HIGH-PRECISION AND PERSONALIZED WEARABLE SENSING SYSTEMS FOR HEALTHCARE APPLICATIONS

By

Linlin Tu

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2022

ABSTRACT

HIGH-PRECISION AND PERSONALIZED WEARABLE SENSING SYSTEMS FOR HEALTHCARE APPLICATIONS

By

Linlin Tu

The cyber-physical system (CPS) has been discussed and studied extensively since 2010. It provides various solutions for monitoring the user's physical and psychological health states, enhancing the user's experience, and improving the lifestyle. A variety of mobile internet devices with built-in sensors, such as accelerators, cameras, PPG sensors, pressure sensors, and the microphone, can be leveraged to build mobile cyber-physical applications that collected sensing data from the real world, had data processed, communicated to the internet services and transformed into behavioral and physiological models. The detected results can be used as feedback to help the user understand his/her behavior, improve the lifestyle, or avoid danger. They can also be delivered to therapists to facilitate their diagnose.

Designing CPS for health monitoring is challenging due to multiple factors. First of all, the high estimation accuracy is necessary for health monitoring. However, some systems suffer irregular noise. For example, PPG sensors for cardiac health state monitoring are extremely vulnerable to motion noise. Second, to include human in the loop, health monitoring systems are required to be user-friendly. However, some systems involve cumbersome equipment for a long time of data collection, which is not feasible for daily monitoring. Most importantly, large-scale high-level health-related monitoring systems, such as the systems for human activity recognition, require high accuracy and communication efficiency. However, with users' raw data uploading to the server, centralized learning fails to protect user's private information and is communication-inefficient.

The research introduced in this dissertation addressed the above three significant challenges in developing health-related monitoring systems. We build a lightweight system for accurate heart rate measurement during exercise, design a smart in-home breathing training

system with bio-Feedback via virtual reality (VR) game, and propose federated learning via dynamic layer sharing for human activity recognition.

Thanks to my family, who always supports me.
Thanks to my advisor, who always leads me.
Thanks to my friend, who shares the laughter with me.
Wish everyone all over the world a healthy and happy life,
and tomorrow will be better.

ACKNOWLEDGEMENTS

Here, I would like to express my deepest gratitude to all – my family, my advisor, my friends, and everyone who helps me in this academic career.

None of my achievement could be possible without the support from my family. Since I was 18, I was away from my hometown. Although they were far away, they always showed their hands and cheered me up when I was down.

Thanks, Dr. Guoliang Xing, my dear advisor. He offered me a chance to explore the academic world. He guided me from the development of a small App to the processing with millions of data points. His insight helps me solving numerous challenging problems in my research.

Thanks, Dr. Jiayu Zhou, who provided valuable suggestions that inspire me to refine my design and give more reliable and useful information for the study of smart system.

Thanks, my friends, old ones and new ones. Life of a researcher is sometimes lonely, but I am so lucky to have all of you on my way. We have worked together, played together, and laughed together. One day in the future, we will get back together and share more stories of our colorful life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The confluence of innovations in sensor development, the emergence of the Internet of Things (IoT), and the ubiquity of mobile devices has given us a variety of new sensors and systems that can be utilized to build mobile cyber-physical applications for the improvement of personal health, well-being, and fitness. Such devices are emerging regularly and address a diverse set of applications, ranging from physical activity, endurance sports, and resistance training to sleep monitoring, mindfulness practice, posture monitoring, weight management, breathing techniques, cardiac health status [128, 80, 113]. Some examples of recent research conducted in the area. Xiao Sun et al. designed a smartphone-based application, which leverages the built-in microphone to unobtrusively detect acoustic events related to respiration symptoms [117]. Shahriar et al. proposed using a sensor-equipped earphone to detect the user's heart rate and built an automated music recommendation system to help the user maintain a target heart rate [91].

When designing state-of-the-art cyber-physical applications for health-related applications, several important considerations must be addressed. The first issue is what optimizations of the measurement technologies are necessary to improve estimation accuracy. Take the detection of cardiac information as an example. In this application, heart rate for fitness is commonly tracked using wrist-worn wearables. However, a major drawback of using these sensors is that significant noise caused by intensive wrist movements can corrupt measurements. As a result, complex filtering algorithms and designs must be created and tailored to each application. To include humans in the loop, a second issue that must be considered is the user-friendliness of applications. Attention must be paid to designing an unobtrusive sensing method that is easy-to-operate and convenient such that users can access the healthcare applications frequently. One example of problems arising from this issue is in the design of RSA-BT (Respiratory Sinus Arrhythmia biofeedback-based Breathing Training), a cardio-respiratory intervention that has been commonly used as a complementary treatment to respiratory diseases and an exercise to help manage stress and anxiety. Despite its health benefits, RSA-BT today still relies on in-person sessions and cumbersome sensing devices in a clinical setting, limiting its accessibility. Furthermore, to design smart systems for large-scale applications, we need to solve several limitations of centralized learning. With users' raw data uploading to the server, centralized learning fails to protect users' private information and is communication-inefficient. Distributed learning [49, 132] has been proposed for large-scale smart systems. The distributed learning paradigm only requires users to upload their model weights for collaborative learning, avoiding sharing users' raw data during the learning process. Several Federated learning systems for Human activity recognition (HAR) [49, 22] have been developed to enable continuous monitoring of human behaviors without sharing users' raw data. However, standard federated learning limits the performance of smart systems, as the accuracy of models learned in this approach can be largely influenced by the diversity of users.

Our research improved the precision of smartwatch-based cardiac measurement, enabled the wrist-band-based unobtrusive and continuous logging of users' cardiac and respiratory information to make the intervention accessible daily in-home, and finally, proposed to use federated learning to train the deep model for HAR. Specifically, first, a lightweight system, Fitbeat, was developed to enable accurate heart rate tracking on wrist-type during intensive exercise [123]. After obtaining accurate physiological signals from the system, we include humans in the loop and design the BreathCoach — a smart and unobtrusive system that enables in-home RSA biofeedback-based Breathing Training (RSA-BT) using smartphone-based virtual reality in conjunction with sensors on a smartwatch [121]. Finally, we propose FedDL, a novel federated learning system for the large-scale HAR, that can dynamically capture the underlying user relationships and apply them to learn personalized learning models for different users [124].

The rest of the thesis is organized as follows: Chapter 2 - the research on high-precision heart rate tracking; Chapter 3 - the research about BreathCoach, an in-home RSA-based Breathing Training system; Chapter 4 - the latest proposed research about Human Activity Recognition using personalized federated deep learning. The last chapter presents the conclusion.

CHAPTER 2

HIGH-PRECISION HEART RATE TRACKING

Tracking heart rate for fitness using wrist-type wearables is challenging, because of the significant noise caused by intensive wrist movements. This chapter presents FitBeat – a lightweight system that enables accurate heart rate tracking on wrist-type wearables during intensive exercises. Unlike existing approaches that rely on computation-intensive signal processing, FitBeat integrates and augments standard filter and spectral analysis tool, which achieves comparable accuracy while significantly reducing computational overhead. FitBeat integrates contact sensing, motion sensing and simple spectral analysis algorithms to suppress various error sources. This chapter is adapted from a publication [123]. The author of the dissertation is the first author of the original work. "We" in this chapter refers to the author of the original publication. This work contains the App design on Android devices. The author recruited all the subjects, then collected and processed the data and the ground truth.

2.1 Background

Recent years have witnessed the proliferation of wrist-type smart wearables. A desirable feature of these devices is tracking heart rate for fitness, which is essential for exercisers to monitor health conditions and control training loads. Wrist-type wearables typically employ photoplethysmogram (PPG) to measure heart rate. Specifically, a PPG sensor consists of a LED and a photo detector. The LED emits light, which is absorbed by blood flow when traveling through the tissue. The photo detector then measures the intensity of reflected light to sense periodic blood flow variation caused by cardiac cycle, which can be used to estimate heart rate.

However, tracking heart rate for fitness using wrist-type wearables poses several key challenges. First, since the capillary network around wrist is relatively sparse, PPG signals

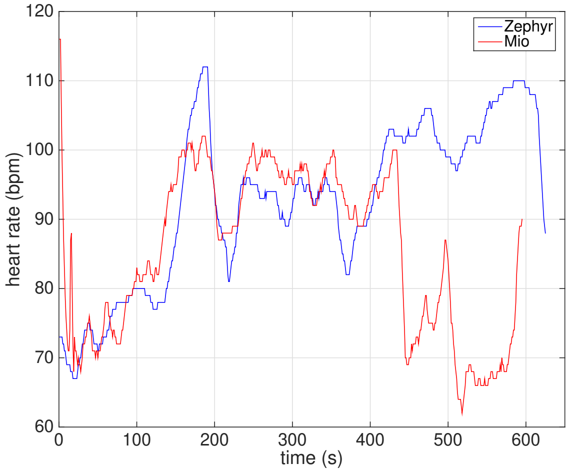


Figure 2.1: Measurement error of the built-in PPG sensor of Moto 360 while the subject is running.

observed by wrist-worn sensors are usually very weak, which makes the signal extremely vulnerable to noise. Second, during intensive exercise the subject's wrist muscle may flex frequently, resulting in an unstable contact between the subject's skin and the PPG sensor, which causes significant noise. Third, in addition to causing unstable contact, intensive wrist motion affects blood flow, which introduces additional noise that may severely degrade the accuracy of heart rate measurement, particularly when the motion-induced noise is overlapping with the desired signal in frequency domain. Our experiments show that popular wearable devices like Mio Alpha and Moto 360 suffer extremely poor performance when measuring heart rate in the presence of intensive wrist movements. For example, as shown in Fig. 2.1, when the subject is running, the heart rate estimation error of Mio Alpha can be as high as 50 beats per minute (bpm), compared with the ground truth measured using a motion-resistant ECG sensor. Similar results were observed on other popular wearable devices like Basis Peak and Fitbit Charge HR [69].

To improve the accuracy of heart rate measurement, numerous approaches have been proposed to remove motion-induced noise, including wavelet transformation, independent component analysis [65], moving average filter, adaptive noise cancellation [118], time frequency methods, and principle component analysis [97]. However, existing approaches are mainly designed for PPG sensors worn on fingertips [104], earlobes [94], or forehead [66]. They perform poorly when sensors are worn on the wrists, because noise caused by wrist motion is much more complex and stronger than those caused by fingers, ears and head. Although there exist a few methods for reducing noise caused by wrist movements, they rely on complex signal processing algorithms. For example, in addition to standard filtering and spectral analysis, TROIKA [135] relies on computation-intensive singular spectrum analysis and FOCUSS algorithm, which significantly increases overhead.

In this chapter, we present Fitbeat – a lightweight system that enables accurate heart rate tracking on wrist-type wearables during intensive exercises. Unlike existing approaches that rely on computation-intensive signal processing [110][38], FitBeat integrates and augments only standard filter and spectral analysis tool, which achieves comparable accuracy while significantly reducing computational overhead. To achieve this goal, FitBeat integrates contact sensing, motion sensing, and simple spectral analysis algorithm to suppress various error sources. Specifically, to remove noise caused by unstable contact between the subject's skin and the PPG sensor, FitBeat performs contact sensing, which measures the amplitude and variance of PPG signal to identify and remove distorted PPG signal samples. To reduce motion artifacts caused by complex and intensive wrist motions, FitBeat exploits accelerometer data to rebuild the waveform of motion-induced noise, and then subtracts it from PPG signal. To extract precise heart rate from raw PPG samples, FitBeat employs a simple pulse identification algorithm, which accurately identifies the spectral peak of heart rate by co-analyzing the spectrum of PPG signal and acceleration data. FitBeat is implemented on Moto 360 – a COTS smartwatch. We evaluate the performance of FitBeat for workouts of different intensities, including walking, running and riding. Experimental results involving 10 subjects show that the average error of FitBeat is around 4 bpm, which improves heart rate accuracy by 10x compared with the default heart rate tracker of Moto 360.

2.2 Related work

To improve the accuracy of heart rate measurement, several approaches have been proposed to remove motion-induced noise from PPG signal. Kim et al. in [65] propose to use independent components analysis (ICA) for reducing motion-induced noise. In ICA, PPG signals are modeled as the combination of PPG signals and motion artifacts.When applied to the contaminated PPG signal, ICA separates the clean PPG signal from noise components. However, ICA assumes that all signal components are mutually independent with each other, which is not true in PPG signal. For example, intensive wrist movements always affect the subject's cardiac activity, which implies that clean PPG signal is correlated with motion-induced noise. Besides, ICA relies on multiple PPG sensors, which are usually not available on COTS wearable devices.

Another approach to reducing motion-induced noise is adaptive noise cancellation (ANC) [130]. ANC estimates motion-induced noise components using acceleration data and then substracts the estimated noise from PPG signal. However, when the hand movements are irregular or the wristband is loosely attached to the subject's skin, the estimated noise may not be well correlated with the noise. Consequently, motion noise can not be removed completely.

There exist two classes of signal processing algorithms to extract heart rate from noise-reduced PPG signal, including moving window and spectral analysis [55]. Previous studies have shown that spectral analysis is more accurate than moving window. Specifically, spectral analysis algorithm estimates heart rate by analyzing the spectrum of PPG signal and then locating the largest spectral peak in the possible range of cardiac cycle. However, this algorithm performs poorly in the presence of residual motion-induced noise, because residual noise may cause multiple peaks around the frequency of cardiac cycle when PPG signal is noisy.

2.3 System Design

FitBeat is designed for accurate heart rate tracking using wrist-type wearables during intensive exercises. To achieve this goal, FitBeat addresses two key challenges. First, during intensive exercises, the subject's wrist muscle may flex frequently, which causes the band of the wearable to tighten and loosen, resulting in an unstable contact between the subject's skin and the PPG sensor that significantly distorts PPG signals. Second, in addition to causing unstable contacts, the wrist motion may affect blood flow, which introduces additional noise that interferes with heart rate measurements.

FitBeat addresses the above challenges by integrating contact sensing, motion sensing, and simple signal processing algorithm to suppress various error sources. The architecture of FitBeat is illustrated in Fig. 2.2. Specifically, FitBeat consists of three major components.

1. Based on the amplitude and variance of PPG signal, the *contact sensing* component continuously monitors the contact between the PPG sensor and the subject's skin, and removes those signal samples distorted by unstable contact.

2. The *noise reduction* component analyzes both PPG signal and accelerometer data to remove motion-induced noise. It exploits accelerometer data to rebuild the waveform of motion-induced noise based on an empirical model, and then subtracts the noise from PPG signal. The empirical model is refined using iterative adaptive filtering to improve accuracy.

3. The *pulse identification* component further reduces the residual motion-induced noise by co-analyzing the spectrum of PPG signal and accelerometer data, and then performs spectral analysis to accurately identify the pulse corresponding to the heart rate.

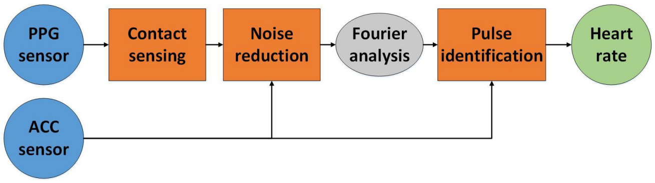


Figure 2.2: The signal processing pipeline of FitBeat.

2.3.1 Contact Sensing

When the subject is moving intensively, his/her wrist muscle may flex frequently, which may tighten and loosen the contact between the PPG sensor and the subject's skin. The impact on PPG signal is two-fold. First, when wrist flex loosens the contact, the PPG sensor will be exposed to an increased level of ambient light, which overwhelms the pulsatility of PPG signal that characterizes cardiac cycles. Second, when the contact varies frequently, a new pulsatile components will be imposed to the original PPG signal, which interferes with heart rate measurements.

To maintain accurate heart rate measurements in the presence of intensive wrist movements, FitBeat continuously senses the contact between the PPG sensor and the subject's skin, and removes those signal samples that are distorted by unstable contact. Specifically, FitBeat identifies distorted PPG signal samples based on their amplitudes and variances. When the PPG sensor is exposed to an increased level of ambient light due to a loosened contact, the amplitude of the PPG signal will experience a disruptive increase. In addition, when the contact between the PPG sensor and the subject's skin is unstable, the PPG signal will exhibit a large variance. For example, Fig. 2.3 shows the amplitude and variance of PPG signals when the subject performs intensive wrist movements. It's obvious that both the amplitude and variance of PPG signals are much larger in the time periods from 30s to 130s and from 280s to 580s, which correspond to the time period when the subject keeps moving his hands.

Based on the above observations, FitBeat removes a PPG signal sample if its amplitude is higher than a pre-defined threshold, and excludes signal samples from heart rate derivation

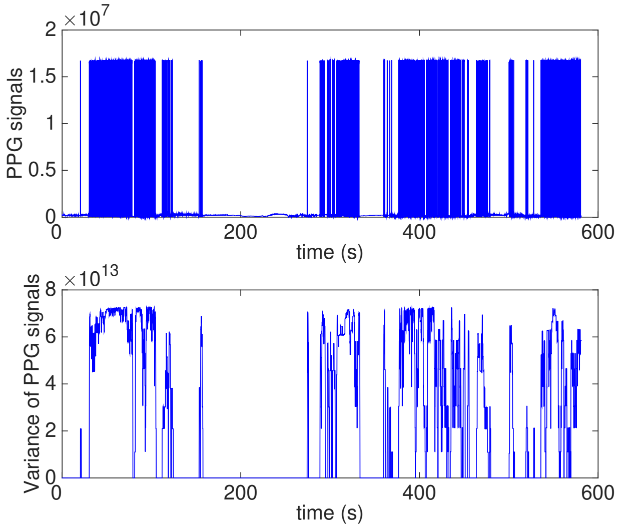


Figure 2.3: The waveform of PPG signal and its variance. The PPG signal are recorded while subject performing wrist movements intensively

if the variance measured in a window is larger than a pre-defined threshold. We determine the thresholds of amplitude and variance based on empirical experiments. According to our measurements, the thresholds of amplitude and variance are set to 10^6 and 5×10^8, respectively.

2.3.2 Noise Reduction

During intensive exercises, the subject's motion may affect his/her blood flow around wrist, imposing another noise component into the original PPG signal. To address this problem, FitBeat exploits the accelerometer of wearable to sense the subject's motion, estimates motion-induced noise, and then subtracts the noise from the original PPG signal. The framework of the noise reduction component is illustrated in Fig. 2.5. Specifically, the input signal $ppg(i)$ is a combination of motion-induced noise $n(i)$ and the desired PPG signal $s(i)$ that characterizes cardiac cycles. The accelerometer data $acc(i)$ is used to derive $n'(i)$ as an approximate estimation of $n(i)$. The adaptive filter then iteratively optimizes its coefficients to improve the accuracy of $n'(i)$. The process typically converges in a few rounds. Finally,

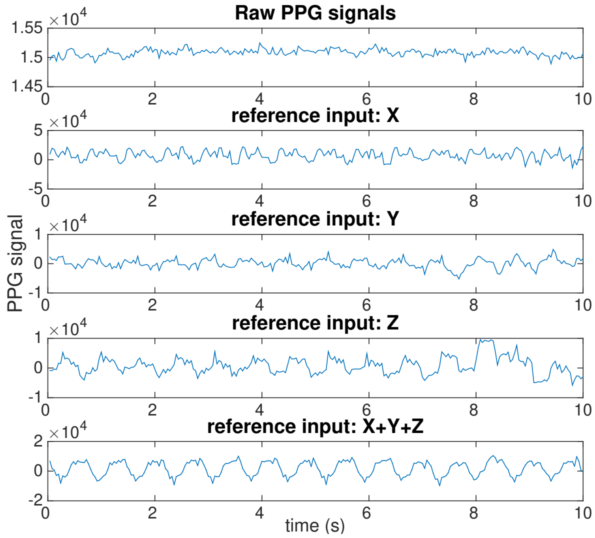


Figure 2.4: the waveform of motion-reduced PPG signal. The last four plots shown the result of adaptive noise cancellation with four different reference input, including acceleration from axis X, Y, Z and a linear summation, $(X + Y + Z)$.

the estimated $n'(i)$ is subtracted from $ppg(i)$ to suppress motion-induced noise.

To re-build the waveform of motion-induced noise, FitBeat models $n'(i)$ as a function of accelerometer data. Specifically, we derive the model based on extensive empirical measurements. First, we sample the X, Y, and Z axis of accelerometer and collect PPG signals for different subjects during typical workouts such as riding, running, and walking, etc. Then, we evaluate different polynomials consisting of X, Y, and Z to study their accuracy when re-building the waveform of motion-induced noise. Based on 20 groups of experiments, we find that the linear combination, i.e., $(X+Y+Z)$ yields the best accuracy. For example, Fig. 2.4 compares the PPG signal generated by the noise reduction component after subtracting noise waveforms modeled using different polynomials. As shown in the figure, the waveform of PPG signal is the smoothest when using $(X + Y + Z)$ to estimate motion-induced noise. We note that this result is different from previous studies [104][94][66], which show that motion-induced noise is best modeled using one axis of accelerometer data when the heart rate sensor is worn on the forehead, earlobe or finger of the subject. This is because, during exercises, the motion of the subject's wrist can be along any possible direction, which is

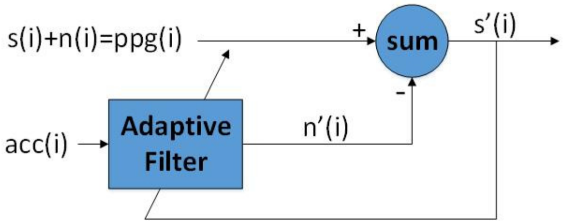


Figure 2.5: the flowchart of Adaptive Noise Cancellation.

much more complex than those of head, ears, and fingers. As a result, models that account for only one axis of accelerometer are not enough to accurately characterize noise caused by wrist movements.

Based on the above observation, we design the noise reduction component shown in Fig. 2.5 as follows.

$$\mathbf{acc}(i) = \mathbf{x}(i) + \mathbf{y}(i) + \mathbf{z}(i) \tag{2.1}$$

$$\mathbf{k}(i) = \frac{\lambda^{-1}\mathbf{P}(i-1)\mathbf{acc}(i)}{1 + \lambda^{-1}\mathbf{acc}^{H}(i)\mathbf{P}(i-1)\mathbf{acc}(i)} \tag{2.2}$$

$$n'(i) = \mathbf{w}^{T}(i)\mathbf{acc}(i) \tag{2.3}$$

$$s'(i) = ppg(i) - n'(i) \tag{2.4}$$

$$\mathbf{w}(i) = \mathbf{w}(i-1) + \mathbf{k}(i)s'(i) \tag{2.5}$$

$$\mathbf{P}(i) = \lambda^{-1}\mathbf{P}(i-1) - \lambda^{-1}\mathbf{k}(i)\mathbf{acc}^{H}(i)\mathbf{P}(i-1) \tag{2.6}$$

where,

- i denotes the current index of time window,
- $\mathbf{acc}(i)$ is the vector of buffered acceleration at step i,
- $\mathbf{P}(i)$ denotes the inversive correlation matrixe at step i,
- $\mathbf{k}(i)$ is the gain vector at step i,
- $\mathbf{w}(i)$ is the vector of filter tap at step i,

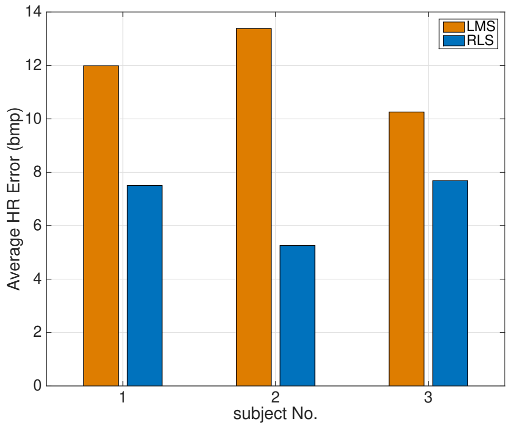


Figure 2.6: Average error of heart rate estimation using PPG signal processed by LMS and RLS filter.

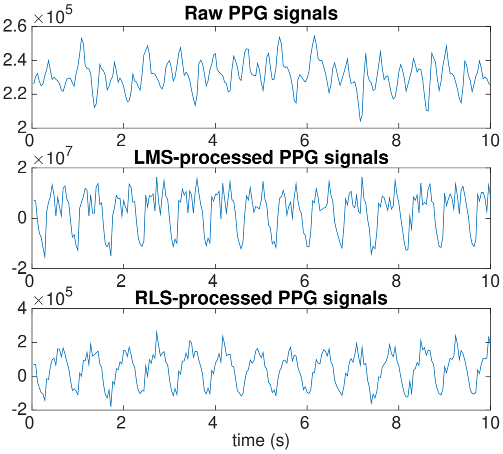


Figure 2.7: The waveform of PPG signal, including the raw PPG signal and the ones processed using LMS filter and RLS filter.

- $n'(i)$ is estimated noise, at step i,
- $s'(i)$ is the noise-reduced PPG signal at step i,
- $ppg(i)$ is the raw PPG signal at step i,
- λ denotes the forgetting factor.

acc, **P** and **w** are all column vectors of the same length.

To improve the accuracy of noise reduction, FitBeat needs to optimize the coefficients of adaptive filter $w(i)$. While previous heart rate monitoring systems typically employ the Least Mean Square (LMS) algorithm to approach this problem, our measurements find that Recursive Least Square (RLS) algorithm [102] performs better in the presence of intensive wrist movements. Fig. 2.6 shows the average estimation error for three subjects during a workout of 10-minute running. As shown in the figure, noise reduction using RLS-based adaptive filter is more accurate. Fig. 2.7 shows the waveform of PPG signal when filter contaminated PPG signals with LMS and RLS-based adaptive filter. As shown in the figure, the signal waveform generated by RLS is much smoother. Based on the above observations, FitBeat employs RLS algorithms to optimize adaptive filter coefficients, and set the forgetting factor of RLS filter to 0.98 based on empirical measurements.

