No.</td><td>Answer to a yes/no question from the system.</td><td>yes,
no.</td><td>(yes|no) .*</td><td>Thank you.
Do you want me to remind you again?</td></tr>
<tr><td>Okay.
Thank you.
Okay thanks.</td><td>To terminate a reminder session.</td><td>okay
ok
thank</td><td>.* (okay | ok | thank) .*</td><td>You are welcome.</td></tr>
<tr><td>Remind me after half hour.
Ask me after one hour. and thirty minute
Please remind me later.</td><td>To ask the system to remind later (after a predefined time interval) or after the interval the user prefers.</td><td>remind,
ask + later
ask + after.</td><td>.* (remind|ask) .* (later | after TIME) .*
TIME $\rightarrow$ NUM hour (and)? NUM minute | NUM hour | NUM minute
NUM$\rightarrow$ number (and half)?</td><td>Ok, thank you.
I'll remind you after half hour.</td></tr>
<tr><td>Don't remind again.
Please don't remind it.</td><td>To ask the system not to remind again.</td><td>don't remind</td><td>.* (don't remind) .*</td><td>Ok, thank you.
Sure!</td></tr>
</table>

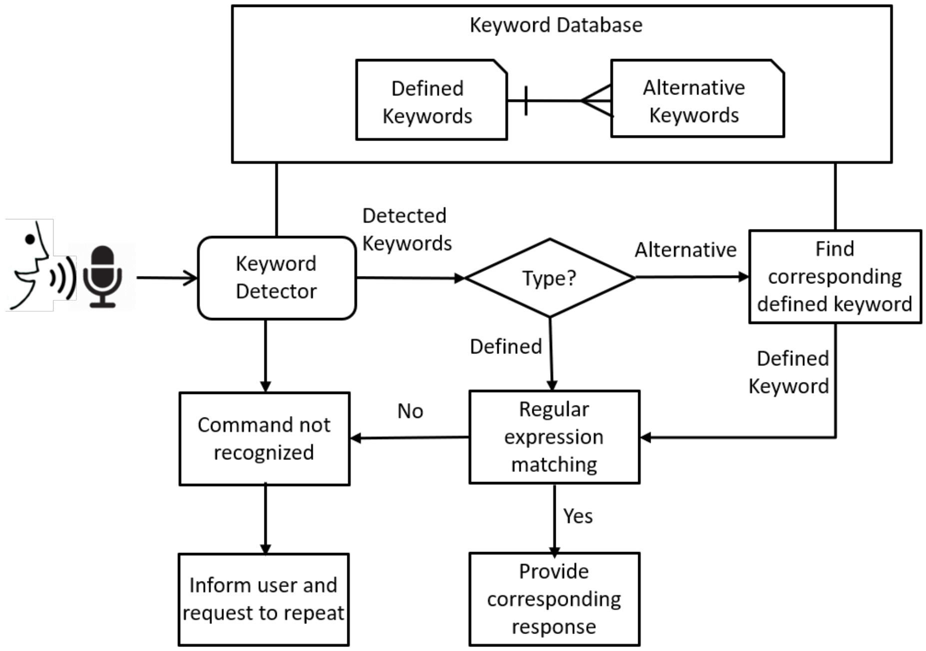


Figure 6.3: Voice command recognition process

"the dates please". Here, the keyword "details" is detected as "the dates" by the recognizer. So, "the dates" is an alternative keyword that is associated with the defined keyword "details". We develop a database of such alternative keywords. When a defined keyword is missed, but an alternative keyword is detected by the recognizer, the corresponding defined keyword is used to match the regular expressions. The process of understanding users' command is depicted in Figure 6.3. When a user talks to the system, it detects the presence of keywords in the speech. If a defined keyword is detected, it is cross-matched with the regular expressions. On the other hand, if an alternative keyword is detected, the corresponding defined keyword is used. In case the system cannot detect any keyword in the command or fails to match a regular expression, it informs the user and asks the users to repeat, as mentioned earlier.

6.1.4 Keyword Database

The set of alternative keywords is very critical to understand the user when the recognizer fails to detect the defined keywords. We use a personalized approach where the database of alternative keywords for a user is empty or pre-populated initially, i.e., when the user starts using the system. The database is populated and updated over time with the alternative keywords. When a command from a user is not recognized by the system, the speech is translated into text and the user is asked to correct the translation. Since the screen of a wrist device is not feasible for text inputs, a device with a larger display like a smartphone can be used for this purpose. The watch is usually connected to a smartphone using Bluetooth. The translated text is compared with the corrected text, and the alternative keywords along with the corresponding defined keyword are extracted. The keyword database is updated with the alternative keywords. Thus, the errors in recognizing the commands are reduced over time. It is not necessary for a user to correct the text immediately or every time the system fails to detect a command. The user can provide this input when feasible. Also, the user can provide input later at a time convenient to the user.

6.2 System Description

Our system is highly flexible and customizable. Different components of the system are described below.

6.2.1 Operating Script

The core of the system is an Operating Script (OS) that contains the list of reminders for a user along with necessary settings and information. The OS can be updated to change reminders and other settings. There are two components of the OS: general

```
{
    "id": 1,
    "type" : "daily",
    "time": "2:00 pm",
    "display symbol": "pill_X",
    "message": "Please take a pill of X ",
    "details": "Take the pill after eating. Report your
    physician immediately if there is any side effect.",
    "repetition interval": "00:20:00",
    "repetition period": "05:00:00",
    "repeat message": "Have you taken a pill of X?"
}
```

Figure 6.4: Example of a reminder entry in the Operating Script (OS) in JSON format.

settings and reminder list. General settings include OS identification number, last date and time the OS was updated, user name, and user preferences that are applicable to all the reminders. Each of the entries of the reminder list contains details of a reminder with fields such as id, type, time, and message. Reminder specific settings are also available in the corresponding entry. An example of an entry in the reminder list is shown in Figure 6.4. The "type" and "time" fields indicate that the reminder is provided daily at 2:00 pm, and the "display symbol" field indicates the symbol that is displayed on the wrist device when the reminder is given. The "message" field contains the message that is provided for the reminder, and the "details" field contains more information about the medication. Some other fields from this example reminder entry are discussed later.

6.2.2 Reminder Life Cycle

Our system allows users to reschedule or postpone a reminder. When a reminder is provided and the user doesn't respond or doesn't confirm that the action (e.g, medication) is taken, the reminder for that medication is rescheduled at a later time according to the settings of the reminder. The reminder is rescheduled repeatedly for a predefined period or number of times. For example, the repetition interval and period of the reminder of Figure 6.4 are defined as 20 minutes and 5 hours, respectively. So, the reminder is given first at the specified time at 2:00 pm, and it is repeated periodically with a 20-minute interval for up to 5 hours if the user doesn't respond to it. The default repetition interval for the reminder is changed according to the user's preferences if there is any. The repetition period as well as the repetition interval can differ among the reminders. In case the user explicitly asks the system to reschedule the reminder at a later time of his/her preference, it is rescheduled according to his/her preference instead of the default settings. As an example, consider a scenario when the user is driving and a reminder is provided, but the user needs one more hour before he/she can take the medications. In this case, the user can ask the system verbally to remind him/her after one hour instead of the default interval. This feature allows the user to avoid unnecessary reminders. The life cycle of a reminder is illustrated in Figure 6.5.

6.2.3 System Architecture

The system uses a microphone and a speaker along with the touch screen of the wrist device for taking inputs from and providing outputs to a user. The microphone and the speaker work as input and output media, respectively, and the touch screen works as both. The system is composed of several modules namely I/O Manager, Network Manager, Schedule Manager, Session Manager, and Storage Manager. The architecture of the system is shown in Figure 6.6.

The Schedule Manager is responsible for scheduling the reminders. It maintains a dynamic list of Repeated Reminders(RR) that contains information about the reminders which need to be repeated. Using the OS and the RR, the Schedule Manager schedules the next reminder session, and details of the next reminder session are sent

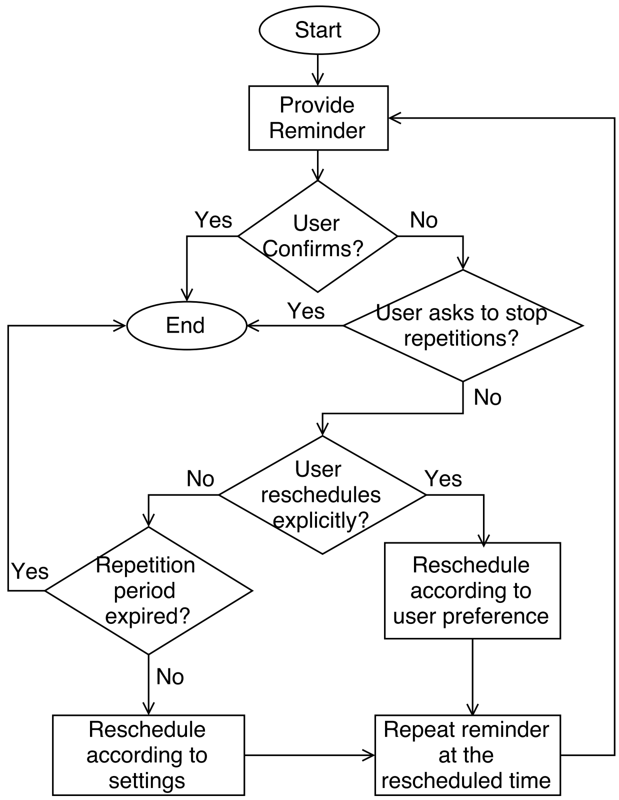


Figure 6.5: Life cycle of a reminder.

