

UNIVERSITÄT OSNABRÜCK

BACHELOR THESIS

---

**Towards Real-world Wireless Sensing on  
the Edge: CSI-based Sleep Measurements  
with ESP32 Devices**

---

*Author:*

Hannah KÖSTER

*Supervisors:*

Dr. Kristoffer APPEL

Prof. Dr. Gordon PIPA

*A thesis submitted in fulfillment of the requirements  
for the degree of Bachelor of Science*

*in*

Cognitive Science

September 18, 2023

## Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

---

Signature

---

City, Date

UNIVERSITÄT OSNABÜCK

## *Abstract*

Cognitive Science  
School of Human Sciences  
Institute of Cognitive Science

Bachelor of Science

### **Towards Real-world Wireless Sensing on the Edge: CSI-based Sleep Measurements with ESP32 Devices**

by Hannah KÖSTER

Sleep plays an important role in people's lives and a low-cost, unobtrusive monitoring solution that can be used for long-term, in-home monitoring to detect potential issues early on is highly desirable. While Wi-Fi sensing shows great promise in this area, previous research has used setups consisting of large, comparatively expensive devices, unsuitable for real-world applications on a larger scale. This work aims to test whether fine-grained sensing is feasible on small, low-cost edge devices, by designing and implementing a system for online, on-device sleep monitoring using ESP32 microcontrollers. The performance of the system's heart and breathing rate estimation is evaluated against the gold standard PSG, and while these initial performance tests show unfavorable results with very high estimation errors, potential causes as well as signal properties that can guide further research were identified.

# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Contents</b>	<b>iii</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Abbreviations</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Preliminaries</b>	<b>4</b>
2.1 Terminology: Wi-Fi, WLAN, and IEEE 802.11 . . . . .	4
2.2 Fundamentals of Wi-Fi . . . . .	5
2.2.1 Channel state information . . . . .	6
2.2.2 Noise . . . . .	7
2.2.3 Signal propagation models . . . . .	9
<b>3 Related works / Current state of the field</b>	<b>10</b>
<b>4 Methods / Implementation</b>	<b>12</b>
4.1 System design & Implementation . . . . .	12
4.1.1 Design goals . . . . .	12
4.1.2 System overview . . . . .	12
4.1.3 Platform . . . . .	13
4.1.4 SSNR . . . . .	15
4.1.5 Preprocessing . . . . .	16
4.1.6 Feature extraction . . . . .	19
4.1.7 Sleep stage NN . . . . .	22
4.2 Experiments . . . . .	23
4.2.1 Experiment design . . . . .	23
4.2.2 Hypothesis / expected results . . . . .	24

4.2.3	Reference device & Performance metrics	25
4.2.4	Experiment overview	25
4.2.5	Collected data	26
4.2.6	Data analysis	28
<b>5</b>	<b>Results</b>	<b>29</b>
5.1	Effect of posture	31
5.2	Effect of blanket	33
5.3	Effect of different system components	34
5.4	Effect of device location	36
<b>6</b>	<b>Discussion</b>	<b>39</b>
6.1	Processing influences CSI	39
6.2	STI	39
6.3	Discussion of the results	40
6.3.1	Effect of posture	40
6.3.2	Effect of device placement	40
6.3.3	Effect of blanket	41
6.4	Problems with the data collection	41
6.5	Limitations & Further work	42
6.6	Conclusion	44
<b>A</b>	<b>Additional tables and plots</b>	<b>45</b>
<b>Bibliography</b>		<b>54</b>

# List of Figures

2.1	Illustration of different types of propagation paths, with static paths in blue, including the line of sight path (green), and the dynamic path in red	8
4.1	Overview of the system. Preprocessing steps with a light blue background are optional	13
4.2	Development boards used with a 1€ coin as a size reference. From left to right: generic ESP32 development board (transmitter), ESP32-S3 development board (receiver), Lolin S3 Pro (receiver)	14
4.3	The three receivers equipped with SD-cards as used for data collection	14
4.4	Setup of the ESPs in the sleep lab	25
5.1	Signal tendency index values over the whole data collection session for the three receivers. Recording periods are marked, with light colors indicating the open condition and more saturated colors the blanket condition. For the postures the colors are matched as follows: supine = blue, left = pink, prone = green, right = orange	30
5.2	Reference breathing signal provided by the abdomen RIP, and the STI and filtered breath amplitude of the right receiver during the left × blanket condition	31
5.3	Breathing rate estimation errors for the different postures across conditions	32
5.4	Heart rate estimation errors for the different postures across conditions	32
5.5	Breathing rate estimation errors for the different conditions across postures	33
5.6	Heart rate estimation errors for the different conditions across postures	34
5.7	Breathing rate estimation errors for the different system setups across all conditions	35
5.8	Heart rate estimation errors for the different system setups across all conditions	35
5.9	Breathing rate estimation errors for the different receivers across all conditions	37
5.10	Heart rate estimation errors for the different receivers across all conditions	37
A.1	Breathing rate estimation errors for the different poses during the open condition	45

A.2 Heart rate estimation errors for the different poses during the open condition . . . . .	48
A.3 Breathing rate estimation errors for the different poses during the blanket condition . . . . .	49
A.4 Heart rate estimation errors for the different poses during the blanket condition . . . . .	50

# List of Tables

4.1	Respiratory and Cardiac cutoff frequencies. Frequencies were chosen to be extreme values encompassing a larger range than the normal rest frequency ranges for healthy adults. . . . .	18
4.2	Standard receiver configuration . . . . .	27
4.3	Time duration of the data used for analysis for each condition . . . . .	28
A.1	Summary statistics of the different breathing rate errors for the different poses . . . . .	46
A.2	Summary statistics of the different heart rate errors for the different poses . . . . .	46
A.3	Summary statistics of the different breathing rate errors for the different conditions . . . . .	47
A.4	Summary statistics of the different heart rate errors for the different conditions . . . . .	47
A.5	Summary statistics of the different breathing rate errors for the different poses during the open condition . . . . .	47
A.6	Summary statistics of the different heart rate errors for the different poses during the open condition . . . . .	48
A.7	Summary statistics of the different breathing rate errors for the different poses during the blanket condition . . . . .	49
A.8	Summary statistics of the different heart rate errors for the different poses during the blanket condition . . . . .	50
A.9	Summary statistics of the different breathing rate errors for the different system settings . . . . .	51
A.10	Summary statistics of the different heart rate errors for the different system settings . . . . .	52
A.11	Summary statistics of the different breathing rate errors for the different devices . . . . .	52
A.12	Summary statistics of the different heart rate errors for the different devices . . . . .	53

# List of Abbreviations

<b>bpm</b>	breaths/beats per minute
<b>CIR</b>	Channel impulse response
<b>CSI</b>	Channel state information
<b>ECG</b>	Electrocardiogram
<b>EEG</b>	Electroencephalography
<b>EMG</b>	Electromyography
<b>EOG</b>	Electrooculography
<b>HRV</b>	Heart rate variability
<b>IMU</b>	Integrated measurement unit
<b>IoT</b>	Internet of Things
<b>LAN</b>	Local area network
<b>LLTF</b>	Legacy long training field
<b>MRC</b>	Maximal-ratio combining
<b>NN</b>	Neural network
<b>OFDM</b>	Orthogonal frequency-division multiplexing
<b>PCA</b>	Principal component analysis
<b>PPG</b>	Photoplethysmography
<b>PSD</b>	Power spectral density
<b>PSG</b>	Polysomnography
<b>PSRAM</b>	Pseudostatic RAM
<b>RIP</b>	Respiratory inductance plethysmography
<b>RX</b>	Receiver
<b>SNR</b>	Signal-to-noise ratio
<b>SSNR</b>	Signal-sensing-to-noise-ratio
<b>STI</b>	Signal tendency index
<b>TX</b>	Transmitter
<b>UDP</b>	User Datagram protocol
<b>WLAN</b>	Wireless LAN

# 1 Introduction

People spend a substantial part of their lives sleeping, and getting enough and especially good quality sleep has important implications for a person's health (Nelson, Davis, and Corbett 2022, Lao et al. 2018, Scott et al. 2021). While sleep research is by no means a new topic and sleep monitoring has been implemented in clinical settings as well as special sleep laboratories to diagnose sleep disorders, the gold standard used in these settings, polysomnography (PSG), requires extensive medical equipment, that is not only expensive but also requires special knowledge and training to set up and use. This is due to PSG recording data from multiple sensors in parallel to obtain various physiological signals, such as EEG for monitoring brain activity, ECG for monitoring the electrical activity of the heart, EOG for monitoring eye moments, and EMG for monitoring skeletal muscle activity, which require correct placement in order to function properly. So, while PSG poses the gold standard for diagnosing sleep disorders in clinical settings and is also often used for conducting fine-grained sleep research, it is unsuitable for long-term as well as in-home monitoring due to its high cost. However, in-home, long-term monitoring is especially desirable because it allows detecting changes in a person's sleep (quality) allowing corresponding actions to be taken early, potentially averting further health issues.

In recent years there has been a surge of research and various commercial products aimed at solving this problem. The approaches or methods can be divided into two categories. First, there are contact-based methods using sensors that require contact and thus mostly need to be worn, for instance using PPG, ECG, accelerometers, IMUs, and pressure sensors which have become less expensive with commercial products ranging from smart watches over smart mattresses to fitness or even special sleep trackers. One major critique of these devices has been that they do not offer fine-grained information or only offer good performance for coarse-grained (e.g. sleep/wake detection) but not for fine-grained information such as sleep staging (Ameen et al. 2019, Chinoy et al. 2022). Another major disadvantage, that will remain even if performance reaches that of clinical standards, is that due to their nature of being contact-based, they require user cooperation and can interfere with a person's sleep. User cooperation is not always feasible, for example, children, elderly, or ill people might not want to or forget why they are wearing such a device and try to take it off during the night.

To solve this problem posed by contact-based methods considerable research has

gone into developing contactless methods for sleep monitoring, for instance using cameras (Deng et al. 2018, Falie and Ichim 2010, Gastel et al. 2021, Meulen et al. 2023, M.-C. Yu et al. 2013), audio (including sonar), such as provided by smartphones (Fang et al. 2023, Ren et al. 2015, Nandakumar, Gollakota, and Watson 2015, Zaffaroni et al. 2019), and radar (Lauteslager et al. 2020, Hsu et al. 2017). While considerable progress has been made, video and audio-based methods have been criticized for being too privacy intrusive (Cay et al. 2022) and requiring large, expensive setups to function well (Ge et al. 2023). And although radar is able to fulfill more of the goals that contactless methods aim to achieve, with the technology having advanced to a level where there are already commercial products available for consumers (Walid et al. 2021), and comes with fewer privacy concerns than video- or audio-based methods, radar-based sleep monitoring still requires special hardware and is therefore quite expensive.

However, another field utilizing radio waves similar to radar-based methods has emerged: Wi-Fi sensing. Specifically, this approach uses Wi-Fi signals for sensing the environment, based on the reasoning that Wi-Fi signals are able to capture information about the environment they travel through, with early work showing its application for movement and ultimately human presence detection (Yunze Zeng et al. 2014, Qian et al. 2014), which has already found its way into consumer products (Mott 2019), and further work presenting the viability of using Wi-Fi for vital sign monitoring (Wang, Yang, and Mao 2017a, Atif et al. 2022), various classification tasks such as activity (J. Yang et al. 2018) and gesture recognition (Kabir, Hasan, and Shin 2022) and fall detection (Palipana et al. 2018), as well as already for sleep monitoring tasks, including sleep staging (Zhang et al. 2021, B. Yu et al. 2021), breathing and heart rate estimation while sleeping (J. Liu et al. 2015, Gu et al. 2019), detection of nocturnal seizures (Korany and Mostofi 2022), and simple detection of sleep pose (X. Liu et al. 2014, J. Liu et al. 2018). One of the big advantages of utilizing Wi-Fi for sensing is that due to Wi-Fi being so widespread—with the number of active devices estimated to reach 19.5 billion this year (The Beacon 2023)—is that devices are cheap, and existing devices could potentially be reused, either through software updates that would enable them to perform or participate in Wi-Fi sensing or by using one device where the CSI for sensing can be retrieved and exploiting polite Wi-Fi (Abedi and Abari 2020) to generate information used for sensing from these devices.

While Wi-Fi sensing shows great promise for achieving various sensing tasks, most research so far serves as providing proof-of-concept prototypes, using large devices in mostly idealized situations or environments, so that the setups are far from being realistic for real-world applications. Previous works often do not analyze the data in real-time, but process it after it has been recorded, with the processing usually taking place on powerful computers—an approach not suitable for large-scale deployment in people’s homes. For these systems to be viable for real-world deployment, a move to smaller, more cost-effective systems is required, which can be easily set up, do not

consume much power, and ideally can be integrated into existing network structures.

It is also desirable for such systems to process the data online, in a real-time fashion, in order for results to be available immediately which is impervious to catching potentially harmful effects of sleep disorders as they happen and be able to intervene quickly. Additionally, an online system is the most sensible approach if processing is to be carried out directly on the edge device, as there might not be enough downtime where no data is collected to perform the processing in chunks. Additionally, for less powerful devices, analyzing a whole night's worth of data at once might take a long time. While it is possible to imagine the Wi-Fi sensing data to be collected by edge devices and sent to a central server for analysis, this requires such a structure to be in place. This might be a solution if Wi-Fi sensing becomes widespread, but requires a central organization to set up the infrastructure. Sending health data to a central server might however raise privacy concerns, and on-device inference might ease some of these concerns.

To take a step towards creating more realistic systems for real-world Wi-Fi sensing applications, this work aims to implement a system performing Wi-Fi sensing directly on an edge device, namely ESP32 microcontrollers. As a potential application, sleep monitoring is selected, as research in this area has important implications and Wi-Fi sensing has already been demonstrated to be a viable approach for sleep monitoring (B. Yu et al. 2021, Li et al. 2016, Gu et al. 2019).

## 2 Preliminaries

### 2.1 Terminology: Wi-Fi, WLAN, and IEEE 802.11

This chapter will clarify the most important terminology and provide the preliminaries required for Wi-Fi sensing. But before turning to the background of those, the terms *Wi-Fi*, *WLAN*, and the *IEEE 802.11 standards* and how these will be used in the remainder of this thesis need to be clarified.

*WLAN* stands for Wireless Local Area Network, meaning a network of devices in a limited area—a local network—connected wirelessly. The term *WLAN* thus only describes that the devices are connected in a wireless manner and provides a rough knowledge of the area covered by the network, but does not specify the protocol of the connection. So, while nowadays most *WLANs* consist of *Wi-Fi*, this does not have to be the case and *WLANs* need not be compatible.

The *IEEE 802.11 standards* are standards for *WLANs*, defining a protocol and how this can be interoperable with existing standards for local, personal, and metropolitan area networks (more specifically the other 802 standards). The first version of the standard was published in 1997, and amendments have been published regularly since, introducing additions or changes to the protocol that allow for more data throughput, introduce the use of other frequency bands, or add features required for compliance with certain regulations. Over the years new versions of the standard have been published, integrating certain amendments into the standard, with the most recent being 802.11-2020 (IEEE 2021).

Because the IEEE only creates and publishes these standards, but does not regulate the implementation of them, the *Wi-Fi Alliance* was created in 1999 (at that time still under the name "Wireless Ethernet Compatibility Alliance") with the aim of certifying that devices complied with the standards and thereby making it easier for consumers to recognize which devices are interoperable. To achieve this the *Wi-Fi Alliance* also coined the term *Wi-Fi*, a made-up word with no actual meaning that was created as a simple and easy-to-remember term to use instead of the long *IEEE 802.11 standards* and amendment designations. Starting in 2018 *Wi-Fi* version names were introduced, which corresponded to different *IEEE* standards or amendments, with *Wi-Fi* standards being numbered with increasing integers, making it easy for consumers to recognize newer standards.

Using these terms interchangeably has been increasingly common practice in research and since essentially all commercially available devices are Wi-Fi certified, and thus WLANs usually follow the IEEE 802.11 standards, they can be reasonably treated as such. For the sake of readability, going forward the term Wi-Fi will be used instead of referring to the IEEE standard except when a specific version needs to be referenced.

## 2.2 Fundamentals of Wi-Fi

Wi-Fi uses radio waves to transmit data. Most notably, the 2.4 and 5 GHz frequency bands are used by most current consumer devices implementing the standard amended by 802.11 a/b/g/n/ac/ax (Wi-Fi 4, 5, 6), although the current standard also defines the uses of other frequency bands (IEEE 2021). Within these large frequency bands smaller bands can be defined, so-called *channels*, the width of which is defined differently depending on the specific physical layer (PHY) used. In the orthogonal frequency division multiplexing (OFDM) modulation scheme, which all the above-mentioned currently most used standards utilize, the channel is further split up into smaller, orthogonal frequency bands, the *subcarriers*. The orthogonality allows data to be transmitted on multiple subcarriers at the same time, thereby increasing throughput.

Of these subcarriers, to reduce interference, the subcarriers at the edges of the frequency band do not transmit data, these are therefore so-called *null subcarriers*. Apart from these null subcarriers, the 0th (DC) subcarrier is not used to avoid issues due to converter offsets in the RF system and is thus also null. Additionally, there are *pilot subcarriers*, spaced evenly throughout the used subcarriers which transmit known symbols to serve as a reference signal used for channel analysis.

*Channel state information* (CSI) is a fundamental part of the PHYs using OFDM, serving multiple purposes such as being used for antenna selection and sounding as well as calibration for beamforming. CSI as used for Wi-Fi sensing is estimated by means of transmitting a sequence of predefined "training" symbols, the OFDM preamble, so that the channel state information can be calculated from the received and the known transmitted signal (see 2.4). As the null subcarriers do not transmit data, no training symbols are transmitted on those frequencies from which CSI could be estimated, and pilot subcarriers also generally do not provide CSI usable for sensing as they are similarly not used for transmitting the training symbols of the preamble (IEEE 2021).