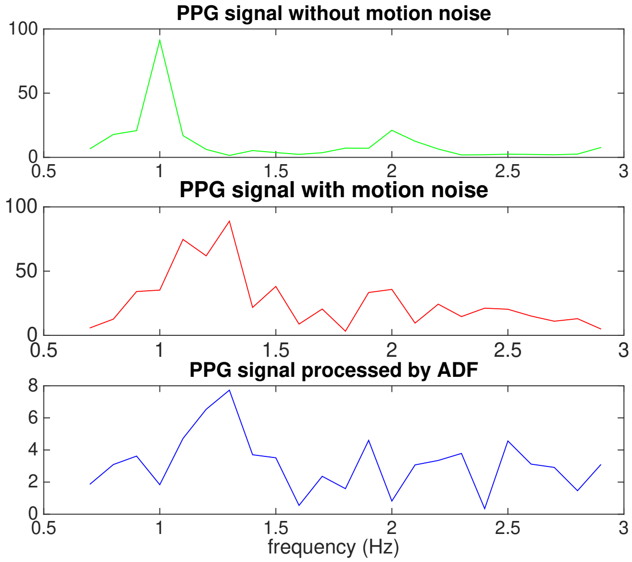


Figure 2.8: The spectrum of 10-second PPG signal, including the ground truth, the raw PPG signal and the noise-reduced one.

2.3.3 Pulse Identification

In the presence of intensive wrist movements, noise reduction cannot completely remove all motion-induced noise components. As an example, Fig. 2.8 compares the spectrum of ground truth, raw PPG signal, and noise-reduced PPG signal when the subject continuously moving his wrist. The ground truth is collected from the other wrist of the subject, which keeps still during measurement. As shown in the figure, in the spectrum of noise-reduced PPG signal, residual motion-induced noise causes multiple peaks in the segment from 0.8 Hz to 3.2 Hz, which significantly disturbs heart rate measurements.

To address the above problem, FitBeat employs a simple spectral analysis algorithm to suppress residual motion-induced noise, which allows it to accurately identify the pulse that corresponds to the heart rate. The basic idea is to co-analyze the spectrum of PPG signal and accelerometer data. Specifically, FitBeat first filters PPG signal and accelerometer data with a Savitzky-Golay (SG) filter to remove high-frequency noise, and then performs Fast Fourier Transform (FFT). To identify the pulse that corresponds to the heart rate, FitBeat performs spectral analysis following two steps.

1. Locate the highest peak between 0.8 Hz and 3.2 Hz in the spectrum of PPG signal. Denote the frequency of the located peak as f_p. If there is only one peak, return $60 \times f_p$ as the heart rate measurement.

2. If there exist multiple peaks, examine the spectrum of accelerometer data, and check if there is a peak at f_p. If not, return $60 \times f_p$ as the heart rate measurement. Otherwise, check the amplitude of the peak at f_p in the spectrum of accelerometer data T. If the amplitude is higher than a pre-defined threshold, remove the peak at f_p from the spectrum of PPG signal, and repeat step 1. Otherwise, return $60 \times f_p$ as a heart rate measurement.

The above algorithm iteratively cleans the PPG signal by removing spectral peaks caused by residual noise. To identify motion-induced peaks, we determine the threshold T based on empirical measurements, and set $T = 10000$ in FitBeat.

2.4 Evaluation

2.4.1 Experiment Settings

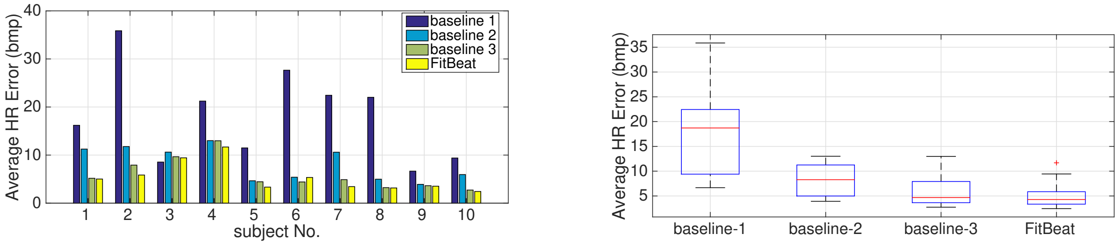
We evaluate FitBeat for typical workouts of different exercise intensities, and compare it with three baselines, including:

- Baseline-1: the default heart rate monitoring app of Moto 360.

- Baseline-2: a variant of FitBeat, where contact sensing and pulse identification are disabled during heart rate measurement.

- Baseline-3: another variant of FitBeat, where only contact sensing is disabled.

We compare FitBeat with baseline-2 and baseline-3 to study the effects of contact sensing and pulse identification. In addition, we employ Zephyr HxM BT – a strap with built-in ECG sensor – to collect ground truth.



((a)) Average estimation error.

((b)) The distribution of average estimation error.

Figure 2.9: Heart rate estimation while walking.

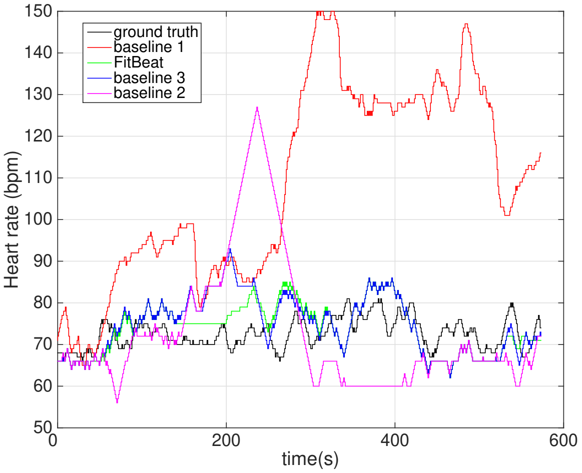


Figure 2.10: Heart rate estimation during a 10-minute walking.

To evaluate FitBeat, we recruited 10 subjects and collected data from different workouts including 10 walks, 3 runs and 3 rides. Our study along with its data collection procedure was approved by the Institutional Review Boards (IRB) at Michigan State University. All the subjects voluntarily agreed to help with data collection, and signed a consent form. In order to collect data, each subject used a smartwatch (Moto 360), a Bluetooth chest strap (Zephyr HxM BT), and a smartphone (Google Nexus 4) while doing exercise. During data collection, the PPG sensor and the accelerometer of Moto 360 are continuously sampled at 25 Hz. At the same time, the heart rate reported by the default app of Moto 360 is recorded at 1 Hz. To obtain ground-truth, we log the heart rate measured by Zephyr HxM BT, which uses ECG sensor that is resistant to body movements. The raw PPG signal and accelerometer data are then transferred to the smartphone for heart rate estimation.

We evaluate FitBeat based on two metrics, including Absolute Heart Rate Error (Err_{abs}), and Average Estimation Error (μ). Specifically, Err_{abs} is the absolute estimation error per minute, and μ is computed as the average of Err_{abs} in a 10-minute window. They are computed as:

$$Err_{abs}(i) = |BPM_{est}(i) - BPM_{true}(i)| \tag{2.7}$$

$$\mu = \frac{1}{N}\sum_{i=1}^{N} Err_{abs}(i) \tag{2.8}$$

where,

- $Err_{abs}(i)$ denotes the absolute estimation error in the i-th time window;
- N is the total number of estimation windows;
- $BPM_{est}(i)$ denotes the estimated heart rate in the i-th time window measured in beats per minute (bpm),
- $BPM_{true}(i)$ denotes the ECG-based heart rate measured in beats per minute (bpm), which is used as ground truth,

2.4.2 FitBeat Performance

In the following, we evaluate FitBeat under different levels of exercise intensity, including walking, running, and riding.

2.4.2.1 Walking

We first evaluate FitBeat while subjects are walking at normal speed. Fig. 2.9(a) shows the average estimation error μ for all subjects. Fig. 2.9(b) compares the distributions of μ for FitBeat and baselines. As shown in Fig. 2.9(b), when using FitBeat, μ ranges from 2.43 to 8.13 with a median of 4.27, outperforming all baselines.

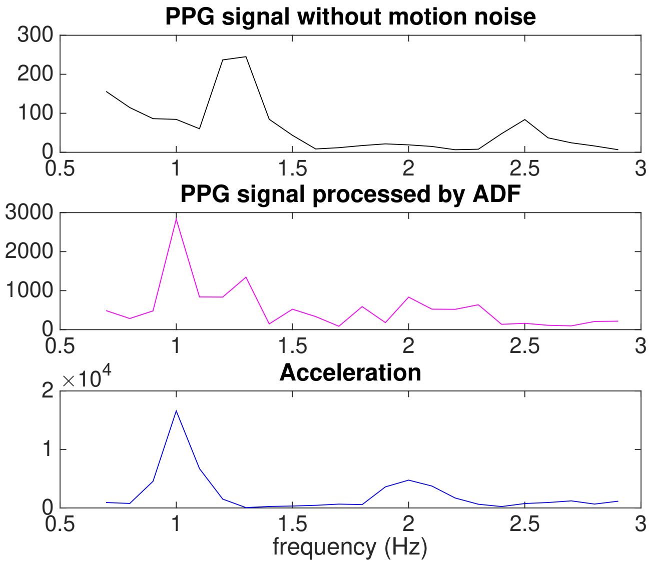


Figure 2.11: The spectrums of reference PPG signal, noise-reduced PPG signal, and accelerometer data.

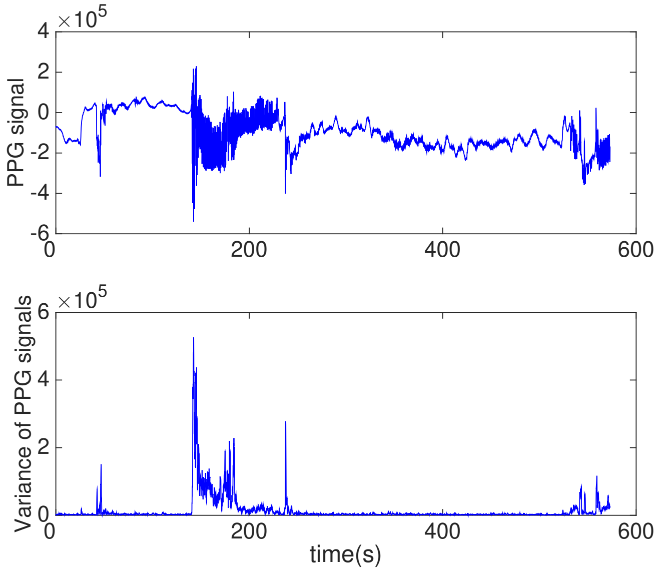


Figure 2.12: The waveform and variance of PPG signal.

To further study the performance of FitBeat, Fig. 2.10 shows the trace of estimated heart rate during a 10-minute walk for one subject. We observe that baseline-1 – which is the default heart rate app of Moto 360 – performs worst among all methods. Its estimation error can be 70 bpm, and is higher than 50 bpm for most of the time. We also observe that baseline-2 performs worse than FitBeat. This is because baseline-2 is more vulnerable

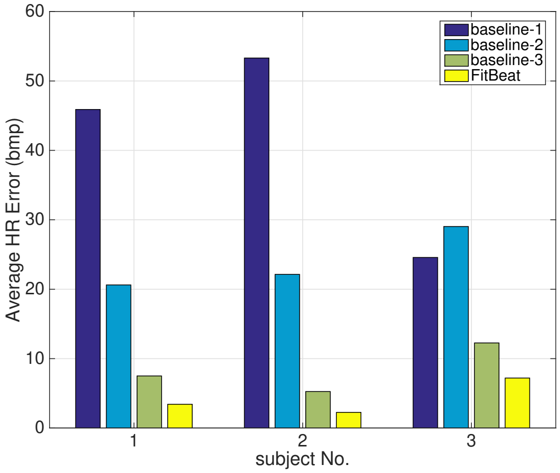
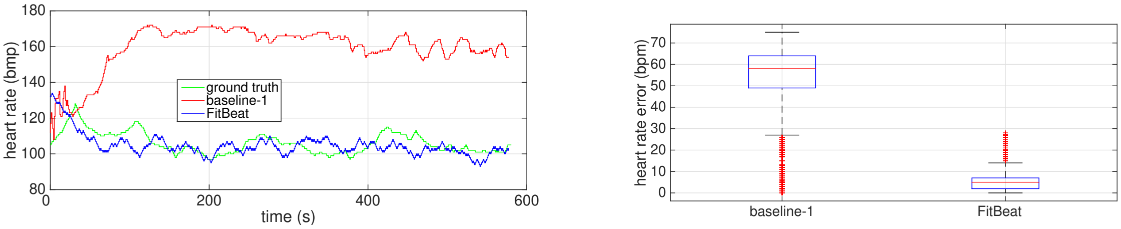


Figure 2.13: Average estimation error while running.

to residual motion-induced interference, as it disables pulse identification during spectral analysis. Specifically, Fig. 2.11 shows the spectrum of reference PPG signal, noise-reduced PPG signal, and accelerometer data. In the spectrum of reference PPG signal, peak corresponding to the heart rate is located at 1.3 Hz. While in the spectrum of motion-reduced PPG signal, the peak of maximum amplitude is located at 1 Hz, which is caused by residual motion-induced noise (as shown in the spectrum of accelerometer data). When using the peak at 1 Hz to estimate heart rate, baseline-2 results in an error of 18 bpm. In comparison, FitBeat uses the pulse identification algorithm to remove peaks caused by residual noise, which allows it to accurately identify the peak corresponds to the heart rate. Moreover, we observe that FitBeat outperforms baseline-3, which disables contact sensing during heart rate measurement. In particular, baseline-3 yields an estimation error of about 15 bpm in the time period between 150s and 240s. As shown in Fig. 2.12, the variance of PPG signal experiences a surge in this time period, which indicates an unstable contact between the subject's skin and the PPG sensor. With contact sensing, FitBeat is able to identify and exclude PPG signal samples corrupted by unstable contacts, which allows it to maintain accurate heart rate measurements.



((a)) Heart rate estimation.

((b)) Distribution of absolute estimation error.

Figure 2.14: Heart rate monitoring while running.

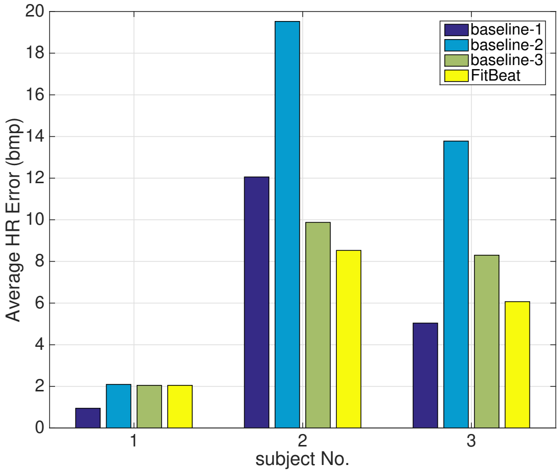
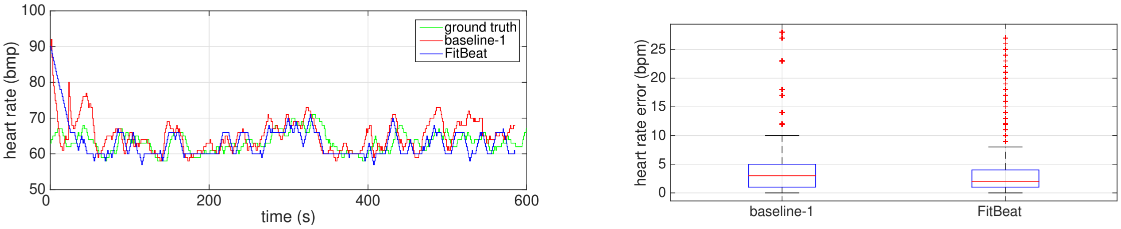


Figure 2.15: Average estimation error while riding.



((a)) Heart rate estimation

((b)) Distribution of absolute estimation error

Figure 2.16: Heart rate monitoring while riding.

2.4.2.2 Running

We now evaluate FitBeat while subjects are running in gyms or outdoors. Fig. 2.13 shows the average estimation error μ for three subjects during a workout of 10-minute running. We observe that FitBeat outperforms baseline-1 by a significant margin. In particular, the average estimation error of FitBeat is below 8 bpm for all subjects. Fig. 2.14(a) shows the trace of heart rate estimation for one subject. As shown in the figure, FitBeat is able to maintain accurate heart rate measurement consistently over time. Fig. 2.14(b) compares the distributions of Err_{abs} for FitBeat and baseline-1. For FitBeat, the median Err_{abs} is around 5 bpm, and the maximum Err_{abs} is below 10 bpm.

2.4.2.3 Riding

We then evaluate FitBeat when subjects are riding on spinning or outdoors. Fig. 2.15 shows the average estimation error μ for three subjects during a workout of 10-minute riding. Compared with the result shown in Fig. 2.13, μ is generally lower during riding, because riding involves less intensive wrist movements. In this case, FitBeat still maintains accurate heart rate measurements, but brings little accuracy improvement when compared with other baselines. For example, Fig. 2.16(a) shows the estimated heart rates for a 10-minute riding on spinning. As shown in the Fig. 2.16(b), the heart rate estimated by FitBeat is close to the ground truth, and the median of Err_{abs} is no more than 2 bpm.

2.5 Conclusion of Study

In this thesis, we present FitBeat – a lightweight system that uses wrist-worn PPG for accurate measurement during intensive exercises. To achieve this goal, FitBeat integrates contact sensing, motion sensing and lightweight signal processing algorithm to suppress various error sources. We implement FitBeat on a COTS smartwatch, and evaluate its performance under different levels of exercise intensity, including walking, running and riding. Experimental results from 10 objects show that FitBeat can accurately measure the heart rate during exercise.

CHAPTER 3

RESPIRATORY SINUS ARRHYTHMIA BIOFEEDBACK-BASED BREATHING TRAINING

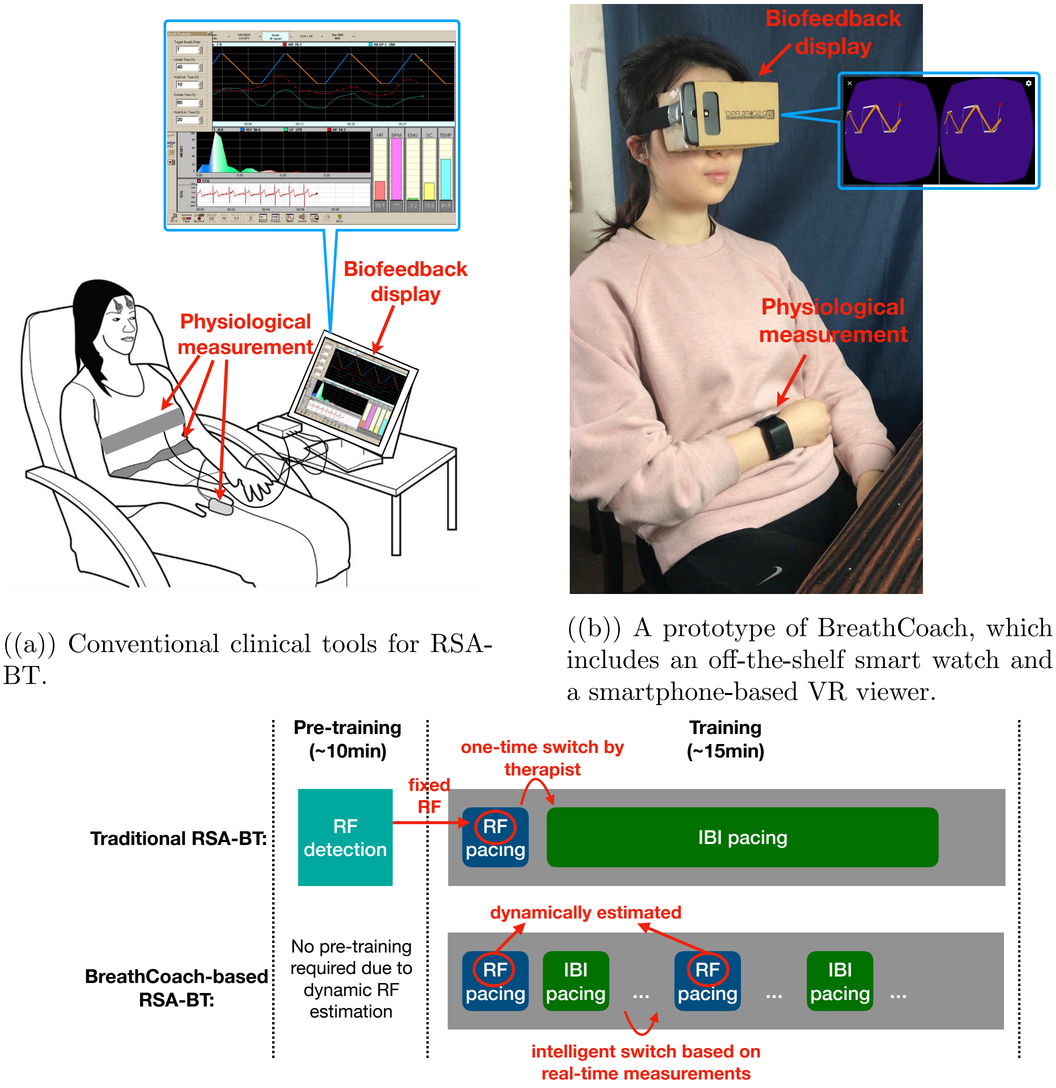
RSA-BT (Respiratory Sinus Arrhythmia biofeedback-based Breathing Training) is a cardio-respiratory intervention that has been commonly used as a complementary treatment to respiratory diseases, as well as an exercise to help manage stress and anxiety. Despite its health benefits, today's RSA-BT still relies on in-person sessions and cumbersome sensing devices in a clinical setting, which limits its accessibility. In this chapter, we introduce BreathCoach, a smart and unobtrusive system that enables effective in-home RSA-BT using sensors on a smartwatch and smartphone-based VR. This chapter is adapted from a publication [122]. The author of the dissertation is the first author of the original work. ”We” in this chapter refers to the author of the original publication. This work contains the software design on Android devices and the algorithm design in Matlab. The author recruited all the subjects, then collected and processed the data and the ground truth.

3.1 Background

Respiratory sinus arrhythmia (RSA) refers to the naturally occurring synchronization between heart beat and respiration — cardio-acceleration during inspiration, and cardio-deceleration during expiration — which is known as a reflection of the regulation of autonomic nervous system [61]. As such, RSA-BT (Respiratory Sinus Arrhythmia biofeedback-based Breathing Training) has been used as a common cardio-respiratory intervention with the goal of guiding trainees to initially breathe at their Resonant Frequency (RF), the frequency at which maximum amplitude of RSA is achieved, and then breathe in phase with heart beat changes with the same goal of RSA maximization [71]. Due to its ability to help improve autonomic control of cardiopulmonary function [72] and emotional self-regulation capacities [64], RSA-BT and its variants have been adopted as a complementary treatment to pulmonary diseases such as Asthma and chronic obstructive pulmonary disease (COPD) [36, 70], or as a relaxation technique especially for mental health conditions such as post-traumatic stress disorder (PTSD) [62].

When used as a clinical therapy, RSA-BT requires a set of instruments and follows a standard procedure administered by a therapist. Figure 3.1(a) shows a typical clinical setting of RSA-BT, where the trainee's physiological signals are measured by ECG electrodes, abdominal strain gauge, and pulse oximeter finger clip sensor. The measurements are then transmitted to a computerized machine to provide bio-feedback for breathing. The training protocol is typically composed of two types of sessions [36] (illustrated in Figure 3.1(a)(c)): (1) RF detection session: the trainee is asked to breathe for 2 minutes at each pre-set pace, e.g. 7, 6.5, 6, 5.5, 5, 4.5 breaths per minute (bpm), to obtain the RF. (2) Biofeedback session: the trainee is instructed to breathe at RF for the first few minutes and then follow a breath pattern (BP) in phase with IBI (Inter-beat Interval) for the rest of the session. Note that the traditional training protocol requires the supervision of a therapist. Specifically, the therapist should decide the RF according to the RSA distribution during the RF detection and suggest the moment to shift to IBI-based breathing basing on trainee's real-time performance during the training.

Despite its health benefits, today's RSA-BT therapy has several limitations. 1) First, it still relies on cumbersome devices and in-person sessions in a lab/clinical setting. Specifically, to measure physiological signs such as BP and IBI, trainees are required to wear sensors on their wrist, chest, and fingertip. In addition, the biofeedback display with a two-dimensional human-computer interface makes it difficult to convey intuitive guidance to trainees, therefore hampering the training experience and effectiveness. 2) Second, the protocol of today's RSA-BT lacks a way of taking poor training performance into account to adjust breathing guidance dynamically. Specifically, during a training session, trainees' breathing is only guided by IBI series, which does not always result in a suitable breathing pattern for trainees to follow, as some trainees may feel uncomfortable to breathe at a certain pace due to their



((a)) Conventional clinical tools for RSA-BT.

((b)) A prototype of BreathCoach, which includes an off-the-shelf smart watch and a smartphone-based VR viewer.

((c)) Comparison of the training procedures of RSA-BT between traditional approach and BreathCoach, showing two major differences: (1) Unlike traditional RSA-BT, no pre-training required in BreathCoach as RF is dynamically estimated during training. (2) During training, traditional RSA-BT relies only on IBI-based pacing after the initial 2-min RF-based pacing, while BreathCoach provide guidance by intelligently switching between RF-based and IBI-based pacing based on real-time measurements.

Figure 3.1: A Comparison between the traditional approach and BreathCoach for respiratory sinus arrhythmia biofeedback-based breathing training (RSA-BT).

physical conditions. Another possible cause of poor training performance is irregularities in measured IBI signals, due to body movements or other sources of interference. 3) Lastly, conventional RSA-BT's dependence on the supervision of a therapist [71] significantly limits its accessibility, therefore making it ill-suited for long-term practice at home.

In this thesis, we present BreathCoach — a smart and unobtrusive system that enables in-home RSA-BT using smartphone-based VR and sensors on a smartwatch (illustrated in Figure 3.1(b)). Specifically, BreathCoach continuously calculates required physiological measurements (i.e., BP, IBI, and RSA) using signals from the accelerometer and the PPG sensor on a smartwatch. These real-time measurements are used to calculate the recommended BP, which is then conveyed through a VR game to provide intuitive and continuous breathing guidance. To further improve the training performance and experience, BreathCoach intelligently switches between two pacing mechanisms based on a dynamic measure of user's difficulty in following the guidance (Figure 3.1(c)).

The key novelties of BreathCoach include:

- The system adopts a suite of lightweight algorithms to extract BP, IBI, and RSA from raw sensor signals in real-time, making it suitable for implementation on smartwatch and smartphone.

- To achieve better effectiveness of training, the system informs the calculation of recommended BP with dynamically estimated RF and RSA thresholds based on both current and historical measurements; and intelligently switches between two feedback mechanisms based on users' difficulty in following the guidance.

- The breathing guidance is conveyed to users in the form of VR game to provide a more intuitive and immersive guidance.