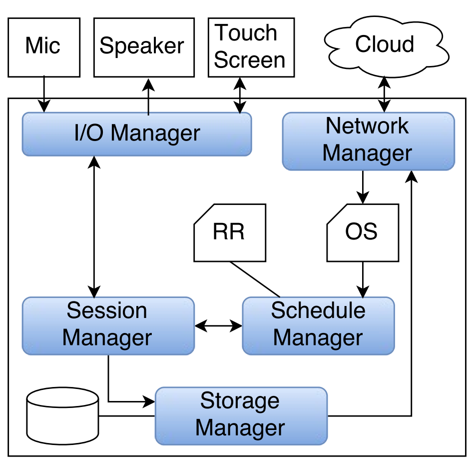


Figure 6.6: System Architecture

to the Session Manager that starts the next session according to the schedule. The Session Manager manages the whole workflow of a reminder session, including the recognition of the voice commands. It stores necessary information like medication intake confirmations. Data is stored temporarily on the device, and the Network Manager uploads the data to the cloud periodically or according to settings. The Network Manager is also responsible to look for and download the updates available in the cloud. The I/O Manager takes inputs from and provides outputs to the user. The Storage Manager helps to organize, store and retrieve data to/from the storage.

6.3 Experiments

We have developed an app for Android smartwatches. For the experiment, we have used ASUS Zenwatch2, an Android powered smartwatch that comes with a microphone, a speaker, and a 1.63-inch touch screen. The application converts the speech from the user into text using the speech recognition engine available on the Android platform, and then searches for the keywords in the commands. If a keyword is found and a regular expression is matched, a predefined response corresponding to the command is provided to the user. If the app cannot match any regular expression, it informs the user that it can not recognize the command. The watch provides the responses verbally using the speaker as well as by showing information on the display.

6.3.1 Data Collection

Data has been collected from 4 native and 6 non-native English speakers. The native speakers are from Untied States, and the non-native speakers are from Bangladesh and China. The participants in the study include undergraduate students, graduate students, a professor, and a housewife, and they are not real patients. The participants are provided with mock reminders on the smart watch, and they interact with the system using voice. The experiments are carried out in a semi-controlled environment where the actual speech of the users and the text generated by the Android Speech Recognizer are recorded by a second person. There is no constraint on the number or order of sentences to be used for each of the reminders. A total of 292 reminders are provided to the participants, and a total of 1142 commands from the participants are recorded for the reminders. 182 of these commands can not be recognized by the application using the Android Speech Recognizer.

6.3.2 Analysis

The performance of the system in recognizing the voice commands is evaluated for two types of training approaches: leave one person out ($LOPO$) training and personalized training. Here, error rate is defined as the ratio of the number of commands the system fails to recognize and the total number of commands given to the system.

In the $LOPO$ training, commands from each subject are tested using the database that is trained by the alternative keywords from other subjects. Figure 6.7 shows the error rates for with and without alternative keywords built from the $LOPO$ training. As shown in the figure, the error rates are 6.43% and 20.9% for the native and non-native speakers, respectively, when no alternative keyword is used. Using the $LOPO$ based alternative keywords, our system reduces the error rates to 2.96%, and 12.88%, respectively. Several of the defined keywords of a subject are not detected as any alternative keyword from the other subjects. It manifests that many of the errors are not generic, rather they are person specific.

For the personalized training, the database of alternative keyword for a user is built and updated with the unrecognized commands from him/her in the temporal order the commands are provided. Figure 6.8 illustrates how errors are reduced with the average number of commands used in training the system. The error rates are

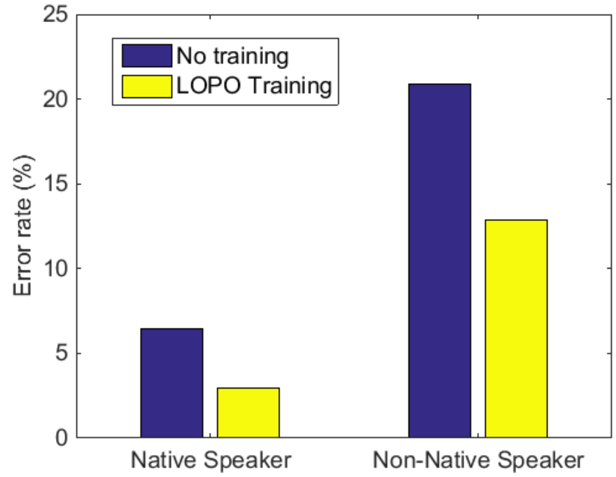


Figure 6.7: Error rates in recognizing intended voice commands using speech recognizer only (No training) and using speech recognizer with leave-one-person-out (LOPO) training

reduced to nearly zero after finding 1.25 and 15 alternative keywords, on average, for a native speaker and a non-native speaker, respectively. Once the system is trained for an unrecognized command, the command is recognized later if the same alternative keyword detected even though the corresponding defined keyword is missed. So, the error rate reduces over time by the training process. The required training effort from a user depends on the variety of command expressions used by the user as well as factors like his/her pronunciation and accent that are associated with speech recognition accuracy. The average number of training commands required to achieve nearly zero error is small due to the fact that a limited number of command keywords are defined in the system, and the varieties of the command expressions used by the users are generally not large.

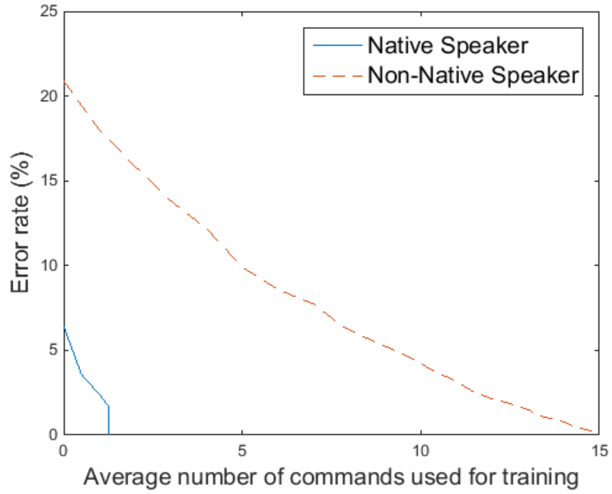


Figure 6.8: Reduction of error rates with personalized training

6.4 Discussion

As expected, the error rate of speech recognition by the state-of-the-art system is much higher for non-native speakers compared to native speakers. The performance of the speech recognizer also differs for native speakers with different accents. The alternative keyword-based training approach enables our system to be robust in recognizing command even from non-native speakers. Though the system is evaluated using the English language only, the design of the system is language independent. We have evaluated the system for medication reminder and tracking, but it can be used for other purposes like providing reminders for exercise and other daily activities, and for tracking the well-being of the users. For instance, the system can be configured to ask the users periodically about their physiological conditions and and record the responses.

One of the limitations of our experiments is the lack of real-world deployment. Previously unseen problems along with new issues are often observed when a system is used in real-world settings. The effectiveness of our system in improving medication adherence has not been explored in this study. However, studies show that automated reminder systems are effective in improving medication adherence, as mentioned before. Considering that our system comes with several useful features, and it overcomes many of the limitations of the existing automated systems, it would be more effective in improving medication adherence compared to the state of the art. Our system can be extremely useful for people with visual impairment, Essential Tremor, Parkinson's Disease, and other disabilities. In the future, different aspects of the system like its effectiveness and usability will be studied through long term and real-world deployments. To reduce interventions in tracking medication intake, automatic medication tracking features can be integrated into the system. The sensors like the accelerometers and the gyroscopes embedded into the wrist devices can be used to monitor medication intake through tracking the hand movement of the users. Automatic tracking of the medication using the wrist device will be explored in future endeavors.

Due to the form factor of the wrist devices, the capacity of the batteries available in the devices is typically very low. So, any system for these devices needs to be energy efficient. Our system uses the display, the microphone, and the speaker only during the reminder sessions. A reminder session typically runs for very short time, usually less than a minute. As stated before, if a user doesn't respond to a reminder within a predefined time period after the reminder is given, the reminder session terminates automatically. The time period is a configurable parameter with a default value of 10 seconds. When a reminder session ends, the next reminder is scheduled, an the app moves to hibernate state. The underlying operating system of the watch wakes the app up according to the schedule. So, the energy consumption by the app on the watch is not significant.