The estimation of channel state information happens at a low level in the Wi-Fi stack, as it is part of the PHY, the physical layer, and therefore CSI is not usually available to applications running at higher levels. However, for some devices or Wi-Fi chipsets modified drivers exist that provide access to CSI (Halperin et al. 2011, Xie, Li, and Li 2015, Gringoli et al. 2019, Gringoli et al. 2021, Jiang et al. 2022), or, as is the case for the ESP32 devices used in this thesis, the manufacturer provides an API for accessing this information ("Wi-Fi Driver - Wi-Fi Channel State Information - ESP32 -

— ESP-IDF Programming Guide v5.1 documentation,” [n.d.](#)). This means that currently only few devices can be used to perform Wi-Fi sensing using CSI out of the box or after changes taking little effort. However, a new standard, IEEE 802.11bf, for Wi-Fi sensing or SENS as it is named in the current draft, is set to be released in September 2024 (Restuccia [2021](#)). With established protocols for performing sensing operations, accessing sensing data, and proper integration into the existing standards, research into Wi-Fi sensing will become easier and more likely to lead to real-world applications (C. Chen et al. [2023](#)).

### 2.2.1 Channel state information

After a Wi-Fi signal is transmitted, it propagates through the environment via different paths, along which it undergoes different attenuation, fading, and scattering. The received signal is therefore a superposition of all multipath signals:

$$V = \sum_{i=1}^N ||V_i|| e^{-j\theta_i} \quad (2.1)$$

with  $V_i$  and  $\theta_i$  being the amplitude and phase of the  $i$ th multipath component.

These paths can be characterized by the channel impulse response (CIR):

$$h(\tau) = \sum_{i=1}^N a_i e^{-j\theta_i} \delta(\tau - \tau_i) \quad (2.2)$$

where  $a_i, \theta_i, \tau_i$  are the amplitude, phase, and time delay of the  $i$ th path respectively, and  $\delta(\tau)$  is the Dirac delta function, whose value is 0 everywhere except at position 0, where it is 1. Thus, each impulse is the complex value representing the amplitude and phase of one path. The transformation of the CIR into the frequency domain leads to the channel frequency response (CFR):

$$H(f) = \sum_{i=1}^N \alpha_i e^{-j2\pi f \tau_i} \quad (2.3)$$

where  $N$  is the number of paths,  $\alpha_i$  is the amplitude attenuation of the  $i$ th path,  $f$  is the carrier frequency, and  $\tau_i$  is the time delay or time of flight of the  $i$ th path.

The CFR can be estimated from the received and transmitted signal through the following relation:

$$R(f) = S(f) \times H(f) + N(f) \quad (2.4)$$

with  $R(f)$  being the received signal,  $S(f)$  being the transmitted signal, and  $N(f)$  being noise.

As previously mentioned, for Wi-Fi this is done by transmitting known sequences of symbols, so-called training fields, which are used to perform channel estimation

to detect variations across subcarriers. This leads to a sampled CFR estimate, which is commonly known as channel state information (CSI), providing the complex CFR value at each subcarrier frequency (over which the training symbols were transmitted).

The channel state information  $H$  can then be defined for each subcarrier  $i$  as follows:

$$H_i = |H_i|e^{\angle H_i} \quad (2.5)$$

with  $|H_i|$  and  $\angle H_i$  being the amplitude and phase of the  $i$ th subcarrier.

From the complex CS value, consisting of an imaginary part  $I$  and a real part  $R$ , the amplitude and phase respectively can be calculated as:

$$A_i = \sqrt{(I(h_i))^2 + (R(h_i))^2} \quad (2.6)$$

$$\phi_i = \text{atan2}(I(h_i), R(h_i)) \quad (2.7)$$

The channel state information therefore describes the attenuation, frequency and phase shift, fading, and scattering the transmitted signal undergoes. As these effects vary depending on the environment the signal travels through, and are for example influenced by movement, CSI contains information about the environment, making it useful for sensing applications.

For human sensing tasks, the signal propagation paths are often grouped into two sets, *static* and *dynamic* paths, as illustrated in figure 2.1. Static paths result from the signal being reflected from the static environment, such as signal reflections off of walls or furniture, and also include the line of sight path between the sender and receiver. In contrast, dynamic paths are affected by changes in the environment, such as human movement, which includes small movements caused by breathing and pulse. Static paths can thus be regarded as constant and might therefore be removed during pre-processing for sensing tasks.

## 2.2.2 Noise

In praxis, the channel state information also contains various types of noise in addition to the actual signal. Specifically, the amplitude contains random noise caused by power control uncertainty while the phase noise consists of the packet boundary detection error, sampling frequency offset, carrier frequency offset, and the unknown initial phase. Most of these are caused by errors in the RF chain, such that they are the same across multiple antennas connected to the same Wi-Fi chip. This property has been used to obtain stable phase information, for example by using the CSI ratio or quotient (Wang, Yang, and Mao 2017a, Youwei Zeng et al. 2019).

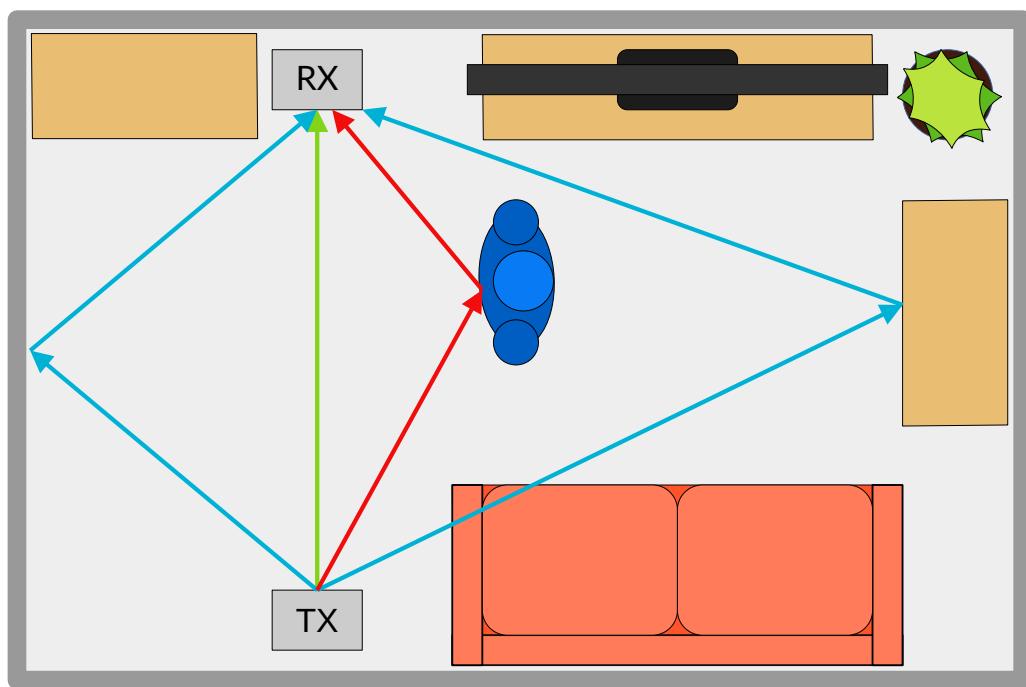


FIGURE 2.1: Illustration of different types of propagation paths, with static paths in blue, including the line of sight path (green), and the dynamic path in red

### 2.2.3 Signal propagation models

To understand and develop better sensing systems, different models have been proposed and used that describe properties of the signal propagation that have an important effect on the sensing capabilities. The arguably most used model is the Fresnel zone model, which describes how an obstacle, such as a human, can have a destructive or constructive effect on the original signal based on where it is located with respect to the transmitter (TX) and receiver (RX). In the Fresnel zone model, the zones are modeled as a series of concentric ellipses, with alternating zones having constructive or destructive effects on the signal. Following this model, the sensing area that can be covered by a transmitter and receiver is an ellipsis stretching from the transmitter to the receiver. A larger sensing coverage could then be achieved by simply increasing the distance between the TX and RX.

X. Wang et al. (2022) have proposed another model specifically for sensing, based on their newly created sensing-signal-to-noise ratio (SSNR), which they propose to use as an alternative to the commonly used signal-to-noise-ratio (SNR), which does not always capture how well a signal is usable for sensing.

### 3 Related works / Current state of the field

Based on the observation that Wi-Fi signals contain information about the environment they travel through, Wi-Fi has been used for several sensing applications, including occupancy/presence detection (J. Yang et al. 2018), movement detection (Yunze Zeng et al. 2014), crowd counting and localization (Choi et al. 2022), activity recognition (W. Wang et al. 2015, J. Yang et al. 2018, Z. Chen et al. 2019), gesture recognition (Kabir, Hasan, and Shin 2022), fall detection (Palipana et al. 2018, Y. Wang et al. 2021), vital sign monitoring (Wang, Yang, and Mao 2017a, Shang and Wu 2016), such as breathing (Atif et al. 2022, Hu et al. 2022, Wang, Yang, and Mao 2017b) and heart rate estimation/classification (Khamis et al. 2018), sleep monitoring (X. Liu et al. 2014, B. Yu et al. 2021, Zhang et al. 2021), and seizure detection (Korany and Mostofi 2022). It cannot only be used for human-centered sensing tasks, there is also research into using Wi-Fi for sensing object properties, such as analyzing soil (Hernandez, Erdag, and Bulut 2021) or determining liquid properties (Song et al. 2020). Importantly, Wi-Fi sensing can also be used for sleep measurements. In this regard, one of the most tackled sleep-related tasks has been breathing rate estimation (J. Liu et al. 2015, Li et al. 2016). As breathing rate is still quite coarse-grained information about a person's breathing, some works have aimed to extract breathing features that might provide more information (B. Yu et al. 2021), such as breathing volume or depth, inhalation and exhalation times and ratios, breathing rate variance, and detection of abnormal breathing, such as in sleep apnea (X. Liu et al. 2016). There have also been a number of works on heart rate estimation during sleep (J. Liu et al. 2015, Gu et al. 2019), which is more challenging than breathing rate estimation due to leading to far smaller displacement, therefore having less of an effect on the wireless signal, and generally resulting in worse performance although some methods still reach average estimation errors below 5 beats per minute (Soto et al. 2022). A small number of works exist for further heart-related measurements, such as estimating the interbeat interval to derive the heart rate variability (Shirakami and Sato 2021, Tsubota, Ohhira, and Hashimoto 2023). While both breathing and heart rate monitoring could be used outside of sleep monitoring, specific to sleep, research has gone into Wi-Fi-based sleep stage classification for which breathing and heart rate have been used as features (Zhang et al. 2021, B. Yu et al. 2021). While SMARS only performs three-stage sleep classification (Wake, NREM, REM), achieving an overall accuracy of

88.4% (Zhang et al. 2021), B. Yu et al. (2021) extend it to four-stage classification for comparison with their WiFi-Sleep system, which achieves an accuracy of 81.8%, outperforming the radar-based RF Sleep and the SMARS adapted for four-stage classification. There has also been work on detecting or classifying movement such as rollovers during sleep, either as a separate feature (Gu et al. 2020, Cao et al. 2020), or to exclude or replace periods containing motion artifacts to improve breathing and heart rate estimation (B. Yu et al. 2021). Although several works have established that sleep pose affects measurements (J. Liu et al. 2015, Gu et al. 2019, Atif et al. 2022), there have been only few efforts in using Wi-Fi for sleep pose classification or estimation (J. Liu et al. 2018, X. Liu et al. 2014). While most reported prototype systems for sleep monitoring work in an offline fashion, collecting data which is then analyzed afterwards, some real-time systems have been proposed (Zhang et al. 2021). For example, there have been several attempts to create a real-time system for monitoring breathing (Li et al. 2016, Wang, Yang, and Mao 2017b) and heart rate (Gu et al. 2019).

To this date, most research aims to show systems that serve as proofs of concept, utilizing setups consisting of multiple PCs, laptops, or full Wi-Fi routers, or combinations of these, limiting real-world applicability. However, some research exists on performing Wi-Fi sensing in ways that are more likely to work in real-world scenarios, performing the sensing directly on edge devices (Soltanaghaei et al. 2020, Hernandez and Bulut 2022a, Hernandez and Bulut 2022b, Hernandez and Bulut 2023), or designing realistic IoT systems utilizing edge devices for collecting the data, compressing it and sending it to a central server for performing the more resource demanding classification or recognition (J. Yang et al. 2018, J. Yang et al. 2022).

As this thesis will utilize ESP32 devices for performing Wi-Fi sensing on the edge, it is to note that to date only few works have used this platform for Wi-Fi sensing (Atif et al. 2022, Choi et al. 2022, Kabir, Hasan, and Shin 2022, Hernandez et al. 2022), although additional frameworks have been proposed that aim to make CSI collection with these devices easier, namely WiESP (Atif et al. 2020), ESP-CSI (Hernandez and Bulut 2020). Notably, while there are works using ESP32 devices for collecting CSI, to my knowledge, this platform has not yet been used for performing realistic sensing tasks directly on the (edge) device, although it is to note that Hernandez and Bulut (2023) performed some benchmarking of specific algorithms on-device to provide guidance for future work utilizing ESP32 devices for on-edge Wi-Fi sensing. This is not surprising, however, as there has been only limited research into Wi-Fi sensing on edge devices in general to date.

# 4 Methods / Implementation

## 4.1 System design & Implementation

### 4.1.1 Design goals

The goal is to implement a real-time sleep monitoring system running solely on ESP32 devices. This system should be fully functional on its own, not relying on other Wi-Fi traffic, thus requiring an active system consisting of a transmitter and a receiver. The system should be realizable using commercial off-the-shelf hardware, such that it is easy to replicate and could be implemented on a larger scale without much effort.

As sleep monitoring can consist of monitoring a multitude of features, a subset of features to be monitored by this system had to be selected for the implementation to be feasible in the time frame of this thesis. The following features were initially selected to implement in this proof-of-concept system to show the suitability of IoT devices, specifically ESP32 devices, for performing intricate sensing tasks such as sleep monitoring on-device:

- human presence/bed occupation
- motion detection
- respiration
- pulse
- sleep stage classification
- sleep posture

The system should thus be capable of detecting whether a person is in range such that the other features can be calculated and produce reasonable results, providing estimates of breathing and heart rate as well as further respiratory and cardiac features that were not specified in advance, and performing sleep stage classification. If the remaining time and the available resources on the ESP permitted, estimation or classification of sleep posture was also to be added to the system implementation.

### 4.1.2 System overview

The system is realized using two ESP32 development boards, with one ESP32 serving as the transmitter, which sends raw WLAN frames to generate CSI, and one ESP32-S3 serving as the receiver, which performs all of the processing and analysis. The functions performed on the receiving ESP consist of preprocessing—i.e. calculating the CSI amplitude, subcarrier fusion, and filtering—feature extraction—including peak/valley

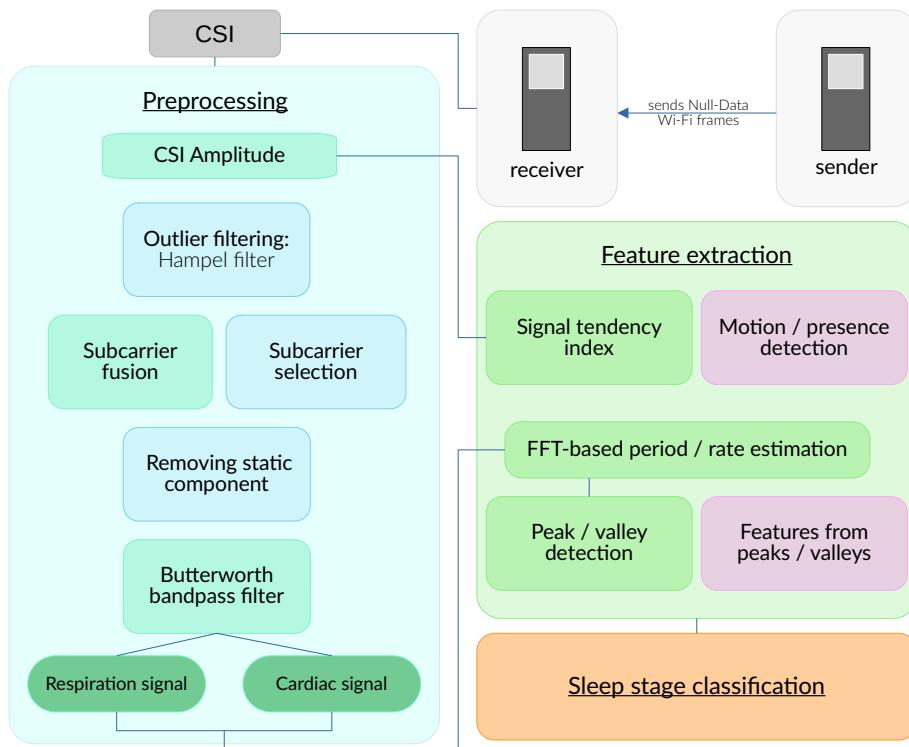


FIGURE 4.1: Overview of the system. Preprocessing steps with a light blue background are optional

detection and calculation of related features—and sleep staging performed by a neural network (NN) which uses the calculated features as input. The following sections will provide a more detailed description of the features the system uses, why they were chosen, and how they were implemented. A graphical overview of the complete system is also provided in figure 4.1.

### 4.1.3 Platform

ESP32 microcontrollers were chosen for the realization of this system such that it could be envisioned to be used in real-world scenarios in a larger scale, which is possible due to these devices being cheap, small, and having a comparatively low energy consumption while still being powerful enough to perform processing directly on the device, including running inference of small neural networks. However, using these devices creates some limitations that have to be considered in the final system design. For instance, a serious hardware limitation is the available RAM of which the original ESP32 microcontroller only has 520 kB, which has to be shared with the operating system and the Wi-Fi stack, limiting the amount of data that can be retained for processing as well as the size of the neural network that has to be loaded into RAM to run inference.