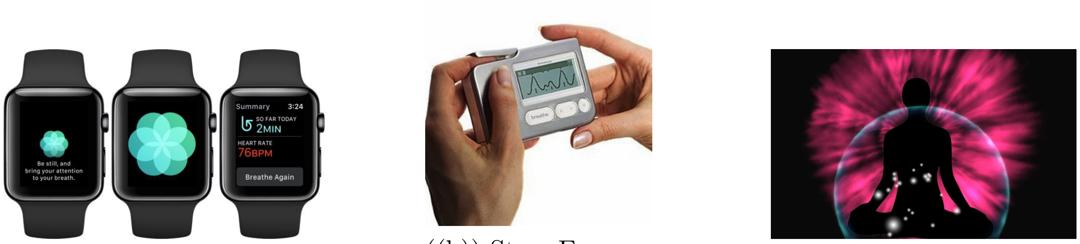
We have implemented a research prototype of BreathCoach with two exploratory VR game designs using a wrist-type wearable (Empatica E4 [2]), a smartphone (Moto G4 [5]) and a VR viewer (Google cardboard [3]). The evaluation of BreathCoach was conducted in three aspects, including the accuracy of physiological measurements, effectiveness of training, and user experience. We have collected both subjective and objective data from experiments where each of 10 participants performed 6 sessions of RSA-BT using either traditional approach or BreathCoach. The results show that BreathCoach is not only able to accurately measure required physiological signs, but also achieves better training performance than the traditional approach.

3.2 related Work

3.2.1 RSA Biofeedback Training

RSA-BT has been implemented for numerous clinical applications, such as treatments for asthma, COPD and various neurotic disorder [36, 70, 139]. Its implementation involves sensing and displaying instruments. The commonly used set of sensing instruments include ECG electrodes, abdominal strain gauge, and pulse oximeter finger clip sensor, as shown in Figure 3.1(a). C-2 biofeedback units with HRDFT software [36] and the cardiotachmeter [71] as shown in Figure 3.1(b) are widely used as displaying instruments for clinical RSA-BT. Figure 3.1(b) also illustrates the biofeedback interface of C-2 biofeedback units. The breathing pacer is a sawtooth-shaped line. A small ball travels along the line from left to right to guide inhalation and exhalation. Heart rate is displayed in the same window as the biofeedback. Besides, the clinical RSA-BT adopted a standard training protocol. Specifically, this protocol consists of two sessions [36]. In the first session, the trainee is asked to breathe for 2 minutes at each pre-set pace, e.g., 7, 6.5, 6, 5.5, 5, 4.5 bpm, to obtain the RF. During the second session, the trainee is instructed to breathe at RF for the first few minutes and then breathe in phase with IBI.

However, these RSA-BT systems have several shortcomings, especially for in-home training. Firstly, the cumbersome sensing and displaying instruments make these systems impractical for in-home RSA-BT. Secondly, training with these systems entails the supervision of the therapist. Specifically, the therapist should decide the RF according to the RSA distribution and suggests the moment to shift to IBI-based breathing basing on trainee's real-time performance in the second session. Moreover, according to the standard RSA-BT protocol, RF detection should be performed every time starting training, which is inefficient. Finally, trainees may feel overwhelmed when failing to breathe in phase with IBI. Trainees'



((a)) Apple Watch Breathing Application [8].

((b)) StressEraser, a popular off-the-shelf device for daily RSA-BT [6].

((c)) The VR interface of SOLAR.

Figure 3.2: Examples of the products on the market with functions related to breathing training.

physiological limit may prevent them from breathing at a low rate, and irregularities in ECG signals also make it difficult for trainees to follow an aperiodic IBI. In these situations, their training performance will be degraded without the therapist's supervision.

An abundance of breathing applications has emerged to serve different functions—from entertainment-oriented games to improving health or well-being. "Breathe" is a native application on Apple watch [8]. As shown in Figure 3.2(a), it uses graphic animation and gentle taps to guide the breathing and help the user focus. The training duration and frequency can be customized. This app is easy to operate and designed for daily breathing training. However, without any biofeedback, this app fails to consider users' training performance. Besides, this app leads users to breathe at a fixed pace and the breathing pace is constant for all users, which makes the training ineffective. On the one hand, the exact cardiac RF varies from person to person [127]. Thus, the breathing pace should be adapted to varied individuals. On the other hand, the RF has been shown to change over time within individuals [71, 33, 74]. Therefore, breathing at a fixed pace is ineffective when the RF has reduced to a slower pace. Instead, by breathing in phase with heartbeat changes, RSA-BT allows each individual to breathe at a rate that is adapted to the rhythms of his/her own body and over time as respiratory function improves.

StressEraser is an off-the-shelf device for daily RSA-BT, which has been commonly used in various treatments and the related research [98, 112]. As shown in Figure 3.2(b), StressEraser is a hand-held biofeedback device that measures HRV from the pulse in your fingertip via an infrared sensor and displays it as a wave to instruct users' respiration. This portable device can be used for in-home RSA-BT. The users often complained about error sensing signals and the failures to deal with irregularities in IBI [1]. To obtain good-quality signals, users are asked to hold finger steady and avoid sunlight. Even so, sometimes it provides a meaningless straight line. Moreover, like a clinical RSA-BT system, it still fails to manage trainees' bad performance resulting from physiological limits and uncertain irregularity in ECG signals. Finally, without a respiratory sensor, StressEraser is unable to detect trainees' real-time respiratory response, and thus leaves trainees unaware of their real-time performance.

Recently, an immersive breathing training system, called AirFlow, has been developed for COPD [100]. It collects respiratory data from sensors on the chest and abdomen and reflects them in immersive breathing training games, including the Balloon Game, Eating Game and Penguin Game. These games are designed to train Pursed-Lip breathing, breathing rhythm and depth respectively. In addition to requiring obtrusive sensing devices, the system is only able to guide users to breathe at a fixed rate.

3.2.2 Breathing training as a stress mitigating intervention

Breathing has a direct effect on RSA and as such plays a fundamental role in regulating the autonomic nervous system and reducing autonomic arousal [46]. Research suggests that each individual has a resonant frequency at which RSA is the greatest. Breathing at resonant frequency stimulates the vagal baroreflex [73]. Frequent high-amplitude stimulation of the baroreflexes by breathing at resonant frequencies increases the efficiency of cardiac reflexes and baroreflexes, and consequently promotes relaxation.

Research shows that breathing training as an effective regulator of autonomic arousal leads to concrete stress reduction effects [24, 23, 126]. Preliminary results suggest that portable RSA biofeedback appears to be a promising treatment adjunct for disorders of autonomic arousal and is easily integrated into treatment [103]. Several studies support that RSA-BT is a promising treatment for several kinds of anxiety disorder, such as post-traumatic stress disorder (PTSD), work stress and perinatal depression [119, 89, 17]. Recently, guided breathing has been utilized as a mindful intervention for drivers to counteract the stress accumulated at work and the additional stress encountered during driving [93].

3.2.3 Respiration pattern measurement

Respiratory inductance plethysmography (RIP) sensor is the most widely used device to evaluate pulmonary ventilation by measuring the movement of the chest and abdominal wall [16, 44]. It consists of two lightweight elastic and adhesive bands, which makes the measurement of respiration pattern cumbersome.

To detect the breathing pattern unobtrusively, the MindfulWatch, a smartwatch-based system for real-time respiration monitoring during meditation, was developed in [44]. It utilizes motion sensors to sense the subtle ”micro” wrist rotation (0.01 rad/s) induced by respiration. MindfulWatch offers reliable real-time respiratory timing measurement using a novel self-adaptive model that tracks changes in both BP and meditation posture over time.

3.2.4 Bio-responsive VR

VR systems have been successfully applied in the treatment of various anxiety disorders including fear of flying, social phobia, PTSD, fear of spiders and fear of heights. There are mainly three principles for the VR design of these mindful games: abstract visual elements, rewarding practice and attention restorative environment. Specifically, abstract visual elements such as images and shapes are less distracting than concrete images such as flower, sky, etc., and thus help participants relax [58]. The use of subtle visual elements as a reminder to focus on stimulus is the preferred form of visible feedback [27]. Additionally, Rewarding practice can motivate users to practice more often and for longer periods of time because of the enjoyment they feel. Finally, attention restorative environment positively affects user's attention [57]. The environments with stimuli that modestly capture attention are preferred. For instance, subtle nature sounds are preferred over traffic noise.

SOLAR is a popular VR game that assists novice users in learning the stress-reducing practice of mindfulness meditation [99]. Its VR is generated by the user's brain activity and respiratory rate. SOLAR asks users to focus their attention on the visual representation of breathing. It is common for the users' mind to wander during meditation. Therefore, we included the user's meditation scores in order to provide gentle feedback to the user when their mind starts to wander. This meditation score was mapped to the color of the meditation circle, positioned behind the silhouette as shown in Figure 3.2(c). Besides, the respiration sensors were placed on the user's thorax and diaphragm. The data received from the sensors were used for generating both audio and visual elements of SOLAR. The respiration sensors are mapped to the breathing circle in front of the silhouette. The breath circle becomes larger and smaller as the user inhales and exhales.

3.3 System Requirements and Challenges

BreathCoach is designed to be an in-home RSA-BT system that continuously tracks physiological variables, calculates the recommended BP in real time and guides users towards the recommended BP through a VR game. To achieve this goal, BreathCoach should meet the following requirements: (1) Since BreathCoach is designed for home and office use, its sensing and displaying instruments should be easy to operate and comfortable to wear. (2) BreathCoach needs to provide accurate and continuous measurement of physiological variables, including BP, IBI and RSA, compared to clinical tools. (3) BreathCoach should automate the procedure of traditional training, and intelligently provide guidance to users without the presence of a therapist. (4) BreathCoach should provide guidance in an intuitive and easy-to-follow fashion.

To meet these requirements, we addressed two major challenges in developing BreathCoach. (1) It is challenging to extract accurate BP, IBI, and RSA in real time from the built-in PPG sensor and accelerometer on the smartwatch. Compare to sensors available (i.e., ECG and RIP) in a clinical setting, smartphone sensors are significantly susceptible to motion artifacts. (2) The pacing mechanism used in traditional RSA-BT only relies on real-time IBI series, which is not suitable due to irregularities caused by interference such as body motion. Therefore, without the supervision of a therapist, it is challenging to create an intelligent pacing mechanism that provides continuous and effective breathing guidance.

3.4 System Design

The architecture of BreathCoach is illustrated in Figure 3.3 with three key components, which are Physiological Measurement, Dynamic Estimation, and Intelligent Pacing. The Physiological Measurement component is responsible for calculating required bio-signals needed for Breathing Pattern Recommendation. Specifically, it takes raw signals from accelerometer and PPG sensor on the smartwatch as input to extract breathing pattern (BP) and inter-beat interval (IBI), which are then used to calculate RSA amplitude. Based on the historical data and current measurements, The Dynamic Estimation keeps track of the resonant frequency (RF) and its corresponding RSA threshold – two key parameters for generating effective breathing recommendation – which typically changes during training. Informed by the results of dynamic estimation, the Intelligent Pacing component selects an optimal pacing mechanism and generates the recommended breathing pattern. Finally, the system presents the resulting breathing pattern in a smartphone-based VR game.

3.4.1 Physiological Measurement

Physiological measurement provides required bio-signals, including BP, IBI and RSA, to BP recommendation. It contains three major components: IBI extraction, breathing pattern extraction and RSA quantification.

As both PPG-based IBI extraction and acceleration-based breathing pattern extraction are sensitive to significant postural change, the system first analyzes the acceleration data to

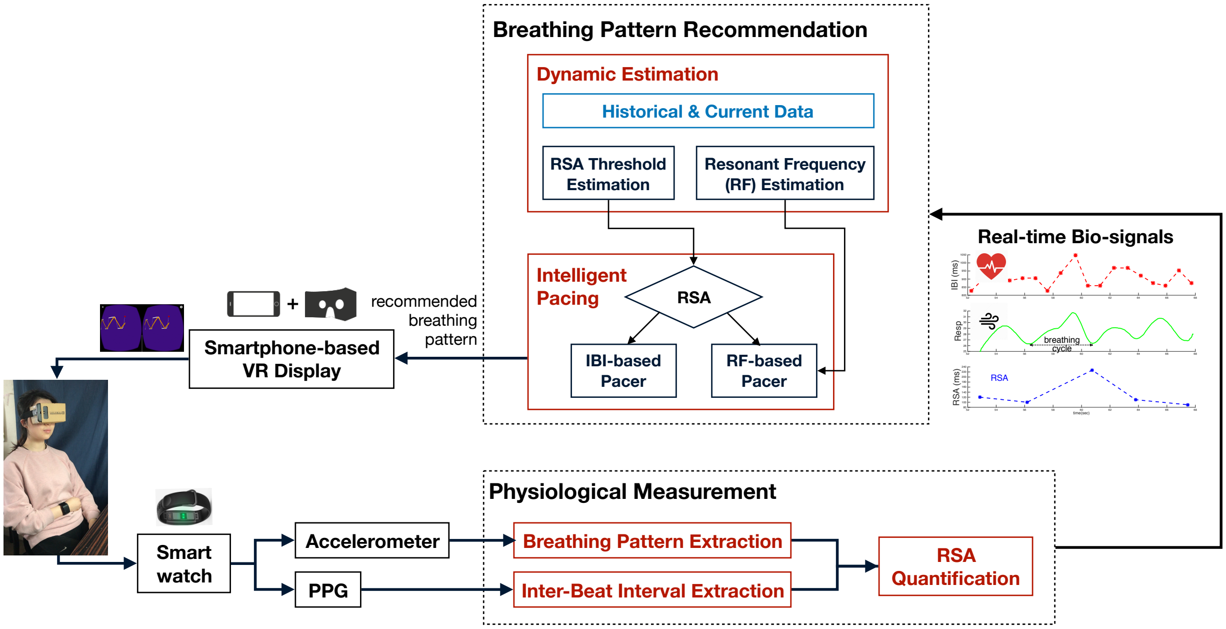

Breathing Pattern Recommendation
Dynamic Estimation
Historical & Current Data
RSA Threshold Estimation
Resonant Frequency (RF) Estimation
Intelligent Pacing
RSA
IBI-based Pacer
RF-based Pacer
recommended breathing pattern
Smartphone-based VR Display
Real-time Bio-signals
IBI (ms)
Resp
breathing cycle
RSA (ms)
RSA
Physiological Measurement
Smart watch
Accelerometer
PPG
Breathing Pattern Extraction
Inter-Beat Interval Extraction
RSA Quantification

Figure 3.3: System overview of BreathCoach.

estimate postural stability before further extracting cardiac and respiratory signals. When low postural stability is detected, the system will pause extracting physiological signals and resume when it goes back to being stable.

The postural stability is assessed using the standard deviation of three-axial acceleration's norm (STD_{acc}), which is calculated over 1 s acceleration series every 0.03s (set according to the sample rate of the accelerometer). As the respiration-induced wrist motion fluctuates subtly and consequently has a low variation in the norm of acceleration compared with a significant postural change, STD_{acc} should stay below a threshold with no significant postural change. Once STD_{acc} exceeds the threshold, physiological measurements are discontinued as it indicates significant postural changes. This threshold is generally defined as 1 g according to our experimental results.

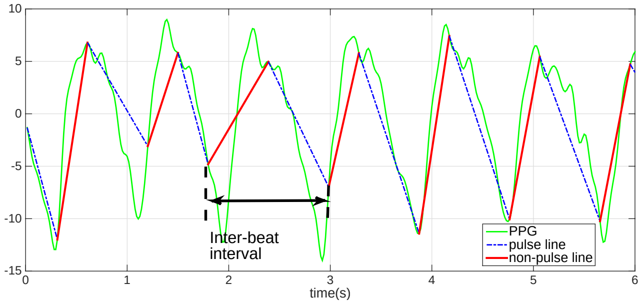


Figure 3.4: An example of inter-beat interval (IBI) extraction based on 6-second pulse wave data from PPG sensor.

3.4.1.1 IBI extraction

The system extracts IBI from PPG signal when the user's posture is relatively stable. Specifically, the raw PPG signals are first filtered from 0.8 Hz to 5 Hz to reduce noise. The filtered PPG is then segmented using Incremental-merge segmentation algorithm (IMS) to calculate IBI [60]. After segmentation, lines are classified as pulse or non-pulse lines. If the interval between an up-slope and the last pulse line is large than a pre-defined threshold, this up-slope is identified as a validated pulse line. The threshold is set to 0.6, as the resting IBI ranges from 0.6 to 1 sec. Finally, the continuous PPG signal is divided into a group of pulse lines (as shown in Figure 3.4). The IBI is calculated as the interval between the ends of consecutive pulse lines.

3.4.1.2 Breathing pattern extraction

With the wrist band being held against the user's abdomen, the respiration can be monitored by analyzing the motion caused by the subtle displacement of the user's abdomen due to respiration. Apple Watch has utilized a similar method for blood pressure measurement, in which the accelerometer would, when held against your chest, detect the heartbeat [7]. To measure breathing, the raw acceleration is first processed by a low-pass filter of 0.4Hz, which aims to highlight the motion due to respiration. The filtered acceleration signal is then used to calculate BP using IMS. As mentioned before, IMS segments acceleration into up-slope

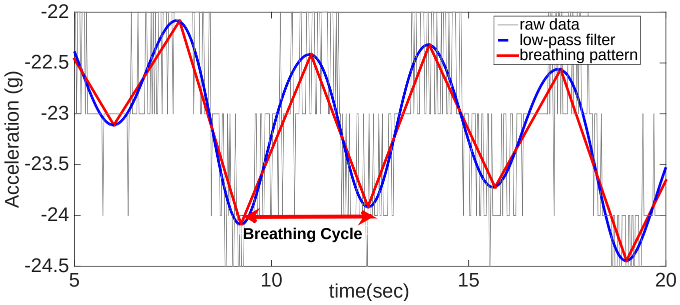


Figure 3.5: An example of breathing pattern extraction based on 15-second acceleration.

and down-slope lines. If the interval between an up-slope and the last expiration line is large than a pre-defined threshold, this up-slope is identified as a validated expiration line. The threshold is set to 3, as the normal resting breathing cycle is no less than 3 sec. Finally, the acceleration is divided into a group of expiration and inspiration lines, and each pair of consecutive expiration and inspiration lines will be identified as a breathing cycle (as shown in Figure 3.5).

3.4.1.3 RSA quantification

RSA refers to synchronization between heart beat and respiration [116]. As a critical parameter for breathing pattern recommendation, the system quantifies RSA by calculating its amplitude on a breath-by-breath basis through Peak-valley algorithm [61] based on the real-time IBI and BP. Specifically, when there are valid minimal and maximal IBI for a breath cycle, RSA is calculated as the difference between the maximum and minimum IBI. Figure 3.6 illustrates the peak-valley method for RSA estimation. Each breathing cycle is detected from the respiration pattern. For each breath, the estimate of RSA is obtained by searching the corresponding segment of IBI series for the maximum and minimum value and then computing their difference.

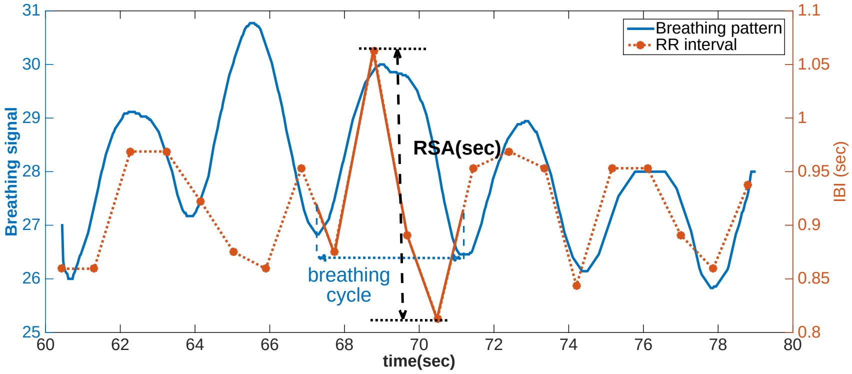


Figure 3.6: An example illustrating Peak-valley algorithm for RSA quantification. Per breath, RSA is calculated as the difference between maximum and minimum inter-beat interval (IBI).

3.4.2 Real-time Breathing Pattern Recommendation

Running on the smartphone, this component takes the continuous measurements of IBI, BP, and RSA to dynamically calculate a recommended BP for optimal performance. Specifically, real-time BP recommendation involves dynamic estimation and intelligent pacing.

3.4.2.1 Intelligent Pacing

The intelligent pacing dynamically chooses the optimal mechanism for BP recommendation between IBI-based where users are guided to breathe in phase with IBI changes, and RF-based pacing mechanism where users are guided to breathe at a fixed pace, i.e., RF.

The dynamic switching is controlled by two RSA thresholds: RSA_{low} and RSA_{high}, which act as the standards for real-time evaluation of training performance. When the user hardly breathes in phase with the IBI wave, which may be irregular at that time, the RSA amplitude will drop below the RSA_{low} indicating a bad training performance. According to experimental results, RSA_{low} is set to 100 ms in BreathCoach to define a bad training performance as a weak synchronization between breathing and IBI with RSA below 100 ms. If the RSA exceeds RSA_{high} while the user breathes following the RF-based pacer, it means the IBI wave acts regular and the user is capable of breathing in phase with

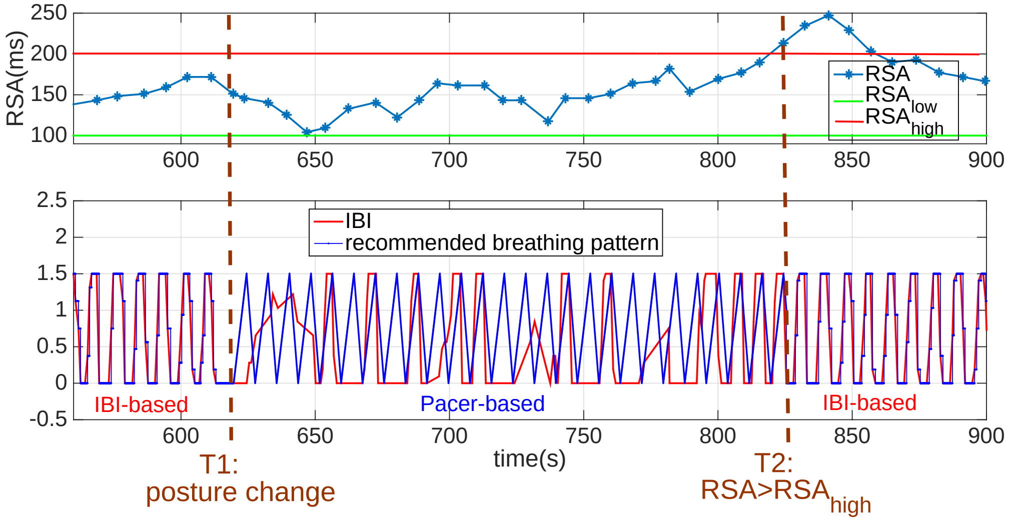
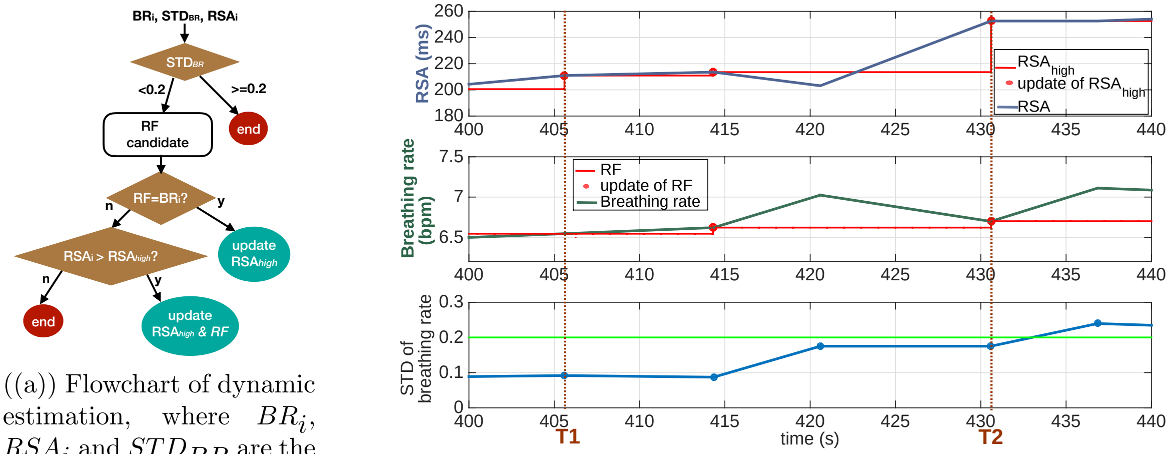


Figure 3.7: An example showing how Intelligent pacing works. At T1, the system switched from IBI-based to RF-based pacer, as significant postural changes interrupted IBI extraction; At T2, the system switched back to IBI-based mechanism, as the RSA exceeded RSA_{high}.

IBI. RSA_{high} is defined as the maximum RSA amplitude achieved when breathing at RF. It should be set and updated during training, because it varies with each individual and changes during the training. Specifically, the system will switch to RF-based pacing mechanism if the current RSA is lower than RSA_{low} or the IBI extraction is interrupted by significant postural changes, and switch back to IBI-based mechanism when the RSA exceeds RSA_{high}.

Figure 3.7 illustrates how intelligent pacing works with a real-world example. At T1, BreathCoach switches from IBI-based to RF-based pacing mechanism as IBI extraction is suspended, and switches back to IBI-based pacing at T2 when RSA is detected greater than RSA_{high}. We can observe that IBI waveform is irregular and RSA stays low from T1 to T2, whereas IBI waveform gets regular and RSA becomes larger at the end of RF-based pacing training, suggesting that RSA as an evaluation of real-time training performance monitors not only the user's capacity to breathe in phase with IBI but also the irregularity of IBI signals.



((a)) Flowchart of dynamic estimation, where BR_i, RSA_i and STD_{BR} are the breathing rate, RSA amplitude and the standard deviation of BR at ith time step, respectively.