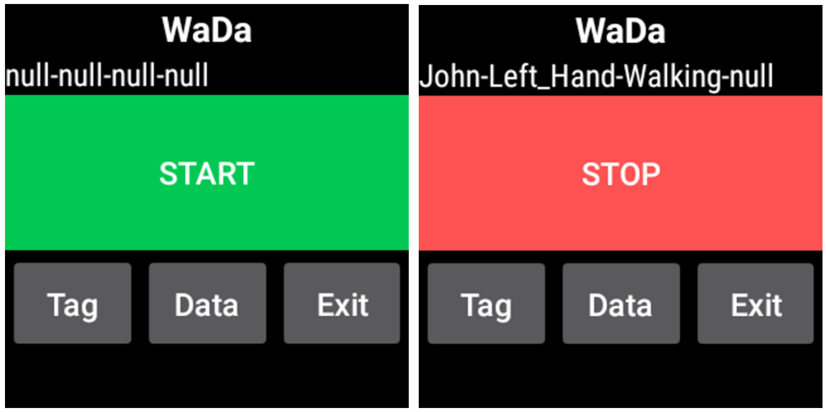
Chapter 7

Data Collection Tool

Activity recognition solutions are generally data-driven, and data is essential to develop and evaluate such solutions. Researchers often spend a significant amount of effort and time in developing devices and/or apps for data collection. Most of the apps and devices are customized with limited options, often for specific sensors and applications. It is not easy or convenient to distribute customized devices to other researchers, and so the use of such a data collection system is usually limited to a small group. We developed WaDa, an easy-to-use app for sensor data collection using commercial off-the-shelf Android smartwatches. The app is publicly available, and it facilitates prompt data collection without requiring expertise and effort for custom device/app development. It has been used by researchers and in various academic courses.

7.1 Data Collection

WaDa runs on the watch, and it does not require a smart phone. This independence makes WaDa more convenient and less constrained compared to the watch apps that require a companion smart phone app. Data collection is started and stopped using



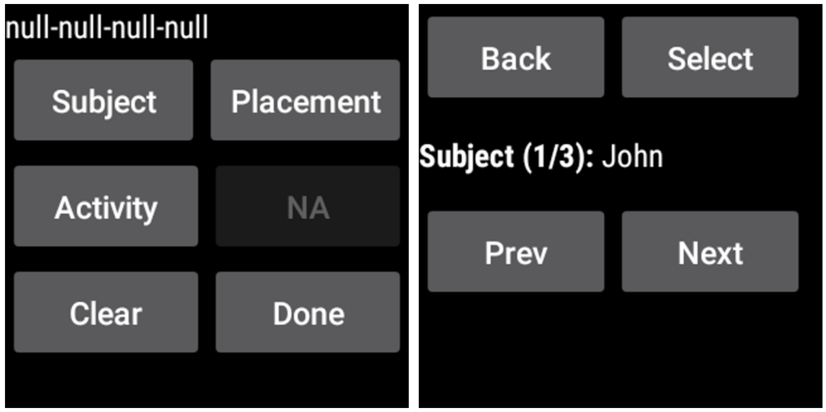
(a) (b)

Figure 7.1: (a) Starting and (b) Stopping data collection in WaDa.

a button available in the app (Figure 7.1). Once started, the app collects data in the background, and the 'STOP' option becomes available. Closing the app does not stop sensor data collection. The 'STOP' option is used explicitly to end data collection. The data is saved in the watch storage.

7.2 Data Labeling

It is very important to label or tag the data, particularly for the ground truth purpose. Wada allows using multiple tags for the data. The tags can represent different ground truth information for the data like the code or name of the participant, the position of the device (e.g., left hand, right hand, waist), the activity for which the data is being collected and so on. Instead of fixed tags, WaDa allows up to four user-defined tags and any number of options for each of the tags. Figure 7.2(a) shows an example where WaDa has been configured for three tags (Subject, Placement, and Activity). WaDa provides easy mechanisms to navigate and select the options for the tags. Figure 7.2(b) shows the interface for a tag (Subject) with 5 options where the 'Prev' and the 'Next' buttons are available for navigating the options. Such an interface is available for each of the tags with a corresponding user-defined option list. Once



(a) (b)

Figure 7.2: (a) List of tags and (b) Option navigation for a tag.

the options are selected, they are displayed just above the 'START/STOP' button (Figure 7.1(b)). The tags are defined before starting data collection, and they are embedded in the name of the file that contains the sensor data. It is not necessary to define or select all the tags. Any tag that is not selected is marked as 'null'.

As mentioned earlier, the name of the tags and their options are not hard coded; rather they are defined by the users according to their preference and the need of the applications. It is very difficult to input text or other options in the watches due to their small form factor. We provide a companion desktop app to define the tags, the options for the tags, the sensors, and the sampling rates for the sensors. Figure 7.4 shows the interface from the desktop app used for these purposes. Up to four different tags can be defined, but there is no limit on the number of options for the tags. In Figure 7.4, we show an example where three tags and some options for each of the tags are defined. The push button is used to transfer the configuration to a watch while the watch and the laptop are connected via a USB cable. The configuration can be named, and then saved in the desktop.

7.3 Sensors and Sampling Rates

In WaDa, a user selects the sensors and their sampling rates. Different models of the Android smart watches have different set of sensors, and new sensors are often added to new models. Also, different applications need data from different set of sensors. Instead of providing an exhaustive list of all possible sensors, WaDa allows the users to create a list of sensors, and then select the sensors needed for the target application. By default three of the most widely used sensors (the Accelerometer, the Gyroscope, and the Magnetometer) are listed in the desktop app. To add a sensor to the list, a user provide the ID of the sensor, as defined by the Android platform [176]. For example, the ID of the Ambient Temperature Sensor is 13. A user can select any number of sensors from the list, and set the sampling rate for each of the sensors. Android has four predefined sampling rates: SENSOR_DELAY_UI, SENSOR_DELAY_NORMAL, SENSOR_DELAY_GAME, and SENSOR_DELAY_FASTEST [177]. Any of these four or a custom (in Hz) rate can be set for a sensor. The list of sensors, their sampling rates and availability (Y/N) can be checked in the watch as shown in Figure 7.5(a).

7.4 Data Navigation

The data collected in a session (between a start and a stop) is saved in a single file in the watch. The list of the files can be navigated in the app (Figure 7.3). The file name contains different information including the serial number of the watch, the name of the configuration file, the selected options of the tags (the labels) and the time when the data collection started. A user can delete an individual file or all the files from the watch when needed. The companion app is used to download the data or files from the watch to a desktop/laptop. The download option is available in the 'Data' panel in the desktop app (Figure 7.4). Once the data files are downloaded to

(a) (b)

Figure 7.3: (a) The total file count and the option for deleting all the files (b) File Navigation with option for deleting individual file.

a desktop/laptop, they should be deleted from the watch to free storage.

7.5 Data Format

The data is saved in CSV (Comma Separated Values) format where each line in the file represents a sample from a sensor, and includes the timestamp of the event in milliseconds (The Unix Epoch time), the ID of the the sensor, the accuracy of the sample, and the sensor readings. The format is like:

$Timestamp, SensorID, Accuracy, [Values]$

The number of readings or values for a sample depends on the sensor. Fore example, an accelerometer provides three values that are the acceleration along the X, Y and Z axes. On the other hand, an Ambient Temperature sensor provides a single value.

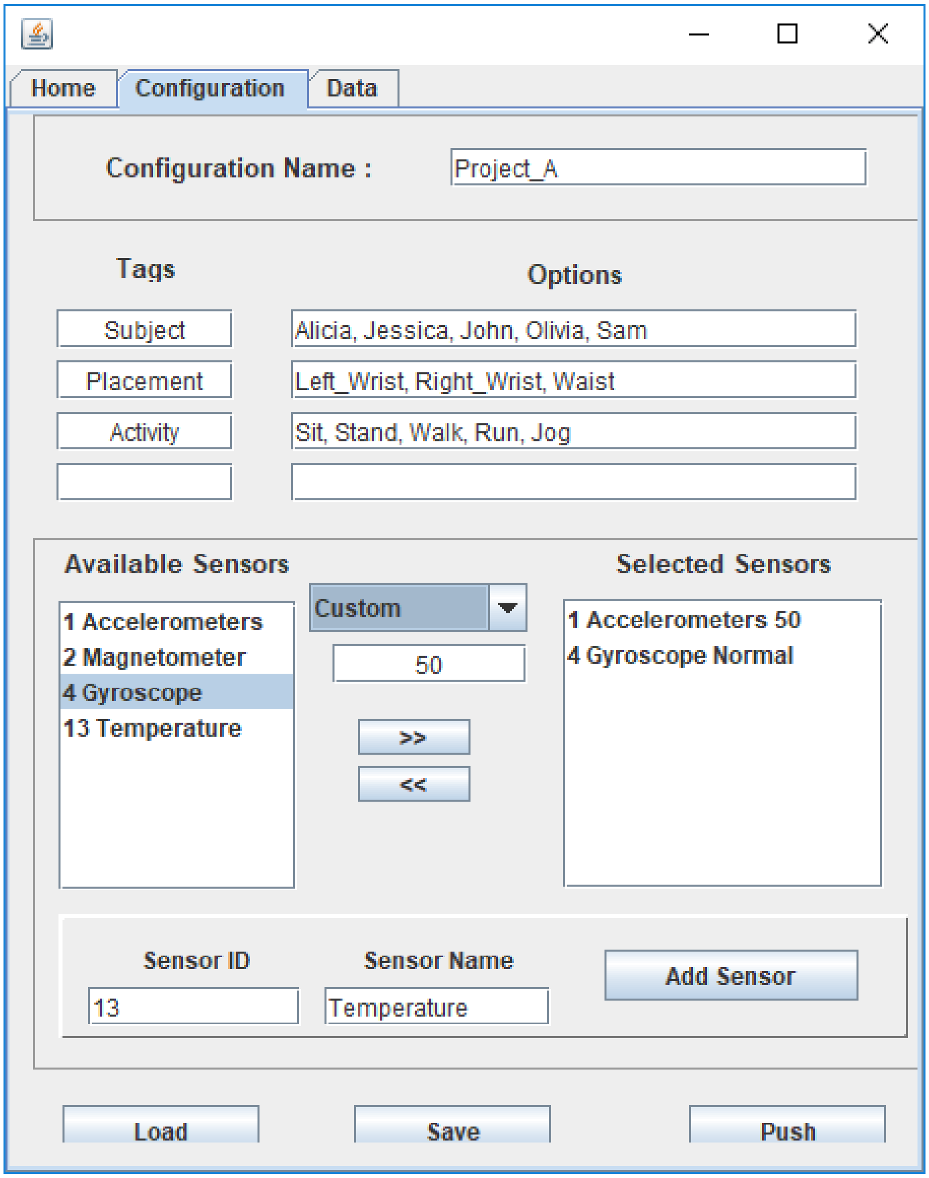


Figure 7.4: Companion desktop app for the configuration and the data download

(a) (b)