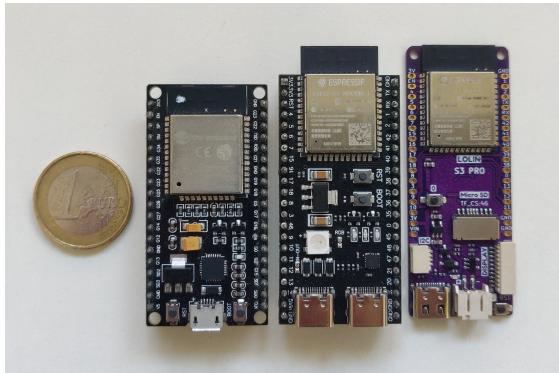


FIGURE 4.2: Development boards used with a 1€ coin as a size reference. From left to right: generic ESP32 development board (transmitter), ESP32-S3 development board (receiver), Lolin S3 Pro (receiver)

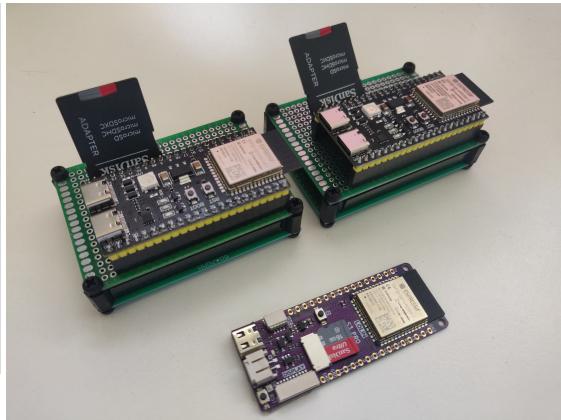


FIGURE 4.3: The three receivers equipped with SD-cards as used for data collection

As the ESP32 can address up to 4 MB of external pseudostatic RAM (PSRAM) and initial testing showed the amount of internal RAM to be very limiting for the processing and analysis, it was deemed necessary for the receiver to use a board that provides at least 4 MB of RAM. Based on current availability, development boards with the ESP32-S3-WROOM-1 module, which includes 8 MB of PSRAM, were selected. This module is part of the newer ESP32-S3 series that can address more than 4 MB of external PSRAM and additionally provides better performance for running inference of neural networks on the device compared to the ESP32 series (Espressif 2022). Specifically, the Lolin S3 Pro<sup>1</sup> and the YD-ESP32-S3 Core Board<sup>2</sup> were used as receivers.

The transmitter consists of a generic ESP32 development board using an ESP-WROOM-32 module without PSRAM, as it is only used for sending null data WLAN frames at a specified rate of 50 Hz and does not require larger amounts of RAM for processing as the receiver does.

Both of the modules used have one PCB antenna used for Wi-Fi, such that the CSI for one received Wi-Fi frame is a vector of the CSI values for the subcarriers.

The system performing the on-device analysis is implemented in C and C++ using the Espressif IoT Development Framework (ESP-IDF), while a live-visualization script and Jupiter notebooks for precalculating parameters and creating and training the neural network are implemented using Python. The source code and corresponding documentation are made available to the public under: <https://github.com/Cl4ryty/bachelor-thesis>

1. [https://www.wemos.cc/en/latest/s3/s3\\_pro.html](https://www.wemos.cc/en/latest/s3/s3_pro.html)

2. <https://github.com/vcc-gnd/YD-ESP32-S3>

### Parameters for CSI collection

To be able to process all of the collected CSI in an online fashion, both devices are configured to run using the 802.11n standard with 20 MHz channel width, which should provide 52 usable subcarriers. Additionally, the receiver is configured to only provide the CSI obtained from the legacy long training field (LLTF), and the rate at which the transmitter transmits frames is set to 50 Hz. Thus, on average, processing for one frame of CSI needs to take less than 20 ms. This very limited amount of time in which processing needs to be completed is one reason not to collect and use more CSI either by utilizing CSI from other LTFs or using 40 MHz channels for more usable subcarriers. Another reason is the limited amount of available RAM, as certain processing methods are calculated over a larger time window, requiring the CSI amplitude for each subcarrier to be retained for such a window, which was deemed unfeasible for more subcarriers or higher sampling rates, especially as RAM needs to be shared between all parts of the analysis.

#### 4.1.4 SSNR

In order to simplify system setup with respect to device placement and orientation, the option to calculate an estimation of the sensing-signal-to-noise-ratio (SSNR) (X. Wang et al. 2022) is provided. The SSNR can be used to model the sensing coverage between a transmitter and receiver, indicating the sensing capabilities of the area and providing a sensing boundary outside of which effective sensing is not possible. The SSNR is defined as the power of the dynamic signal reflected off the target  $P_d$  divided by the power of the other signal components  $P_i$  which includes the static signal, noise, interference, and dynamic effects caused by interferers:

$$SSNR = \frac{P_d}{P_i} \quad (4.1)$$

For this simple implementation, it is assumed that there is only one target and that there are no interferers. The static power is then estimated as the mean amplitude in the observation window, while the interference power is calculated as the difference of the amplitude before and after outlier filtering using the Hampel filter. The dynamic power is estimated by taking the maximum absolute difference between the (outlier filtered) amplitude and the static power within the observation window. As these powers are calculated over an observation window, this feature is quite memory intensive as the values for the chosen observation window size have to be kept in RAM to calculate the SSNR. This feature can therefore not be run parallel with all other features, such as subcarrier fusion and sleep stage classification which also have hefty memory requirements.

### 4.1.5 Preprocessing

This work only uses the CSI amplitude, disregarding the phase information as the raw phase information is practically unusable due to consisting not only of the theoretical phase but also different kinds of random phase noise (Z. Yang et al. 2021). While other works use the phase difference between two antennas, which eliminates noise caused by the RF-chain and has been shown to be stable (Wang, Yang, and Mao 2017a), that solution cannot be applied in this case as the ESP32 can only use one antenna at a time.

For each received CSI frame the amplitude for each usable subcarrier is calculated as in equation 2.6. Null and pilot subcarriers are not disregarded, as collected data indicated that while the ESPs were set to use the 802.11n standard, and the received packets were reported to be of that type, they might actually use the a or g standard as subcarriers 27, 28, -27, and -28 always provided values of zero, indicating that they were null subcarriers, as would be expected for the a and g, but not for the n standard. Furthermore, the pilot subcarrier values appeared to be usable, not overpowering the other subcarriers, such that it was deemed reasonable to use all subcarriers in the further processing and allow the filtering and fusion to use or discard the amplitudes of these subcarriers based on the properties of the signal.

#### Outlier filtering

Outliers are removed by filtering the amplitude of each subcarrier with a Hampel filter. For each value  $x$  in the signal, a window consisting of  $x$  and  $n$  adjacent values on each side is considered and the median and median absolute deviation of this window are calculated. If the absolute deviation of  $x$  from the window median is larger than  $3\sigma$  with  $\sigma \approx 1.4828 \times MAD$ ,  $x$  is considered an outlier and replaced with the median of the window. Outlier filtering is implemented mainly as a preprocessing step for the variance-based subcarrier selection which selects the subcarrier with the highest variance—a metric that can be distorted by outliers.

#### Subcarrier fusion

In order to improve the signal quality and as a means of dimensionality reduction, fusion of the amplitudes of all subcarriers is performed. This is done using MRC-PCA (B. Yu et al. 2021), which maximizes the contribution of signals in the expected signal frequency spectrum, smoothing out noise while retaining the desired signal. MRC-PCA fuses signals according to maximal-ratio combining (MRC) by providing a different gain for each signal depending on the signal's signal-to-noise ratio (SNR), but as CSI amplitudes of different subcarriers can exhibit opposite directions, principal component analysis (PCA) is used to determine the direction for each subcarrier and correct it to avoid the signals canceling each other out when fusing them by means of a weighted average. More specifically, to perform subcarrier fusion using MRC-PCA, first the SNR

for each subcarrier is estimated by obtaining the power spectral density (PSD) of the subcarrier amplitude and calculating the ratio of the signal energy within the range of the normal respiratory/cardiac rate and the signal energy outside of this range. Mathematically, the signal-to-noise ratio estimate for subcarrier  $i$ ,  $SNR_i$ , is

$$SNR_i = \frac{E_{signal_i}}{E_{noise_i}} \quad (4.2)$$

with  $E_{signal_i}$  and  $E_{noise_i}$  being the signal energy within the range of the normal respiratory/cardiac rate and the signal energy outside of this range for subcarrier  $i$  respectively. Secondly, the amplitude of each subcarrier is filtered with a bandpass filter with cutoffs corresponding to the normal respiratory/cardiac range, and the first principal component of the filtered signal is calculated. The sign of this principal component is then added to the previously estimated SNR to obtain the final MRC gain. The MRC gain  $m_i$  for subcarrier  $i$  is therefore

$$m_i = \begin{cases} SNR_i & \text{if } p_{1i} \geq 0 \\ -SNR_i & \text{if } p_{1i} \leq 0 \end{cases} \quad (4.3)$$

with  $SNR_i$  being the SNR estimation for the  $i$ th subcarrier and  $p_{1i}$  the first principal component of the filtered amplitude of the  $i$ th subcarrier. Is

The original amplitudes of all of the  $N$  subcarriers can then be fused into one amplitude  $a_{fused}$  by calculating the weighted average, weighting the subcarrier's amplitude  $a_i$  according to its MRC gain  $m_i$ :

$$a_{fused} = \sum_{i=0}^N \frac{m_i * a_i}{\sum_{j=0}^N m_j} \quad (4.4)$$

As the MRC-PCA includes a filtering step, where the signal is bandpass filtered before performing PCA in order to ensure that the PCA maximizes the amplitude of the actual signal and not the noise, subcarrier fusion is performed separately for obtaining the respiration and cardiac waveform using bandpass filters with the corresponding cutoff frequencies (see 4.1 for the specific values). For the sake of keeping the implementation simple, the Butterworth bandpass filter used in a later step for filtering the signal to obtain the respiration and cardiac waveforms is used as bandpass filter for this step in the fusion process.

#### Reference implementation: variance-based subcarrier selection

To be able to evaluate the effectiveness of the MRC-PCA subcarrier fusion technique, variance-based subcarrier selection was implemented as a baseline to compare against.

With this method, only the amplitude of one of the subcarriers is used for further processing, and the subcarrier is selected based on the variance of the amplitude in a specific time window, usually selecting the subcarrier with the maximum amplitude variance in the window. This simple method has previously been used successfully in other works (Gu et al. 2019) and initial experiments have shown it to also be effective for naive breathing rate estimation based on CSI collected from an ESP32 device<sup>3</sup>.

In this implementation, the subcarrier with the highest amplitude variance in the window that is kept for analysis (with the default window size being 30 seconds) is selected. The selection is performed every X seconds with the default X value being 30 seconds. To avoid issues in the further processing and feature extraction due to the sudden change in the signal if a different subcarrier is selected, in such a case the stored amplitude of that subcarrier is bandpass filtered and replaces the array of filtered amplitudes from the previously selected subcarrier. Further, the FFT rate estimation is performed, using the filtered amplitude of the newly selected subcarrier, and the peak-valley detection is reset such that previously detected points do not influence the following detection.

TABLE 4.1: Respiratory and Cardiac cutoff frequencies. Frequencies were chosen to be extreme values encompassing a larger range than the normal rest frequency ranges for healthy adults.

	Respiration		Cardiac	
	Frequency (Hz)	Rate (bpm)	Frequency (Hz)	Rate (bpm)
Low cutoff	0.1	6	0.75	45
High cutoff	0.5	30	2.0	120

### Removal of the static component

To enhance sensing performance, the static CSI component is removed after subcarrier fusion or selection. As the static component remains the same over time, it is estimated as the mean amplitude over the time window that is kept for analysis. If the amplitude of a single selected subcarrier is used for further processing, the static component is estimated as the mean of the amplitude of the selected subcarrier in the time window kept for analysis, which is then subtracted from the current amplitude value for each new CSI sample. If subcarrier fusion is performed, the mean of the fused signal in the time window kept for analysis is used as an estimate of the static component and subtracted from the current fused amplitude value for respiratory as well as cardiac features, and this is done for each new CSI sample.

3. <https://youtu.be/bW1JCm654fE>

## Filtering

The fused amplitude waveforms—or the amplitude of the selected subcarrier for the reference implementation—are filtered with a Butterworth bandpass filter with cutoff frequencies corresponding to the normal range of cardiac and respiratory rate (see 4.1) to obtain the respiration and cardiac signal.

### 4.1.6 Feature extraction

#### Presence and motion detection

Following J. Yang et al. (2018), the signal tendency index (STI) of the subcarrier amplitudes is used to implement a simple presence detection. Originally, Doppler-MUSIC was to be used for motion detection. However, during the implementation, it became clear that this is unsuitable for online processing as the necessary calculation could not be performed fast enough on the ESP<sup>4</sup>. After initial tests showed that the STI appears to be higher for larger motions than for small ones, it is also used to implement a simple method for motion detection—differentiating between small and large movements. For these three detection tasks—presence, small, and large movement—thresholds have to be set such that presence or movement is detected when the STI value is larger than the corresponding threshold. Thresholds can be calibrated directly on the device, by receiving signals from another device (e.g. a smartphone or laptop) via UDP broadcasts, with certain signals corresponding to starting calibration and the activities as well as ending the calibration. Once the "end calibration" signal is received, the optimal thresholds are calculated by maximizing the F-score for the corresponding activity. Calculated thresholds are stored in non-volatile storage and are automatically retrieved from there when the ESP starts, allowing for restarts of the receiver without requiring recalibration. However, as the STI values can change depending on the environment and the orientation of the ESPs, calibration should be performed if the devices were moved or the environment changed.

#### Peak and valley detection

For detecting peaks and valleys, minima and maxima between moving average curve crossings were considered (Lu et al. 2006), a method which has been successfully used for peak and valley detection of respiration and cardiac waveforms generated from radar (Sharma et al. 2020), which are similar to the waveforms extracted from the CSI amplitude especially when it comes to contained noise. As the algorithm described by Lu et al. (2006) is designed for offline processing, some adaptations were necessary to perform the detection in an online system. Firstly, the signal period  $T$  is estimated by

4. more specifically the eigenvalue decomposition of the 50x50 correlation matrix for 1 second of data took around 6 seconds

applying FFT to the last 1024 samples of the waveform, which corresponds to around 20 seconds of data at the sampling rate of 50 Hz. This value was chosen as at least one cycle needs to be included in this window, the FFT implementation used requires the number of samples to be a power of 2, and the window size determines the frequency resolution of the FFT with larger windows providing higher resolution. The chosen window size gives a resolution of around 3 bpm for the rate estimates. To keep the implementation simple this value is used both for respiratory and cardiac waveforms. As in the reference paper, the mean is subtracted from the waveform before performing FFT, then the location of the peak in the spectrum is used as an estimation of the signal frequency  $f$ , such that the period  $T$  is thus  $T = 1/f$ . Secondly, the moving average curve is calculated at every time point  $t$  as

$$MAC(t) = \overline{x(\tau)}|_t^{t-2T}, \quad (4.5)$$

with  $x$  being the signal,  $\tau$  being the time, and  $\overline{x(\tau)}|_{t_1}^{t_2}$  being the average of  $x$  during  $[t_1, t_2]$ .

Intercepts are detected as described by Lu et al. (2006), with the addition that no intercept is identified if  $MAC(t) = x(t) = x(t-1)$ . Thus, up-intercepts are found when the following condition holds

$$\{x(t-1) \leq MAC(t-1)\} \cap \{x(t) \geq MAC(t)\} \cap \overline{\{MAC(t) = x(t) = x(t-1)\}}, \quad (4.6)$$

while down-intercepts are detected if the following condition is met:

$$\{x(t-1) \geq MAC(t-1)\} \cap \{x(t) \leq MAC(t)\} \cap \overline{\{MAC(t) = x(t) = x(t-1)\}}, \quad (4.7)$$

A peak is then the maximum value of the signal between an up- and a down-intercept, while a valley is the minimum value of the signal between a down- and an up-intercept. Two of the criteria for removing irregular peaks/valleys additionally introduced by Lu et al. (2006) were implemented, namely discarding intercepts within  $T/20$  of the previous intercept, as well as peaks and valleys and the corresponding new intercept if the amplitude was smaller than 20% of the mean peak to valley amplitude. Different from the original paper, for consecutive intercepts of the same type, the new intercept is discarded, as removing the first would also require removing the associated peak or valley, which would require the corresponding features to be recalculated. As the choice to remove the first rather than the second intercept was arbitrary (Lu et al. 2006), and the fact that, due to the addition to the intercept criteria, consecutive intercepts of the same type should never occur normally, but can only occur if an intercept was deemed invalid due to the amplitude criterion, this was deemed a reasonable adaption for the ease of processing in an online system.

## Further features

The following further features are calculated both for the peaks and valleys detected from the cardiac as well as the respiration waveform:

- Instantaneous Peak Rate
- Instantaneous Valley Rate
- Mean Peak Rate over Window
- Mean Valley Rate over Window
- FFT Rate over Window
- Variance of Peak Rate in Window
- Variance of Valley Rate in Window
- Mean Up-Stroke Length
- Mean Down-Stroke Length
- Up-Stroke Length Variance
- Down-Stroke Length Variance
- Up-to-Down Length Ratio
- Fractional Up-Stroke Time
- Mean Up-Stroke Amplitude
- Mean Down-Stroke Amplitude
- Up-Stroke Amplitude Variance
- Down-Stroke Amplitude Variance
- Up-to-Down Amplitude Ratio
- Fractional Up-Stroke Amplitude

All features are calculated again when a new point relevant to the feature is found, the list is color-coded to show which features are calculated after which kind of point is found. Specifically, the green items are calculated when a peak is found, the orange items are calculated when a valley is found, and the black items, which are all variance features, are calculated after either a peak or valley is found. The FFT rate estimate (red) is calculated irrespective of found points every ten seconds and is used as the period estimate  $T$  in the peak and valley detection algorithm. The FFT-based rate estimation also serves as a baseline to which the breathing and heart rate estimates derived from the peak-valley-detection can be compared.