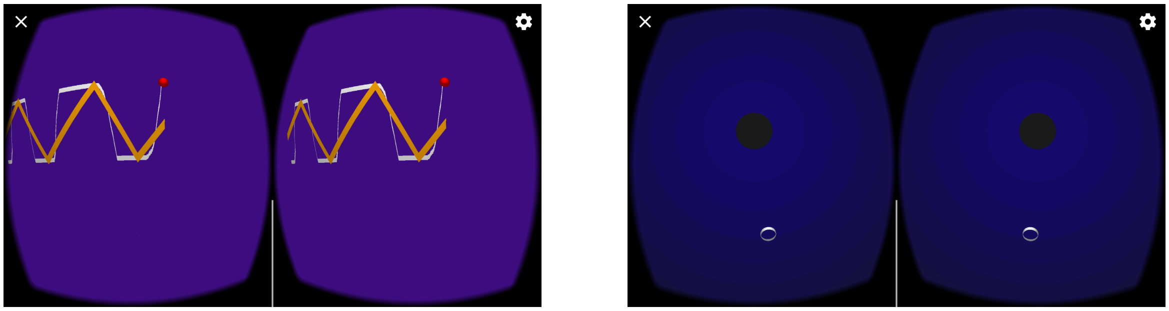
((b)) An example of Dynamic estimation of RF and RSA_{high}. Both RF and RSA_{high} are updated at T2, as the STD_{BR} is lower than 0.2 and RSA exceeds RSA_{high}. Only RSA_{high} is modified at T1, as the BR with a STD_{BR} below 0.2 is equal to RF at this point.

Figure 3.8: Dynamic estimation.

3.4.2.2 Dynamic Estimation

RF and its corresponding maximum RSA amplitude (i.e.,RSA_{high}) changes during training in two scenarios. In the first scenario, a respiration frequency with its corresponding RSA amplitude higher than RSA_{high} is detected to be the new RF. In another scenario, the RSA amplitude observed when the user breathes at RF is different from RSA_{high}, suggesting RSA_{high} should be updated to this RSA amplitude. In order to adapt breathing pattern recommendation to such changes, RF and RSA_{high} are dynamically updated by analyzing historical data, including BP and RSA.

By analyzing BP, BreathCoach monitors user's breathing rate (BR) and its stability through the standard deviation of BR (STD_{BR}) to identify RF candidates. BR is calculated breath by breath as 60 divided by the average of previous 5 breathing cycles. Its corresponding STD_{BR} is also calculated each breath in a 5-breathing-cycle window. Since the detection of RF entails a long-term observation, STD_{BR} is necessary to make sure users maintain a BR long enough that this BR can be a potential RF. A STD_{BR} lower than the pre-defined threshold (set to 0.2) makes the corresponding BR an RF candidate, suggesting the user has kept breathing at this BR during previous five cycles.



((a)) Balloon, where the player controls the movement of the balloon through respiration to follow the recommended breathing pattern represented by the yellow track.

((b)) Pilot, where the player breathes in and out with the shrinkage and expansion of the white circle to make the flight as fast and straight as possible.

Figure 3.9: Screenshots of two proof-of-concept VR games.

BreathCoach recognizes the new RF and RSA_{high} by observing the RSA of RF candidates. Same as BR, its corresponding RSA amplitude (RSA_{BR}) is calculated breath by breath in a 5-breathing-cycle window. As shown in Figure 3.8(a), for each RF candidate, if BR equals RF, RSA_{high} is updated to the RSA_{BR}. Otherwise, RSA_{high} and RF will be updated to RSA_{BR} and BR respectively if RSA_{BR} is greater than RSA_{high}. Figure 3.8(b) shows the dynamic estimation of RF and RSA_{high} in practice. As shown in this figure, both RF and RSA_{high} are updated at T2 as the STD_{BR} is lower than 0.2 and RSA exceeds RSA_{high} at this moment. Only RSA_{high} is modified at T1, since BR with a STD_{BR} below 0.2 is equal to RF at this point.

3.5 VR Game

To provide an immersive and intuitive guidance, BreathCoach presents bio-feedback through VR game, in which a pacing stimulus is driven by the recommended BP to instruct breathing. In this section, we describe two exploratory proof-of-concept VR games implemented as part of BreathCoach system.

3.5.1 Balloon

As illustrated in Figure 3.9(a), the goal of this game is to guide users to breathe in sync with the yellow track to make the red balloon move along the track as precisely as possible in 15 minutes. The dynamic track, as the pacing stimulus, represents the recommended BP. The player controls the movement of the balloon through respiration, and the trail of the balloon reflects the player's breathing pattern. The degree of alignment between the trail of the balloon and the track indicates the player's performance, which is also used to change the game's background color to give users feedback on their performance.

3.5.2 Pilot

The Pilot game is designed to guide users to breathe in and out with the shrinkage and expansion of the white circle to make the flight as fast and straight as possible, as shown in Figure 3.9(b). As the pacing stimulus, the white circle near the bottom of the screen expands and shrinks according to the recommended BP. The flight altitude, speed, and the game's background color are controlled by RSA amplitude, i.e., a proxy of the player's real-time performance. The higher and stabler the RSA estimations are, the farther and straighter the player will fly. Different from Balloon, the Pilot game translates a proxy of training performance into actions in the game, instead of directly revealing real-time respiration and performance to users.

3.6 Evaluation

In this section, we present the evaluation of BreathCoach based on a set of in-lab controlled experiments. First, we evaluate the accuracy of physiological measurements that the system uses to generate recommendations (Section 6.2). Second, we investigate the effectiveness of BreathCoach's real-time breathing pattern recommendation with respect to RSA amplitude maximization throughout breathing training [71] and an essential use case of RSA-BT, i.e. stress reduction [59, 96, 112, 139] (Section 6.3). Finally, we explore the effect of different game design (Section 6.4).

3.6.1 Experiment settings

The evaluation adopts a repeated-measures design, with the training protocol (i.e., traditional and BreathCoach) and game design (i.e., Balloon and Pilot) as within-subjects variables. Subjects were required to conduct RSA-BT using three types of protocol-game combination, including *Traditional-Balloon (traditional breathing training protocol plus Balloon)*, *BreathCoach-Balloon*, and *BreathCoach-Pilot*. Such experiment design allowed us to compare the BreathCoach-Balloon training with the traditional-Balloon training to assess the effects of intelligent breathing pattern recommendation module in BreathCoach and compare the BreathCoach-Balloon with BreathCoach-Pilot training to study the game design of BreathCoach. Our study along with its data collection procedure was approved by the Institutional Review Boards (IRB). All the subjects voluntarily agreed to help with data collection and signed a consent form.

We have recruited 10 subjects, and each participated in our data collection consisting of six 45-minute RSA-BT sessions scheduled in different days. As shown in 3.10, in each session, the participants were exposed to a different and randomly selected breathing training setup. Note that the six sessions consist of two for each kind of training setup and participants randomly arranged their sequence for the six-day training. Each experiment begins with a tutorial during which the study administrator explained each part of the session and gave subjects a live demonstration of the breathing training system. After that, participants started the six daily sessions. As illustrated in Figure 3.10, each session includes 5 stages:(1) RF detection, (2) pre-training task, (3) breathing training, (4) post-training task, (5) survey. Specifically, the participants are initially left alone in the workspace to accomplish a 10-min RF detection, a procedure required in traditional RSA-BT protocol to manually estimate user's in-situ RF. Subsequently, participants are asked to perform cognitive tasks, including a standard Stroop Test, followed by a restorative break. This task is widely utilized to

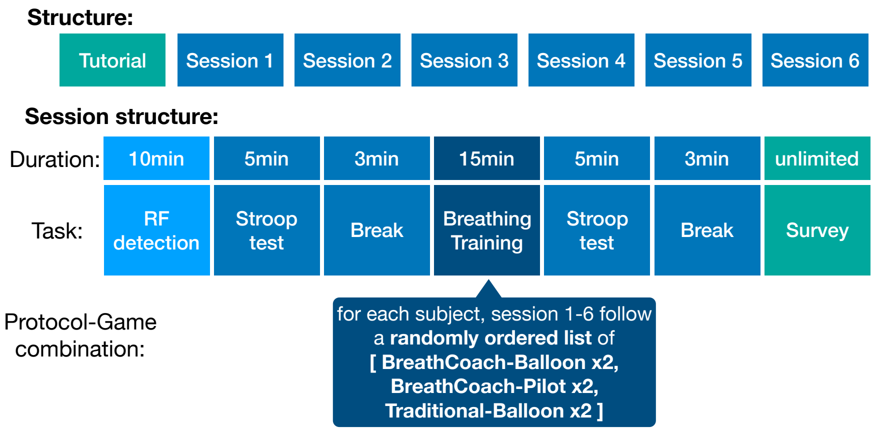


Figure 3.10: Schematic illustration of the study protocol.

simulate a focused and stress-eliciting work situation and the recovery from it [98, 112]. After cognitive tasks, participants are left alone in the workspace for the 15-min breathing training with specific training setup. Upon finishing the training, participants will be asked to perform cognitive tasks again. At the end of each experimental session, participants are presented with a survey for training experience. The scale explores users' training experience and game preference via the following questions:

1. How often have you been distracted from breathing during the training?
2. How often have you felt hard to follow pacing stimulus?
3. How often have you felt anxious while training?
4. How often have you tried too hard while breathing?
5. Which game do you prefer, Balloon or Pilot, and Why?

The subject assesses the frequency on a 0-4 scale (0 = Never, 4 = Very Often). Besides, subjects' physiological responses, such as RSA, BR and IBI, were recorded in all procedures.

In order to collect data, each subject was asked to wear an off-the-shelf wrist-type wearable (Empatica E4 [2]) and a smartphone (Moto G4 [5]) with a VR viewer (Google Cardboard [3]) during breathing training as shown in Figure 3.1(b). Both BreathCoach and traditional protocol are implemented using the Empatica E4, Moto G4 and Google Cardboard. During data collection, the PPG sensor and the accelerometer of Empatica E4 are continuously sampled at 64 Hz and 32 Hz, respectively. The ground truth for IBI and BP measurements is collected from Hexoskin [4] – a smart shirt with built-in ECG and RIP sensors.

3.6.2 Evaluation of physiological measurement

We evaluate the algorithms for BP and IBI measurements by comparing them with the ground truth.

3.6.2.1 Evaluation of breathing pattern extraction

We first evaluate the performance of BreathCoach in detecting the breathing pattern and measuring the complete breathing cycles. The evaluation is based on the metric: estimation error of Breathing cycle duration (Dur_{bc}). BreathCoach extracts users' BP from acceleration. To evaluate its accuracy, we compare BreathCoach's measurement for each breath cycle with the corresponding one from the ground truth (same data measured using the RIP sensor), and use their differences in Dur_{bc} as performance metrics. Figure 3.11 shows the error distribution of the breath-by-breath detection result collected from 10 subjects during their RF detection. We can see that the distribution of Dur_{bc} error is mostly symmetric around 0, indicating that the error does not accumulate over time. Specifically, the average absolute error of Dur_{bc} is 0.61 s, with 80.07% of the absolute errors under 1 second, as shown in the cumulative distribution function (CDF) of absolute Dur_{bc} error. We believe that this accuracy of complete breathing cycle detection is sufficient for deriving RSA and BR as users' breathing rates range from 4 to 10 breath per minute (bpm) during breathing training.

3.6.2.2 Evaluation of IBI extraction

To evaluate BreathCoach's performance in measuring IBI, we compare all the IBI produced by BreathCoach with those obtained from the ground truth (same data measured using the

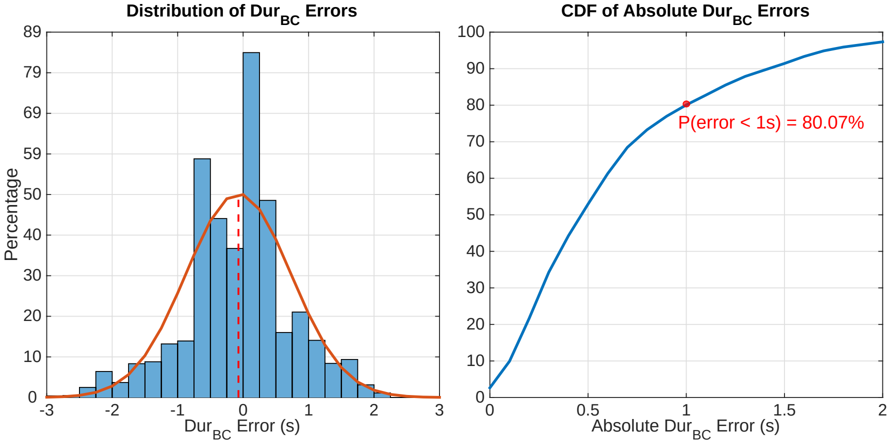


Figure 3.11: The error distribution (left) and CDF (right) of the breath-by-breath detection result of BreathCoach collected from 10 subjects. The average absolute error of breathing cycle duration (Dur_{bc}), which is used to derive RSA, is 0.61 s.

ECG on Hexoskin), and use their pairwise differences, i.e. IBI errors, as evaluation metrics. Figure 3.12 shows the error distribution of IBI collected from 10 subjects during their RF detection. We can observe that the distribution of IBI error is almost symmetric around 0, suggesting that the error does not accumulate over time. Specifically, the average absolute error of IBI is 9.6 ms, with 81.48% of the absolute errors under 15 ms, as shown in the CDF of absolute IBI error. Therefore, we believe that BreathCoach's accuracy and reliability in measuring IBI are sufficient for generating feedbacks and deriving RSA.

3.6.3 Evaluation of Intelligent Breathing pattern recommendation

In this subsection, we evaluate BreathCoach in two aspects: training effectiveness and subjects' training experience. To study the effect of intelligent breathing pattern recommendation, we compare the BreathCoach-Balloon training with the baseline, traditional-Balloon training.

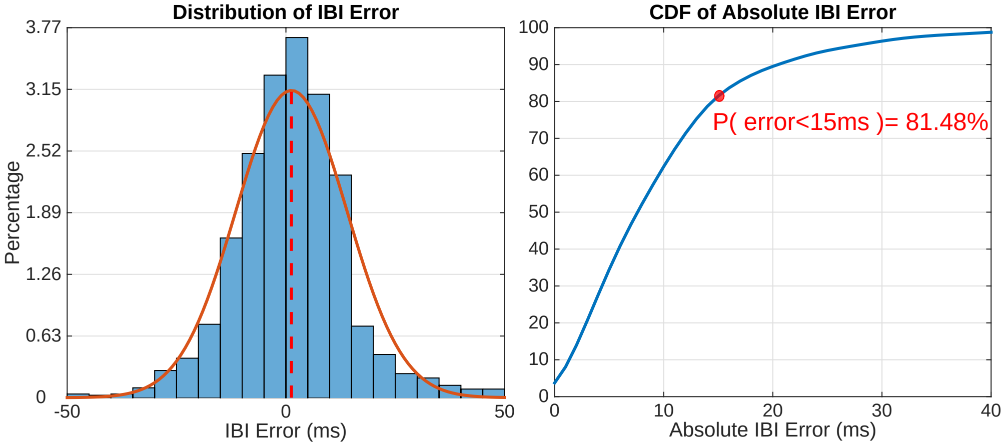


Figure 3.12: The error distribution (left) and CDF (right) of the BreathCoach's inter-beat interval (IBI) extraction from 10 subjects. The average absolute error of IBI, which is used for RSA assessment and real-time breathing pattern recommendation, is 9.6 ms.

3.6.3.1 RSA maximization

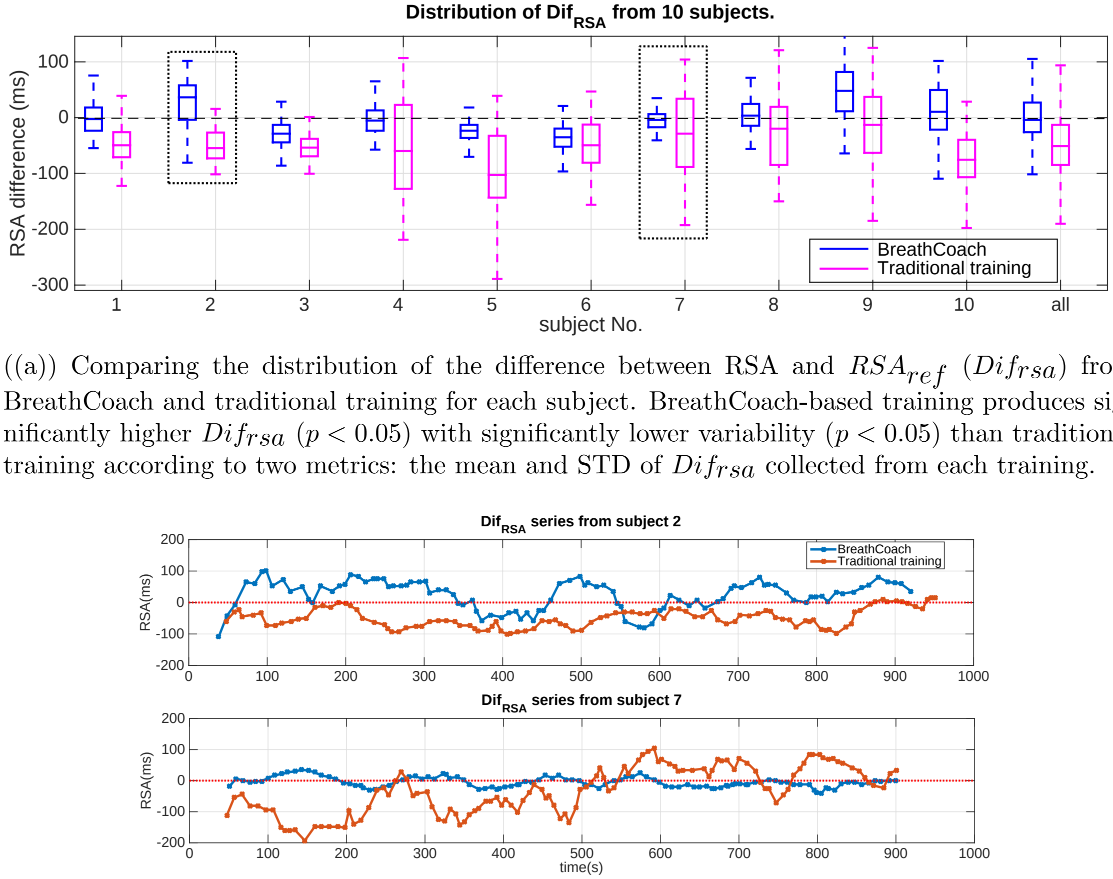
We evaluate the effect of BreathCoach on RSA maximization, as the direct objective of RSA-BT is to maximize the RSA amplitude throughout the training to achieve better health outcome. The evaluation is based on Dif_{rsa}, the difference between RSA and RSA_{ref}. RSA_{ref} is the maximum RSA amplitude achieved by breathing at RF during RF detection, which is considered as a reference in the assessment of the effect on RSA maximization. It is obtained from the 10-min RF detection before each training. Dif_{rsa} gauges how close user's RSA is to the maximum RSA amplitude. Dif_{rsa} is computed as:

$$Dif_{rsa}(i) = RSA(i) - RSA_{ref} \tag{3.1}$$

where $RSA(i)$ denotes the user's RSA in the i-th breathing cycle. $Dif_{rsa}(i)$ denotes the difference between user's RSA in the i-th breathing cycle and the maximum RSA, RSA_{ref}. It is worth noting that Dif_{rsa} has a sign, determining whether recorded values fall below or above RSA_{ref}. Specifically, a non-negative Dif_{rsa} suggests the RSA amplitude is currently maximized and a high negative Dif_{rsa} indicates the current RSA fall closely below the maximum RSA. Therefore, high Dif_{rsa} implies well performance in maximizing RSA amplitude.

Figure 3.13(a) compares the distribution of Dif_{rsa} from BreathCoach and traditional training for each subject. We observe that there is a smaller variability of Dif_{rsa} for BreathCoach as well as greater medians, suggesting that for most of the subjects RSA consistently falls more closely below RSA_{ref} and is more likely to exceed RSA_{ref} in BreathCoach-based training than traditional training. For subject 2, Dif_{rsa} from BreathCoach are generally higher with about 50% above 0, suggesting that RSA has been maximized for most of the time during the BreathCoach-based training, see Figure 3.13(b). For subject 7, although the median of Dif_{rsa} from BreathCoach is lower than 0, it fluctuates within a narrow range, which indicates that RSA falls more closely to the RSA_{ref} in BreathCoach than in the traditional training, see Figure 3.13(b). Thus, BreathCoach still outperforms traditional training in RSA maximization for subject 7. Also, we used paired t-tests to reveal significant ($p < 0.05$) differences between the effects of BreathCoach and traditional training on RSA maximization according to two metrics: the mean and STD of Dif_{rsa} collected from each training. The result shows BreathCoach-based training produces significantly higher Dif_{rsa} ($Mean(Dif_{rsa}) : p = 0.00001$) with significantly lower variability ($STD(Dif_{rsa}) : p = 0.0086$) than traditional training.

Figure 3.14 compares the distribution of Dif_{rsa} collected from all BreathCoach-based training with the one obtained from traditional training. We can see that, compared with Dif_{rsa} from traditional training, those from BreathCoach distribute more intensively around an average closer to 0, suggesting that RSA collected from BreathCoach-based training consistently fall closely below or above RSA_{ref}. Specifically, the average and STD of Dif_{rsa} from BreathCoach are 2.37 and 42.94 ms respectively, with 70% of Dif_{rsa} above -20 ms and 50% of Dif_{rsa} above 0 ms. For traditional training, the average and STD of Dif_{rsa} are -49.9 and 63.82 ms respectively, with only 32% of Dif_{rsa} above -20 ms and 70% above -85 ms, see Figure 3.14. As the RSA_{ref} is usually greater than 200 ms, an absolute Dif_{rsa} below 20 ms is sufficient to suggest an RSA highly close to the maximum value. Therefore, we believe that training using BreathCoach enable users to perform well in maximizing RSA



((a)) Comparing the distribution of the difference between RSA and RSA_{ref} ($Difrsa$) from BreathCoach and traditional training for each subject. BreathCoach-based training produces significantly higher $Difrsa$ ($p < 0.05$) with significantly lower variability ($p < 0.05$) than traditional training according to two metrics: the mean and STD of $Difrsa$ collected from each training.

((b)) Illustrating the $Difrsa$ series of subject 2 (upper) and 7 (lower). For each subject, compare the $Difrsa$ series collected from BreathCoach-based and traditional training

Figure 3.13: Evaluating the effect of BreathCoach on RSA maximization by observing the difference between RSA and RSA_{ref} ($Difrsa$). RSA_{ref}, the maximum RSA amplitude achieved by breathing at RF during RF detection, acts as a reference in the assessment of the effect on RSA maximization.

throughout the training.

3.6.3.2 Stress Reduction

The stress reduction is studied based on heart rate variability (HRV), which is an established psycho-physiological measure for stress development and restoration [120, 19]. We used the standard deviation of normal to normal R-R intervals (SDNN) method to compute HRV. SDNN is calculated for consecutive overlapping sections of 1-min IBI data. We defined three metrics from HRV time series, including the mean of HRV during the cognitive task

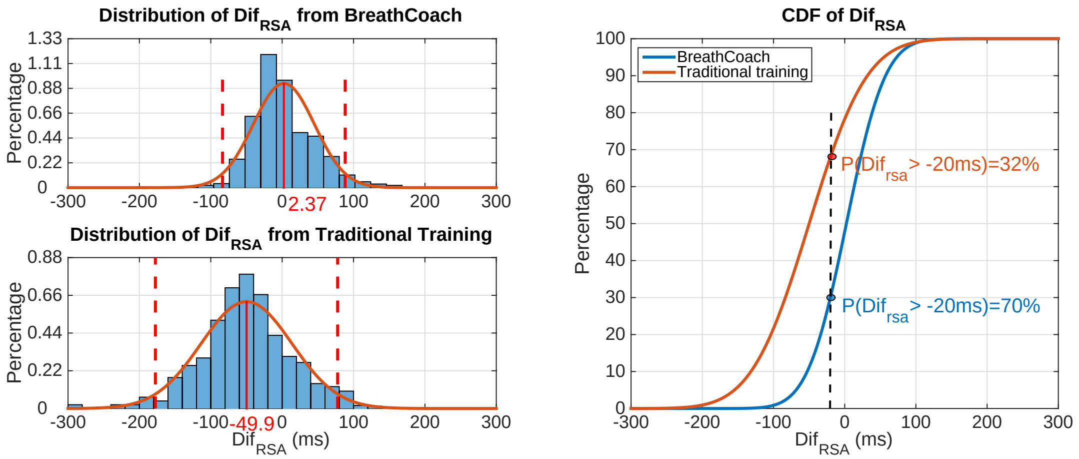


Figure 3.14: The distribution (left) and CDF (right) of the difference between RSA and RSA_{ref} (Dif_{rsa}) collected from BreathCoach-based training, showing that BreathCoach significantly improves the performance in maximizing users' RSA throughout the training compared with traditional training approach ($p < 0.05$).

(μ_{HRV}), recovery speed and amplitude of HRV during the post-task rest ($SpeedRec_{HRV}$ and $AmpRec_{HRV}$). Specifically, greater μ_{HRV} is associated with enhanced executive function resulting in faster reaction time and more correct responses to cognitive tasks [43, 42]. $SpeedRec_{HRV}$ and $AmpRec_{HRV}$ act as indicators for stress recovery. High amplitude of HRV is generally believed to promote emotional self-regulation [59, 96]. To evaluate the post-training improvements in stress reduction, we study the difference between pre- and post-training metrics and perform a series of t-tests on the difference.

Each plot in Figure 3.15 compares the 8-min HRV series of pre-training tasks and post-training tasks for subject 1 with the left from BreathCoach and the right from traditional training. We can see that HRV stays low during the first 5-min cognitive task and is elevated during the subsequent break, which supports that HRV is an indicator of stress recovery. Comparing pre- and post-training HRV series, we find that, after training with BreathCoach, there is an increment in three features: HRV amplitude during Stroop test, the speed of HRV increasing to the maximum amplitude right after 5-min task and the maximum recovery amplitude during break, suggesting an improvement in the ability to recover from stressful situation. However, these gains are hardly observed after traditional training.