Figure 7.5: (a) Summary of the configuration and (b) The Unix Epoch time in the watch

7.6 Time Synchronization

The timestamp of the sensor sample represents the time from the watch when the sensor is sampled. However, the time of the watch might not be synchronized with other devices (e.g., the camera used for the ground truth or other devices used together for data collection). It is important that the times from multiple devices are synchronized. WaDa displays the epoch time of the watch in milliseconds that is updated every 100 ms (Figure 7.5(b)). This time can be recorded using a camera, and be used to synchronize the watch with other devices including the camera itself.

7.7 Discussion

Smart watches are designed to be used on the wrists. However, a watch, due to its small form factor, can be placed on different parts of the body (e.g., the waist, the chest, the legs, and the upper arms) using appropriate straps. So, WaDa can be used to collect data not only from the wrist, but also from other parts of the body. WaDa is easy to use, and it supports prompt data collection without requiring the time, the effort and the expertise needed for custom app/device development. WaDa is a generic app for data collection, and it has the potential to be used for many applications in different domains including ubiquitous computing, health care, elderly monitoring, and behavioral science. Earlier versions of the app have been used to collect data for eating (Chapter 3) and hand washing (Chapter 4) detection, and movement visualization (Chapter 5). The app has been used for hands-on experiments in three different courses at undergraduate and graduate levels at the University of Virginia. Students with no or little prior experience of sensor data collection or using smart watches found it simple and easy to use. The app along with the detailed instructions and related information are available in a public repository [18].

Chapter 8

Summary and Future Work

The focus of this thesis is to utilize the orientation of the device and the distribution of the data to develop more efficient, robust, and accurate solutions for activity recognition as well as to develop a useful visualization technique. We have developed novel solutions for activity recognition, and evaluated the solutions for three activities of daily living, namely eating, hand washing, and smoking. Our technique for visualization is useful in understanding movement and orientation. Reminders are often an integral part of activity recognition systems. Though the focus of the thesis is activity recognition and movement visualization, we developed a novel and interactive reminder that closes the loop. Here, we summarize the solutions and the findings from the experiments. This chapter concludes by providing directions for future research.

8.1 Summary

8.1.1 Monitoring Family Eating Dynamics

- We have developed an efficient and effective solution for eating gesture detection leveraging the orientation of the wrist. This solution has been used in MFED, a system that we designed and developed for monitoring family eating dynamics.

- Our algorithm detects eating gestures with a F1-score of 0.74, 0.12 and 0.21 more when compared to the methods proposed by Thomaz et al. [58] and Mirtchouk et al. [59] that give F1-scores of 0.62 and 0.53, respectively. Our method discards more that 80% of the data using a threshold, and so it requires less than 20% computation compared to other methods.

- We deployed the MFED system for approximately two weeks in 20 real homes with a total of 74 participants. The system sent a total of 14413 EMAs to the participants with 13776 and 637 EMAs for mood and eating, respectively. The participants responded to 4750 of the EMAs (4224 for mood and 526 for eating).

- The EMA responses from the participants justifies the phenomenon that people eat meals together more than alone and eat snacks alone more than with others.

- The collaborative approach gives ground truth for 170 instances for which first-person ground truth is not available. It is about 32% of all the eating EMAs responded by the participants. So, the collaborative approach is significantly effective in collecting ground truth data.

8.1.2 Hand Washing Detection

- We developed a robust and effective solution for hand washing detection using smartwatches.

- The outputs of the penultimate layer for most of the hand washing instances are clustered to a limited area, whereas instances detected falsely by the neural network, particularly from unseen activities, spread over a larger area. We leverage the data distribution of hand washing to detect false positives.

- Our method reduces the false positives from the WISDM dataset (unseen) by about 77% and improves overall F1-score by 30%.

- The distance threshold parameter can be adjusted to balance the trade-off between precision and recall.

8.1.3 Movement Visualization

- We have developed a novel method for movement visualization using wearable inertial sensors. Our visualization method helps to analyze and understand data that can further help in developing effective and efficient solutions.

- We analyzed a dataset related to smoking activity recognition and provided several insights through representing the data using our method.

- We developed an efficient solution for smoking puff detection using the orientation of the wrist. Using a threshold on the gravitational acceleration along X axis, we can reduce computation by about 80% for detecting smoking puffs.

- The visualization shows that the orientations during smoking puffs are limited to a specific area on the sphere, and we can further filter out non-smoking gestures by leveraging the distribution of the orientations. It reduces the overall computation.

- We illustrated how our method can be used to monitor movement patterns related to several rehabilitation exercises.

8.1.4 Interactive Reminder

- We designed and developed a voice based interactive reminder system by focusing on a few keywords to understand users' commands.

- The error rates of using the off-the-shelf speech recognition systems are 6.43% and 20.9% for the native and the non-native speakers, respectively.
- Our solution can reduce the error rate to nearly zero for both the native and the non-native speakers with only 1.25 and 15 training commands on average, respectively.

8.1.5 Data Collection Tool

- We have developed an easy-to-use app for sensor data collection from commercial off-the-shelf Android smartwatches.
- The app comes with handy features such as user-defined labels, time synchronization, sensor selection, and data navigation.
- The app is publicly available. It has been used by researchers and in various academic courses.

8.2 Directions for Future Research

Research is a continuous endeavor, and there are always scopes for further improvements or experimentation. The solutions presented in this dissertation are efficient and effective, but they have limitations that lead to the path for future research. Here, we list some directions for future research works.

- The data collected by the MFED system can be used to model family eating dynamics. Such a model would facilitate just-in-time interventions that could reduce obesity. Our MFED ssytem and the dataset would enable future works in modeling family eating dynamics and just-in-time interventions.

- The MFED dataset contains sensor data from real deployments with responses from the participants about eating events. The data can be used to develop better solutions for eating gesture and event detection.

- Locating the MFED users at home is out of the scope of this dissertation. However, we have collected data from Bluetooth beacons that in combination with other data would be useful in developing and evaluating solutions for localization. It can be incorporated in the MFED system as well as for modeling FED.

- We have not evaluated the solution for hand washing detection in the real world. A system can be developed to monitor hand washing practices and to provide reminders when necessary. We can learn many issues from a real-world deployment that would help to improve the solution. Future works include deploying and evaluating the solution in hospitals, food businesses, and daily life, and using the findings to further improve the solution.

- We have evaluated the hand washing detection solution for feed-forward networks only. Future works include evaluating it for different types of networks including convolutional and recurrent neural networks.

- Our visualization method can help in developing better solutions for different activity recognition tasks. We have developed a solution for smoking puff detection. Using our method in developing better solutions for other activities is a potential direction for future works.

- We have not evaluated the utility of our visualization methods involving real stakeholders. For example, the effectiveness of our visualization method for monitoring different types of exercises need to be evaluated involving real patients and physicians/therapists.

- We have not evaluated the reminder system with real users. Future works include evaluating the reminder system in the real-world for different purposes like medication and daily routines.

- A reminder can be combined with activity recognition to provide context and activity aware reminder or interventions. For example, it can be used to provide reminders if the user forgets to wash hands properly or to intervene the user during binge eating events.

- Though the data collection tools have been used by other researchers and in classes, we haven't evaluated its usefulness, ease of use and other features through formal study. A formal study for evaluation and getting feedback from users will help to improve the tool.

Bibliography

[1] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster. "Wearable activity tracking in car manufacturing". In: *IEEE Pervasive Computing* 2 (2008), pp. 42–50.

[2] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey". In: *Architecture of computing systems (ARCS), 2010 23rd international conference on*. VDE. 2010, pp. 1–10.

[3] P. Cottone, S. Gaglio, G. L. Re, and M. Ortolani. "User activity recognition for energy saving in smart homes". In: *Pervasive and Mobile Computing* 16 (2015), pp. 156–170.