Thus, there are 19 features for each signal type calculated based on the cardiac and respiration waveforms.

All features except ratios, fractional times, and fractional amplitudes are calculated for both peaks and valleys as the signal might be flipped compared to the reference signal.

Features were chosen for breathing based on WiFi-Sleep (B. Yu et al. 2021), which utilized respiratory rate (corresponding to instantaneous and mean rate), average inspiratory length and amplitude (mean stroke length and amplitude), fractional inspiratory amplitude (Fractional Up-Stroke Amplitude), and the inspiration-to-expiration-ratio (Up-to-Down Amplitude Ratio), as well as variance of these features to perform sleep staging based on these respiratory as well as additional motion features. To keep the system implementation simple, the same features were used to extract both respiratory and cardiac features.

#### 4.1.7 Sleep stage NN

For the sleep stage classification, two different neural networks were designed, a fully connected dense neural network to be used as a reference and a more complex network, including a convolutional and recurrent part. Originally, the fully connected dense NN was chosen as the final network design based on the assumption that a more complex and for the nature of the data more fitting recurrent neural network would be too large to fit into the RAM of the ESP and too computationally expensive to run inference at a reasonable rate and without affecting other processing. This initial assumption is based on the experiments carried out by Hernandez and Bulut (2023) who tested fully connected models on an ESP32 both with and without PSRAM and reported that the largest model that would run to have a size of 3,147.5 kB and using INT8 quantization, for a model with four layers with 100 neurons each, and an input of size  $64 \times 100$ . However, the ESP32 used in these experiments can only address a maximum of 4 MB of PSRAM at a time, while the ESP32-S3, which was selected for the final implementation to accommodate the RAM requirements of this system, does not have this limitation. After initial tests confirmed that the ESP32-S3 allowed for much larger networks to be run, with the largest tested to be working having a compiled size of 13,154.25 kB, an LSTM-based model was implemented. TensorFlow Lite for Microcontrollers is used as the framework for implementing the NN. This allows creating and training the NN on a more powerful device and then converting and storing it on the ESP to run inference directly on the microcontroller. TensorFlow Lite for Microcontrollers currently does not support training on the microcontroller, but it is not uncommon for neural networks to be only trained before deployment. Due to the limitations of microcontrollers, TensorFlow Lite for Microcontrollers only supports a subset of all TensorFlow operations, and the size, and thereby also the architecture of the NN, is limited by the amount of RAM of the microcontroller, as the NN has to be loaded into RAM to run inference.

The previously extracted features are used as input for the NN to be able to reduce the network size. This sort of hybrid approach where classical approaches are used to extract features that are then provided to a NN which performs the desired sensing task is often used in Wi-Fi sensing to reduce network size and training time as well as required training data, that would be necessary if the NN was trained to perform the sensing task on raw CSI. As a default inference is set to be performed every 30 seconds.

#### Network architecture

The dense network for the sleep stage classification consists of four "hidden" layers, with 500 neurons each activated using the ReLU (rectified linear unit) activation function, and an output layer consisting of 6 neurons, using the softmax activation function to obtain probabilities of the output classes. The input is a  $64 \times 45$  matrix consisting of 64 feature vectors representing the last 32 seconds of data, as the current features

are added to this sliding feature matrix every 0.5 seconds. The feature vectors consist of the 42 calculated features, namely the current STI value, the 3 presence/movement detection results, and the respiratory and cardiac features (19 each). The output of the network is a probability vector of length 6 of which the maximum is taken to obtain the classification result, with the six categories being no person detected, awake, N1, N2, N3, REM.

The other network consists of a multi-scale convolutional neural network with three convolution and max pooling layers, an LSTM, and a dense output layer. The design is based on the network proposed by B. Yu et al. (2021), with adaptions to allow inference on the ESP. This means that due to the RAM limitation, only a smaller window of data can be processed and used as input features and consequently fewer convolution layers are used. Furthermore, TensorFlow Lite for Microcontrollers does not currently support bidirectional LSTMs, and CFRs are also not available, such that a unidirectional LSTM and a dense output layer were used instead.

No feature scaling is performed due to the system performing online processing, resulting in the full range of the feature values over a night being unknown during the processing.

### Data collection and training

Due to time and resource limitations, it was not possible to collect training data, train, deploy, and evaluate the neural networks. However, a pipeline for data collection, training, and deployment was implemented and is made available with detailed documentation alongside the other system code<sup>5</sup>. Further, to generate more training data and make the system more robust, it is suggested that training data is collected by multiple receivers at the same time while the ground truth is collected with the PSG device. Consequently, one transmitter is placed on one side of the bed, while the receivers are placed on the other side of the bed spaced apart equally. The CSI at each of the receivers will therefore be slightly different, while still containing the desired information.

## 4.2 Experiments

### 4.2.1 Experiment design

Experiments are conducted to measure the performance of the system for sleep monitoring against the gold standard PSG. Specifically, the breathing and heart rate estimates are evaluated. To test the influence of different sleep postures on the performance of the respiration and cardiac feature calculation data is collected for four different common postures while the test person is awake to control the posture, and the rate estimates for the collected data are evaluated. In order to test the effect of a blanket in

5. <https://github.com/Cl4ryty/bachelor-thesis>

addition to the sleep postures, data for all postures is collected for two conditions, with and without a blanket covering the target, and the rate estimates evaluated. To validate the design decisions, the effect the certain system components on the performance is tested by processing a subset of the collected data with these components turned off or replaced. Specifically, the effect of the following components on the system performance is evaluated:

- removing the static CSI component
- outlier filtering using the Hampel filter
- subcarrier fusion versus variance-based subcarrier selection
- peak/valley detection amplitude criterion

Additionally, the effect of different device placement on the monitoring performance is evaluated.

#### 4.2.2 Hypothesis / expected results

It is expected that the blanket will negatively impact the performance due to muffling the movement caused by respiration and pulse, making it harder to detect and therefore estimation more difficult. Based on previous research, it is also expected that performance will differ for the different sleep postures. Specifically, it is expected that the system will perform best for the supine, followed by the lateral postures, with the prone position being expected to lead to the largest estimation errors. These expectations are based on the device placement and direction of the motion caused by breathing and pulse, which is very pronounced in the supine position and likely to have a larger effect on the CSI than for other postures due to being roughly orthogonal to the LOS path for the proposed device placement. In contrast, the movement caused by breathing and pulse is smaller in the prone position, being mostly absorbed by the mattress, thus sensing performance should be worse. Regarding the test of the system components, it is expected that removing the Hampel filter will lead to worse performance due to unfiltered outliers negatively impacting peak/valley detection. Based on recommendations from the literature, removing the static CSI component should lead to better sensing performance. Subcarrier fusion is expected to outperform the system utilizing variance-bases subcarrier selection, due to the former making use of information from all subcarriers, which should make it more robust to noise that might only occur on some frequencies.

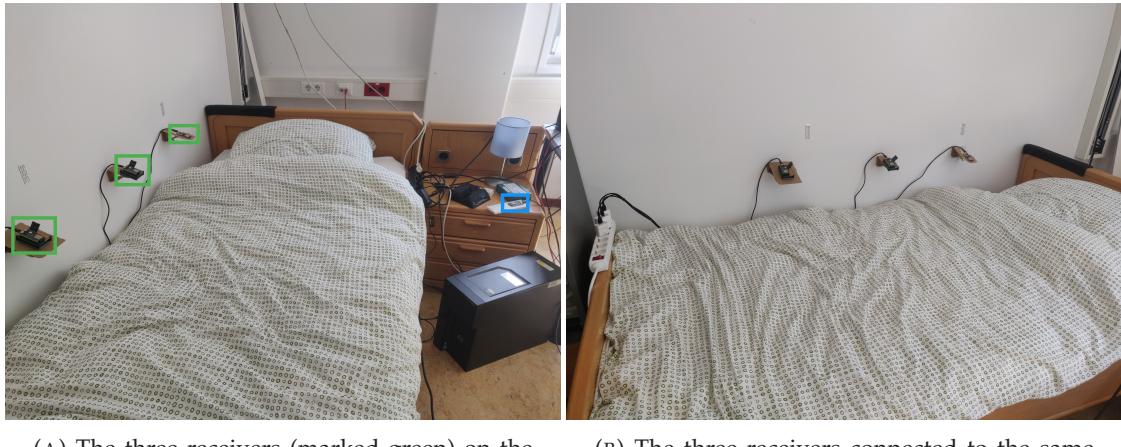


FIGURE 4.4: Setup of the ESPs in the sleep lab

#### 4.2.3 Reference device & Performance metrics

The system is evaluated against the gold standard polysomnography. Specifically, the SOMNO HD Eco from SOMNOmedics is used, and the manufacturer's DOMINO software to automatically analyze and export the data for further analysis of the system performance.

To evaluate the system performance, the breathing and heart rates provided by the PSG and corresponding software are compared to the system's estimates. The breathing and heart rates provided by the reference device are the instantaneous rates for each detected breath or pulse. To evaluate the system's performance the rate error and absolute rate error are used and calculated for both the peak rate estimates and the FFT-based estimates.

#### 4.2.4 Experiment overview

To evaluate the system's robustness to different device placement, multiple receivers are employed that collect data simultaneously. Specifically, one transmitter is placed on one side of the bed, and three receivers are placed on the wall on the other side of the bed, with the receivers all spaced approximately 40 cm apart. The receivers are powered from the same power strip, such that they all can be powered on at the same time by turning on the power strip. This is done to ensure simultaneous start-up of the ESPs such that their clocks are synchronized and the amount of collected data should be equal for all of them to facilitate synchronization for the analysis. A photo of the setup is provided as figure 4.4.

The SOMNO HD eco is used to record a 1-lead ECG and the signal from respiratory inductance plethysmography (RIP) bands around the chest and abdomen from which heart and respiration rates are calculated respectively. The ECG is recorded at a rate of

256 Hz and the signal from the RIP bands has a resolution of 32 readings per second. For the ESP32-based monitoring system, the transmitter is set to send frames at a frequency of 50 Hz and the receivers write all features as well as the raw CSI and information about the Wi-Fi frame to a file on the SD card for each received CSI sample.

Data from the receivers is collected throughout the whole session, while the reference data from the PSG device is recorded separately for each of the eight conditions. To facilitate synchronization of the data from the PSG and the ESPs a button on one of the receivers—hereafter also referred to as reference receiver—is pressed when the recordings from the PSG device are started and ended, and the button press is logged to the SD card with the next processed sample.

All receivers are set to the same standard configuration (see table 4.2), with the exception that the reference receiver—the rightmost one, also referred to as right—also logs to UDP to allow live visualization during the data collection. For the evaluation of specific system components, the CSV file containing the collected data from the reference receiver was brought into the expected format by removing all unexpected columns and loaded onto the SD card for processing. The configuration was changed from the standard one, setting the Process CSI from file option to true and changing one of the other options according to the component to be tested.

#### 4.2.5 Collected data

Data was collected from one subject in one session, first for the open and then for the blanket condition, with four collection phases each for the posture sub-conditions—supine, left lateral, prone, right lateral—for a total of eight different test settings.

As initial data from the PSG showed an abnormally low breathing rate of the subject that falls below the range of rates the system is designed to detect (less than 6 bpm), the subject was instructed to breathe more quickly to achieve a more normal breathing rate. Importantly, the subject was still free to choose and vary their breathing rate during the data collection.

For each condition, data was collected for approximately five minutes, with a 30 to 60 second period before the start of the recording where the person was already in the tested posture and instructed to move very little to allow the signals of both tested and the reference device to stabilize such that movement artifacts before the measurement do not negatively impact the recorded performance.

As starting and ending the recording involved movement which can lead to artifacts for both the reference device and the system to be tested, these portions of the recorded data were disregarded for the analysis. Specifically, the first 15 seconds of recorded data for each condition were not used for the analysis, and the end point for each condition was chosen manually based on visual inspection of the raw RIP and ECG data provided by the PSG device. The total duration of data used for analysis for each condition is

TABLE 4.2: Standard receiver configuration

Configuration Option	Value
Process CSI from file	false
Frames per second	50
Time range of samples to keep	30
Calculate SSNR	false
Perform outlier filtering	y
Hampel filter window size	3
Remove static component	false
FFT every x seconds	10
Subcarrier selection instead of fusion	false
MRC-PCA on timer	y
MRC-PCA every x seconds	120
MRC-PCA after large movement	false
MRC-PCA on new presence	false
Enable peak/valley amplitude criterium	y
Collect training data for NN	false
Run inference	false
Number of samples for NN input	64
WiFi Channel	1
Log to SD card	y
Log to serial	false
Log to UDP broadcast	false
Log different things to different outputs	false
print CSI	y
print amplitudes	false
print sti value	y
print presence/movement detection results	y
print presence/movement thresholds	y
print fused heart amplitude value	y
print fused breath amplitude value	y
print filtered heart value	y
print filtered breath value	y
print heart features	y
print breathing features	y
print heart poi	y
print breathing poi	y

given in table 4.3. Notably, less data was recorded for the blanket  $\times$  prone condition, which can be attributed to human error in the process of starting and stopping the recording.

TABLE 4.3: Time duration of the data used for analysis for each condition

	Supine	Left	Prone	Right
Open	4:20	4:42	1:46	4:38
Blanket	4:37	4:34	4:38	4:48

#### 4.2.6 Data analysis

As the start and end points for each condition, which are subsequently used to synchronize the data from the ESP and the ground truth, are marked via button press for only the reference ESP, these markers have to be transferred to the data collected by the other receivers in order to compare the performance for the different device placements. Originally, it was assumed that this could be achieved by marking the samples of the same indices, which would correspond to the CSI obtained from the same Wi-Fi frame for devices that received and recorded the same amount of samples. However, as the number of recorded samples differed for the three receivers, the button press is synchronized according to the timestamp of the received CSI, such that the first sample with a larger timestamp than the timestamp of the reference ESP sample, for which the button press was logged, is treated as if a button press was recorded for this sample.

Results are then calculated as follows. First, the data obtained from the PSG device is synchronized with the data from the ESPs according to the start of the recording by assuming that the button press corresponds to the start of the recording. The timestamp of the Wi-Fi frame for which the button press was recorded is then subtracted from the following timestamps such that each recording is taken to start at time 0. Timestamps for the ground truth provided by the PSG are similarly normalized. As the breathing and heart rate estimates provided by the reference device are instantaneous estimates and provided for each detected peak, and the estimates from the ESPs are provided for each received CSI frame, the sampling rate of the estimates differs and has to be matched for further analysis. For this, the rate estimates provided by the PSG device are upsampled by taking the estimate for a specific time to be the most recent estimate. In this manner, a ground truth value for the breathing and heart rate is assigned to each sample collected by the receivers. Then the error and absolute error are calculated as the estimated rate minus the ground truth and the absolute value of the former. This is done for both the mean peak rate and the FFT rate estimate. The analysis for the effect of posture, blanket, and system components is performed on the data from the reference receiver, while the data from the other two receivers is only used to evaluate the effect of the device position on the performance.

## 5 Results

In the following, plots and tables of the collected and analyzed data most relevant to the tested hypotheses are presented, but for the sake of completeness, additional tables with the mean, minimum, maximum, and median errors for different condition and posture groupings than shown here, as well as additional plots are provided in the appendix. For the sake of readability, values mentioned in the text are rounded to two decimal points.

The STI values collected for the total recording session are plotted in 5.1 for each device. The intervals during which the individual conditions were recorded are marked. As can be seen, the STI remains relatively stable throughout the recording of each condition, while fluctuating during the periods between conditions, during which the target moved. It can also be noted that the STI value is not always around a specific value for recording periods during which the target lying still, and is different for each device. Specifically, for the rightmost receiver, the STI value is higher for the conditions in which the subject is in the right lateral posture than for all other postures  $\times$  conditions pairings. This is relevant for the proposed presence and movement detection, which is based on thresholding the STI, but which was not explicitly tested in these experiments.

Noticeably, there are spikes in the STI values that appear at regular intervals of 120 seconds, the same interval at which the MRC-PCA coefficients are calculated. While they do not appear after every 120 second interval, and there are some spikes in between these intervals, this pattern of spikes occurs across devices, with the difference that for the right ESP, which served as the reference receiver, there are spikes at the beginning and end of the recording sections, corresponding to the calculation of the subcarrier fusion coefficients after a button press, and the following spikes appear after intervals of 120 seconds after these and not in sync with the spikes at the other receivers.

A more detailed look at the STI plots reveals that the STI exhibits a sinusoidal pattern for several portions of the recording, which seems to correlate to the breathing of the target. One such instance is plotted in figure 5.2, which shows the reference signal from the abdominal RIP band, and the STI and filtered breath amplitude of the right receiver during the left  $\times$  blanket condition.