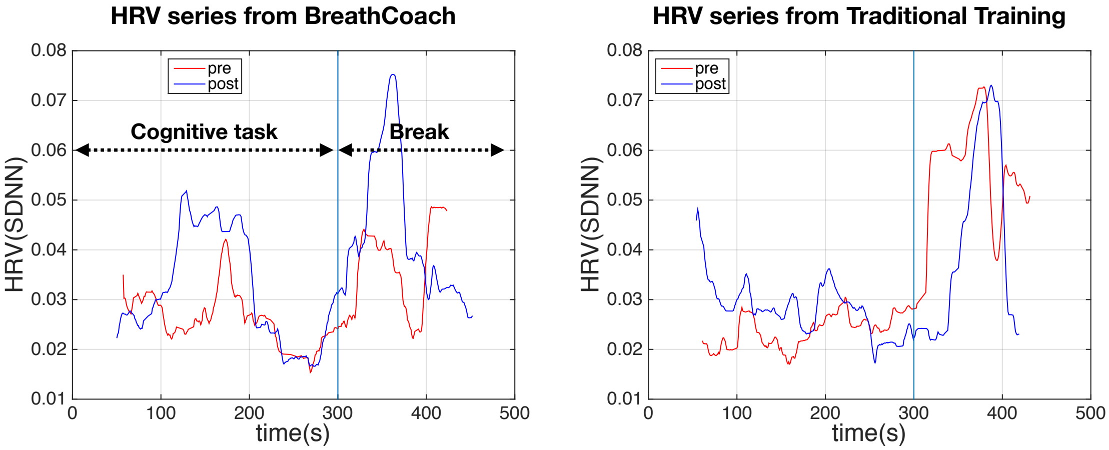


Figure 3.15: Compare the 8-min HRV series of pre-training task and post-training task for subject 1 with the left from BreathCoach and the right from traditional training. After training with BreathCoach (right), there is an increment in three features: HRV amplitude during cognitive task, the speed of HRV increasing to the maximum amplitude right after 5-min task and the maximum recovery amplitude during break. However, these gains are hardly observed after traditional training (left).

Figure 3.16 visualizes the change in μ_{HRV}, $SpeedRec_{HRV}$ and $AmpRec_{HRV}$ after BreathCoach-based training and traditional training for each subject. We can observe that μ_{HRV}, $SpeedRec_{HRV}$ and $AmpRec_{HRV}$ increased after BreathCoach-based training for most of participants, while very few participants have these three indices improved after traditional training. Specifically, when training with BreathCoach, there is a significant post-training improvements in stress reduction according to the three metrics: μ_{HRV} ($p = 0.0052$), $SpeedRec_{HRV}$ ($p = 0.0006$) and $AmpRec_{HRV}$ ($p = 0.0031$). However, the significant improvement is not observed after traditional breathing training (μ_{HRV}: $p = 0.52$, $SpeedRec_{HRV}$: $p = 0.29$ and $AmpRec_{HRV}$: $p = 0.73$).

In conclusion, our results suggest that BreathCoach is more effective than Traditional training when comes to RSA maximization, cognitive function and stress reduction. BreathCoach can improve cognitive performance while concurrently aiding stress reduction.

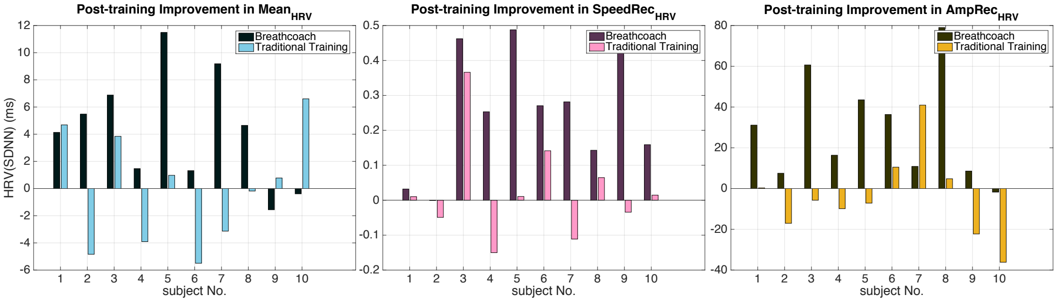


Figure 3.16: Visualize the change in the mean of HRV during the cognitive task (μ_{HRV}), recovery speed and amplitude of HRV during the post-task rest ($SpeedRec_{HRV}$ and $AmpRec_{HRV}$) after BreathCoach-based training and traditional training for each subject. When training with BreathCoach, there is a significant post-training improvements in stress reduction according to the three metric: μ_{HRV} ($p < 0.05$), $SpeedRec_{HRV}$ ($p < 0.05$) and $AmpRec_{HRV}$ ($p < 0.05$). However, the significant improvement is not observed after traditional breathing training.

3.6.3.3 Training Experience

A good training experience of RSA-BT involves participants' relaxed and stable respiration and sustained attention during training. In this subsection, training performance is studies based on both subjective and objective measurements. The self-reported scale for training experience is taken as the subjective assessment of the training experience. To examine physiological responses in relation to subjective perception, we analyze BR distribution collected during training.

Training experience is assessed subjectively through a 6-item self-report measure, which asks users the frequency of they being distracted, feeling hard to follow pacing stimulus, feeling anxious while training and trying too hard while breathing, etc. The survey is performed right after each training, as shown in Figure 3.10. Table 3.1 statistically analyzes the difference of self-reported training experience between BreathCoach-Balloon and Traditional-Balloon training using paired t-tests. We can see that, compared with traditional training, the frequency of feeling distracted, anxious, hard to follow stimulus and breathing too deeply significantly decreases when training with BreathCoach ($p < 0.05$).

Moreover, Figure 3.17 compares the distribution of BR from BreathCoach and tradi-

Table 3.1: Assess the difference of self-reported training experience between BreathCoach-Balloon and Traditional-Balloon training using paired t-tests. Compared with traditional training, the frequency of feeling distracted, anxious, hard to follow stimulus and breathing too deeply significantly decreases when training with BreathCoach ($p < 0.05$).

Frequency of.	BreathCoach M(STD)	Traditional M(STD)	p
Being distracted	0.85 (0.74)	1.35 (0.67)	0.0234
Feeling hard to follow pacing stimulus	0.8 (0.76)	2.2 (1.1)	0.00005
Feeling anxious while training	0.95 (0.6)	1.7 (0.92)	0.0004
Trying too hard while breathing	1.05 (0.75)	2.4 (0.88)	0.00001

tional training for each subject. We observe that there is a smaller variability of BR for BreathCoach as well as lower medians, suggesting that BreathCoach enables users to keep the breath steady while slowing their respiration. Traditional training can also provide a steady breathing experience, like for subject 2. However, it can not ensure steady respiration as BreathCoach do. For subject 5, BR from BreathCoach fluctuates within a narrow range, while the one from traditional training has a large variability and falls far above the RF. Additionally, we extract the STD of BR (STD_{BR}) for each training and compare the STD_{BR} collected from all BreathCoach sessions with the one collected from traditional training through paired t-test. It turns out STD_{BR} from BreathCoach is significantly lower than the one from traditional training, suggesting that BreathCoach enables users to breathe significantly more steady than traditional training does($STD_{BR} : p = 0.0019$). Given the above, BreathCoach ensures users' steady respiration, which is in agreement with users' subjective ratings.

3.6.4 Discussion of game designs

To explore the game design, we compare the BreathCoach-Balloon and BreathCoach-Pilot training for each subject. Paired t-tests reveal no significant ($p < 0.05$) differences of training effectiveness between BreathCoach-Balloon and BreathCoach-Pilot. Additionally, we collect users' game preference through the last question of the scale. There are 7 out of 10 participants who prefer Balloon over Pilot. According to the survey, this is mainly because Balloon

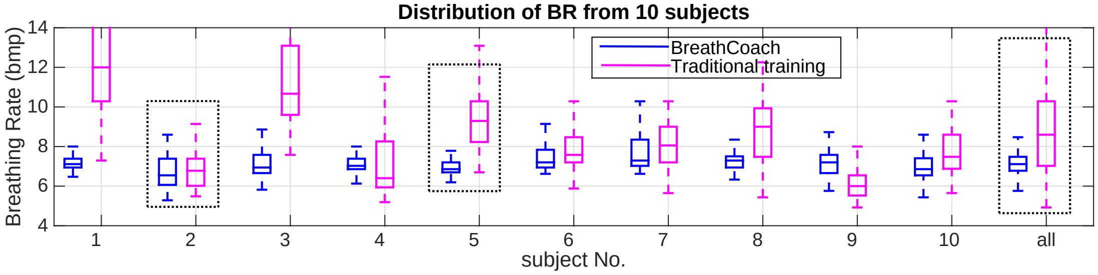


Figure 3.17: Compare the distributions of BR from BreathCoach and traditional training for each subject. It shows that BreathCoach enables users to breath significantly more steady while slowing their respiration according to the metric, the STD of BR for each training ($p = 0.0019$).

presents players their real-time training performance (i.e., how well the user's respiration is in phase with the recommended BP.) by displaying not only the recommended BP but also their respiration trace, which helps users maintain or improve training performance by adjusting their breathing. We leave further investigation of the effects of game designs as future work.

3.7 Conclusion of Study

In this thesis, we present BreathCoach – a smart and unobtrusive system that enables in-home RSA-BT using sensors on smartwatch and smartphone-based VR. To achieve this goal, BreathCoach adopts a suite of lightweight algorithms to continuously monitors BP, IBI and RSA using raw acceleration and PPG signals collected from the smartwatch. The system uses these real-time measurements to intelligently switch between two feedback mechanisms, IBI-based and RF-based, in order to derive the optimal BP. The recommended BP is then conveyed to users in the form of VR game to provide an intuitive training experience. We implemented BreathCoach using an off-the-shelf wrist-type wearable, a smartphone and a VR viewer, and designed two exploratory VR games. BreathCoach is evaluated in three aspects, including accuracy of physiological measurements, effectiveness of training, and user experience. Our experimental results collected from 10 subjects with each one performs both traditional and BreathCoach-based training indicate that BreathCoach is able to provide accurate physiological measurements with breathing cycle duration and IBI errors lower than 0.61s and 15ms respectively. Moreover, compared to traditional RSA-BT protocol, BreathCoach achieves significant improvement ($p < 0.05$) on training effectiveness and experience.

CHAPTER 4

PERSONALIZED FEDERATED LEARNING FOR HUMAN ACTIVITY RECOGNITION

This chapter introduces FedDL, a novel federated learning system for human activity recognition that can capture the underlying user relationships and apply them to learn personalized models for different users dynamically. This chapter is adapted from a publication [124]. The author of the dissertation is the first author of the original work. ”We” in this chapter refers to the author of the original publication. This work contains the phototype implementation on Amazon Elastic Compute Cloud (Amazon EC2) and the algorithm design in Tensorflow.

4.1 Background

Human activity recognition (HAR) is a key enabling technology for a wide range of applications, including smart home, health surveillance, and medical assistance [52, 133, 51]. For instance, it has been shown that longitudinal monitoring of daily routine activities, such as indoor/outdoor time, meals with/without family, and sleeping, can help to detect early onsets of Alzheimer's Disease in aged population [108, 75]. Similarly, smart home systems can conserve home energy consumption and improve residents' comfort/safety by recognizing complex home activities (e.g., eating, taking a shower, washing dishes, etc.) [32, 50].

Deep learning has recently been applied to HAR thanks to its better generalization and the ability of automatic feature extraction with less human effort [107, 41, 45]. However, several major challenges have not been addressed. The data collected from each user is usually unbalanced and sparse. Activities such as taking a shower, shopping, and biking, usually take place in a relatively low frequency. Applying deep learning to sparse and unbalanced data is likely to result in severe under-sampling artifacts. Training a global model for HAR in the cloud in a centralized manner may reduce the effect of data sparsity. However, the sensing data for HAR is often privacy-sensitive and hence cannot be shared or uploaded [29, 105].

Federated Learning (FL) is an emerging technique used to collaboratively learn a global model, such as by computing an average aggregation of local models, without exposing users' raw data [84, 115, 22, 90, 83]. Existing FL paradigms learn a single global model that however fails to capture the statistical diversity of users' data. Such statistical diversity of users' data not only leads to significant convergence delay but also poor model accuracy [13, 25, 47]. Several FL approaches have been proposed to address this problem by learning personalized models which capture both general and personal features of users [31, 28, 82, 15]. In [15], users share only lower layers of their models and leave upper layers user-specific to retain personal features. However, this approach assumes a pre-defined number of model layers shared among users, which is determined by empirical perception of user data distributions and their correlations. As a result, it suffers poor performance when the users' data distributions are highly dynamic and time-varying [109]. The post-personalized FL approach is proposed to further fine-tune the global federated model on the nodes' local data [54, 35]. However, the performance of such an approach is largely influenced by the accuracy of the global model.

4.2 Related Work

4.2.1 Deep learning for HAR.

Deep learning has been applied to improve the accuracy of human activity recognition and eliminate the human efforts of handcrafted feature extractions [101, 20, 63]. However, since many daily events, like taking a shower, shopping, and biking, only occur occasionally, one user usually has limited and unbalanced training samples, which can cause overfitting in training deep learning models [39, 30]. Data augmentation techniques may address the issue by expanding the local datasets. However, they will fail to discover the new activities in HAR when users' patterns of activities change largely. For instance, users without exercise habit start to do sports, which is not the situation that data augmentation works. As data augmentation cannot produce the data for a previous unknown activity, the model trained with data augmentation will fail to discover the new activity. Training a global model for HAR at the server is proposed to reduce the effect of data sparsity [138]. However, centralized methods require uploading users' sensing data to the cloud, leading to risk of privacy breach.

4.2.2 Federated learning (FL)

[49, 132] is an emerging learning paradigm that only requires users to upload their model weights for collaborative learning, avoiding sharing user's raw data during the learning process. A typical FL approach named FedAvg [49, 22] averages all models from users to learn a single global model, which proves to suffer significant accuracy degradation under heterogeneous data distributions of users [137, 76]. Recently several personalized FL approaches are proposed to address this issue. Dinh et al. add a regularized term to the loss function of each user's local model during the FL process to reduce the distance between the local and global models (average of all models) [31, 54, 35]. However, the accuracy of models learned in this approach can be largely influenced by the diversity of users. Moreover, other studies [28, 82] tend to introduce a post-training procedure that personalizes the learned global model on each user's local data. However, careful fine-tuning is required in this approach to balance the local and global models, which varies among different applications and hence is hard to generalize. Compared with existing personalized FL approaches, FedDL is able to learn users' relationships during the FL process and utilize them to dynamically aggregate the local models in a layer-wise manner, which is applicable to different applications with highly diverse data distributions.

4.2.3 FL personalization via model sharing.

In the FL approaches proposed in [26, 15], the lower layers between all users are shared, while several upper layers are user-specific. This design is motivated by the observation that the lower layers capture more general features, and hence can be shared across multiple tasks, whereas the top layers capture features at a higher level of abstraction and hence are more user-specific [134]. The above methods have been extended and applied to multi-task deep learning [79, 86], where the goal is to learn multiple different models. However, these multi-task methods rely on a pre-defined structure for model sharing. As network architectures become deep and the user relationship becomes more complex in large-scale HAR applications, finding the right level of feature sharing across local models through hand-crafted network branches is impractical. Moreover, most multi-task deep learning methods [95, 81] are centralized and do not address the communication efficiency of the learning process. To reduce the communication overhead of FL (especially for transferring deep learning models), previous solutions mainly focused on the techniques for model quantization [67, 111] or model compression [40]. FedDL reduces the communication overhead through the dynamic layer-wise sharing scheme, as each model merging at the server only involves the parameters of users' lower model layers, which is orthogonal to the model quantization or compression techniques. In a recent work [92], the authors show significant similarity exists among users in a number of real-world datasets, which is similar to our finding in Section 4.3. However, in [92], the clustering structure is formulated as part of the learning objective, and the local models are required to share all the layers in their multi-task learning framework. On the contrary, FedDL dynamically captures the users' relationship while learning different models for users with a partial sharing structure, which leads to better model accuracy and lower communication overhead.

4.3 A Motivation Study

In this section, we use an open real-world dataset, HARBox [9], to motivate the approach of FedDL in two aspects. First, there often exists underlying similarity amongst users' patterns of activities due to their habits of behavior or environments [114, 136, 68, 92], which can be utilized to improve the learned model accuracy by facilitating collaborations among similar users. Second, the degree of similarity among users' deep models reduces from

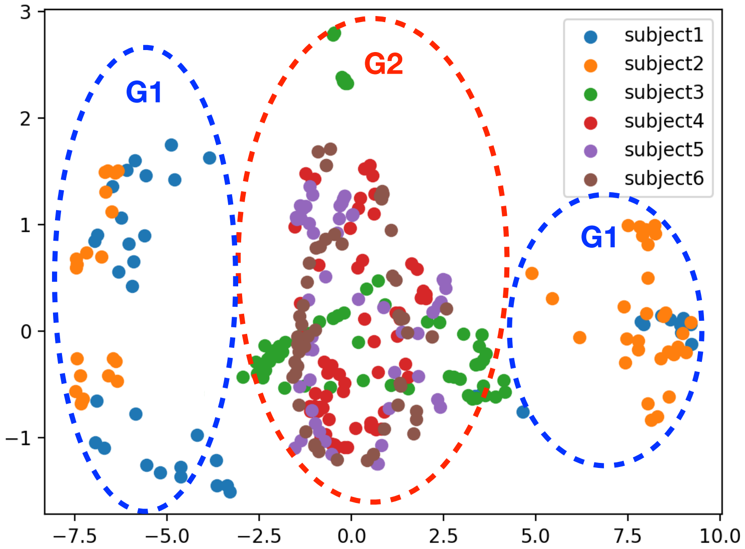


Figure 4.1: The data of "typing" from the HARBox dataset after reducing dimension to 2D using PCA. There exists a clear group relationship among different subjects' data.

the bottom up [134, 78, 88], which suggests that we may exploit such similarity of models and aggregate them in an iterative, layer-wise manner, rather than aggregating whole models. We show that such an approach improves the model accuracy and reduces communication overhead between users and the server since only partial models need to be transmitted.

The HARBox dataset is collected in real-world federated settings [92]. The 9-axis IMU data from 121 users' smartphones is recorded when the users conduct five activities of daily life (ADL), including walking, hopping, phone calls, waving, and typing. To visualize the data distribution, we plot the data of "typing" from 6 users in the HARBox dataset after reducing the dimension of features to 2D using Principal Component Analysis. As shown in Fig. 4.1, there exists a clear grouping relationship among the 6 subjects' data, with $G_1 = (n_1, n_2)$ and $G_2 = (n_3, n_4, n_5, n_6)$. We note that such similarity among users is also reported on other HAR datasets [14, 37, 53].

Our goal is to exploit the similarity among users' data to personalize their models. A natural idea is to share some model layers between similar users [82, 15]. We now explore different model sharing schemes for each user group and their impact on the shared model accuracy. Fig. 4.3 shows three sharing schemes of deep learning models for a specific user

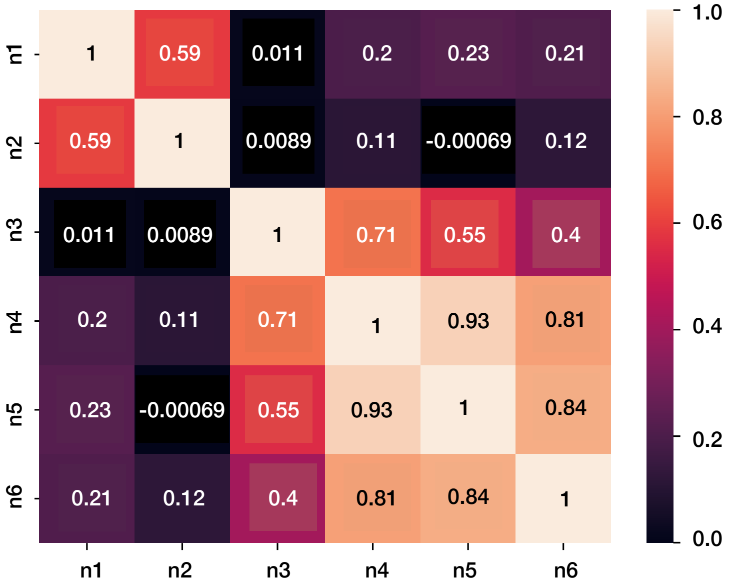


Figure 4.2: Correlation matrix of 6 users' HARBOX data. Each number is the Pearson correlation coefficient (PCC), measuring the linear correlation between two users' data. It is obvious there are two groups, (n_1, n_2) and (n_3, n_4, n_5, n_6). However, the users within each group are of different degrees of similarity.

group. The "all-sharing" scheme shares all layers of the users' models within each group. The K-sharing scheme shares only the lowest K layers of the users' models, where the number of shared lower layers K is usually empirically pre-set and fixed during the learning process. This baseline is similar to several existing FL personalization methods [15, 79]. In the experiments, we set $K = 3$ for the two groups. However, we will show that the K-sharing scheme cannot accurately capture the complicated relationship among users' data distribution. Some users are closely related enough to share more than K layers, while others with a large difference in their data distributions may benefit from sharing fewer than K layers. We visualize the relationship among data of 6 users from the HARBOX datasets through a correlation matrix by computing the Pearson correlation coefficients (PCC) between each pair of users' data. As shown in Fig. 4.2, we see there are two groups, $G_1 = (n_1, n_2)$ and $G_2 = (n_3, n_4, n_5, n_6)$. However, the users within each group are of different degrees of similarity. For instance, n_3 is less related with the other users in G_2 (the statistically independent variables have correlation coefficients close to zero). This observation inspires a dynamic sharing structure, where only users with similar data distributions should collaborate in learning and users who are more closely related to each other will share more layers of their models. Based on this idea, we design a new scheme “layer-wise sharing” shown in Fig. 4.3, which is derived according to the correlation matrix of the six users (shown in Fig. 4.2) with closer-related users sharing more model layers. Specifically, n_4, n_5 and n_6 should share more layers than n_3, since they are more closely related to each other than n_3. Shown in Fig. 4.3, n_4, n_5 and n_6 share their lower 3 layers in this example, while n_3 only shares the lower two layers with them.

We also implement a baseline “global” method where all the six users share the same global model by averaging all their layers [83], and compare its performance on HAR with three sharing schemes (shown in Fig. 4.3): all-sharing, K-sharing, and the layer-wise sharing structure derived from the correlation matrix in Fig. 4.2. Fig. 4.4 presents the model accuracy performance of n_3 when trained under different sharing schemes. We see that the model based on the layer-wise sharing structure gives the highest testing accuracy.

Motivated by this result, we attempt to generate the layer-wise sharing structure from user relationships to improve the model accuracy. However, the correlation matrix of users' data in Fig. 4.2 is global information that cannot be obtained on the server without accessing the data of users. Thus, we design a dynamic sharing scheme to learn the similarity of users' model weights and generate the layer-wise model sharing structure accordingly during FL to improve the model accuracy. Specifically, FedDL learns the grouping relationship of the local models and then merges only the lower layers of models in a bottom-up layer-wise manner. In Section 4.5, we will elaborate on the proposed dynamic sharing scheme.

In addition to the possible improvement in the training accuracy and efficiency of FL, another key advantage of our dynamic sharing scheme is that it reduces communication overhead as it is unnecessary for users to upload their user-specific layers to the server for model merging during the distributed learning process.

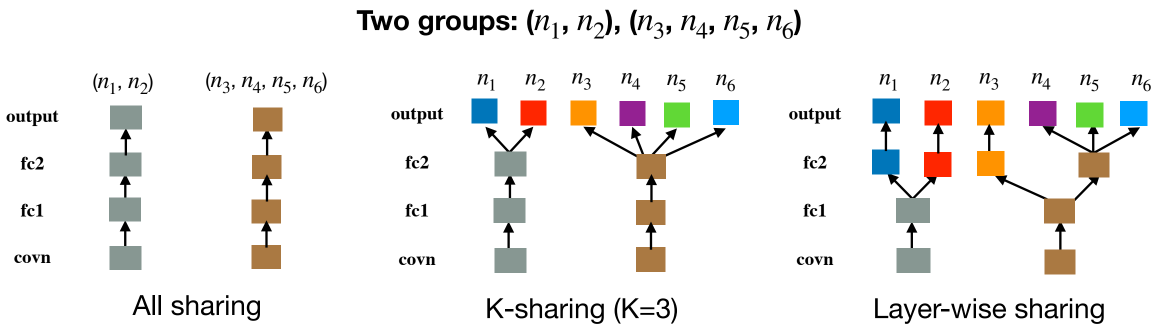


Figure 4.3: Illustration of three sharing schemes for a group.

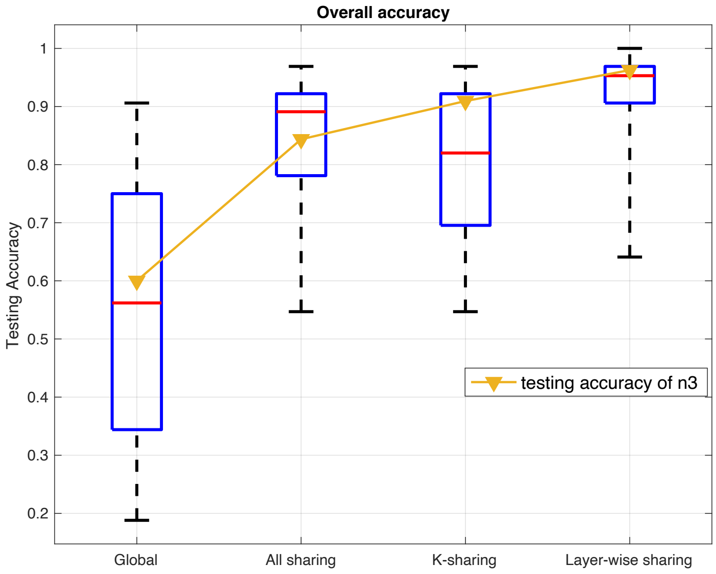


Figure 4.4: Illustration of the performance of federated learning under four sharing schemes. Layer-wise sharing scheme outperforms other sharing schemes in overall accuracy.

4.4 System Overview

This section presents an overview of the proposed Federated Learning via Dyanmic Layer Sharing (FedDL). FedDL aims to enable accurate daily activity recognition through communication-efficient deep FL, based on the underlying affinities among users' activity patterns. In this section, we first briefly introduce the application scenarios of FedDL, and then describe its system architecture.

FedDL is designed for monitoring a wide range of daily activities using sensors built in wearables or deployed in natural living environments. Representative applications include healthcare monitoring and smart home systems [32, 50]. These systems are usually designed to recognize a wide range of activities, like medicine taking, indoor/outdoor activities, and meal events, using ambient sensors and body-worn sensors [125, 106, 21]. However, since many events only occur occasionally, users tend to have limited and unbalanced training samples, which can cause overfitting in training deep learning models. Moreover, the sensing data for HAR is mostly privacy-sensitive and hence cannot be shared or uploaded. To address this issue, FedDL adopts the FL paradigm, utilizing a central server to collect local models and aggregate them, while avoiding the exposure of users' raw data during the learning process. However, models learned by FL may deliver unsatisfactory performance on recognition of each user's activities, due to the statistical diversity of users' data. To improve the model accuracy, FedDL learns the underlying relationship among users dynamically and merges the local models partially based on the degree of similarity among users in a layer-wise manner. Since the users' data distribution may change over time, FedDL will periodically update the layer-wise sharing structure and models.