[4] L. Liu, C. Karatas, H. Li, S. Tan, M. Gruteser, J. Yang, Y. Chen, and R. P. Martin. "Toward detection of unsafe driving with wearables". In: *Proceedings of the 2015 workshop on Wearable Systems and Applications*. ACM. 2015, pp. 27–32.

[5] D. Riboni, C. Bettini, G. Civitarese, Z. H. Janjua, and V. Bulgari. "From lab to life: Fine-grained behavior monitoring in the elderly's home". In: *Pervasive Computing and Communication Workshops (PerCom Workshops), 2015 IEEE International Conference on*. IEEE. 2015, pp. 342–347.

[6] T. Fawcett and F. Provost. "Activity monitoring: Noticing interesting changes in behavior". In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 1999, pp. 53–62.

[7] S. J. Biddle and N. Mutrie. *Psychology of physical activity: Determinants, well-being and interventions*. Routledge, 2007.

[8] M Hayakawa, Y Uchimura, K Omae, K Waki, H Fujita, K Ohe, et al. "A smartphone-based medication self-management system with realtime medication monitoring". In: *Appl Clin Inform* 4.1 (2013), pp. 37–52.

[9] *Forecast wearables unit shipments worldwide*. `https://www.statista.com/statistics/437871/wearables-worldwide-shipments`. last accessed Dec 14, 2019. 2019.

[10] A. Bulling, U. Blanke, and B. Schiele. "A tutorial on human activity recognition using body-worn inertial sensors". In: *ACM Computing Surveys (CSUR)* 46.3 (2014), p. 33.

[11] S. Zhang, R. Alharbi, M. Nicholson, and N. Alshurafa. "When generalized eating detection machine learning models fail in the field". In: *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM. 2017, pp. 613–622.

[12] F. J. Ordóñez and D. Roggen. "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition". In: *Sensors* 16.1 (2016), p. 115.

[13] N. Y. Hammerla, S. Halloran, and T. Plötz. "Deep, convolutional, and recurrent models for human activity recognition using wearables". In: *arXiv preprint arXiv:1604.08880* (2016).

[14] Y. Guan and T. Ploetz. “Ensembles of Deep LSTM Learners for Activity Recognition using Wearables”. In: *arXiv preprint arXiv:1703.09370* (2017).

[15] Y. Bi, M. Lv, C. Song, W. Xu, N. Guan, and W. Yi. “Autodietary: A wearable acoustic sensor system for food intake recognition in daily life”. In: *IEEE Sensors Journal* 16.3 (2015), pp. 806–816.

[16] S. O. Arik, M. Kliegl, R. Child, J. Hestness, A. Gibiansky, C. Fougner, R. Prenger, and A. Coates. “Convolutional recurrent neural networks for small-footprint keyword spotting”. In: *arXiv preprint arXiv:1703.05390* (2017).

[17] C. Shan, J. Zhang, Y. Wang, and L. Xie. “Attention-based end-to-end models for small-footprint keyword spotting”. In: *arXiv preprint arXiv:1803.10916* (2018).

[18] *Wada App Repository*. `https://github.com/abumondol/WaDa`. 2018.

[19] J. M. Fontana and E. Sazonov. “Detection and characterization of food intake by wearable sensors”. In: *Wearable Sensors*. Elsevier, 2015, pp. 591–616.

[20] E. Thomaz, I. A. Essa, and G. D. Abowd. “Challenges and Opportunities in Automated Detection of Eating Activity”. In: *Mobile Health*. Springer, 2017, pp. 151–174.

[21] T. Vu, F. Lin, N. Alshurafa, and W. Xu. “Wearable food intake monitoring technologies: A comprehensive review”. In: *Computers* 6.1 (2017), p. 4.

[22] H. Hassannejad, G. Matrella, P. Ciampolini, I. De Munari, M. Mordonini, and S. Cagnoni. “Automatic diet monitoring: A review of computer vision and wearable sensor-based methods”. In: *International journal of food sciences and nutrition* 68.6 (2017), pp. 656–670.

[23] G. Schiboni and O. Amft. "Automatic Dietary Monitoring Using Wearable Accessories". In: *Seamless Healthcare Monitoring: Advancements in Wearable, Attachable, and Invisible Devices* (2017), p. 369.

[24] F. Zhu, M. Bosch, C. J. Boushey, and E. J. Delp. "An image analysis system for dietary assessment and evaluation". In: *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE. 2010, pp. 1853–1856.

[25] F. Kong and J. Tan. "DietCam: Automatic dietary assessment with mobile camera phones". In: *Pervasive and Mobile Computing* 8.1 (2012), pp. 147–163.

[26] E. Thomaz, A. Parnami, I. Essa, and G. D. Abowd. "Feasibility of identifying eating moments from first-person images leveraging human computation". In: *Proceedings of the 4th International SenseCam & Pervasive Imaging Conference*. ACM. 2013, pp. 26–33.

[27] E. Thomaz, A. Parnami, J. Bidwell, I. Essa, and G. D. Abowd. "Technological approaches for addressing privacy concerns when recognizing eating behaviors with wearable cameras". In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM. 2013, pp. 739–748.

[28] O. Amft, M. Stäger, P. Lukowicz, and G. Tröster. "Analysis of chewing sounds for dietary monitoring". In: *International Conference on Ubiquitous Computing*. Springer. 2005, pp. 56–72.

[29] O. Amft and G. Troster. "Methods for detection and classification of normal swallowing from muscle activation and sound". In: *Pervasive Health Conference and Workshops, 2006*. IEEE. 2006, pp. 1–10.

[30] E. Sazonov, S. Schuckers, P. Lopez-Meyer, O. Makeyev, N. Sazonova, E. L. Melanson, and M. Neuman. "Non-invasive monitoring of chewing and swal-

lowing for objective quantification of ingestive behavior". In: *Physiological measurement* 29.5 (2008), p. 525.

[31] E. S. Sazonov, O. Makeyev, S. Schuckers, P. Lopez-Meyer, E. L. Melanson, and M. R. Neuman. "Automatic detection of swallowing events by acoustical means for applications of monitoring of ingestive behavior". In: *IEEE Transactions on Biomedical Engineering* 57.3 (2010), pp. 626–633.

[32] P. Lopez-Meyer, S. Schuckers, O. Makeyev, J. M. Fontana, and E. Sazonov. "Automatic identification of the number of food items in a meal using clustering techniques based on the monitoring of swallowing and chewing". In: *Biomedical signal processing and control* 7.5 (2012), pp. 474–480.

[33] S. Päßler and W.-J. Fischer. "Acoustical method for objective food intake monitoring using a wearable sensor system". In: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*. IEEE. 2011, pp. 266–269.

[34] S. Päßler, M. Wolff, and W.-J. Fischer. "Food intake monitoring: an acoustical approach to automated food intake activity detection and classification of consumed food". In: *Physiological measurement* 33.6 (2012), p. 1073.

[35] S. Päßler and W.-J. Fischer. "Food intake monitoring: Automated chew event detection in chewing sounds". In: *IEEE journal of biomedical and health informatics* 18.1 (2014), pp. 278–289.

[36] K. Yatani and K. N. Truong. "BodyScope: a wearable acoustic sensor for activity recognition". In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM. 2012, pp. 341–350.

[37] T. Rahman, A. T. Adams, M. Zhang, E. Cherry, B. Zhou, H. Peng, and T. Choudhury. "BodyBeat: a mobile system for sensing non-speech body sounds." In: *MobiSys*. Vol. 14. 2014, pp. 2–13.

[38] T. Rahman, M. Czerwinski, R. Gilad-Bachrach, and P. Johns. "Predicting About-to-Eat Moments for Just-in-Time Eating Intervention". In: *Proceedings of the 6th International Conference on Digital Health Conference*. ACM. 2016, pp. 141–150.

[39] E. Thomaz, C. Zhang, I. Essa, and G. D. Abowd. "Inferring meal eating activities in real world settings from ambient sounds: A feasibility study". In: *Proceedings of the 20th International Conference on Intelligent User Interfaces*. ACM. 2015, pp. 427–431.

[40] E. S. Sazonov and J. M. Fontana. "A sensor system for automatic detection of food intake through non-invasive monitoring of chewing". In: *IEEE sensors journal* 12.5 (2012), pp. 1340–1348.

[41] M. Farooq, J. M. Fontana, and E. Sazonov. "A novel approach for food intake detection using electroglottography". In: *Physiological measurement* 35.5 (2014), p. 739.

[42] R. Zhang, S. Bernhart, and O. Amft. "Diet eyeglasses: Recognising food chewing using EMG and smart eyeglasses". In: *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on*. IEEE. 2016, pp. 7–12.

[43] J. Liu, E. Johns, L. Atallah, C. Pettitt, B. Lo, G. Frost, and G.-Z. Yang. "An intelligent food-intake monitoring system using wearable sensors". In: *Wearable and Implantable Body Sensor Networks (BSN), 2012 Ninth International Conference on*. IEEE. 2012, pp. 154–160.

[44] H. Kalantarian, B. Mortazavi, N. Alshurafa, C. Sideris, T. Le, and M. Sarrafzadeh. "A comparison of piezoelectric-based inertial sensing and audio-based detection of swallows". In: *Obesity Medicine* 1 (2016), pp. 6–14.