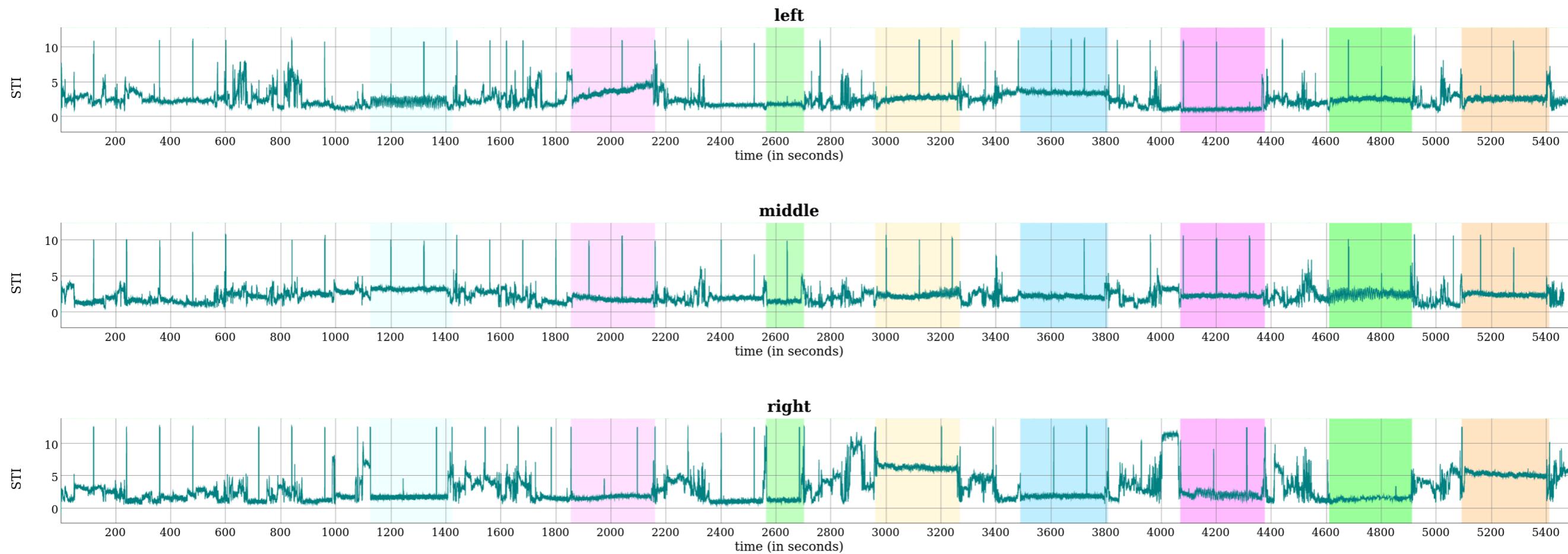


FIGURE 5.1: Signal tendency index values over the whole data collection session for the three receivers. Recording periods are marked, with light colors indicating the open condition and more saturated colors the blanket condition. For the postures the colors are matched as follows: supine = blue, left = pink, prone = green, right = orange

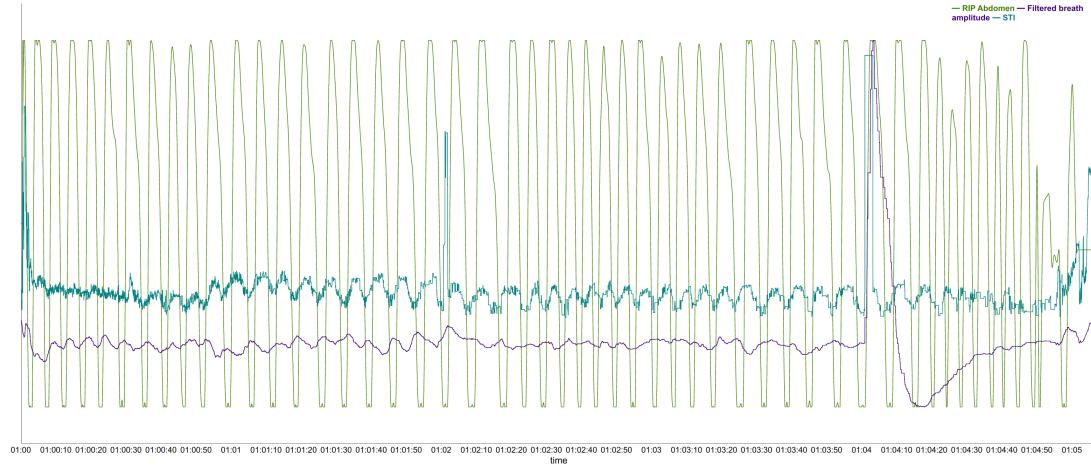


FIGURE 5.2: Reference breathing signal provided by the abdomen RIP, and the STI and filtered breath amplitude of the right receiver during the left  $\times$  blanket condition

## 5.1 Effect of posture

Figure 5.3 shows the distribution of the breathing rate errors for the different postures. As can be seen, the peak rate error, which is the difference between the mean peak rate provided by the receiver and the ground truth recorded by the PSG device, has a high variance and the error is mostly positive with the largest error being 145.31 bpm for the supine posture, meaning that the mean peak rate tends to underestimate the real breathing rate. The performance is best for the prone and left lateral postures, which have a median peak error and absolute peak error rate of 21.82 and 16.79 bpm respectively. The FFT rate error has a lower variance, although the error range is still larger than the normal range for breathing, with the errors for the prone posture ranging from -18.14 to 12.51 bpm for the smallest range and from -65.07 to 28.16 bpm for the right lateral posture as the largest range. The FFT rate errors are mostly centered around zero, with the FFT rate estimates neither primarily over- or underestimating the true breathing rate. The FFT rate estimates are better than the peak rate estimates, with the mean and medians of the absolute FFT rate errors for the two postures with the best results—left and prone—lying below 10 bpm compared to the lowest mean and median for the absolute peak error rate of 16.79 (left) and 26.31 bpm (prone).

As can be seen in figure 5.4, similar as for the breathing rate error, the peak-based heart rate estimation errors have a large variance, and peak rate estimates seem to mostly underestimate the true heart rate, with the estimates for the supine and right lateral postures always underestimating the true rate. As for the breathing rate errors, the FFT rate errors have a smaller range than the peak rate errors, but for the heart rate

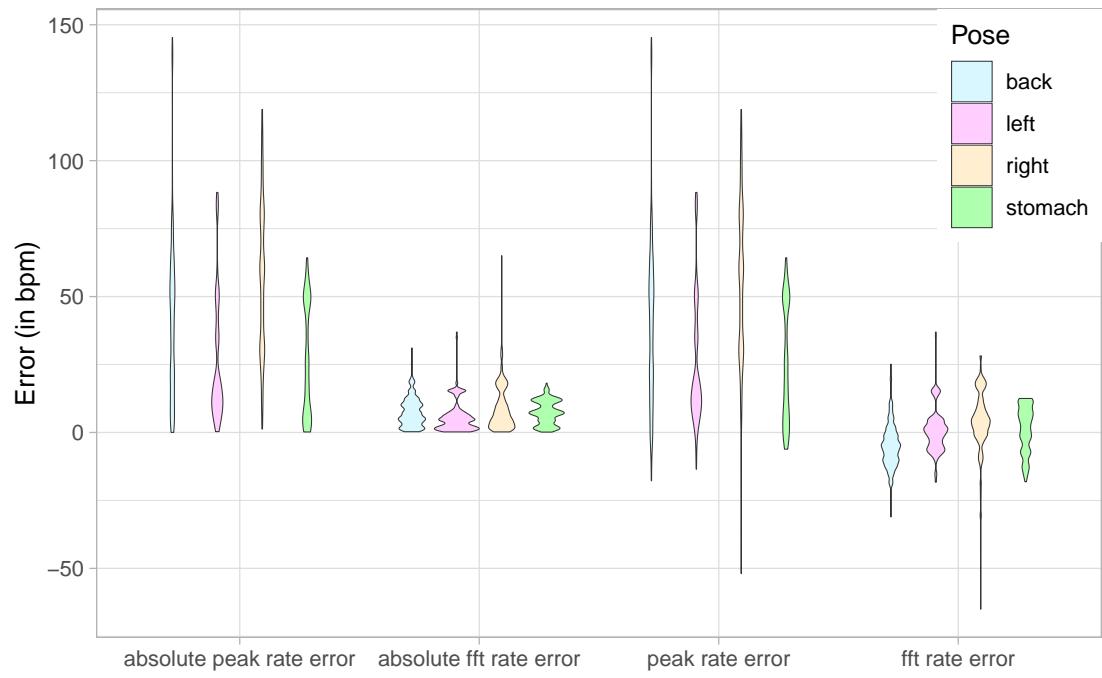


FIGURE 5.3: Breathing rate estimation errors for the different postures across conditions

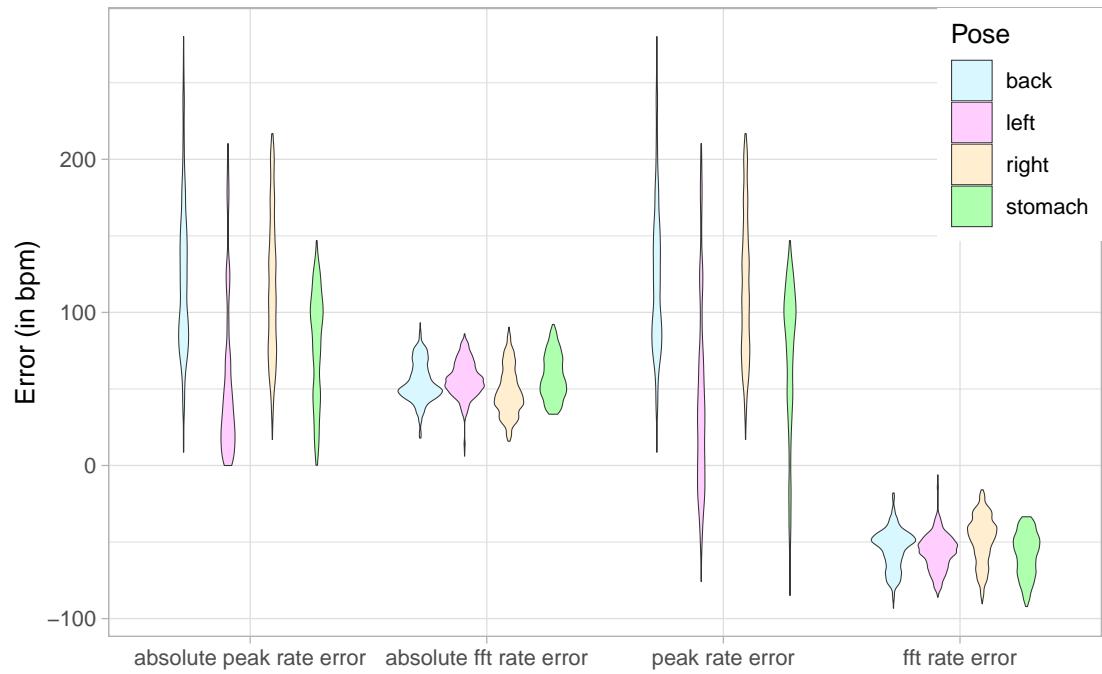


FIGURE 5.4: Heart rate estimation errors for the different postures across conditions

estimates, they are all approximately centered around -50 bpm, indicating that the FFT-based rate estimation overestimates the true heart rate by around 50 bpm on average.

## 5.2 Effect of blanket

Figure 5.5 shows the breathing rate estimation errors for the different conditions, blanket, during which the subject's torso was covered with a blanket, and open, during which the subject was lying on the bed without being covered by a blanket. As can be seen in that plot, the FFT-based rate estimation performs better for the open condition, with the FFT rate errors for the blanket conditions having a higher variance with more extreme errors, although both are centered around zero. This difference also becomes apparent from the mean and median absolute FFT rate errors, which are lower for the open condition (mean: 6.44 bpm, median: 5.21 bpm) compared to the blanket condition (mean: 8.80 bpm, median: 7.21 bpm). Regarding the peak-based rate estimates, these seem to be better for the blanket condition, as it is evident that there is more underestimation for the open condition, with a median of 49.39 compared to 23.10 bpm for the blanket condition for the peak rate error and medians of 49.39 and 24.03 bpm for the absolute peak rate error, showing that most of the error for both conditions is positive, thus indicating underestimation of the true breathing rate.

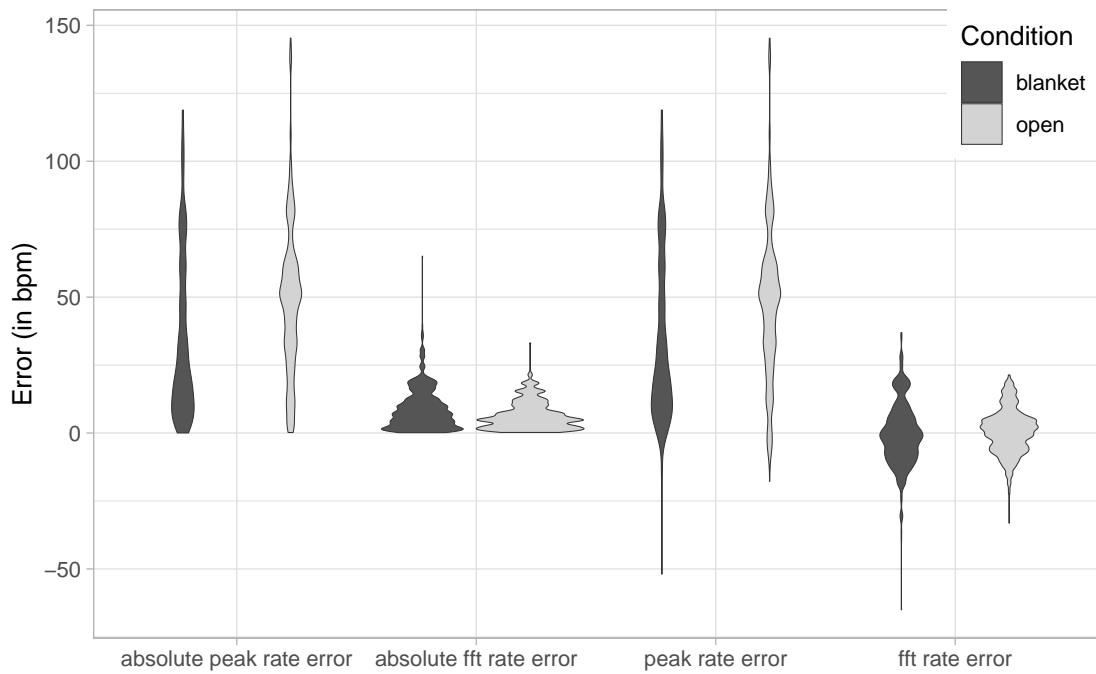


FIGURE 5.5: Breathing rate estimation errors for the different conditions across postures

The effect of the conditions on the performance of the heart rate estimation is similar to the effect seen for the breathing rate estimations as evidenced by figure 5.6, with

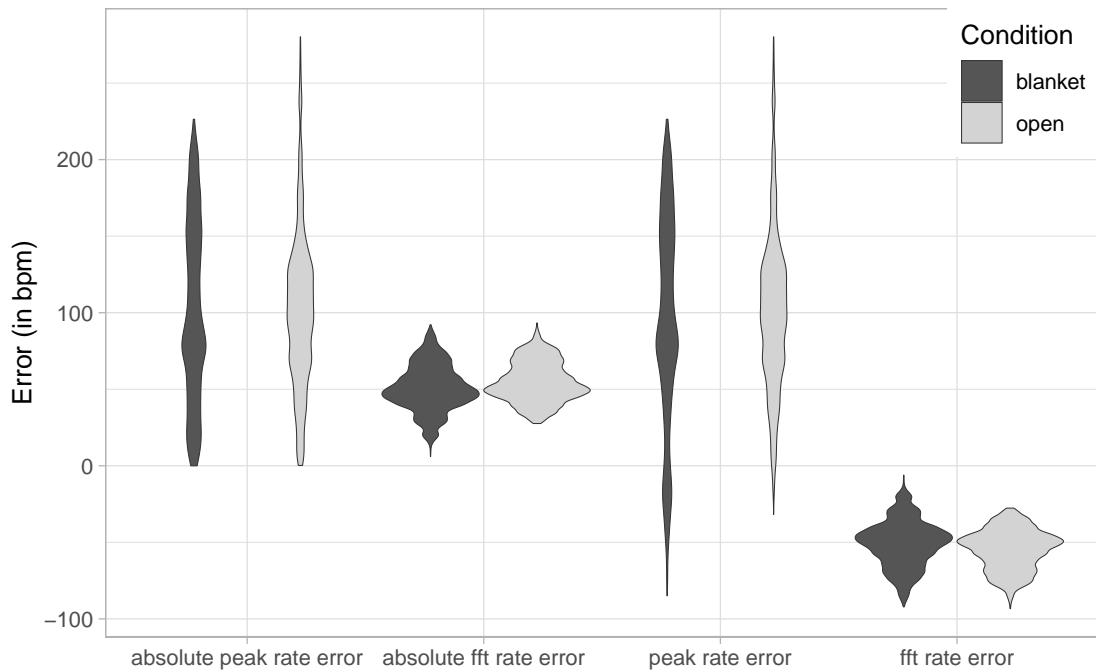


FIGURE 5.6: Heart rate estimation errors for the different conditions across postures

lower FFT rate errors for the open and lower peak rate errors for the blanket condition. As seen before in figure 5.4, depicting the heart rate errors for the different postures, the FFT rate errors are centered around -50 bpm, meaning that the FFT-based heart rate estimate overestimates the true rate by that amount, while the peak rate errors vary a lot, with absolute means and medians around 100 bpm and mostly underestimating the true heart rate.

### 5.3 Effect of different system components

Figure 5.7 and 5.8 show the estimation errors for different system setups for breathing and heart rate respectively. As can be seen from these two figures, removing the amplitude criterion for the peak/valley detection has almost no effect in terms of breathing and heart rate estimation errors when compared to the default system with the amplitude criterion.

Disabling the Hampel filter while keeping the other system parameters the same as the default and rerunning the processing on the data collected from the reference receiver leads to better FFT-based breathing rate estimations, as evidenced by the errors being more concentrated around zero than for the default setup where the FFT-based rate estimation tends to underestimate the true breathing rate a bit more, as also becomes clear from the mean and median FFT rate errors which are -2.13 and 0.79 bpm for the system without Hampel filter and 5.72 and 7.44 bpm for the default system.