FedDL features a dynamic and hierarchical FL framework that improves accuracy and communication efficiency by capturing the intrinsic relationship among users and applying it to learn layer-wise personalized models for different users. Fig. 4.5 depicts the hierarchical training procedure of FedDL. First of all, the local model of each user is optionally initialized randomly or from a pre-trained model. Then FedDL performs **model grouping** and **model merging** in a bottom-up layer-wise manner. Specifically, the server groups users based on the model affinities obtained from models' testing results on a common sample set using Kullback–Leibler divergence (KLD) (shown in Fig. 4.6(3.1)). It then performs model-merging to obtain stable models with the lower layers shared within each group. The merging process is implemented by calculating a weighted average of local models' parameters at the server over multiple rounds. Each model merging round involves 4 steps, as shown in Fig. 4.6. Users perform multiple epochs of local training and then upload local models to the server. The server computes the weighted average of local models based on grouping results. It

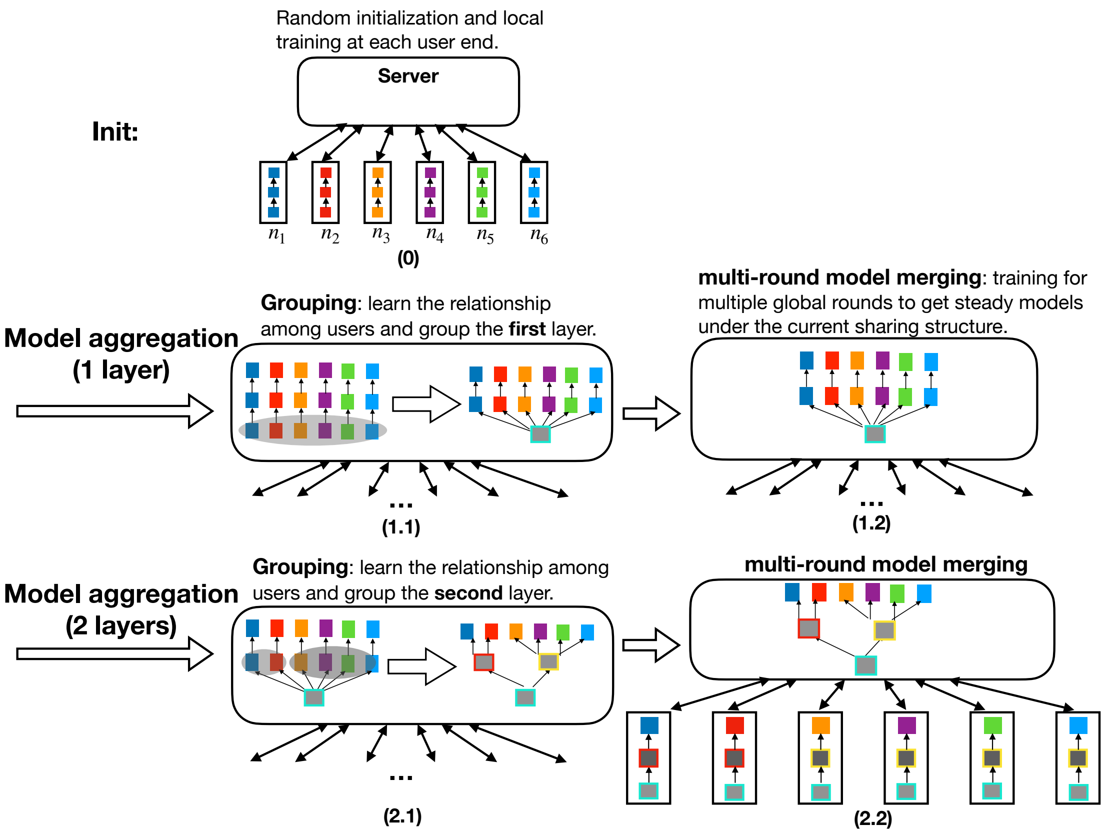

Random initialization and local training at each user end.
Server
Init:
n_1
n_2
n_3
n_4
n_5
n_6
(0)
Model aggregation (1 layer)
Grouping: learn the relationship among users and group the first layer.
multi-round model merging: training for multiple global rounds to get steady models under the current sharing structure.
...
(1.1)
(1.2)
Model aggregation (2 layers)
Grouping: learn the relationship among users and group the second layer.
multi-round model merging
(2.1)
(2.2)

Figure 4.5: Illustration of the dynamic and hierarchical federated learning framework of FedDL when learning 3-layer models for 6 users.

then generates further personalized models through the weighted average of local models and their corresponding averaged models. Finally, the server transmits personalized models back to users for local training. This model grouping and model merging process repeats till reaching the output layer (i.e., the top layer), as FedDL leaves the output layer user-specific without sharing between users.

It is challenging to learn the intrinsic relationship among users without accessing the users' data. FedDL learns the relationship among users based on their local models, and generates the sharing structure by grouping the lower model layers of closely related users, and keeps exploring the grouping relationship layer by layer within each group from the bottom up till reaching the top layer. Section 4.5.1 describes the model affinity-based grouping in detail. Moreover, based on the iteratively learned sharing structure, FedDL performs layer-wise model merging after each model grouping process to obtain stable models un-

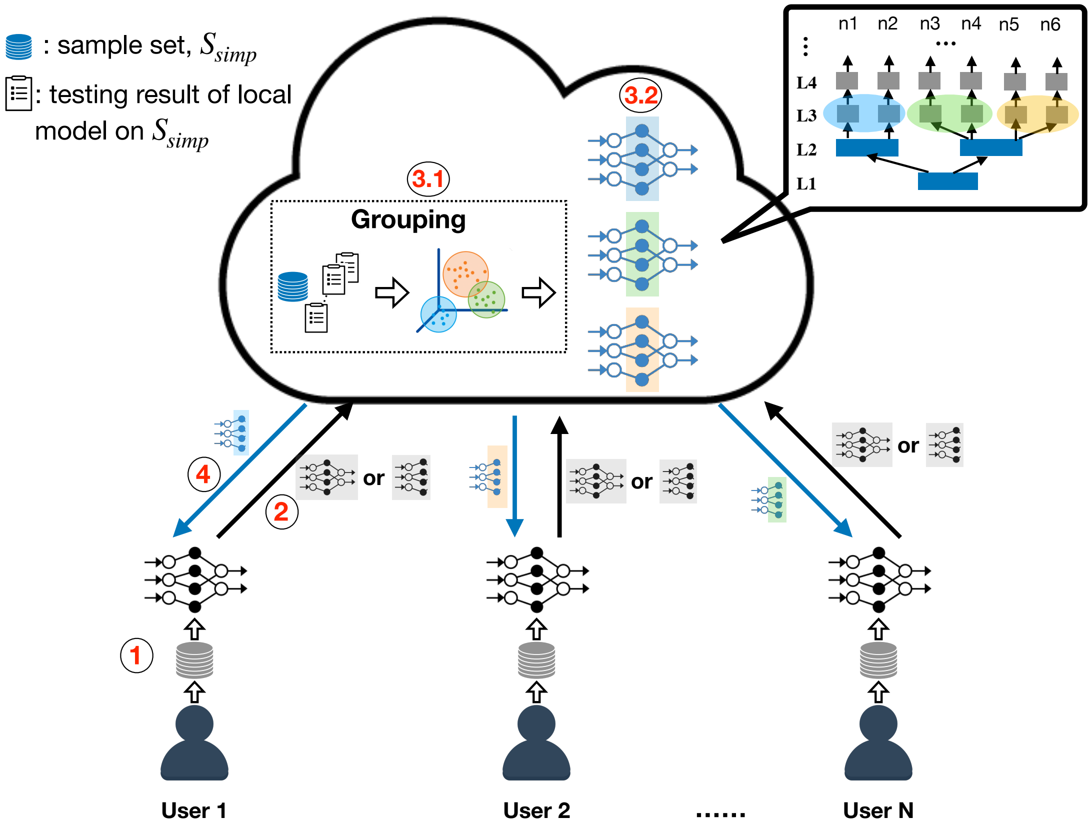


Figure 4.6: The system architecture of FedDL. Each grouping / model-merging round mainly consists of 4 steps.

der the sharing structure. Section 4.5.2 presents the design of intra-group layer-wise model merging. FedDL generates shared models in a bottom-up layer-wise manner using a greedy algorithm. Section 4.5.3 describes the detail of bottom-up layer-wise model aggregation.

The layer-wise model aggregation of FedDL improves the model accuracy through dynamic sharing within groups and reduces communication overhead by only transmitting the merged layers rather than entire models. As shown in Fig. 4.6, except for the grouping iteration when whole local models are uploaded to the server, most of the global communication involves only their lower layers, which significantly reduces the communication overhead during the FL process.

4.5 Dynamic Layer-wise Federated deep learning framework

FedDL is a federated learning framework that learns personalized deep models for users with limited or unbalanced data in HAR applications. Specifically, FedDL learns the relationship among users, generates the dynamic sharing structure for models' lower layers based on the user relationship, and merges the models according to the sharing structure iteratively. In Section 4.5.1 and 4.5.2, we presents how to group users using their deep models and how to dynamically merge different layers of models for users in the same group, respectively. In Section 4.5.3, we describe the procedure of the bottom-up layer-wise model aggregation. Finally, we introduce the design on communication efficiency in Section 4.5.4.

4.5.1 Model Affinity-based User Grouping

FedDL learns the underlying relationship of users based on their model affinities. Specifically, FedDL measures model affinities using Kullback–Leibler divergence (KLD), which estimates how one probability distribution is different from the reference one and is recently used for knowledge distillation of deep learning models [12, 10]. As demonstrated in [48, 34], element-wise weight distances (e.g., L1/L2 norms) have severe limitations in modeling affinities of deep models since the neurons of each layer in hierarchical models are permutable. Besides, it is computational inefficient to measure the norm distance of high-dimensional weights for complex hierarchical models. Therefore, instead of directly analyzing the weight matrices, FedDL tests all local models on a reference distribution in the form of a common sample set, and then measures the model affinities using the KLD of the different model outputs, as shown in Fig. 4.7. Specifically, the KLD for a pair of models, $\boldsymbol{w}_p$ and $\boldsymbol{w}_q$, is calculated as follows:

$$D_{kl}(\boldsymbol{w}_p, \boldsymbol{w}_q) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(\delta_{p,i}\log\frac{\delta_{p,i}}{\delta_{ref,i}} + \delta_{q,i}\log\frac{\delta_{q,i}}{\delta_{ref,i}}) \tag{4.1}$$

$$\delta_{p,i} = \delta(\boldsymbol{w}_p, \boldsymbol{x}_i) \tag{4.2}$$

$$\delta_{ref,i} = \frac{1}{2}[\delta(\boldsymbol{w}_p, \boldsymbol{x}_i) + \delta(\boldsymbol{w}_q, \boldsymbol{x}_i)] \tag{4.3}$$

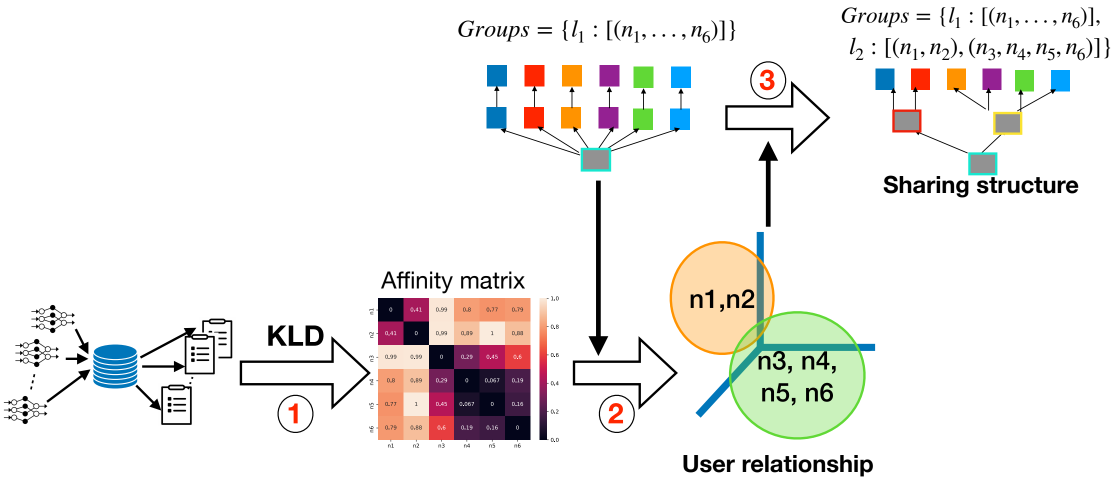

Groups = {l_1 : [(n_1, . . . , n_6)]}
Groups = {l_1 : [(n_1, . . . , n_6)],
l_2 : [(n_1, n_2), (n_3, n_4, n_5, n_6)]}
3
Sharing structure
KLD
1
Affinity matrix
n1 n2 n3 n4 n5 n6
n1 0 0.41 0.99 0.8 0.77 0.79
n2 0.41 0 0.99 0.89 1 0.88
n3 0.99 0.99 0 0.29 0.45 0.6
n4 0.8 0.89 0.29 0 0.067 0.19
n5 0.77 1 0.45 0.067 0 0.16
n6 0.79 0.88 0.6 0.19 0.16 0
1.0 0.8 0.6 0.4 0.2 0.0
2
n1,n2
n3, n4,
n5, n6
User relationship

Figure 4.7: The procedure of model-affinity-based grouping. It consists of three steps: 1. Calculate the affinity matrix; 2. Group users based on the affinity matrix and previous grouping results; 3. Update the layer-wise sharing structure.

where $\delta_{q,i}$ denotes the softmax outputs of the model w_q on the ith record, x_i, of the common sample set. $\delta_{ref,i}$ is the reference distribution. We take the average of the two models' outputs as the reference distribution and measure how these two models are different from the reference, where a lower D_{kl} value indicates a higher model affinity. Instead of directly using the KLD of P over Q, we adopt this symmetric metric for similarity measurement, which is more suitable for user grouping.

In the next, FedDL performs grouping at the l-th layer based on the model affinity and previous grouping results. Specifically, FedDL maintains an affinity matrix, M_a with $a_{(p,q)} = D_{kl}(\boldsymbol{w}_p, \boldsymbol{w}_q)$, and keeps the grouping results of lower l layers in the dictionary, *Groups*, to represent the dynamic sharing structure, as follows:

$$\begin{aligned} Groups = \{\ & 1 : [G_{1,1}, G_{1,2}, ..., G_{1,k_1}] \\ & 2 : [G_{2,1}, G_{2,2}, ..., G_{2,k_2}] \\ & ... \\ & l : [G_{l,1}, G_{l,2}, ..., G_{l,k_l}]\ \} \end{aligned}$$

where *Groups* keeps the layer index as the key and a list of groups at this layer as the value, respectively. $G_{l,i}$ denotes the i-th group for the aggregation of the l-th layer. k_l is the number of groups at the l-th layer.

With the affinity matrix, M_a, and previous grouping results *Groups*, FedDL groups the users at the server as shown in Fig. 4.7. Specifically, the l-th round of grouping operation only happens within groups that are obtained from the previous grouping round ($G_{l-1,k}$). To group users within $G_{l-1,k}$, FedDL checks the affinity between each pair of users, i and j ($i, j \in G_{l-1,k}$), and compares it with the threshold, θ_G, to decide if their models are related enough to be grouped together. We take the average of the affinities between users in $G_{l-1,k}$ as the adaptive threshold θ_G for the grouping within this group. It is noted that two less-related users may be grouped together as long as they are closely related to the same user. To differentiate the degree that users are related to their group, we consider not only the group members (m_h) but also their corresponding frequency ($freq_{m_h}$) as shown in Equation 4.4. $freq_{m_h}$ is the times the user being accessed during the procedure of grouping. A higher $freq_{m_h}$ indicates that the group member, m_h, is closely related to more users within the group. This information will be utilized to improve the accuracy of the model merging.

$$G_{l,i} = [(m_1, freq_{m_1}), (m_2, freq_{m_2}), ...(m_h, freq_{m_h})]. \tag{4.4}$$

Based on the group relationship among users, FedDL updates the layer-wise sharing structure by sharing one upper layer of users' models within each group, as shown in Fig. 4.7. FedDL performs the grouping operation periodically till the output layer. Moreover, we can stop the grouping operation in FedDL earlier, when the number of groups at a layer equals the number of users, i.e. $k_l = N$.

4.5.2 Intra-group Layer-wise Model Merging

Based on the grouping results, *Groups*, FedDL merges the local models in a layer-wise manner. Fig. 4.8 illustrates the layer-wise model sharing at the server. Based on the grouping results of the lower 3 layers, the local models from users in the same group are merged layer

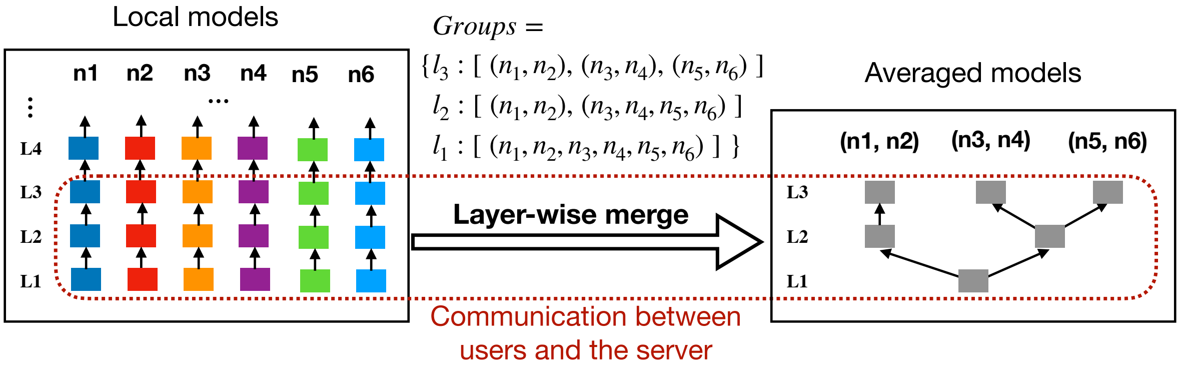

Local models
n1
n2
n3
n4
n5
n6
L4
L3
L2
L1
Groups =
{l3 : [(n1, n2), (n3, n4), (n5, n6)]
l2 : [(n1, n2), (n3, n4, n5, n6)]
l1 : [(n1, n2, n3, n4, n5, n6)] }
Layer-wise merge
Averaged models
(n1, n2)
(n3, n4)
(n5, n6)
L3
L2
L1
Communication between users and the server

Figure 4.8: Illustration of the layer-wise model merging based on the grouping results, *Groups*. Only lower 3 layers of models are transferred between six users and the server for model merging.

by layer, as follows:

$$\boldsymbol{W}_{G_{l,k}} = \sum_{i \in G_{l,k}} \mu_i \boldsymbol{W}_{i,l} \tag{4.5}$$

$$\mu_i = \frac{freq_i}{\sum_{j \in G_{l,k}} freq_j} \tag{4.6}$$

where $\boldsymbol{W}_{G_{l,k}}$ is the weights shared by the users in $G_{l,k}$, the k-th group at l-th layer. $\boldsymbol{W}_{G_{l,k}}$ is a weighted average of all the group members' layer weights, $\boldsymbol{W}_{i,l}$. The weighted average coefficient, μ_i, of each group member is calculated based on the $freq_i$, which indicates how close the member is tied to the group. As a result, the models with higher $freq$ will contribute more to the group model.

After merging the layers of the models into shared models, the server further personalizes the shared models for each user by aligning each local model with its corresponding group model as follows,

$$\boldsymbol{W}'_i = (1 - \lambda_i)\boldsymbol{W}_i + \lambda_i \boldsymbol{W}_{G_{l,k}} \tag{4.7}$$

$$\lambda_i = min(1, \frac{\mu_i}{1/sizeof(G_{l,k})}) \tag{4.8}$$

where $i \in G_{l,k}$. λ_i indicates, from the user's stand, how closely local model $\boldsymbol{W}_i$ is related to the group model, $\boldsymbol{W}_{G_{l,k}}$. This alignment makes the models trained using FedDL robust to boundary cases, where the least related users are still included in a group (i.e., with the smallest μ_i). These users are likely to become a separate group in another training process.

For instance, at a certain layer, three users are grouped as $\{1 : 0.5, 2 : 0.49, 3 : 0.01\}$, and the training process produces the grouping result $\{1 : 0.5, 2 : 0.5\}$ and $\{3 : 1\}$ at the same layer. Without alignment, the models of user 3 obtained from the two training processes are significantly different with $\boldsymbol{W}'_3 = \boldsymbol{W}_G$ and $\boldsymbol{W}'_3 = \boldsymbol{W}_3$ respectively. However, after the alignment between the shared models and users' local models, the models of user 3 under these two grouping results become similar with $\boldsymbol{W}'_3 = 0.03\boldsymbol{W}_G + 0.97\boldsymbol{W}_3$ and $\boldsymbol{W}'_3 = \boldsymbol{W}_3$ respectively.

As illustrated in Fig. 4.8, in this model-merging round, we get three shared models: [$\boldsymbol{W}_{G_{1,1}}$, $\boldsymbol{W}_{G_{2,1}}$, $\boldsymbol{W}_{G_{3,1}}$], [$\boldsymbol{W}_{G_{1,1}}$, $\boldsymbol{W}_{G_{2,2}}$, $\boldsymbol{W}_{G_{3,2}}$] and [$\boldsymbol{W}_{G_{1,1}}$, $\boldsymbol{W}_{G_{2,2}}$, $\boldsymbol{W}_{G_{3,3}}$] shared within the three groups, $(n1, n2)$, $(n3, n4)$ and $(n5, n6)$ respectively. Finally, the three shared models are aligned with their corresponding users' local models. For example, the second shared model will be aligned with the local models of n_3 and n_4 and sent to them, respectively. Moreover, only the lower layers of models are necessarily transferred between server and users during the model-merging iterations, which will significantly reduce the overall communication overhead during the FL process.

4.5.3 Bottom-up Layer-wise Model Aggregation

At the core of FedDL is the multi-round greedy model aggregation in a bottom-up layer-wise fashion.

Consider a situation where there are N users. Initially, all the users start with the same neural network model and initialize it randomly or from a pre-trained model. After users perform multiple epochs (denoted as R) of local updates, the server will receive the latest N local models from all the users. The model aggregation operation of the server starts from the bottom layer, l_1. It will first group the N branches into k_1 groups where $k_1 \leq N$. After that, FedDL will greedily perform the bottom-up model aggregation within their groups. We note that finding the optimal sharing structure is combinatorial prohibitive. A brute-force method would need to train and test all the $((C_N^N)^{L-1}$ possible structures for finding the optimal aggregation scheme for N users with L-layer models. Our approach is more efficient since it only takes $O(N \log N * L)$ time. For each round of model grouping, it takes $O(N \log N)$ time, and takes a total of $O(N \log N) * (L - 1)$ time at the worse case to form the sharing structure.

For L-layer deep models, FedDL will perform L rounds of user grouping with each grouping round followed by *intvl* rounds of model merging, as shown in Fig.4.5. At each grouping round, FedDL learns the affinity relationship of local models and groups one upper layer of the users' models into groups. After that, FedDL performs multiple rounds of model merging within each group according to the current sharing structure. It is noted that the interval between grouping rounds, *intvl*, decreases with a decay rate, λ. When more layers of local models are merged according to their layer-wise similarity, the divergence among local models reduces gradually. Therefore it takes fewer global training rounds for the shared models to converge [78, 88]. Specifically, the procedure of model aggregation for the lower l layers is as follow:

1. **Grouping round**: the users send their complete models to the server. The server groups the users based on current model affinities and the grouping results from $l-1$th iteration, $Groups[l-1]$. It was noted that the grouping operation at lth iteration only happens within each group obtained from $(l-1)$th iteration, i.e., the users being separated into different groups in the first $l-1$ rounds no longer share their upper layers. Moreover, the grouping result will be added to *Groups*, where the grouping results of all the lower layers are kept for the model merging process.

2. **Model-merging round**: After the model grouping, FedDL performs *intvl* rounds of layer-wise model merging within each group. For each model-merging round, the clients perform R epochs of local training and then upload the lower l layers of their local models to the server. Upon receiving all local models, the server weighted averages the lower l layers of local models' within each group based on the grouping structure,

> *Groups*, and then aligns each local model with its corresponding group model to generate the shared model for each user. At the end, the server sends the shared models to their corresponding clients. It is noted that, for global communication, only the lower l layers of models are transferred between users and the server, which makes the FedDL very communication-efficient.

The grouping operation stops at the layer before the output layer. As a result, the higher layers of each model will be user-specific, while the lower shared layers will ensure generality across similar users. Moreover, the grouping structure and the models of FedDL will be updated periodically with continuously collected data.

Fig. 4.5 shows the procedure of learning a 3-layer model for 6 users. As shown in Fig. 4.5(1.1), after grouping based on the initial local training models, all the users are grouped together for model aggregation. As a result, the first layers of their models are merged as the group model. After that, FedDL performs the model aggregation operation for the lower two layers within the groups obtained from the previous round, i.e., $\{n_{1\ 6}\}$, as shown in Fig. 4.5(2.1) and (2.2). The server groups the users into two groups, $\{n_1, n_2\}$ and $\{n_3, n_4, n_5, n_6\}$ and updates the sharing structure with parameters of the second layer shared within each group. The dynamic sharing structure is finalized after the 2-round model aggregation, and FedDL keeps the output layer user-specific.

4.5.4 Reducing Communication Overhead

In typical FL systems [83, 115, 22, 90], a large number of global communication rounds between users and the server is required, which can be the bottleneck of the learning process. FedDL takes advantage of the dynamic layer-wise sharing scheme to improve communication performance. Specifically, FedDL reduces the number of parameters that each user needs to upload to the server as well as the number of global training rounds.