[45] C. Merck, C. Maher, M. Mirtchouk, M. Zheng, Y. Huang, and S. Kleinberg. "Multimodality sensing for eating recognition". In: *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare.* ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2016, pp. 130–137.

[46] H. Junker, O. Amft, P. Lukowicz, and G. Tröster. "Gesture spotting with body-worn inertial sensors to detect user activities". In: *Pattern Recognition* 41.6 (2008), pp. 2010–2024.

[47] O. Amft and G. Tröster. "Recognition of dietary activity events using on-body sensors". In: *Artificial intelligence in medicine* 42.2 (2008), pp. 121–136.

[48] J. M. Fontana, M. Farooq, and E. Sazonov. "Automatic ingestion monitor: a novel wearable device for monitoring of ingestive behavior". In: *IEEE Transactions on Biomedical Engineering* 61.6 (2014), pp. 1772–1779.

[49] X. Ye, G. Chen, and Y. Cao. "Automatic eating detection using head-mount and wrist-worn accelerometers". In: *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on.* IEEE. 2015, pp. 578–581.

[50] M. Farooq and E. Sazonov. "Accelerometer-Based Detection of Food Intake in Free-living Individuals". In: *IEEE Sensors Journal* (2018).

[51] R. I. Ramos-Garcia, E. R. Muth, J. N. Gowdy, and A. W. Hoover. "Improving the recognition of eating gestures using intergesture sequential dependencies".

In: *IEEE journal of biomedical and health informatics* 19.3 (2015), pp. 825–831.

[52] Y. Shen, J. Salley, E. Muth, and A. Hoover. “Assessing the accuracy of a wrist motion tracking method for counting bites across demographic and food variables”. In: *IEEE journal of biomedical and health informatics* 21.3 (2017), pp. 599–606.

[53] E. R. Muth and A. Hoover. “Designing Mobile Health Technologies for Self-Monitoring: The Bite Counter as a Case Study”. In: *Mobile Health*. Springer, 2017, pp. 101–120.

[54] S. Zhang, W. Stogin, and N. Alshurafa. “I sense overeating: Motif-based machine learning framework to detect overeating using wrist-worn sensing”. In: *Information Fusion* 41 (2018), pp. 37–47.

[55] K. Kyritsis, C. L. Tatli, C. Diou, and A. Delopoulos. “Automated analysis of in meal eating behavior using a commercial wristband IMU sensor”. In: *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*. IEEE. 2017, pp. 2843–2846.

[56] Y. Dong, A. Hoover, J. Scisco, and E. Muth. “A new method for measuring meal intake in humans via automated wrist motion tracking”. In: *Applied psychophysiology and biofeedback* 37.3 (2012), pp. 205–215.

[57] Y. Dong, J. Scisco, M. Wilson, E. Muth, and A. Hoover. “Detecting periods of eating during free-living by tracking wrist motion”. In: *IEEE journal of biomedical and health informatics* 18.4 (2014), pp. 1253–1260.

[58] E. Thomaz, I. Essa, and G. D. Abowd. “A practical approach for recognizing eating moments with wrist-mounted inertial sensing”. In: *Proceedings of*

the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM. 2015, pp. 1029–1040.

[59] M. Mirtchouk, D. Lustig, A. Smith, I. Ching, M. Zheng, and S. Kleinberg. "Recognizing eating from body-worn sensors: Combining free-living and laboratory data". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (2017), p. 85.

[60] *ACE Dataset*. `http://healthailab.org/data.html`. 2017.

[61] S. Flutura, A. Seiderer, I. Aslan, C.-T. Dang, R. Schwarz, D. Schiller, and E. André. "DrinkWatch: A Mobile Wellbeing Application Based on Interactive and Cooperative Machine Learning". In: *Proceedings of the 2018 International Conference on Digital Health*. 2018, pp. 65–74.

[62] K. S. Chun, S. Bhattacharya, and E. Thomaz. "Detecting eating episodes by tracking jawbone movements with a non-contact wearable sensor". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (2018), p. 4.

[63] A. Bedri, R. Li, M. Haynes, R. P. Kosaraju, I. Grover, T. Prioleau, M. Y. Beh, M. Goel, T. Starner, and G. Abowd. "EarBit: using wearable sensors to detect eating episodes in unconstrained environments". In: *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 1.3 (2017), p. 37.

[64] J. Reichenberger, P. Kuppens, M. Liedlgruber, F. H. Wilhelm, M. Tiefengrabner, S. Ginzinger, and J. Blechert. "No haste, more taste: An EMA study of the effects of stress, negative and positive emotions on eating behavior". In: *Biological psychology* 131 (2018), pp. 54–62.

[65] G. F. Dunton, E. Dzubur, J. Huh, B. R. Belcher, J. P. Maher, S. O'Connor, and G. Margolin. "Daily associations of stress and eating in mother–child dyads". In: *Health Education & Behavior* 44.3 (2017), pp. 365–369.

[66] A. B. Goldschmidt, R. D. Crosby, L. Cao, C. M. Pearson, L. M. Utzinger, C. R. Pacanowski, T. B. Mason, L. A. Berner, S. G. Engel, S. A. Wonderlich, et al. "Contextual factors associated with eating in the absence of hunger among adults with obesity". In: *Eating behaviors* 26 (2017), pp. 33–39.

[67] K. Alabduljader, M. Cliffe, F. Sartor, G. Papini, W. M. Cox, and H.-P. Kubis. "Ecological momentary assessment of food perceptions and eating behavior using a novel phone application in adults with or without obesity". In: *Eating behaviors* 30 (2018), pp. 35–41.

[68] C. Hlady, M. Severson, A. Segre, and P. Polgreen. "A mobile handheld computing application for recording hand hygiene observations". In: *Infection control and hospital epidemiology: the official journal of the Society of Hospital Epidemiologists of America* 31.9 (2010), p. 975.

[69] C. Davis. "Infection-free surgery: how to improve hand-hygiene compliance and eradicate methicillin-resistant Staphylococcus aureus from surgical wards". In: *Annals of the Royal College of Surgeons of England* 92.4 (2010), p. 316.

[70] G. Kinsella, A. Thomas, and R. Taylor. "Electronic surveillance of wall-mounted soap and alcohol gel dispensers in an intensive care unit". In: *Journal of Hospital Infection* 66.1 (2007), pp. 34–39.

[71] M. Edmond, A. Goodell, W. Zuelzer, K. Sanogo, K. Elam, and G. Bearman. "Successful use of alcohol sensor technology to monitor and report hand hygiene compliance". In: *Journal of Hospital Infection* 76.4 (2010), pp. 364–365.

[72] P. Polgreen, C. Hlady, M. Severson, A. Segre, and T. Herman. "Method for automated monitoring of hand hygiene adherence without radio-frequency identification". In: *Infection control and hospital epidemiology: the official journal of the Society of Hospital Epidemiologists of America* 31.12 (2010), p. 1294.

[73] S. Mohamed Rabeek, A. Norman, M. Je, M. Raja, R. Peh, and M. Dempsey. "A Reliable Handwash Detector for Automated Hand Hygiene Documentation and Reminder System in Hospitals". In: *The 15th International Conference on Biomedical Engineering*. Vol. 43. IFMBE Proceedings. Springer International Publishing, 2014, pp. 892–895. ISBN: 978-3-319-02912-2.

[74] D. Llorca, I. Parra, M. Sotelo, and G. Lacey. "A vision-based system for automatic hand washing quality assessment". In: *Machine Vision and Applications* 22.2 (2011), pp. 219–234.

[75] T. Palmore and D. Henderson. "Big brother is washing... video surveillance for hand hygiene adherence, through the lenses of efficacy and privacy". In: *Clinical infectious diseases* (2011), cir781.

[76] F. Shahmohammadi, A. Hosseini, C. E. King, and M. Sarrafzadeh. "Smartwatch based activity recognition using active learning". In: *2017 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE. 2017, pp. 321–329.

[77] V. Galluzzi, T. Herman, and P. Polgreen. "Hand hygiene duration and technique recognition using wrist-worn sensors". In: *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM. 2015, pp. 106–117.

[78] H. Li, S. Chawla, R. Li, S. Jain, G. D. Abowd, T. Starner, C. Zhang, and T. Plötz. "Wristwash: towards automatic handwashing assessment using a wrist-

worn device". In: *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. ACM. 2018, pp. 132–139.

[79] *World Health Organization (WHO) Guidelineson Hand Hygiene in Health Care*. http://whqlibdoc.who.int/hq/2009/who_ier_psp_2009.07_eng.pdf. Last Accessed: 2014-10-07.

[80] D. Hendrycks and K. Gimpel. "A baseline for detecting misclassified and out-of-distribution examples in neural networks". In: *arXiv preprint arXiv:1610.02136* (2016).

[81] K. Lee, K. Lee, H. Lee, and J. Shin. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7167–7177.

[82] A. Krizhevsky, G. Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. Citeseer, 2009.

[83] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 248–255.

[84] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop". In: *arXiv preprint arXiv:1506.03365* (2015).

[85] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. "Reading digits in natural images with unsupervised feature learning". In: (2011).

[86] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner. "The operational global icosahedral–hexagonal gridpoint model GME: description and high-resolution tests". In: *Monthly Weather Review* 130.2 (2002), pp. 319–338.