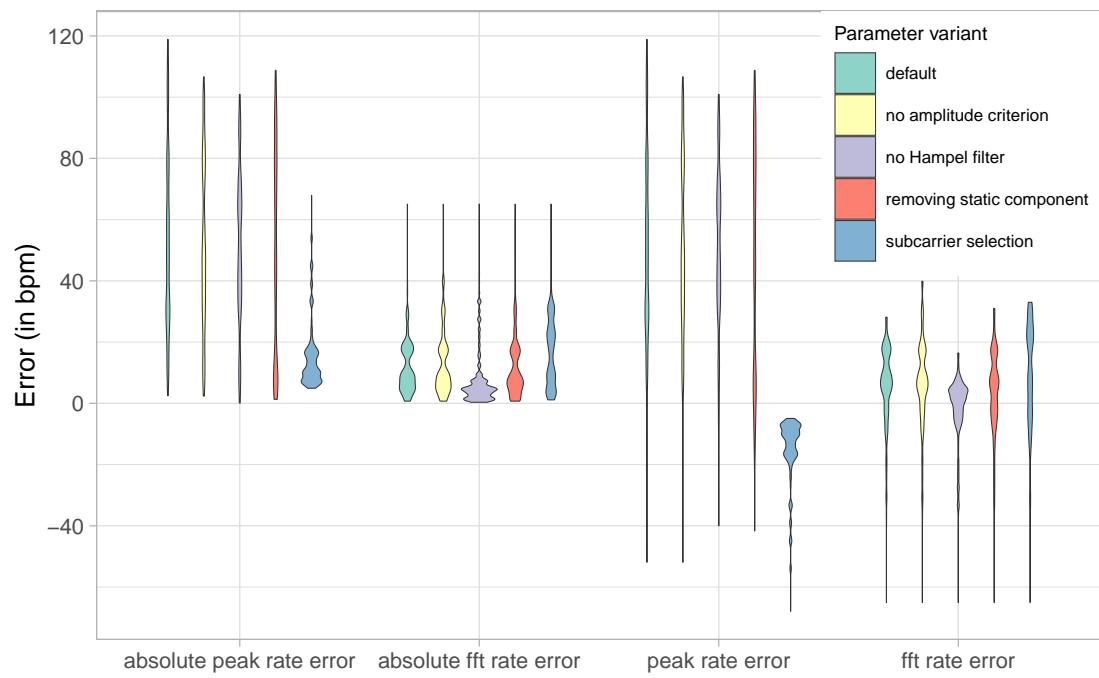


FIGURE 5.7: Breathing rate estimation errors for the different system setups across all conditions

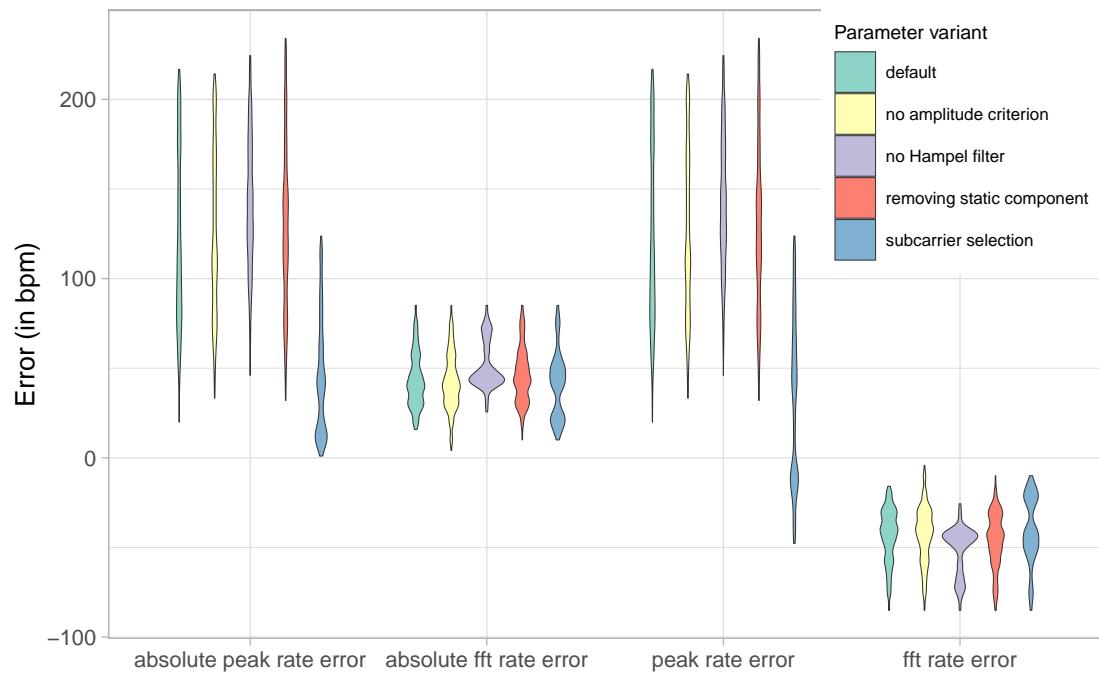


FIGURE 5.8: Heart rate estimation errors for the different system setups across all conditions

The peak rate errors for breathing, however, are similar for both variants. For the heart rate estimation, removing the Hampel filter leads to worse peak-based rate estimates, with the mean absolute error being around 20 bpm higher than the default, at 143.08 compared to 123.01 bpm, although even the latter error is very high. This setting also exhibits the worst FFT-based rate estimates of all tested parameter settings, with the FFT rate errors being more concentrated around a median of -46.28 bpm.

The performance of the system set to remove the static CSI component is similar to the default system for both breathing and heart rate estimation, with the rate errors being not very noticeably different.

Reanalyzing the data with the system configured to perform variance-based subcarrier selection instead of subcarrier fusion, seems to most change the performance with respect to breathing and heart rate estimation compared to all test the system configurations. As seen in figure 5.7, this parameter variant leads to peak rate errors being much more concentrated in the range from around -5 to -20 bpm, indicating that the peak-based breathing rate estimation tends to overestimate the true rate, with a median of -10.93 bpm compared to the default system, where this estimate tends to underestimate the true rate (median 48.87 bpm). The FFT rate error is not that different from the one for the default parameter variant, although performance is slightly worse with mean and median of 16.69 and 17.16 bpm compared to 11.65 and 9.44 bpm. For the heart rate estimations, the FFT-rate estimation performs similarly for this parameter variant as for the default one, but the peak rate estimation has much better performance, with errors being more distributed around zero, over- and underestimating the true rate while the peak-based rate estimation always underestimates the true rate for the other test the system variants. However, the absolute peak rate error is still quite large with a mean of 45.29, a median of 41.31, and a maximum of 123.79 bpm.

## 5.4 Effect of device location

Figures 5.9 and 5.10 show the breathing and heart rate estimation errors for the three receivers. As can be seen, there are differences between the three devices. Specifically, for the peak-based breathing rate estimation, the left receiver performs best, underestimating the true breathing rate less, with the median peak rate errors for the left, middle, and right device being 18.81, 41.53, and 48.87 bpm and the absolute peak rate errors being 19.89, 41.53, and 49.43 bpm respectively. The FFT rate errors are similar for the left and middle device, while the rightmost receiver performs worse, with the estimation errors being less concentrated around zero, tending to underestimate with a mean error of 5.24 and median error of 7.44 bpm compared to -2.62 and -3.38 bpm for the left and -0.35 and -1.21 bpm for the middle device.

For the heart rate estimations, the image is similar for the peak rate estimation error, with the left outperforming the middle outperforming the rightmost device with

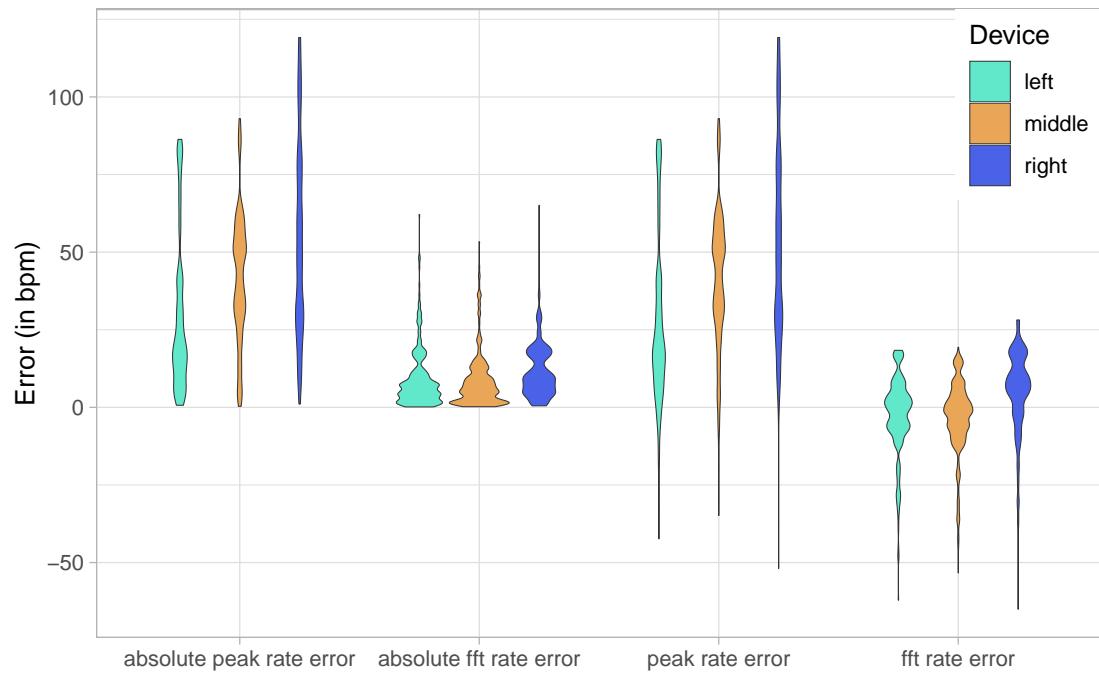


FIGURE 5.9: Breathing rate estimation errors for the different receivers across all conditions

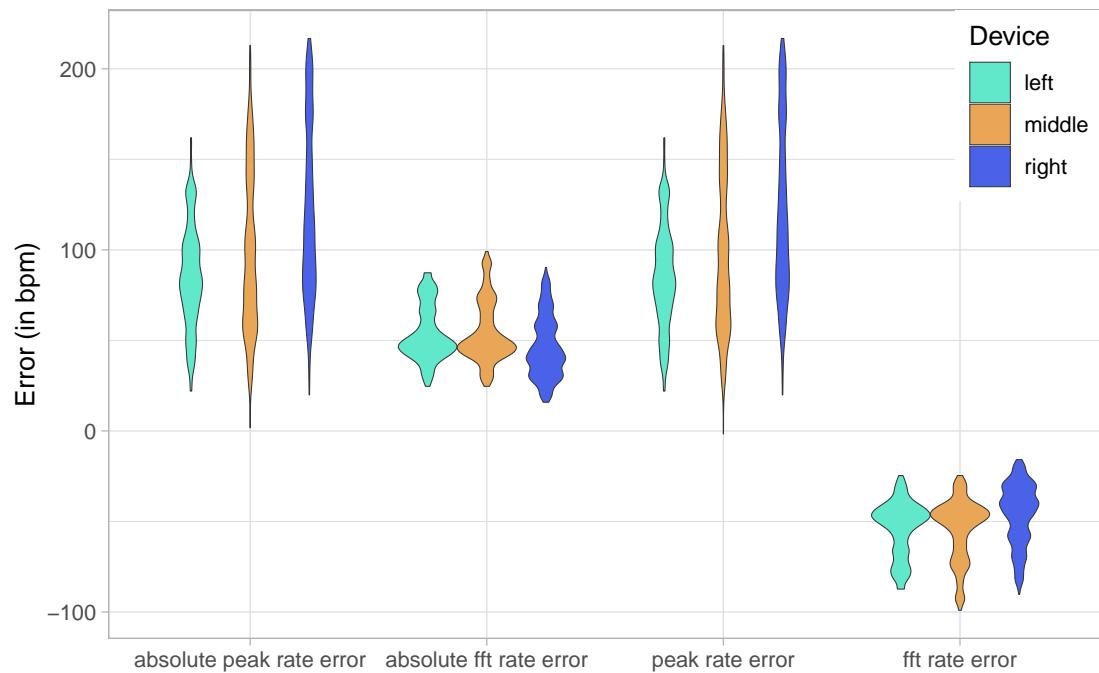


FIGURE 5.10: Heart rate estimation errors for the different receivers across all conditions

smaller estimation errors on average. As for the breathing rate estimation, the left and middle receivers have similar FFT rate error distributions, however, different from the breathing rate estimation, the right device exhibits a better performance for the heart rate estimation than the other two devices, with a median FFT error rate of -43.28 compared to -49.21 bpm.

# 6 Discussion

## 6.1 Processing influences CSI

The pattern of the spikes in the STI plots of the receivers, with spikes occurring in 120 second intervals for the left and middle receiver and for the right—reference—receiver, with spikes occurring at the beginning and end of a recording period and following spikes also occurring in 120 second intervals, indicates that the calculation of the MRC-PCA coefficients influences the collected CSI. As this calculation is computationally expensive, this finding suggests that heavy computations can affect simultaneously collected CSI, which would have important implications for further research using ESPs for CSI-based sensing. If heavy computational load does indeed affect CSI measurements as drastically as demonstrated in this experiment, this would place an additional restriction on the design of on-device sensing systems, potentially requiring them to monitor processor load in order to keep the load low enough for accurate CSI measurements. While it is unclear how processing can affect the collected CSI, it could be hypothesized that this effect is caused by high core load resulting from the computation-intensive calculations, which in turn could result in changes in the system’s power draw potentially impacting the Wi-Fi chip or causing electric interference and thus affecting the CSI. However, further work is required to identify the exact mechanism behind this phenomenon and quantify the impact this has on on-device sensing applications.

## 6.2 STI

One unanticipated finding was that the STI seems to, at least under certain circumstances, capture the breathing and even provide a more robust signal than the one given by subcarrier fusion and filtering. This property of the STI has, to the best of my knowledge, not been described before, and from the limited data collected for this experiment, it is unclear under which circumstances it exhibits this property. Although this pattern does not occur for all conditions in a single device, a sinusoidal curve that seems to correlate with the breathing of the subject is present for all devices over some intervals of the recording periods. Therefore it is possible that this property depends on the device placement and subject orientation, possibly occurring only for certain constellations of sender, receiver, and sleep posture of the target. This observation of this

phenomenon warrants further research into the properties of the STI, and whether it could be used for vital sign estimation.

## 6.3 Discussion of the results

The performance of the system's breathing and heart rate estimation as reported in the results section is a lot worse than expected based on existing literature (B. Yu et al. 2021, J. Liu et al. 2015, Gu et al. 2019) and initial experiments performed with a simpler system. Specifically, the estimation errors of this system are much higher than those reported in the literature, even compared to the worst of the results reported in the literature. However, these highly different results might be rooted in issues of the collected CSI rather than the processing. Specifically, the collected data has a much lower sampling rate than expected and, perhaps even more detrimental to the performance, heavy processing seems to cause spikes in the CSI amplitude disrupting the sensing signal and likely being the main cause of these performance issues. How drastically this system-created noise affects the sensing signals can be seen in figure 5.2, where the noise in the amplitude resulting from the calculation of the subcarrier fusion coefficients at the four-minute mark causes a large spike and following drop in the filtered breathing signal, with the signal being impacted for around a minute following the noise in the amplitude, due to the nature of the bandpass filter used.

### 6.3.1 Effect of posture

The results show that the performance of the breathing and heart rate estimation differs for the four tested sleep postures. This is in line with previous research (J. Liu et al. 2015, X. Liu et al. 2014, Gu et al. 2019, Atif et al. 2022) that has already established that different sleep postures impact performance. However, the results indicate that the system performs best for the postures left and prone, while—based on the results of previous research and the similar device placement—it was expected that the error would be lowest for the supine posture, followed by the side postures, with the prone posture being expected to result in the highest error. A possible explanation for this difference is that due to the proposed system not performing very well, resulting in large estimation errors, the results might not reflect the effect of posture well.

### 6.3.2 Effect of device placement

The results show that different device placement can affect performance, even for relatively small differences where the device is placed 40 cm from the first position. Interestingly, for the collected data, one placement can lead to improved heart rate estimation but decreased performance of the breathing rate estimation compared to the other placement options.

Environmental factors also influence the propagation of the Wi-Fi signal and therefore the CSI used for sensing, such that optimal or more desirable device placements depend on the environment in which the system is to be deployed. However, the effect of the device placement on the sensing performance might be exaggerated by the directionality of the onboard Wi-Fi antennas used, such that a system using devices with external antennas might be more robust to changes in device placement and orientation. The use of external antennas has previously been shown to improve sensing in terms of the accuracy of vital sign estimations (Atif et al. 2022).

### 6.3.3 Effect of blanket

In line with the hypothesis, the subject being covered with a blanket led to larger errors in the rate estimates compared with the non-blanket condition for the FFT-based rate estimation. However, contrary to expectations the opposite was found for the peak-based rate estimation. As the peak-based estimates are far from the true rates for both conditions, indicating that the system was not able to correctly detect the movement caused by the respiration and pulse and identify peaks to estimate the breathing and heart rate, this result might not actually reflect a truly positive effect.