As shown in Fig. 4.5, after learning the grouping results for the l-th layer, only the lower l layers of local models need to be merged at the server for each global training. Thus, FedDL uploads the lower l layers of local models for the global model merging where l increases from 1 to $L-1$ during the training process, largely reducing the amount of data transferred. Besides, FedDL further reduces the communication overhead by stopping the upload of each model's user-specific layers whenever possible. As shown in the left of Fig. 4.3, at the third layer, n_1, n_2 and n_3 no longer belong to any group, i.e. their layers are user-specific. After obtaining the fourth layer's grouping results, unlike $n_4 - n_6$, n_1, n_2 and n_3 need to upload only the lower 2 layers to the server during all the following model merging rounds.

Moreover, the models trained using FedDL converge fast even with a small number of local training rounds, which is detailed in Section 4.6. Thus, FedDL can use a small number of global training rounds to reduce communication costs. As shown in Section 4.6, FedDL-based models can always converge within 10 global rounds with different settings of local rounds R, while other FL methods may take more than 30 rounds to converge. Therefore FedDL largely reduces the communication overhead.

4.6 Evaluation

In this section, we evaluate the performance of FedDL from three aspects, including the performance on different datasets, the scalability of the system, and its performance with different local computation rounds. For each evaluation, we compare the performance of FedDL with four baselines as follows:

1. **FedAvg** [83]: the standard FL method, where all users share one global model.

2. **FedPer** [15]: a federated deep learning approach, where all the users share their lower K layers and leave their upper layers user-specific. This approach adopts a K-sharing scheme and pre-sets the value of K empirically, as mentioned in Section 4.3. In our experiments, we set K to be 3.

3. **pFedMe** [31]: an algorithm for personalized FL using the distance between the global model and the user's local model as the user's regularized loss functions. The global

Table 4.1: Five HAR datasets (UWB, Depth Images, HARBOX-IMU, IMU and LiDAR).

Application	Tasks	Sensor	Data Dimension	Number of subjects	Number of records per subject
Human movement detection	with/without human movement	UWB	50x1	8	$\sim$ 80
Hand Gesture Recognition	good/ok/victory/stop/fist	Depth camera	36x36x1	9	$\sim$ 400
Activity of Daily Life (ADL) recognition using IMU	walking/hopping/phone calls/waving/typing	IMU	100x9x1	121	$\sim$ 300
Human Activity Recognition using IMU	walking-upstair/ walking-downstair/ walking/sitting /standing/laying.	IMU	128x3x2	30	$\sim$ 300
Human Activity Recognition using LiDAR	walking/bending/phone calls/sitting/standing/ checking watch.	Livox Horizon LiDAR	60x30x1	10	$\sim$ 600

model is an average aggregation of all the local models at the server.

4. **Local training**: the model learned from local data at each user.

4.6.1 Datasets

In our evaluation, we use one self-collected dataset and four public real-world datasets (Table 4.1) for deep learning. We use the self-collected LiDAR data for two main reasons. First, most of the existing HAR datasets lack detailed information about subjects, such as gender, height, and weight. Such information is critical to understand the underlying similarity of users' data and hence is important to validate the design of FedDL. Second, LiDAR has a long detection distance, which facilitates recognizing whole-body movements, like bending and falling. At the same time, compared with RGB images, Lidar data is more spare presents a major challenge in achieving high model accuracy, which motivates the adoption of FL to enable collaborative learning from multiple users.

Moreover, we choose additional four public datasets for evaluation as they are collected in real-world settings with significant dynamics. Besides, these datasets are collected from various HAR tasks based on different sensors, like depth sensor and IMU (inertial measurement units). Moreover, some datasets are of large scale, which can be utilized to evaluate systems' scalability.

1. **Human Activity Recognition using LiDAR**[1] : We record the point cloud data of 6 types of human activities (walking/sitting/standing/bending/checking watch/phone calls) conducted by 10 subjects using a Livox Horizon LiDAR [77] in an indoor environment. The LiDAR collects point clouds at 10Hz, and each activity of a subject lasts for 2 minutes. Fig. 4.9 shows the preprocessing steps for the collected point clouds, which are first proposed in [18, 87]. First, we conduct the cylinder projection to project the 3D point cloud to a range image of 120 × 30 pixels, where each pixel's grayscale represents the range value (the whiter, the farther). Then we average every 25 consecutive range images in sliding windows of 2.5 seconds and 50% overlap to form each data record. After that, the ROIs (region of interest) of each image are retrieved, and then we down-sample the original image to 60×30 pixels and normalize the depth value to 0-1. This dataset has a large number of data records (6560 records in total), and each data record's dimension is relatively high, thus increasing the difficulty of activity recognition.

2. **Human Movement Detection using UWB** [9]: To detect if there were human movements in a specific area, two UWB (Ultra Wide Band) nodes are deployed 3m away from each other in 3 different environments (i.e., parking lot, corridor, room) with or without a person walking between them. This dataset is collected using 8 subjects, with each one walking randomly in the area for 10 minutes. The two-way ranging at 5Hz is captured and labeled manually. Then the data is sampled in sliding windows

[1] The data collection was approved by IRB of the authors' institution.

of 10 seconds and 50% overlap (50 readings/window) to form each data record (50×1 dimensions).

3. **Hand Gesture Recognition using Depth Camera** [9]: Five types of gestures (good/ok/victory/stop/fist) are conducted by 8 subjects using a depth camera. The region of interest of the depth image is retrieved, and then we down-sample the original image to 40×40 pixels and normalize the depth value to 0-1.

4. **Activity of Daily Life (ADL) Recognition using Smartphones** [9]: The "HARBOX" App is developed to collect 9-axis IMU (inertial measurement units) data from users' smartphones when the user conducts five types of ADL, including walking, hopping, phone calls, waving and typing. Labeled IMU data from 121 users is collected in total. The data from each user is filtered and then sliced into multiple frames (100×9 dimensions) using a window of 2 seconds and 50% overlap.

5. **Human Activity Recognition using Smartphones** [2]: this online dataset is collected from 30 subjects performing six activities (walking, walking_upstairs, walking_downstairs, sitting, standing, lying) while carrying a waist-mounted smartphone (Samsung Galaxy S II) with embedded IMU. Specifically, the 3-axial linear acceleration and 3-axial angular velocity are captured at a constant rate of 50Hz and are labeled manually through video records. The 6-dimension sensor signals were pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 seconds and 50% overlap (128 readings/window) to form each data frame with a size of 128×6.

4.6.2 Implementation

We design and implement a FedDL phototype on Amazon Elastic Compute Cloud (Amazon EC2). This EC2 instance is built on the Ubuntu platform and has 96 virtual CPUs (3.1

[2] https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

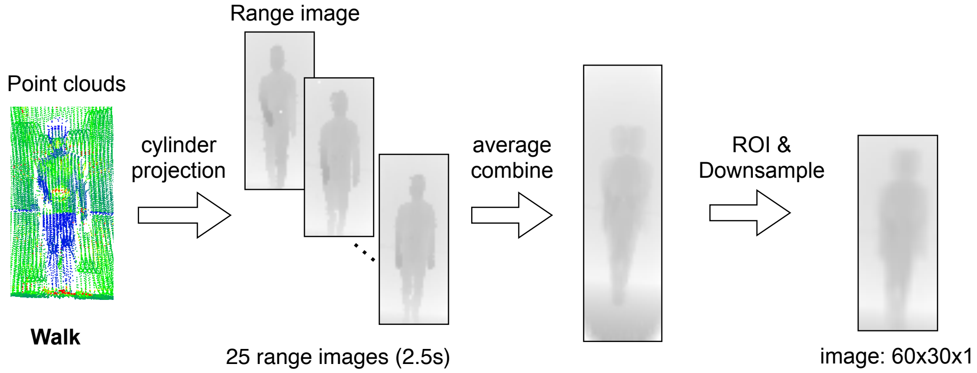


Figure 4.9: The preprocessing of LiDAR data for the recognition of activities, including walking, sitting, standing, bending, checking the watch and phone calls.

GHz) and 768 GB memory. We build a server on the instance and run each user end on one CPU to simulate the FL. The communication between the server and users is implemented locally using sockets. The system is implemented in Python3.

We adopt randomly initialized convolutional neural networks (CNN) for the human activity recognition tasks of the five datasets. The CNN network is composed of 2 convolutional layers, 2 full-connect layers, and one softmax output layer. It uses mini-batch Stochastic Gradient Descent (SGD) for optimization. For the data samples of each subject, we use 75% of the local data for model training, while the rest 25% is for model testing. We set the initial learning rate to be 0.01 with periodical decay and the batch size to be 32. Although with the same depth, the CNN models for different datasets will have various network structures (e.g., input dimension, kernel size, stride, and padding) depending on the data characteristics and the tasks.

4.6.3 Validation on LiDAR Dataset

In this section, we validate the design of FedDL on the LiDAR dataset. Specifically, we compare the performance of FedDL with four baselines, FedAvg, FedPer, pFedMe, and local training. We set the local communication rounds (R) to be 30 and the global computation rounds (T) to be 40. We involve totally 10 users for the FL on the LiDAR dataset.

Fig. 4.10 shows the overall accuracy and the communication overhead of different ap-

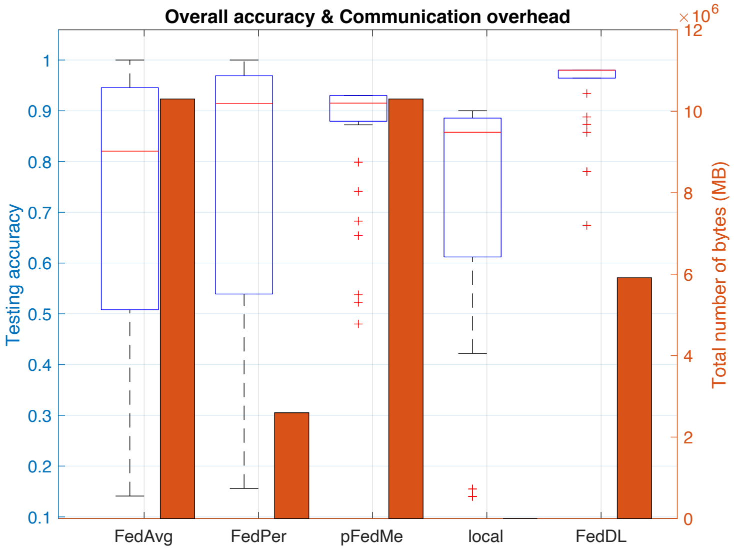


Figure 4.10: Comparison of different approaches' performance on the LiDAR dataset. FedDL outperforms other approaches in accuracy performance by more than 15%, and save about 42.6% communication overhead compared with approaches that share the whole models (Fedavg and pFedMe).

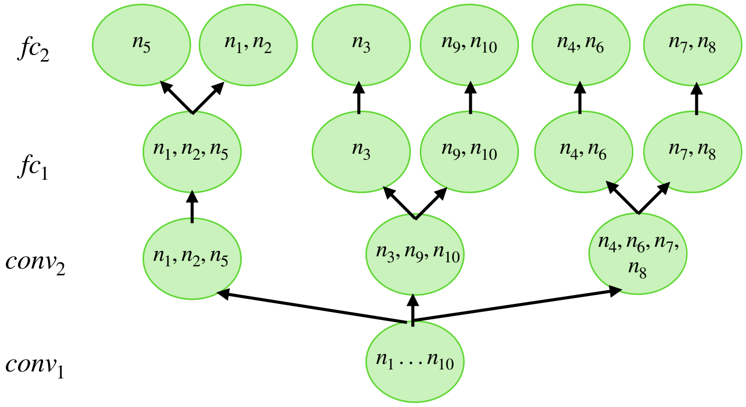


Figure 4.11: The sharing structure for 10 users, which is dynamically learned by FedDL. n_2, and n_1 share more layers as they have similar behavior habits and biological features.

proaches on the LiDAR dataset. We evaluate the overall accuracy by observing the distribution of testing accuracy after 30 rounds of global training for all the users. From Fig. 4.10, we can see that FedDL achieves the best accuracy performance with $mean_{acc} = 0.98$ and the interquartile range $IQR = 0.025$. Compared with other methods, FedDL improves the mean testing accuracy by more than 15% and reduces the variation significantly by more than 94%, suggesting that FedDL can converge fast to a steady and accurate model for most users. In contrast, FedAvg and FedPer yield larger testing accuracy variations as models of some users barely converge even after 30 rounds of global training. FedDL achieves a significantly lower variation as it facilitates the collaboration among users with similar data distributions, which mitigates the noise/outliers from other users, improving the convergence rate and accuracy. Fig. 4.10 also compares the communication overhead of FedDL with the other three FL methods. We measure the communication overhead (Q_{comm}) by calculating the total amount of data transferred between the server and users during the training procedure. It is shown that FedDL saves about 42.6% communication overhead compared with FedAvg and pFedMe, which share the entire models during FL.

To better understand the above results, we take a closer look at the sharing structure dynamically learned by FedDL (shown in Fig. 4.11). From the figure, we can see, n_3, n_9, and n_{10} share the lower two layers, which is consistent with the fact that they are the only three subjects using the left hand to make phone calls. Among these three subjects, n_9 and n_{10} are females (n_9: heights 1.66m, weights 50kg; n_{10}: heights 1.63m, weights 48kg), while n_3 is a male with the height 1.78m and weight 66kg. It is shown that n_9 shares more layers with n_{10} than with n_3, which can be attributed to the distinct effects of the body shapes on the collected LiDAR data. The effect of biological features on the LiDAR data is also reflected on n_5. Users n_5, n_2, and n_1 use both the left and right hands to answer phone calls, and they are all males. However, n_5 (height 1.93m, weights 95kg) is much taller and heavier than the other two subjects. In the sharing structure, n_5 shares the lower 3 layers with n_2 and n_1, while n_2 and n_1 keep sharing more upper layers.

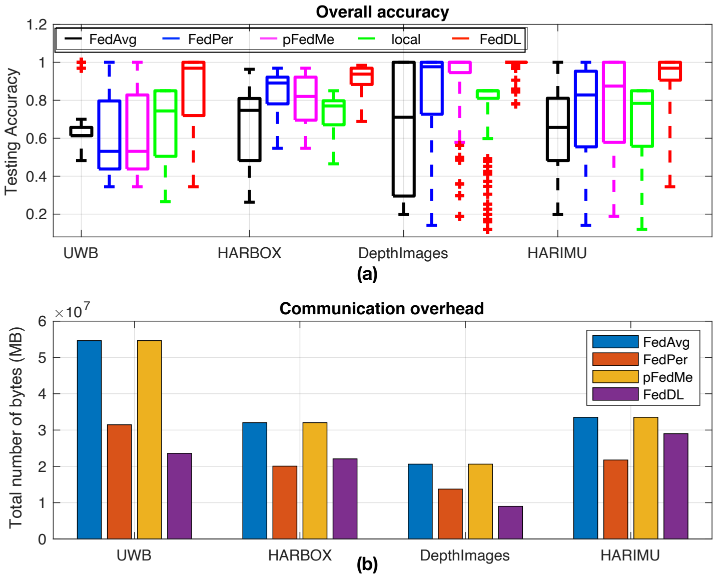


Figure 4.12: Comparison of different approaches' performance on four datasets, UWB, HARBOX, Depth Images and IMU. FedDL outperforms other approaches in accuracy performance and has a lower communication overhead than approaches that share the whole models (Fedavg and pFedMe).

The above results confirm that FedDL can capture the different degrees of similarity among users' data due to behavior habits or biological features, and can effectively apply them to layer-wise model merging to improve model accuracy and communication efficiency.

4.6.4 Performance on Different Datasets

In this section, we evaluate the performance of FedDL on different datasets, UWB, HARBOX-IMU, depth images, and IMU (Table .4.1). Specifically, for each dataset, we compare the overall accuracy and communication overhead of FedDL with four baselines, FedAvg, FedPer, pFedMe, and local training. We fix the local communication rounds (R) to be 30 and the global computation rounds (T) to be 40 for all the approaches. Also, we involve 8 users for the FL on each dataset, where the number of data samples varies for different users to simulate an unbalance data setting in FL. It is noted that we evaluate the scalability of FedDL on HARBOX dataset involving up to 90 users in Section 4.6.4.

Overall accuracy. Fig. 4.12(a) compares the testing accuracy of different approaches for the four datasets. It is shown that compared with four baselines, FedDL achieves the best and stable accuracy performance on the four datasets with a high mean value ($mean_{acc} > 90\%$) and $IQR < 0.2$. Specifically, compared with local training ($0.05 < IQR < 0.4$, $75\% < mean_{acc} < 85\%$), FedDL, FedPer, and pFedMe improve the accuracy of the model while FedAvg fails, as the data distributions of users are too heterogeneous to learn a good global model. Specifically for the UWB dataset, FedAvg barely converges within 40 global rounds. FedPer and pFedMe also fail to improve the accuracy as their model aggregation schemes are oblivious to the underlying relationship among users. Moreover, FedDL outperforms them, as FedDL can capture the intrinsic relationship among users dynamically and aggregate users' models within each group in a layer-wise manner.

Communication overhead. Fig. 4.12(b) compares the communication overhead of different methods for the four datasets. In our experiments, we set the number of global rounds $T = 40$. The communication overhead measures the total amount of the parameters transferred between users and the server during the whole FL process, which is determined by the sharing scheme and the size of the CNN model. From the figure, we can see FedDL is able to maintain a relatively low communication overhead, which suggests our dynamic bottom-up layer-wise model aggregation strategy improves the communication efficiency. Specifically, FedDL and FedPer have a relatively low communication cost for all the datasets, as they only share part of model layers among users. FedPer combines the lower 3 layes of local models and FedDL merges models according to layer-wise grouping results. In particular, FedDL outperforms FedPer for UWB and depth images datasets. The reason is that the data distributions of users are so heterogeneous in these two datasets that most of the users' upper layers are user-specific in FedDL's grouping results, i.e., they share and upload less than 3 lower layers.

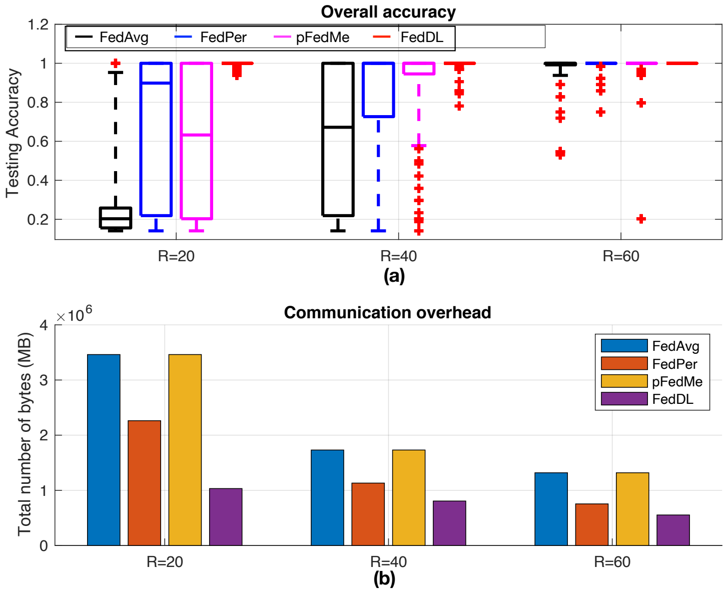


Figure 4.13: Comparison of different approaches' performance on Depth images datasets with different number of local computation rounds ($R = 20, 40, 60$). All the methods benefits from a larger R, and FedDL maintains the best accuracy and communication performance with different numbers of R.

4.6.5 Scalability

To evaluate the scalability of FedDL, we compare the performance of different approaches (FedDL, FedAvg, FedPer, pFedMe) when training on the data of 30, 60, 90 users from the HARBOX dataset.

4.6.5.1 Overall accuracy

. Fig. 4.14 shows the experiment results with different number of users. From Fig. 4.14(a), It is obvious that the overall accuracy of FedAvg decreases with the increase of the number of users, as the heterogeneity of users' data becomes higher. In this case, FedAvg performs the worst among all approaches and can not even converge within 40 global rounds when 90 users are involved. Besides, FedDL outperforms FedAvg, FedPer and pFedMe under different settings as FedDL can capture the relationship among users and dynamically merge user' models within each group in a layer-wise manner. On the contrary, FedPer adopts a static sharing scheme that shares the lower 3 layers of models for all the users, which

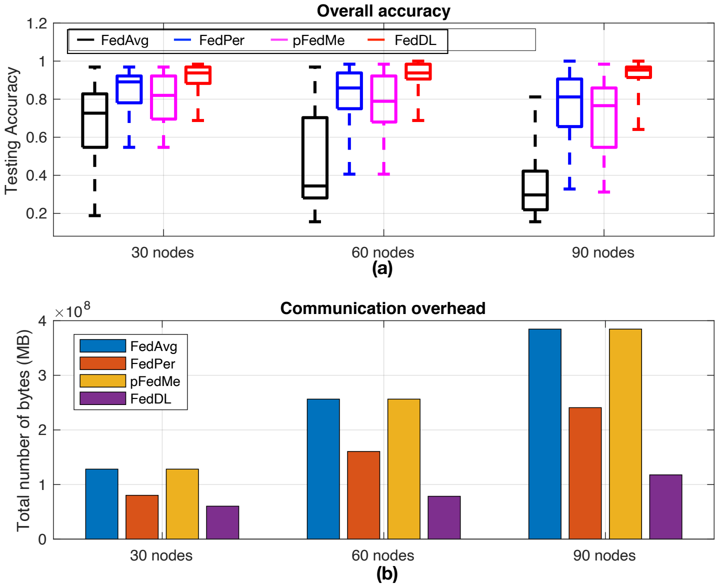


Figure 4.14: Comparison of different approaches' performance on 30-, 60- and 90-user HARBOX datasets. FedDL outperforms FedAvg, FedPer and pFedMe in both overall accuracy and communication overhead.

fails to capture the complicated user relationship, resulting in worse performance when the number of involved users is large. pFedMe aligns each user's local model with the averaged global models, which makes the overall accuracy partially dependent on the global model's performance, which is hence largely influenced by users' data heterogeneity. Moreover, the accuracy of FedDL is more stable (with small IQRs) as the number of users increases, which shows the advantage of its group-based dynamic model aggregation scheme.

4.6.5.2 Communication overhead

. Fig. 4.14(b) compares the communication overhead of different approaches with the data of 30, 60, 90 users from the HARBOX datasets. We can see that the communication overhead of FedAvg, pFedMe and FedPer increases dramatically in proportion to the number of users involved in the training procedure. However, FedDL always maintains a relatively low communication overhead, as FedDL can stop uploading the parameters of models' upper layers earlier when the users' data is significantly heterogeneous.

The above results suggest that FedDL exhibits satisfactory scalability by maintaining relatively high accuracy and low communication overhead and performs better on large-scale datasets.

4.6.6 Impact of Local Computation Rounds

The number of local computation rounds, R, is a critical hyperparameter in FL. The setting of R shows a trade-off between the computation and communication: a larger R requires more computations at local devices of users, while a smaller R means more global communication rounds to converge. To understand how R affects the convergence of different FL methods, we conduct the experiments on an 8-user Depth Images dataset with ($R = 20, T = 30$), ($R = 40, T = 15$) and ($R = 60, T = 10$, respectively. It is noted that, for all the baselines, we only change the value of R with the model structure and all the other settings of the models stay the same. Specifically, the initial learning rate is set to be 0.01 with periodic decay and the batch size is set to be 32.

Fig. 4.13 illustrates the performance of different methods with different settings of local computation rounds R. It shows that a larger value of R will improve the performance on the accuracy and communication overhead of both the personalized and the global models. Fig. 4.15 visualizes the change of training loss and testing accuracy over global rounds with different settings of R for a specific user. We can see that all the methods have improvements in convergence when R is larger. For example, FedAvg takes a much smaller number of global communication rounds to converge (reduce from more than 30 to 10 rounds) when R increases from 20 to 40. However, FedDL will always converge fastest (with the smallest number of global rounds), especially when the local computation round R is set small (e.g., R=20).

4.7 Discussion and Future Work

4.7.1 Convergence of FedDL

. In our experiments (discussed in Section 4.6.3-4.6.6), FedDL is demonstrated to converge on the five real-world HAR datasets. In particular, it converges fast even when training

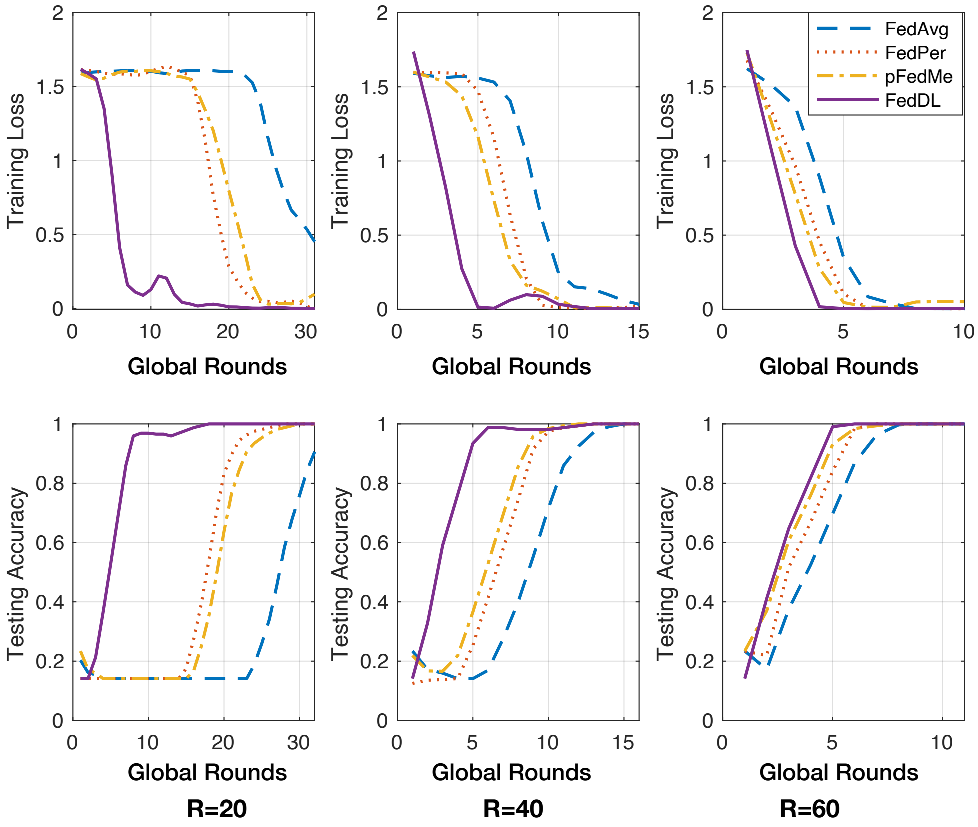


Figure 4.15: The training loss and testing accuracy of a specific user's model changing over global rounds with different settings of R. Larger R improves convergence, especially for FedAvg. However, FedDL will always converge fastest with different local computation rounds R.

on 90 users with a limited number of local rounds. We now provide some insights into the convergence guarantee of FedDL. Firstly, FedDL groups users with similar data distributions, which mitigates the impact of noise/outliers from other users, thus improving the convergence performance. Second, the intra-group model merging entails a weighted average of the local models (see section 4.5.2), where the weights quantify how closely each local model is to the group model. In FedDL, the weights of users whose models lie at the border or intersection of multiple groups are relatively small, and hence the models will contribute less to the intra-group model merging. Thus, such a design mitigates the impact of dynamic grouping on model convergence.