[87] D. Majewski, D. Liermann, P. Prohl, B. Ritter, M. Buchhold, T. Hanisch, G. Paul, W. Wergen, and J. Baumgardner. “The new global icosahedral–hexagonal grid point model GME of the Deutscher Wetterdienst”. In: *Proc. ECMWF Seminar on Recent Developments in Numerical Methods for Atmospheric Modelling, ECMWF, Reading, United Kingdom.* Citeseer. 1998.

[88] L. v. d. Maaten and G. Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[89] L. Liu, C. Karatas, H. Li, S. Tan, M. Gruteser, J. Yang, Y. Chen, and R. P. Martin. “Toward detection of unsafe driving with wearables”. In: *Proceedings of the 2015 workshop on Wearable Systems and Applications.* ACM. 2015, pp. 27–32.

[90] D. Pittet, B. Allegranzi, and J. Boyce. “The World Health Organization guidelines on hand hygiene in health care and their consensus recommendations”. In: *Infection Control & Hospital Epidemiology* 30.07 (2009), pp. 611–622.

[91] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff. “Simultaneous localization and recognition of dynamic hand gestures”. In: *Application of Computer Vision, 2005. WACV/MOTIONS’05 Volume 1. Seventh IEEE Workshops on.* Vol. 2. IEEE. 2005, pp. 254–260.

[92] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis. “Risq: Recognizing smoking gestures with inertial sensors on a wristband”. In: *Proceedings of the 12th annual international conference on Mobile systems, applications, and services.* ACM. 2014, pp. 149–161.

[93] A. W. Armstrong, A. J. Watson, M. Makredes, J. E. Frangos, A. B. Kimball, and J. C. Kvedar. “Text-message reminders to improve sunscreen use:

a randomized, controlled trial using electronic monitoring". In: *Archives of dermatology* 145.11 (2009), pp. 1230–1236.

[94] E. O. Kharbanda, M. S. Stockwell, H. W. Fox, and V. I. Rickert. "Text4Health: a qualitative evaluation of parental readiness for text message immunization reminders". In: *American journal of public health* 99.12 (2009), pp. 2176–2178.

[95] M. Y. Hou, S. Hurwitz, E. Kavanagh, J. Fortin, and A. B. Goldberg. "Using daily text-message reminders to improve adherence with oral contraceptives: a randomized controlled trial". In: *Obstetrics & Gynecology* 116.3 (2010), pp. 633–640.

[96] *Android App Store*. `https://play.google.com/store/apps?hl=en`.

[97] *Apple App Store*. `https://itunes.apple.com/us/genre/ios/id36?mt=8`.

[98] K. Stawarz, A. L. Cox, and A. Blandford. "Don't forget your pill!: Designing effective medication reminder apps that support users' daily routines". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2014, pp. 2269–2278.

[99] M.-Y. Wang, P. Tsai, J. W.-S. Liu, and J. K. Zao. "Wedjat: a mobile phone based medicine in-take reminder and monitor". In: *Bioinformatics and Bio-Engineering, 2009. BIBE'09. Ninth IEEE International Conference on*. IEEE. 2009, pp. 423–430.

[100] J. M. Silva, A. Mouttham, and A. El Saddik. "UbiMeds: a mobile application to improve accessibility and support medication adherence". In: *Proceedings of the 1st ACM SIGMM international workshop on Media studies and implementations that help improving access to disabled users*. ACM. 2009, pp. 71–78.

[101] E. Årsand, M. Muzny, M. Bradway, J. Muzik, and G. Hartvigsen. "Performance of the first combined smartwatch and smartphone diabetes diary application study". In: *Journal of diabetes science and technology* 9.3 (2015), pp. 556–563.

[102] V. Sharma, K. Mankodiya, F. De La Torre, A. Zhang, N. Ryan, T. G. Ton, R. Gandhi, and S. Jain. "SPARK: personalized parkinson disease interventions through synergy between a smartphone and a smartwatch". In: *Design, User Experience, and Usability. User Experience Design for Everyday Life Applications and Services*. Springer, 2014, pp. 103–114.

[103] F. Sailer, M. Pobiruchin, M. Wiesner, and G. Meixner. "An Approach to Improve Medication Adherence by Smart Watches". In: *Studies in health technology and informatics* 210 (2015), p. 956.

[104] *Medisafe Meds & Pill Reminder*. `https://play.google.com/store/apps/details?id=com.medisafe.android.client&hl=en`.

[105] M. Shoaib, H. Scholten, P. J. Havinga, and O. D. Incel. "A hierarchical lazy smoking detection algorithm using smartwatch sensors". In: *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE. 2016, pp. 1–6.

[106] G. M. Weiss, K. Yoneda, and T. Hayajneh. "Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living". In: *IEEE Access* 7 (2019), pp. 133190–133202.

[107] S. Mohammed and W. Gomaa. *HAD-AW Data-set Benchmark For Human Activity Recognition Using Apple Watch*. Mar. 2018. DOI: `10.13140/RG.2.2.18186.57280`.

[108] *G-sensor Logger*. https://play.google.com/store/apps/details?id=com.peterhohsy.gsensor_debug&hl=en_US. last accessed Dec 28, 2019. 2019.

[109] *Sensor Log*. https://play.google.com/store/apps/details?id=com.hfalan.activitylog&hl=en_US. last accessed Dec 28, 2019. 2019.

[110] *Sensor Sense Toolbox*. https://play.google.com/store/apps/details?id=com.kristofjannes.sensorsense&hl=en_US. last accessed Dec 28, 2019. 2019.

[111] *Sensor Data Logger*. https://play.google.com/store/apps/details?id=net.steppschuh.sensordatalogger&hl=en_US. last accessed Dec 28, 2019. 2019.

[112] Y. Vaizman, K. Ellis, G. Lanckriet, and N. Weibel. "Extrasensory app: Data collection in-the-wild with rich user interface to self-report behavior". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 554.

[113] D. Ginelli, D. Micucci, M. Mobilio, and P. Napoletano. "UniMiB AAL: An Android Sensor Data Acquisition and Labeling Suite". In: *Applied Sciences* 8.8 (2018), p. 1265.

[114] *Swear*. https://play.google.com/store/apps/details?id=uva.swear&hl=en_US. last accessed Dec 28, 2019. 2019.

[115] J. Upadhyay, O. Farr, N. Perakakis, W. Ghaly, and C. Mantzoros. "Obesity as a disease". In: *Medical Clinics* 102.1 (2018), pp. 13–33.

[116] *Health Risks of Being Overweight*. https://www.niddk.nih.gov/health-information/weight-management/health-risks-overweight. 2019.

[117] L. A. Lytle, M. O. Hearst, J. Fulkerson, D. M. Murray, B. Martinson, E. Klein, K. Pasch, and A. Samuelson. "Examining the relationships between family meal practices, family stressors, and the weight of youth in the family". In: *Annals of Behavioral Medicine* 41.3 (2010), pp. 353–362.

[118] K. N. Boutelle, A. S. Birnbaum, L. A. Lytle, D. M. Murray, and M. Story. "Associations between perceived family meal environment and parent intake of fruit, vegetables, and fat". In: *Journal of nutrition education and behavior* 35.1 (2003), pp. 24–29.

[119] E. Moens, C. Braet, G. Bosmans, and Y. Rosseel. "Unfavourable family characteristics and their associations with childhood obesity: a cross-sectional study". In: *European Eating Disorders Review: The Professional Journal of the Eating Disorders Association* 17.4 (2009), pp. 315–323.

[120] *Short Dietary Assessment Instruments*. `https://epi.grants.cancer.gov/diet/screeners/`. 2019.

[121] N. Khanna, C. J. Boushey, D. Kerr, M. Okos, D. S. Ebert, and E. J. Delp. "An overview of the technology assisted dietary assessment project at Purdue University". In: *Multimedia (ISM), 2010 IEEE International Symposium on*. IEEE. 2010, pp. 290–295.

[122] G. X. Ayala, B. Baquero, E. M. Arredondo, N. Campbell, S. Larios, and J. P. Elder. "Association between family variables and Mexican American children's dietary behaviors". In: *Journal of Nutrition Education and Behavior* 39.2 (2007), pp. 62–69.

[123] S. Munsch, K. Hasenboehler, T. Michael, A. H. Meyer, B. Roth, E. Biedert, and J. Margraf. "Restrained eating in overweight children: Does eating style run in families?" In: *International journal of pediatric obesity* 2.2 (2007), pp. 97–103.

[124] M. A. Pachucki, P. F. Jacques, and N. A. Christakis. "Social network concordance in food choice among spouses, friends, and siblings". In: *American journal of public health* 101.11 (2011), pp. 2170–2177.

[125] A. A. Gorin, R. R. Wing, J. L. Fava, J. M. Jakicic, R. Jeffery, D. S. West, K. Brelje, and V. G. DiLillo. "Weight loss treatment influences untreated spouses and the home environment: evidence of a ripple effect". In: *International Journal of Obesity* 32.11 (2008), p. 1678.

[126] *Estimote beacon*. `https://estimote.com/`. 2019.

[127] *Sony SmartWatch 3*. `https://www.sonymobile.com/global-en/products/smart-products/smartwatch-3-swr50`. 2019.