The results show that model-based approaches, that rely on hand-crafted solutions based on known signal properties, taken as assumptions, fail if these assumptions do not hold, and might not always lead to the best results. Specifically, on visual inspection using the STI values instead of the amplitude might have led to a better breathing rate estimation due to seemingly capturing the breathing better and with less noise than the designed feature extraction. In such cases, a learning-based approach might have led to better results given enough data, as the most relevant features would be learned instead of painstakingly hand-crafted. However, learning-based approaches do require training, which in turn usually requires large amounts of training data, which has to be collected first, and they are often more computationally expensive. This means that implementing them in a system performing the monitoring on-device on such a resource-constrained device such as the ESP32 is a challenge. To date, most research on Wi-Fi-based vital sign and sleep monitoring has focused on model-based or hybrid approaches. For Wi-Fi sensing to become ubiquitous, further research into learning-based approaches is required, especially into less resource-intensive learning or hybrid approaches that would be feasible to run on IoT devices.

## 6.4 Problems with the data collection

As already mentioned, there was an issue with the data collection as far fewer CSI samples were collected than was expected. As the recorded data does not indicate that frames were dropped due to limitations in the processing, it can be assumed that a

portion of the Wi-Fi frames that were sent to generate the CSI were either not received by the receiver or not written to the SD card.

Specifically, for the collected data, only around 20 out of the 50 frames per second that were sent seem to have reached the receiver. This means that a large quantity of the frames was lost, likely due to interference. While the PSG device uses Bluetooth to transmit the data to the computer for processing and analysis, and Bluetooth uses the same 2.4 GHz frequency band as the ESP used for Wi-Fi, such a high packet loss is unusual due to interference from Bluetooth. This packet loss has consequences for the processing of the collected CSI. Importantly, the assumption was made that the receiver would receive 50 samples per second, and the band pass filters were calculated beforehand based on this assumption. If the sampling rate that was specified in the creation of the filters is not matched and is not even a constant rate, the filters will not work correctly. However, other parts of the processing might still work, although the assumed window size would then also not be correct.

Two possible causes for the unexpectedly low amount of recorded samples can be identified: packet loss, especially caused by interference, and failure to write samples to the SD card.

The main cause seems to be packet loss caused by interference from the PSG device. Importantly, the PSG device continuously transmits the collected data to the computer for live visualization and recording using Bluetooth, which operates within the same 2.4 GHz frequency band the Wi-Fi chip of the ESPs. This causes interference which can lead to packet loss. And while such a high percentage of packet loss due to Bluetooth interference is unusual, it is likely to account for a large part of the lost samples. This seems to also be indicated by the observation that the live visualization of the monitoring transmitted via UDP broadcasts received more samples during the setup of the ESPs when the PSG device was turned off compared to the data collection phase during which the PSG collected and transmitted data via Bluetooth. As mentioned, interference from Bluetooth and resulting packet loss alone is unlikely to lead to such a large percentage of lost samples, and a significantly lower amount of collected samples on the reference ESP compared to the other receivers indicates issues on the reference receiver when writing the samples to the SD card. This could be caused by a faulty SD card or defects of the development board.

## 6.5 Limitations & Further work

The system is designed for monitoring only one person, such that another person in the sensing area can interfere with the monitoring. Using CSI-based Wi-Fi sensing to monitor multiple people for fine-grained sensing tasks such as breathing and heart rate monitoring is a known problem, that is yet to be solved. While there have been attempts for multi-person respiration monitoring, these usually require multiple antenna

systems or assume that the individuals have different breathing rates (J. Liu et al. 2015, Gao et al. 2020). In these cases, attributing the estimated breathing rates to the individuals is another problem that would have to be solved. In the case of this system, the effect of interference by other people could potentially be reduced by better device placement following the sensing area model introduced by X. Wang et al. (2022), who showed that placing the sender at a distance of over ten meters away from the receiver improved breathing rate estimation when an interferer was present compared to placing the devices closer together.

The use of ESPs with a PCB antenna, which has some directionality, means that device placement and especially orientation can greatly affect sensing performance. This could be remedied by using an external antenna. While this might lead to a general improvement of the sensing performance (Atif et al. 2022), it also adds costs. However, given that the required components are available, the costs for one sender and receiver pair would still likely not exceed 100€, such that this would still be one of the most affordable Wi-Fi sensing systems.

The system only uses the amplitude of the CSI, discarding any phase information as this is assumed to be unstable and containing various types of noise. However, phase has been shown to be complementary to the amplitude, and using both can eliminate blind spots and lead to better respiration monitoring performance. Further work could therefore examine the phase information provided by the ESP32 and try to find a way to use it for sensing tasks. This could potentially be done by utilizing a NN for denoising and phase unwrapping or directly extracting features from the phase information.

Only very little data was collected for each condition and used to evaluate the system's performance, and the data came from only one subject. The results should therefore only be seen as a first indication of the performance that could guide improvements to both the system and experiment design for further, more extensive testing.

However, while the results for the breathing and heart rate estimations are not very promising, valuable insights can be gleaned from the collected data. For one thing, it was found that the on-device online processing on the ESP32-S3 can affect the collect CSI, in that computationally demanding tasks can cause noise. For another thing, the reference device, using Bluetooth for data transmission can cause interference resulting in high packet loss, impacting the performance of the system.

The unusually high packet loss due the Bluetooth interference could be due to the Bluetooth device being located directly between the sender and receiver and especially being fixed to the body of the subject such that the source of the interference is located where the movement to be measured takes place. This observation has important implications for further research, as it imposes constraints on the experimental setup used for evaluating such Wi-Fi sensing systems. Further experiments are warranted to quantify how the presence or absence of a reference device transmitting data via Bluetooth

as well as its placement and transmission rate affect Wi-Fi packet loss and sensing performance of a Wi-Fi sensing system such as the one proposed in this thesis.

## 6.6 Conclusion

While the experiments showed that the proposed system is not suitable to use for sleep monitoring due to the high errors of the breathing and especially the heart rate estimation, valuable insights could be gained that can guide further research for designing systems for online, on-device Wi-Fi sensing on ESP32 devices. Specifically, it was found that heavy processing affects the collected channel state information, and the signal tendency index seems to reflect the breathing signal and could potentially be used in future sensing applications.

## A Additional tables and plots

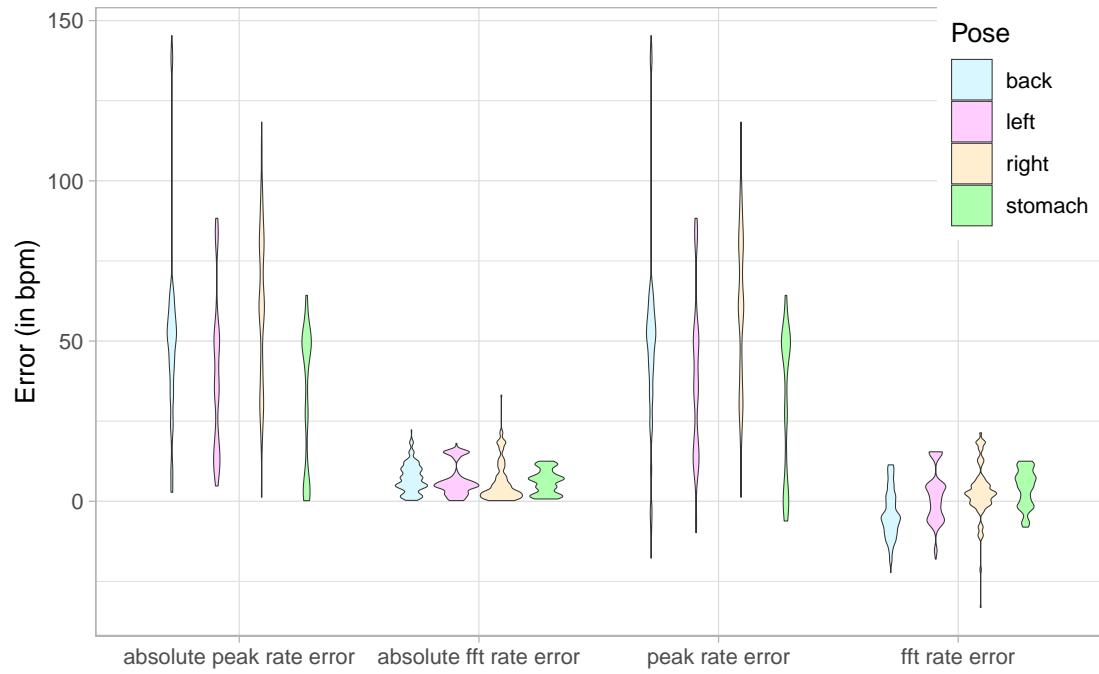


FIGURE A.1: Breathing rate estimation errors for the different poses during the open condition

TABLE A.1: Summary statistics of the different breathing rate errors for the different poses

error_type	pose	minimum	mean	median	maximum
abs_error	back	0.025185	40.5968108	39.614357	145.30673
abs_error	left	0.319481	26.3629003	16.792925	88.31953
abs_error	right	1.247812	56.7195227	58.246552	118.82622
abs_error	stomach	0.183388	26.3104260	21.823097	64.26710
abs_fft_error	back	0.281250	7.5352453	6.507812	31.07031
abs_fft_error	left	0.210938	5.5751129	4.718750	36.94531
abs_fft_error	right	0.210938	8.8215285	6.437500	65.07031
abs_fft_error	stomach	0.140625	7.4174305	7.070312	18.14062
error	back	-17.804763	39.7872566	39.614357	145.30673
error	left	-13.545647	25.8743639	16.792925	88.31953
error	right	-51.959333	56.2622441	58.246552	118.82622
error	stomach	-6.159328	25.5744919	21.823097	64.26710
fft_error	back	-31.070312	-4.6964823	-5.281250	25.08594
fft_error	left	-18.281250	0.4700521	-0.210938	36.94531
fft_error	right	-65.070312	4.4331164	4.648438	28.15625
fft_error	stomach	-18.140625	1.0829518	1.718750	12.50781

TABLE A.2: Summary statistics of the different heart rate errors for the different poses

error_type	pose	minimum	mean	median	maximum
abs_error	back	8.613029	121.76505	116.17191	280.221893
abs_error	left	0.000927	54.60472	36.96507	210.270111
abs_error	right	16.903816	116.20847	111.11568	216.759491
abs_error	stomach	0.127525	78.24278	85.42616	146.910980
abs_fft_error	back	17.914062	54.18650	51.28125	93.281250
abs_fft_error	left	6.054688	56.28289	55.56250	86.070312
abs_fft_error	right	15.843750	48.18139	45.63281	90.351562
abs_fft_error	stomach	33.492188	58.02410	56.21094	92.140625
error	back	8.613029	121.76505	116.17191	280.221893
error	left	-75.841234	42.08565	31.37151	210.270111
error	right	16.903816	116.20847	111.11568	216.759491
error	stomach	-84.891201	69.68509	85.42616	146.910980
fft_error	back	-93.281250	-54.18650	-51.28125	-17.914062
fft_error	left	-86.070312	-56.28289	-55.56250	-6.054688
fft_error	right	-90.351562	-48.18139	-45.63281	-15.843750
fft_error	stomach	-92.140625	-58.02410	-56.21094	-33.492188

TABLE A.3: Summary statistics of the different breathing rate errors for the different conditions

error_type	condition	minimum	mean	median	maximum
abs_error	blanket	0.025185	34.8949094	24.028870	118.82622
abs_error	open	0.183388	48.3716266	49.392700	145.30673
abs_fft_error	blanket	0.140625	8.8028497	7.210938	65.07031
abs_fft_error	open	0.210938	6.4446378	5.210938	33.14062
error	blanket	-51.959333	34.1589891	23.099537	118.82622
error	open	-17.804763	47.8759287	49.392700	145.30673
fft_error	blanket	-65.070312	-0.3529653	-1.281250	36.94531
fft_error	open	-33.140625	0.7197898	0.789062	21.36719

TABLE A.4: Summary statistics of the different heart rate errors for the different conditions

error_type	condition	minimum	mean	median	maximum
abs_error	blanket	0.000927	101.72967	93.02611	226.406830
abs_error	open	0.259094	101.22030	99.37256	280.221893
abs_fft_error	blanket	6.054688	50.81001	49.35156	92.140625
abs_fft_error	open	27.632812	54.80423	53.14062	93.281250
error	blanket	-84.891201	94.80046	93.02611	226.406830
error	open	-31.842152	100.95656	99.37256	280.221893
fft_error	blanket	-92.140625	-50.81001	-49.35156	-6.054688
fft_error	open	-93.281250	-54.80423	-53.14062	-27.632812

TABLE A.5: Summary statistics of the different breathing rate errors for the different poses during the open condition

error_type	pose	minimum	mean	median	maximum
abs_error	back	2.804763	49.479532	50.614357	145.30673
abs_error	left	4.772370	38.474826	34.903690	88.31953
abs_error	right	1.247812	60.226013	61.782574	118.37210
abs_error	stomach	0.183388	31.827883	40.826336	64.26710
abs_fft_error	back	0.281250	7.101256	7.281250	22.28125
abs_fft_error	left	0.210938	6.541818	4.718750	18.07031
abs_fft_error	right	0.210938	5.883573	3.140625	33.14062
abs_fft_error	stomach	0.718750	6.124674	6.578125	12.50781
error	back	-17.804763	48.678672	50.614357	145.30673
error	left	-9.772370	37.919327	34.903690	88.31953
error	right	1.247812	60.226013	61.782574	118.37210
error	stomach	-6.159328	30.881226	40.826336	64.26710
fft_error	back	-22.281250	-3.792971	-5.210938	11.36719
fft_error	left	-18.070312	1.559857	1.718750	15.43750
fft_error	right	-33.140625	3.098466	2.648438	21.36719
fft_error	stomach	-8.070312	4.141674	4.648438	12.50781

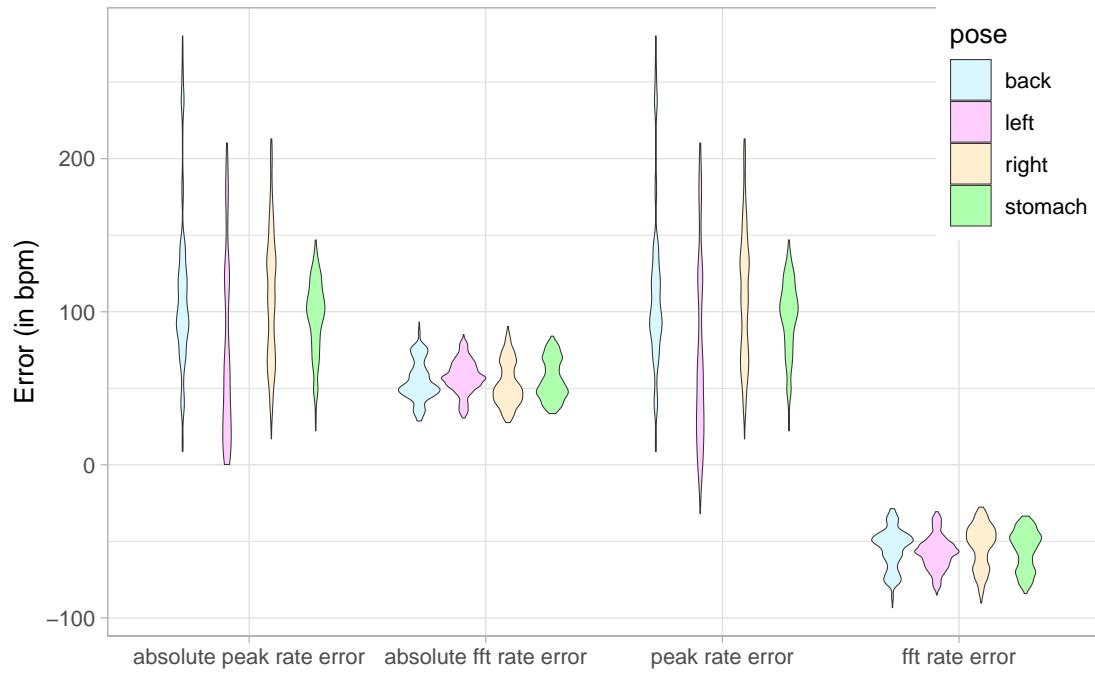


FIGURE A.2: Heart rate estimation errors for the different poses during the open condition

TABLE A.6: Summary statistics of the different heart rate errors for the different poses during the open condition

error_type	pose	minimum	mean	median	maximum
abs_error	back	8.613029	114.17738	101.92203	280.22189
abs_error	left	0.259094	73.96274	59.86017	210.27011
abs_error	right	16.903816	109.15174	110.31357	212.92545
abs_error	stomach	22.141518	96.77378	100.50922	146.91098
abs_fft_error	back	28.632812	54.53966	51.28125	93.28125
abs_fft_error	left	30.562500	57.29156	57.21094	85.07031
abs_fft_error	right	27.632812	53.06845	50.14062	90.35156
abs_fft_error	stomach	33.492188	55.61412	53.21094	84.07031
error	back	8.613029	114.17738	101.92203	280.22189
error	left	-31.842152	72.79584	59.86017	210.27011
error	right	16.903816	109.15174	110.31357	212.92545
error	stomach	22.141518	96.77378	100.50922	146.91098
fft_error	back	-93.281250	-54.53966	-51.28125	-28.63281
fft_error	left	-85.070312	-57.29156	-57.21094	-30.56250
fft_error	right	-90.351562	-53.06845	-50.14062	-27.63281
fft_error	stomach	-84.070312	-55.61412	-53.21094	-33.49219