4.7.2 Scalability of FedDL

. FedDL is generally more scalable as a clustering-based approach since the number of user groups (who share some degree of similarity among their data) may not increase drastically with the number of users. For the scenarios where users arrive dynamically, FedDL merges the new users in the sharing structure instead of retraining the sharing structure for all the users for scratch, which substantially reduces the compute and communication overhead. Specifically, FedDL considers each group as one user and learns the new users' relationship with existing groups to update the sharing structure by merging the new users into different groups.

4.7.3 Future work

. Firstly, the local models transmitted in FedDL may reveal certain information about user activities [56, 131]. In the future, we will integrate additional mechanisms, like differential privacy [129], in FedDL to provide stronger privacy protection. However, such privacy-preserving mechanisms can have a complicated impact on the overall performance. We will conduct a comprehensive study of privacy-preserving techniques and the trade-off between the privacy and performance of FedDL. Besides, we will extend FedDL to other applications where the users' data has a high level of dynamics while exhibiting significant similarity. For example, FedDL can be applied to applications like health monitoring [11] and road traffic prediction [85], where the data of nodes (e.g., users or cars) share spatial-temporal similarity due to spatial proximity, models of devices/cars, user routines, etc. Finally, as the real-world HAR applications may involve high-dimension data (e.g., images or videos), deeper or wider neural network models are required to avoid underfitting. We will evaluate how the model complexity, including the depth and width of the model, affects the convergence and accuracy of FedDL.

4.8 Conclusion of Study

This thesis proposes FedDL, a novel federated deep learning system for HAR that captures the similarity of users' models and generates personalized user models through dynamic layer sharing in an iterative layer-wise manner. We evaluate the performance of FedDL for the recognition of various activities on five datasets collected from 178 users in total. The experimental results show that FedDL outperforms the other methods in terms of overall accuracy (e.g., by 24.05%, 16.67%, 19.51%, and more than 30.67%, to local training, pFedMe, FedPer, and FedAvg respectively). Moreover, FedDL saves more than 50% communication overhead when there is a large number of users and achieves a high convergence rate even with a small number of local computation rounds. As future work, we will deploy FedDL on edge devices, like smartphones, to evaluate the system overhead of FedDL. Moreover, we will also explore the application scenarios with intrinsic statistical heterogeneity beyond HAR by leveraging domain adaptation techniques.

CHAPTER 5

CONCLUSION

This thesis introduce three studies for CPS to first address the inaccuracy of the sensing data due to the noise and the dynamics of the context, then include human in the loop of smart systems, and finally, design distributed learning platform for large-scale applications.

First of all, the accuracy of cardiac signals is extremely essential for daily health monitoring. FitBeat enables accurate heart rate tracking on wrist-type wearables during intensive exercises. It integrates and augments standard filter and spectral analysis tool, which achieves comparable accuracy while significantly reducing computational overhead. Experimental results involving 10 subjects show that the average error of FitBeat is around 4 beats per minute, which improves heart rate accuracy of the default heart rate tracker of Moto 360 by 10x.

After that, we include human in the loop and design the smart health applications. To make the RSA-based breathing training, which relies on in-person sessions and cumbersome sensing devices, accessible at home, we propose the BreathCoach - a smart and unobtrusive system which enables effective in-home RSA-BT using sensors on a smartwatch and smartphone-based VR. Specifically, BreathCoach continuously measures key bio-signals including breathing pattern (BP), inter-beat interval (IBI), amplitude of RSA, and intelligently calculates the optimal BP based on current and historical measurements. The recommended BP is conveyed to users through a VR game to provide intuitive guidance. The experimental results suggest that BreathCoach is able to reliably measure needed bio-signals and intelligently calculate BP recommendations which result in improved performance compared with the traditional approach.

Finally, we build the smart system using federated learning for large-scale applications. Federated Learning (FL) enables the collaborative learning of a global model without exposing users' raw data. However, existing FL approaches yield unsatisfactory HAR performance as they fail to dynamically aggregate models according to the statistical diversity of users' data. In out study, we propose FedDL, a novel federated learning system for HAR that can capture the underlying user relationships and apply them to learn personalized models for different users dynamically. We have implemented FedDL and evaluated using a new data set we collected using LiDAR and four public real-world datasets involving 178 users in total. The results show that FedDL outperforms several state-of-the-art FL paradigms in terms of model accuracy (by more than 15%), converging rate (by more than 70%), and communication overhead (about 30% reduction). Moreover, the testing results on the datasets of different scales show that FedDL has high scalability and hence can be deployed for large-scale real-world applications.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Customer review of stresseraser.

[2] Empatica e4 wristband. `https://www.empatica.com/en-int/research/e4/`.

[3] Google cardboard.

[4] Hexoskin.

[5] Moto g4. `https://www.motorola.com/us/products/moto-g`.

[6] Stresseraser.

[7] Apple is developing watch technology to detect heart abnormalities and now blood pressure. `http://www.patentlyapple.com/patently-apple/2017/10/`, 2017.

[8] Use the breathe app. `https://support.apple.com/en-us/HT206999`, 2018. [Online; accessed 17-October-2018].

[9] Federated learning datasets for human activity recognition. 2021.

[10] K. T. Abou-Moustafa and F. P. Ferrie. A note on metric properties for some divergence measures: The gaussian case. In *Asian Conference on Machine Learning*, pages 1–15. PMLR, 2012.

[11] K. Alam, S. Qureshi, and T. Blaschke. Monitoring spatio-temporal aerosol patterns over pakistan based on modis, toms and misr satellite data and a hysplit model. *Atmospheric environment*, 45(27):4641–4651, 2011.

[12] E. Aljalbout, V. Golkov, Y. Siddiqui, M. Strobel, and D. Cremers. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*, 2018.

[13] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer, 2012.

[14] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, page 3, 2013.

[15] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[16] R. Bari, R. J. Adams, M. M. Rahman, M. B. Parsons, E. H. Buder, and S. Kumar. rconverse: Moment by moment conversation detection using a mobile respiration sensor. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):2, 2018.

[17] A. J. Beckham, T. B. Greene, and S. Meltzer-Brody. A pilot study of heart rate variability biofeedback therapy in the treatment of perinatal depression on a specialized perinatal psychiatry inpatient unit. *Archives of women's mental health*, 16(1):59–65, 2013.

[18] C. Benedek, B. Gálai, B. Nagy, and Z. Jankó. Lidar-based gait analysis and activity recognition in a 4d surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(1):101–113, 2016.

[19] G. G. Berntson, J. T. Bigger, D. L. Eckberg, P. Grossman, P. G. Kaufmann, M. Malik, H. N. Nagaraja, S. W. Porges, J. P. Saul, P. H. Stone, et al. Heart rate variability: origins, methods, and interpretive caveats. *Psychophysiology*, 34(6):623–648, 1997.

[20] S. Bhattacharya and N. D. Lane. From smart to deep: Robust activity recognition on smartwatches using deep learning. In *2016 IEEE International conference on pervasive computing and communication workshops (PerCom Workshops)*, pages 1–6. IEEE, 2016.

[21] C. Bi, G. Xing, T. Hao, J. Huh, W. Peng, and M. Ma. Familylog: A mobile system for monitoring family mealtime activities. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 21–30. IEEE, 2017.

[22] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[23] R. P. Brown and P. L. Gerbarg. Sudarshan kriya yogic breathing in the treatment of stress, anxiety, and depression: part i—neurophysiologic model. *Journal of Alternative & Complementary Medicine*, 11(1):189–201, 2005.

[24] T. E. Brown, L. A. Beightol, J. Koh, and D. L. Eckberg. Important influence of respiration on human rr interval power spectra is largely ignored. *Journal of Applied Physiology*, 75(5):2310–2317, 1993.

[25] L. Cao, Y. Wang, B. Zhang, Q. Jin, and A. V. Vasilakos. Gchar: An efficient group-based context—aware human activity recognition on smartphone. *Journal of Parallel and Distributed Computing*, 118:67–80, 2018.

[26] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[27] K. Cochrane and T. Schiphorst. Developing design considerations for mobile and wearable technology m-health applications that can support recovery in mental health disorders. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2015 9th International Conference on*, pages 29–36. IEEE, 2015.

[28] Y. Deng, M. M. Kamani, and M. Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[29] M. Dimiccoli, J. Marín, and E. Thomaz. Mitigating bystander privacy concerns in egocentric activity recognition with deep learning and intentional image degradation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–18, 2018.

[30] S. Ding, Z. Chen, T. Zheng, and J. Luo. Rf-net: a unified meta-learning framework for rf-enabled one-shot human activity recognition. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 517–530, 2020.

[31] C. T. Dinh, N. H. Tran, and T. D. Nguyen. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*, 2020.

[32] Y. Du, Y. Lim, and Y. Tan. A novel human activity recognition and prediction in smart home based on interaction. *Sensors*, 19(20):4474, 2019.

[33] F. Estève, N. Blanc-Gras, J. Gallego, and G. Benchetrit. The effects of breathing pattern training on ventilatory function in patients with copd. *Biofeedback and Self-regulation*, 21(4):311–321, 1996.

[34] T. Evgeniou, C. A. Micchelli, M. Pontil, and J. Shawe-Taylor. Learning multiple tasks with kernel methods. *Journal of machine learning research*, 6(4), 2005.

[35] A. Fallah, A. Mokhtari, and A. Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[36] N. D. Giardino, L. Chan, and S. Borson. Combined heart rate variability and pulse oximetry biofeedback for chronic obstructive pulmonary disease: preliminary findings. *Applied psychophysiology and biofeedback*, 29(2):121–133, 2004.

[37] G. Glass and K. Hopkins. Statistical methods in education and psychology. *Psyccritiques*, 41(12), 1996.

[38] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, Mar 1997.

[39] Y. Guan and T. Plötz. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–28, 2017.

[40] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 2350–2358. PMLR, 2021.

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

[42] A. L. Hansen, B. H. Johnsen, J. J. Sollers, K. Stenvik, and J. F. Thayer. Heart rate variability and its relation to prefrontal cognitive function: the effects of training and detraining. *European journal of applied physiology*, 93(3):263–272, 2004.

[43] A. L. Hansen, B. H. Johnsen, and J. F. Thayer. Vagal influence on working memory and attention. *International journal of psychophysiology*, 48(3):263–274, 2003.

[44] T. Hao, C. Bi, G. Xing, R. Chan, and L. Tu. Mindfulwatch: A smartwatch-based system for real-time respiration monitoring during meditation. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):57, 2017.

[45] M. M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren. A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307–313, 2018.

[46] J. A. Hirsch and B. Bishop. Respiratory sinus arrhythmia in humans: how breathing pattern modulates heart rate. *American Journal of Physiology-Heart and Circulatory Physiology*, 241(4):H620–H629, 1981.

[47] A. Ignatov. Real-time human activity recognition from accelerometer data using convolutional neural networks. *Applied Soft Computing*, 62:915–922, 2018.

[48] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. *arXiv preprint arXiv:0809.2085*, 2008.

[49] M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-efficient distributed dual coordinate ascent. *arXiv preprint arXiv:1409.1458*, 2014.

[50] A. Jalal and S. Kamal. Real-time life logging via a depth silhouette-based human activity recognition system for smart home services. In *2014 11th IEEE International conference on advanced video and signal based surveillance (AVSS)*, pages 74–80. IEEE, 2014.

[51] A. Jalal, M. Z. Uddin, and T.-S. Kim. Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home. *IEEE Transactions on Consumer Electronics*, 58(3):863–871, 2012.

[52] Y. Jia. Diatetic and exercise therapy against diabetes mellitus. In *2009 Second International Conference on Intelligent Networks and Intelligent Systems*, pages 693–696. IEEE, 2009.

[53] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, et al. Towards environment independent device free human activity recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 289–304, 2018.

[54] Y. Jiang, J. Konečnỳ, K. Rush, and S. Kannan. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*, 2019.

[55] W. S. Johnston. *Development of a signal processing library for extraction of SpO2, HR, HRV, and RR from photoplethysmographic waveforms.* PhD thesis, Worcester Polytechnic Institute, 2006.

[56] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[57] S. Kaplan. Meditation, restoration, and the management of mental fatigue. *Environment and behavior*, 33(4):480–506, 2001.

[58] M. Karamnejad. Virtual reality and health informatics for management of chronic pain. *Simon Fraser University*, 2014.

[59] M. K. Karavidas, P. M. Lehrer, E. Vaschillo, B. Vaschillo, H. Marin, S. Buyske, I. Malinovsky, D. Radvanski, and A. Hassett. Preliminary results of an open label study of heart rate variability biofeedback for the treatment of major depression. *Applied psychophysiology and biofeedback*, 32(1):19–30, 2007.

[60] W. Karlen, J. M. Ansermino, and G. Dumont. Adaptive pulse segmentation and artifact detection in photoplethysmography for mobile applications. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 3131–3134. IEEE, 2012.

[61] P. G. Katona and F. Jih. Respiratory sinus arrhythmia: noninvasive measure of parasympathetic cardiac control. *Journal of applied physiology*, 39(5):801–805, 1975.

[62] A. H. Kemp, D. S. Quintana, K. L. Felmingham, S. Matthews, and H. F. Jelinek. Depression, comorbid anxiety disorders, and heart rate variability in physically healthy, unmedicated patients: implications for cardiovascular risk. *PloS one*, 7(2):e30777, 2012.

[63] M. A. A. H. Khan, N. Roy, and A. Misra. Scaling human activity recognition via deep learning-based domain adaptation. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–9. IEEE, 2018.

[64] I. Z. Khazan. *The clinical handbook of biofeedback: A step-by-step guide for training and practice with mindfulness.* John Wiley & Sons, 2013.

[65] B. S. Kim and S. K. Yoo. Motion artifact reduction in photoplethysmography using independent component analysis. *IEEE Transactions on Biomedical Engineering*, 53(3):566–568, March 2006.

[66] S. H. Kim, D. W. Ryoo, and C. Bae. Adaptive noise cancellation using accelerometers for the ppg signal from forehead. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2564–2567, Aug 2007.

[67] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[68] N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks (csn). In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 355–364, 2011.

[69] L. Leger and M. Thivierge. Heart rate monitors: Validity, stability, and functionality. *The Physician and Sportsmedicine*, 16(5):143–151, 1988.

[70] P. Lehrer, A. Smetankin, and T. Potapova. Respiratory sinus arrhythmia biofeedback therapy for asthma: A report of 20 unmedicated pediatric cases using the smetankin method. *Applied psychophysiology and biofeedback*, 25(3):193–200, 2000.

[71] P. M. Lehrer, E. Vaschillo, and B. Vaschillo. Resonant frequency biofeedback training to increase cardiac variability: Rationale and manual for training. *Applied psychophysiology and biofeedback*, 25(3):177–191, 2000.

[72] P. M. Lehrer, E. Vaschillo, and B. Vaschillo. Resonant frequency biofeedback training to increase cardiac variability: Rationale and manual for training. *Applied psychophysiology and biofeedback*, 25(3):177–191, 2000.

[73] P. M. Lehrer, E. Vaschillo, B. Vaschillo, S.-E. Lu, D. L. Eckberg, R. Edelberg, W. J. Shih, Y. Lin, T. A. Kuusela, K. U. Tahvanainen, et al. Heart rate variability biofeedback increases baroreflex gain and peak expiratory flow. *Psychosomatic medicine*, 65(5):796–805, 2003.

[74] R. Ley. The modification of breathing behavior: Pavlovian and operant control in emotion and cognition. *Behavior Modification*, 23(3):441–479, 1999.

[75] J. Li, Y. Rong, H. Meng, Z. Lu, T. Kwok, and H. Cheng. Tatc: predicting alzheimer's disease with actigraphy data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 509–518, 2018.

[76] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

[77] Z. Liu, F. Zhang, and X. Hong. Low-cost retina-like robotic lidars based on incommensurable scanning. *IEEE/ASME Transactions on Mechatronics*, 2021.

[78] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3071–3085, 2018.

[79] M. Long, Z. Cao, J. Wang, and P. S. Yu. Learning multiple tasks with multilinear relationship networks. *arXiv preprint arXiv:1506.02117*, 2015.

[80] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal. Secure and scalable cloud-based architecture for e-health wireless sensor networks. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7. IEEE, 2012.

[81] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5334–5343, 2017.

[82] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

[83] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[84] H. B. McMahan and D. Ramage. 2017.

[85] W. Min and L. Wynter. Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, 19(4):606–616, 2011.

[86] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[87] M. Moencks, V. De Silva, J. Roche, and A. Kondoz. Adaptive feature processing for robust human activity recognition on a novel multi-modal dataset. *arXiv preprint arXiv:1901.02858*, 2019.

[88] L. Mou, Z. Meng, R. Yan, G. Li, Y. Xu, L. Zhang, and Z. Jin. How transferable are neural networks in nlp applications? *arXiv preprint arXiv:1603.06111*, 2016.

[89] M. Munafo, E. Patron, and D. Palomba. Improving managers' psychophysical well-being: effectiveness of respiratory sinus arrhythmia biofeedback. *Applied psychophysiology and biofeedback*, 41(2):129–139, 2016.

[90] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand. A performance evaluation of federated learning algorithms. In *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning*, pages 1–8, 2018.

[91] S. Nirjon, R. F. Dickerson, Q. Li, P. Asare, J. A. Stankovic, D. Hong, B. Zhang, X. Jiang, G. Shen, and F. Zhao. Musicalheart: A hearty way of listening to music. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 43–56, 2012.

[92] X. Ouyang, Z. Xie, J. Zhou, J. Huang, and G. Xing. Clusterfl: a similarity-aware federated learning system for human activity recognition. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 54–66, 2021.

[93] P. E. Paredes, Y. Zhou, N. A.-H. Hamdan, S. Balters, E. Murnane, W. Ju, and J. A. Landay. Just breathe: In-car interventions for guided slow breathing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(1):28, 2018.

[94] J. A. C. Patterson, D. C. McIlwraith, and G. Z. Yang. A flexible, low noise reflective ppg sensor platform for ear-worn heart rate monitoring. In *2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, pages 286–291, June 2009.

[95] L. Peng, L. Chen, Z. Ye, and Y. Zhang. Aroma: A deep multi-task learning based simple and complex human activity recognition method using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):1–16, 2018.

[96] S. W. Porges, J. A. Doussard-Roosevelt, and A. K. Maiti. Vagal tone and the physiological regulation of emotion. *Monographs of the society for research in child development*, 59(2-3):167–186, 1994.

[97] B. Prathyusha, T. S. Rao, and D. Asha. Extraction of respiratory rate from ppg signals using pca and emd.

[98] G. E. Prinsloo, H. L. Rauch, M. I. Lambert, F. Muench, T. D. Noakes, and W. E. Derman. The effect of short duration heart rate variability (hrv) biofeedback on cognitive performance during laboratory induced cognitive stress. *Applied Cognitive Psychology*, 25(5):792–801, 2011.

[99] M. Prpa, K. Cochrane, and B. E. Riecke. Hacking alternatives in 21st century: designing a bio-responsive virtual environment for stress reduction. In *International Symposium on Pervasive Computing Paradigms for Mental Health*, pages 34–39. Springer, 2015.

[100] Y. Qin, C. J. Vincent, N. Bianchi-Berthouze, and Y. Shi. Airflow: designing immersive breathing training games for copd. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 2419–2424. ACM, 2014.

[101] V. Radu, N. D. Lane, S. Bhattacharya, C. Mascolo, M. K. Marina, and F. Kawsar. Towards multimodal deep learning for activity recognition on mobile devices. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 185–188, 2016.

[102] M. A. D. Raya and L. G. Sison. Adaptive noise cancelling of motion artifact in stress ecg signals using accelerometer. In *Proceedings of the Second Joint 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society*, volume 2, pages 1756–1757 vol.2, 2002.

[103] R. Reiner. Integrating a portable biofeedback device into clinical practice for patients with anxiety disorders: Results of a pilot study. *Applied Psychophysiology and Biofeedback*, 33(1):55–61, 2008.

[104] S. Rhee, B.-H. Yang, and H. H. Asada. Artifact-resistant power-efficient design of finger-ring plethysmographic sensors. *IEEE Transactions on Biomedical Engineering*, 48(7):795–805, July 2001.

[105] D. Riboni and C. Bettini. Cosar: hybrid reasoning for context-aware activity recognition. *Personal and Ubiquitous Computing*, 15(3):271–289, 2011.

[106] D. J. L. F. d. V. Rodrigues. *Risk Assessment for Alzheimer Patients, using GPS and Accelerometers with a Machine Learning Approach*. PhD thesis, 2019.

[107] C. A. Ronao and S.-B. Cho. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, 59:235–244, 2016.

[108] P. C. Roy, S. Giroux, B. Bouchard, A. Bouzouane, C. Phua, A. Tolstikov, and J. Biswas. A possibilistic approach for activity recognition in smart homes for cognitive assistance to alzheimer's patients. In *Activity Recognition in Pervasive Intelligent Environments*, pages 33–58. Springer, 2011.

[109] S. Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[110] A. L. Rukhin. Analysis of time series structure ssa and related techniques. *Technometrics*, 44(3):290–290, 2002.

[111] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems*, 2019.

[112] L. Sherlin, R. Gevirtz, S. Wyckoff, and F. Muench. Effects of respiratory sinus arrhythmia biofeedback versus passive biofeedback control. *International Journal of Stress Management*, 16(3):233, 2009.

[113] J. Shi, J. Wan, H. Yan, and H. Suo. A survey of cyber-physical systems. In *2011 international conference on wireless communications and signal processing (WCSP)*, pages 1–6. IEEE, 2011.

[114] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[115] K. Sozinov, V. Vlassov, and S. Girdzijauskas. Human activity recognition using federated learning. In *2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom)*, pages 1103–1111. IEEE, 2018.

[116] H. M. Stauss. Heart rate variability. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 285(5):R927–R931, 2003.

[117] X. Sun, Z. Lu, W. Hu, and G. Cao. Symdetector: detecting sound-related respiratory symptoms using smartphones. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 97–108, 2015.

[118] T. Tamura, Y. Maeda, M. Sekine, and M. Yoshida. Wearable photoplethysmographic sensors—past and present. *Electronics*, 3(2):282, 2014.

[119] G. Tan, T. K. Dao, L. Farmer, R. J. Sutherland, and R. Gevirtz. Heart rate variability (hrv) and posttraumatic stress disorder (ptsd): a pilot study. *Applied psychophysiology and biofeedback*, 36(1):27–35, 2011.

[120] M. P. Tarvainen, J.-P. Niskanen, J. A. Lipponen, P. O. Ranta-Aho, and P. A. Karjalainen. Kubios hrv–heart rate variability analysis software. *Computer methods and programs in biomedicine*, 113(1):210–220, 2014.

[121] L. Tu, T. Hao, C. Bi, and G. Xing. Breathcoach: A smart in-home breathing training system with bio-feedback via vr game. *Smart Health*, 16:100090, 2020.

[122] L. Tu, T. Hao, C. Bi, and G. Xing. Breathcoach: A smart in-home breathing training system with bio-feedback via vr game. *Smart Health*, 16:100090, 2020.

[123] L. Tu, J. Huang, C. Bi, and G. Xing. Fitbeat: A lightweight system for accurate heart rate measurement during exercise. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–8. IEEE, 2017.

[124] L. Tu, X. Ouyang, J. Zhou, Y. He, and G. Xing. Feddl: Federated learning via dynamic layer sharing for human activity recognition. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, page 15–28, New York, NY, USA, 2021. Association for Computing Machinery.

[125] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal. Iot healthcare analytics: The importance of anomaly detection. In *2016 IEEE 30th international conference on advanced information networking and applications (AINA)*, pages 994–997. IEEE, 2016.

[126] I. Van Diest, K. Verstappen, A. E. Aubert, D. Widjaja, D. Vansteenwegen, and E. Vlemincx. Inhalation/exhalation ratio modulates the effect of slow breathing on heart rate variability and relaxation. *Applied psychophysiology and biofeedback*, 39(3-4):171–180, 2014.

[127] E. G. Vaschillo, B. Vaschillo, and P. M. Lehrer. Characteristics of resonance in heart rate variability stimulated by biofeedback. *Applied psychophysiology and biofeedback*, 31(2):129–142, 2006.

[128] J. Wang, H. Abid, S. Lee, L. Shu, and F. Xia. A secured health care application architecture for cyber-physical systems. *arXiv preprint arXiv:1201.0213*, 2011.

[129] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, and H. V. Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[130] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. E. Dong, and R. C. Goodlin. Adaptive noise cancelling: Principles and applications. *Proceedings of the IEEE*, 63(12):1692–1716, Dec 1975.

[131] L. Xie, I. M. Baytas, K. Lin, and J. Zhou. Privacy-preserving distributed multi-task learning with asynchronous updates. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1195–1204, 2017.

[132] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[133] J. Yin, Q. Yang, and J. J. Pan. Sensor-based abnormal human-activity detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1082–1090, 2008.

[134] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[135] Z. Zhang, Z. Pi, and B. Liu. Troika: A general framework for heart rate monitoring using wrist-type photoplethysmographic signals during intensive physical exercise. *IEEE Transactions on Biomedical Engineering*, 62(2):522–531, Feb 2015.

[136] S. Zhao, W. Li, and J. Cao. A user-adaptive algorithm for activity recognition based on k-means clustering, local outlier factor, and multivariate gaussian distribution. *Sensors*, 18(6):1850, 2018.

[137] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[138] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. 2014.

[139] T. L. Zucker, K. W. Samuelson, F. Muench, M. A. Greenberg, and R. N. Gevirtz. The effects of respiratory sinus arrhythmia biofeedback on heart rate variability and posttraumatic stress disorder symptoms: a pilot study. *Applied psychophysiology and biofeedback*, 34(2):135, 2009.