[128] M. Ma, R. Alam, B. Bell, K. de la Haye, D. Spruijt-Metz, J. Lach, and J. Stankovic. "Mˆ 2G: A Monitor of Monitoring Systems with Ground Truth Validation Features for Research-Oriented Residential Applications". In: *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE. 2017, pp. 10–18.

[129] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy. "Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition." In: *Ijcai*. Vol. 15. 2015, pp. 3995–4001.

[130] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao. "Time series classification using multi-channels deep convolutional neural networks". In: *International Conference on Web-Age Information Management*. Springer. 2014, pp. 298–310.

[131] M. A. S. Mondol and J. A. Stankovic. "Harmony: A hand wash monitoring and reminder system using smart watches". In: *12th EAI International Conference*

on Mobile and Ubiquitous Systems: Computing, Networking and Services. 2015, pp. 11–20.

[132] S. Shiffman, A. A. Stone, and M. R. Hufford. "Ecological momentary assessment". In: *Annu. Rev. Clin. Psychol.* 4 (2008), pp. 1–32.

[133] J. Laurent, S. J. Catanzaro, T. E. Joiner Jr, K. D. Rudolph, K. I. Potter, S. Lambert, L. Osborne, and T. Gathright. "A measure of positive and negative affect for children: Scale development and preliminary validation." In: *Psychological assessment* 11.3 (1999), p. 326.

[134] S. Cohen, T. Kamarck, and R. Mermelstein. "A global measure of perceived stress". In: *Journal of health and social behavior* (1983), pp. 385–396.

[135] A. V. Cardello, H. G. Schutz, L. L. Lesher, and E. Merrill. "Development and testing of a labeled magnitude scale of perceived satiety". In: *Appetite* 44.1 (2005), pp. 1–13.

[136] *ChronoViz*. `http://chronoviz.com/`. 2019.

[137] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[138] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song. "E-gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices". In: *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2011, pp. 260–273.

[139] K. Katevas, H. Haddadi, and L. Tokarchuk. "Sensingkit: Evaluating the sensor power consumption in ios devices". In: *2016 12th International Conference on Intelligent Environments (IE)*. IEEE. 2016, pp. 222–225.

[140] V. Sandulescu, S. Andrews, D. Ellis, N. Bellotto, and O. M. Mozos. "Stress detection using wearable physiological sensors". In: *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer. 2015, pp. 526–532.

[141] A. Zenonos, A. Khan, G. Kalogridis, S. Vatsikas, T. Lewis, and M. Sooriyabandara. "HealthyOffice: Mood recognition at work using smartphones and wearable sensors". In: *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. IEEE. 2016, pp. 1–6.

[142] A. S. Mondol, I. A. Emi, and J. A. Stankovic. "MedRem: An interactive medication reminder and tracking system on wrist devices". In: *2016 IEEE Wireless Health (WH)*. IEEE. 2016, pp. 1–8.

[143] B. Allegranzi and D. Pittet. "Role of hand hygiene in healthcare-associated infection prevention". In: *Journal of Hospital Infection* 73.4 (2009), pp. 305–315.

[144] D. Pittet, B. Allegranzi, H. Sax, S. Dharan, C. L. Pessoa-Silva, L. Donaldson, and J. Boyce. "Evidence-based model for hand transmission during patient care and the role of improved practices". In: *The Lancet Infectious Diseases* 6.10 (2006), pp. 641–652.

[145] B. Simmons, J. Bryant, K. Neiman, L. Spencer, and K. Arheart. "The role of handwashing in prevention of endemic intensive care unit infections". In: *Infection Control and Hospital Epidemiology* (1990), pp. 589–594.

[146] *CDC HAI Prevalence Survey*. `http://www.cdc.gov/hai/surveillance/`. Last Accessed: 2014-10-07.

[147] R. Scott. “The direct medical costs of healthcare-associated infections in US hospitals and the benefits of prevention”. In: (2009).

[148] *World Health Organization (WHO)Fact Sheet on Healthcare Associated Infections*. http://www.who.int/gpsc/country_work/gpsc_ccisc_fact_sheet_en.pdf. Last Accessed: 2014-10-07.

[149] M. Ross and J. Guzewich. “Evaluation of risks related to microbiological contamination of ready-to-eat food by food preparation workers and the effectiveness of interventions to minimize those risks”. In: *FDA White Paper, FDA, CFSAN* (1999).

[150] D. Pittet, A. Simon, S. Hugonnet, C. Pessoa-Silva, V. Sauvan, and T. Perneger. “Hand hygiene among physicians: performance, beliefs, and perceptions”. In: *Annals of internal medicine* 141.1 (2004), pp. 1–8.

[151] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. “The mahalanobis distance”. In: *Chemometrics and intelligent laboratory systems* 50.1 (2000), pp. 1–18.

[152] A. M. Sabatini. “Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing”. In: *IEEE Transactions on Biomedical Engineering* 53.7 (2006), pp. 1346–1356.

[153] *Android Sensors*. https://developer.android.com/reference/android/hardware/Sensor.html. Last accessed: 2017-03-30. 2017.

[154] J. P. Zacny and M. L. Stitzer. “Human smoking patterns”. In: *Smoking and tobacco control monograph* 7 (1996), pp. 151–60.

[155] F. Alharbi and K. Farrahi. “A Convolutional Neural Network for Smoking Activity Recognition”. In: *2018 IEEE 20th International Conference on e-*

Health Networking, Applications and Services (Healthcom). IEEE. 2018, pp. 1–6.

[156] *Rehabilitation Exercises*. http://orthoinfo.org/topic.cfm?topic=A00672. Last accessed: 2017-05-03. 2017.

[157] G. Chen, C. Parada, and G. Heigold. "Small-footprint keyword spotting using deep neural networks". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2014, pp. 4087–4091.

[158] M. Sun, A. Raju, G. Tucker, S. Panchapagesan, G. Fu, A. Mandal, S. Matsoukas, N. Strom, and S. Vitaladevuni. "Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting". In: *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2016, pp. 474–480.

[159] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni. "Model Compression Applied to Small-Footprint Keyword Spotting." In: *INTERSPEECH*. 2016, pp. 1878–1882.

[160] Y. Zhang, N. Suda, L. Lai, and V. Chandra. "Hello edge: Keyword spotting on microcontrollers". In: *arXiv preprint arXiv:1711.07128* (2017).

[161] Y. Zhang, W. Chan, and N. Jaitly. "Very deep convolutional networks for end-to-end speech recognition". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 4845–4849.

[162] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin". In: *International conference on machine learning*. 2016, pp. 173–182.

[163] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. "Attention-based models for speech recognition". In: *Advances in neural information processing systems*. 2015, pp. 577–585.

[164] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, et al. "State-of-the-art speech recognition with sequence-to-sequence models". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 4774–4778.

[165] E. Sabaté. *Adherence to long-term therapies: evidence for action*. World Health Organization, 2003.

[166] A. Bozek and J. Jarzab. "Adherence to asthma therapy in elderly patients". In: *Journal of Asthma* 47.2 (2010), pp. 162–165.

[167] P. Latry, M. Pinet, A. Labat, J.-P. Magand, C. Peter, P. Robinson, K. Martin-Latry, and M. Molimard. "Adherence to anti-inflammatory treatment for asthma in clinical practice in France". In: *Clinical therapeutics* 30 (2008), pp. 1058–1068.

[168] L. Osterberg and T. Blaschke. "Adherence to medication". In: *New England Journal of Medicine* 353.5 (2005), pp. 487–497.

[169] J. D. Smith and D. Oakley. "Why Do Women Miss Oral Contraceptive Pills? An Analysis of Women's Self-Described Reasons for Missed Pills". In: *Journal of Midwifery & Women's Health* 50.5 (2005), pp. 380–385.

[170] M. Vervloet, A. J. Linn, J. C. van Weert, D. H. De Bakker, M. L. Bouvy, and L. Van Dijk. "The effectiveness of interventions using electronic reminders to improve adherence to chronic medication: a systematic review of the lit-

erature”. In: *Journal of the American Medical Informatics Association* 19.5 (2012), pp. 696–704.

[171] R. E. Sarabi, F. Sadoughi, R. J. Orak, K. Bahaadinbeigy, et al. “The Effectiveness of Mobile Phone Text Messaging in Improving Medication Adherence for Patients with Chronic Diseases: A Systematic Review”. In: *Iranian Red Crescent Medical Journal* In Press (2016).

[172] S. L. Spoelstra, C. W. Given, A. Sikorskii, C. K. Coursaris, A. Majumder, T. DeKoekkoek, M. Schueller, and B. A. Given. “Feasibility of a Text Messaging Intervention to Promote Self-Management for Patients Prescribed Oral Anticancer Agents.” In: *Oncology nursing forum*. Vol. 42. 6. 2015, pp. 647–657.

[173] *Amazon Alexa*. `https://en.wikipedia.org/wiki/Amazon_Alexa`. 2019.

[174] *Google Home*. `https://en.wikipedia.org/wiki/Google_Home`. 2019.

[175] *Apple Siri*. `https://en.wikipedia.org/wiki/Siri`. 2019.

[176] *Android Sensors*. `https://developer.android.com/reference/android/hardware/Sensor`. 2018.

[177] *Android Sensor manger*. `https://developer.android.com/reference/android/hardware/SensorManager`. 2018.