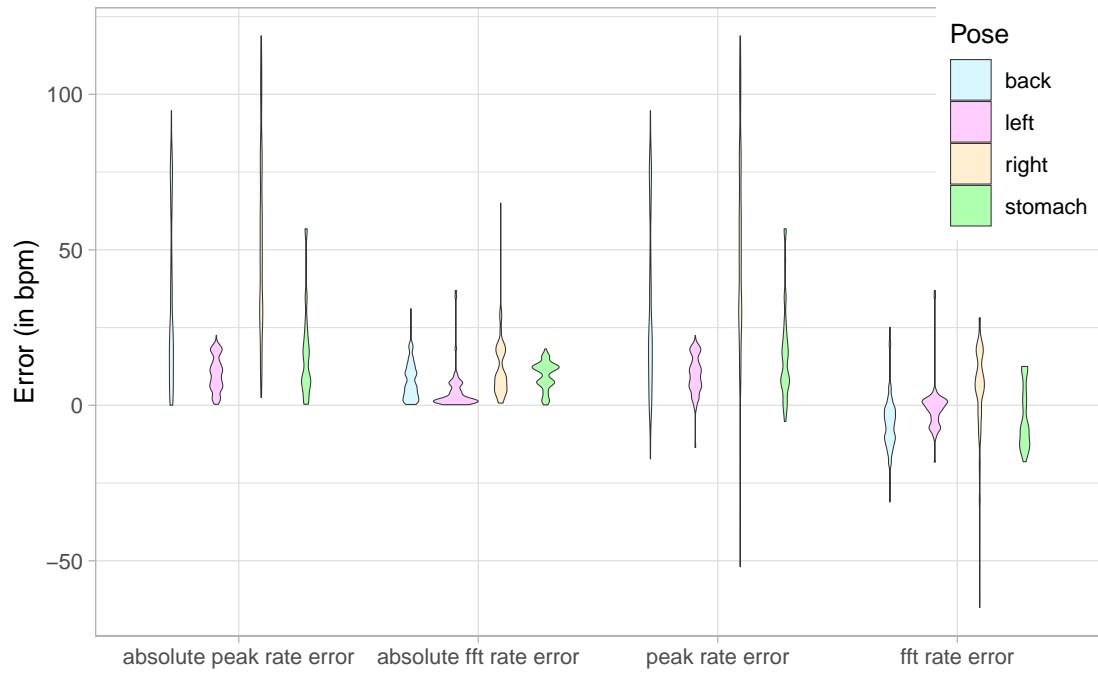


FIGURE A.3: Breathing rate estimation errors for the different poses during the blanket condition

TABLE A.7: Summary statistics of the different breathing rate errors for the different poses during the blanket condition

error_type	pose	minimum	mean	median	maximum
abs_error	back	0.025185	32.2012711	22.689022	94.77341
abs_error	left	0.319481	10.1640182	10.346684	22.48487
abs_error	right	2.483006	53.3414102	48.869728	118.82622
abs_error	stomach	0.345117	17.0369249	14.528465	56.77664
abs_fft_error	back	0.281250	7.9454318	6.351562	31.07031
abs_fft_error	left	0.210938	4.2822105	1.789062	36.94531
abs_fft_error	right	0.718750	11.6519204	9.437500	65.07031
abs_fft_error	stomach	0.140625	9.5902389	11.507812	18.14062
error	back	-17.259880	31.3834992	22.689022	94.77341
error	left	-13.545647	9.7650405	9.893929	22.48487
error	right	-51.959333	52.4435947	48.869728	118.82622
error	stomach	-5.199214	16.6551645	14.528465	56.77664
fft_error	back	-31.070312	-5.5504398	-6.210938	25.08594
fft_error	left	-18.281250	-0.9874878	-0.281250	36.94531
fft_error	right	-65.070312	5.7189035	7.437500	28.15625
fft_error	stomach	-18.140625	-4.0580159	-7.070312	12.50781

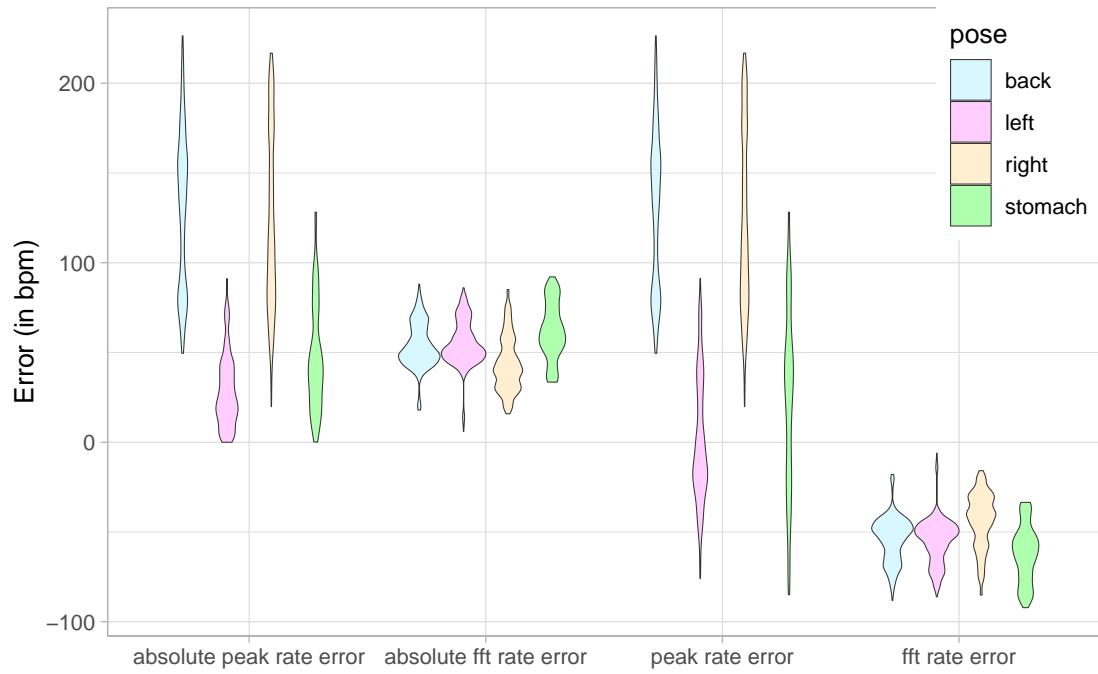


FIGURE A.4: Heart rate estimation errors for the different poses during the blanket condition

TABLE A.8: Summary statistics of the different heart rate errors for the different poses during the blanket condition

error_type	pose	minimum	mean	median	maximum
abs_error	back	49.555786	128.936575	135.40649	226.406830
abs_error	left	0.000927	28.714678	25.31783	91.182098
abs_error	right	19.853065	123.006845	113.66861	216.759491
abs_error	stomach	0.127525	47.096685	41.22869	128.204559
abs_fft_error	back	17.914062	53.852719	51.28125	88.140625
abs_fft_error	left	6.054688	54.933872	52.21094	86.070312
abs_fft_error	right	15.843750	43.473248	41.28125	85.070312
abs_fft_error	stomach	33.492188	62.074701	60.49219	92.140625
error	back	49.555786	128.936575	135.40649	226.406830
error	left	-75.841234	1.012843	-7.27821	91.182098
error	right	19.853065	123.006845	113.66861	216.759491
error	stomach	-84.891201	24.155587	30.26141	128.204559
fft_error	back	-88.140625	-53.852719	-51.28125	-17.914062
fft_error	left	-86.070312	-54.933872	-52.21094	-6.054688
fft_error	right	-85.070312	-43.473248	-41.28125	-15.843750
fft_error	stomach	-92.140625	-62.074701	-60.49219	-33.492188

TABLE A.9: Summary statistics of the different breathing rate errors for the different system settings

error_type	filename	minimum	mean	median	maximum
abs_error	CSI	2.483006	53.341410	48.869728	118.826218
abs_error	Sc_selection	4.925067	15.121286	10.925067	67.925067
abs_error	no_amp_criterion	2.380226	51.224188	48.869720	106.655228
abs_error	no_hampel	0.095029	51.070458	50.364471	100.888367
abs_error	removing_static	1.301270	47.773110	45.770401	108.728561
abs_fft_error	CSI	0.718750	11.651920	9.437500	65.070312
abs_fft_error	Sc_selection	1.140625	16.688171	17.156250	65.070312
abs_fft_error	no_amp_criterion	0.718750	12.695425	9.437500	65.070312
abs_fft_error	no_hampel	0.351562	7.321419	4.718750	65.070312
abs_fft_error	removing_static	0.718750	11.555600	8.070312	65.070312
error	CSI	-51.959333	52.443595	48.869728	118.826218
error	Sc_selection	-67.925067	-15.121286	-10.925067	-4.925067
error	no_amp_criterion	-51.959348	50.086445	48.869720	106.655228
error	no_hampel	-40.051981	50.060603	50.364471	100.888367
error	removing_static	-41.698730	44.710998	45.770401	108.728561
fft_error	CSI	-65.070312	5.718903	7.437500	28.156250
fft_error	Sc_selection	-65.070312	8.477535	9.648438	33.015625
fft_error	no_amp_criterion	-65.070312	6.437480	6.648438	39.875000
fft_error	no_hampel	-65.070312	-2.133556	0.789062	16.437500
fft_error	removing_static	-65.070312	3.041594	4.718750	31.015625

TABLE A.10: Summary statistics of the different heart rate errors for the different system settings

error_type	filename	minimum	mean	median	maximum
abs_error	CSI	19.853065	123.00685	113.66861	216.759491
abs_error	Sc_selection	0.988377	45.29394	41.31373	123.793701
abs_error	no_amp_criterion	33.254372	124.99917	117.06602	214.181000
abs_error	no_hampel	45.935921	143.07597	142.11806	224.450378
abs_error	removing_static	32.022285	129.48311	125.31569	233.974243
abs_fft_error	CSI	15.843750	43.47325	41.28125	85.070312
abs_fft_error	Sc_selection	9.984375	40.71462	42.35156	85.070312
abs_fft_error	no_amp_criterion	4.125000	42.75467	40.49219	85.070312
abs_fft_error	no_hampel	25.632812	51.32571	46.28125	85.070312
abs_fft_error	removing_static	9.984375	46.15056	44.28125	85.070312
error	CSI	19.853065	123.00685	113.66861	216.759491
error	Sc_selection	-47.781180	31.03287	38.01041	123.793701
error	no_amp_criterion	33.254372	124.99917	117.06602	214.181000
error	no_hampel	45.935921	143.07597	142.11806	224.450378
error	removing_static	32.022285	129.48311	125.31569	233.974243
fft_error	CSI	-85.070312	-43.47325	-41.28125	-15.843750
fft_error	Sc_selection	-85.070312	-40.71462	-42.35156	-9.984375
fft_error	no_amp_criterion	-85.070312	-42.75467	-40.49219	-4.125000
fft_error	no_hampel	-85.070312	-51.32571	-46.28125	-25.632812
fft_error	removing_static	-85.070312	-46.15056	-44.28125	-9.984375

TABLE A.11: Summary statistics of the different breathing rate errors for the different devices

error_type	device	minimum	mean	median	maximum
abs_error	left	0.672590	29.790039	19.894057	86.33158
abs_error	middle	0.324673	41.674599	41.531334	93.00812
abs_error	right	1.065002	54.575001	49.425102	119.14980
abs_fft_error	left	0.140625	9.063156	6.437500	62.14062
abs_fft_error	middle	0.210938	8.536895	6.210938	53.35156
abs_fft_error	right	0.492188	11.521586	9.492188	65.07031
error	left	-42.370885	27.583453	18.811367	86.33158
error	middle	-34.837029	40.797035	41.531334	93.00812
error	right	-51.959333	53.746203	48.869728	119.14980
fft_error	left	-62.140625	-2.621684	-0.351562	18.36719
fft_error	middle	-53.351562	-3.382134	-1.210938	19.36719
fft_error	right	-65.070312	5.246573	7.437500	28.15625

TABLE A.12: Summary statistics of the different heart rate errors for the different devices

error_type	device	minimum	mean	median	maximum
abs_error	left	21.997787	85.58625	84.11639	161.91510
abs_error	middle	1.687912	98.56283	91.70428	212.87470
abs_error	right	19.853065	121.37736	112.30205	216.75949
abs_fft_error	left	24.632812	53.58102	49.21094	87.35156
abs_fft_error	middle	24.632812	54.25804	49.21094	99.07031
abs_fft_error	right	15.843750	46.02652	43.28125	90.28125
error	left	21.997787	85.58625	84.11639	161.91510
error	middle	-1.687912	98.55845	91.70428	212.87470
error	right	19.853065	121.37736	112.30205	216.75949
fft_error	left	-87.351562	-53.58102	-49.21094	-24.63281
fft_error	middle	-99.070312	-54.25804	-49.21094	-24.63281
fft_error	right	-90.281250	-46.02652	-43.28125	-15.84375

# Bibliography

- Abedi, Ali, and Omid Abari. 2020. "WiFi Says "Hi!" Back to Strangers!" In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks*, 132–138. HotNets '20. <https://doi.org/10.1145/3422604.3425951>.
- Ameen, Mohamed S., Lok Man Cheung, Theresa Hauser, Michael A. Hahn, and Manuel Schabus. 2019. "About the Accuracy and Problems of Consumer Devices in the Assessment of Sleep." *Sensors* 19 (19). <https://doi.org/10.3390/s19194160>.
- Atif, Muhammad, Shapna Muralidharan, Heedong Ko, and Byounghyun Yoo. 2020. "Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS)." *Journal of Computational Design and Engineering* 7, no. 5 (May): 644–656. <https://doi.org/10.1093/jcde/qwaa048>.
- . 2022. "COVID-Beat: a low-cost breath monitoring approach for people in quarantine during the pandemic." *Journal of Computational Design and Engineering* 9, no. 3 (May): 992–1006. <https://doi.org/10.1093/jcde/qwac037>.
- Cao, Yangjie, Fuchao Wang, Xinxin Lu, Nan Lin, Bo Zhang, Zhi Liu, and Stephan Sigg. 2020. "Contactless Body Movement Recognition During Sleep via WiFi Signals." *IEEE Internet of Things Journal* 7 (3): 2028–2037. <https://doi.org/10.1109/JIOT.2019.2960823>.
- Cay, Gozde, Vignesh Ravichandran, Shehjar Sadhu, Alyssa H. Zisk, Amy L. Salisbury, Dhaval Solanki, and Kunal Mankodiya. 2022. "Recent Advancement in Sleep Technologies: A Literature Review on Clinical Standards, Sensors, Apps, and AI Methods." *IEEE Access* 10:104737–104756. <https://doi.org/10.1109/ACCESS.2022.3210518>.
- Chen, Cheng, Hao Song, Qinghua Li, Francesca Meneghelli, Francesco Restuccia, and Carlos Cordeiro. 2023. "Wi-Fi Sensing Based on IEEE 802.11bf." *IEEE Communications Magazine* 61 (1): 121–127. <https://doi.org/10.1109/MCOM.007.2200347>.
- Chen, Zhenghua, Le Zhang, Chaoyang Jiang, Zhiguang Cao, and Wei Cui. 2019. "WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM." *IEEE Transactions on Mobile Computing* 18 (11): 2714–2724. <https://doi.org/10.1109/TMC.2018.2878233>.

- Chinoy, Evan D, Joseph A Cuellar, Jason T Jameson, and Rachel R Markwald. 2022. "Performance of Four Commercial Wearable Sleep-Tracking Devices Tested Under Unrestricted Conditions at Home in Healthy Young Adults." *Nature and Science of Sleep* 14:493–516. <https://doi.org/10.2147/NSS.S348795>.
- Choi, Hyuckjin, Manato Fujimoto, Tomokazu Matsui, Shinya Misaki, and Keiichi Yamamoto. 2022. "Wi-CaL: WiFi Sensing and Machine Learning Based Device-Free Crowd Counting and Localization." *IEEE Access* 10 (March): 24395–24410. <https://doi.org/10.1109/access.2022.3155812>.
- Deng, Fei, Jianwu Dong, Xiangyu Wang, Ying Fang, Yu Liu, Zhaofei Yu, Jing Liu, and Feng Chen. 2018. "Design and Implementation of a Noncontact Sleep Monitoring System Using Infrared Cameras and Motion Sensor." *IEEE Transactions on Instrumentation and Measurement* 67, no. 7 (July): 1555–1563. <https://doi.org/10.1109/tim.2017.2779358>.
- Espressif. 2022. *TensorFlow Lite Micro for Espressif Chipsets - Performance comparison*. <https://web.archive.org/web/20230313130657/https://github.com/espressif/tflite-micro-esp-examples#performance-comparison>. Accessed: 2023-09-13.
- Falie, D., and M. Ichim. 2010. "Sleep monitoring and sleep apnea event detection using a 3D camera." *2010 8th International Conference on Communications*, <https://doi.org/10.1109/iccomm.2010.5509015>.
- Fang, Yu, Dongbo Liu, Zhongwei Jiang, and Haibin Wang. 2023. "Monitoring of sleep breathing states based on audio sensor utilizing Mel-scale features in Home Healthcare." *Journal of Healthcare Engineering* 2023. <https://doi.org/10.1155/2023/6197564>.
- Gao, Qinghua, Jingyu Tong, Jie Wang, Zhouhua Ran, and Miao Pan. 2020. "Device-Free Multi-Person Respiration Monitoring Using WiFi." *IEEE Transactions on Vehicular Technology* 69 (11): 14083–14087. <https://doi.org/10.1109/TVT.2020.3020180>.
- Gastel, Mark van, Sander Stuijk, Sebastiaan Overeem, Johannes P. van Dijk, Merel M. van Gilst, and Gerard de Haan. 2021. "Camera-Based Vital Signs Monitoring During Sleep – A Proof of Concept Study." *IEEE Journal of Biomedical and Health Informatics* 25, no. 5 (May): 1409–1418. <https://doi.org/10.1109/jbhi.2020.3045859>.
- Ge, Yao, Ahmad Taha, Syed Aziz Shah, Kia Dashtipour, Shuyuan Zhu, Jonathan Cooper, Qammer H. Abbasi, and Muhammad Ali Imran. 2023. "Contactless WiFi Sensing and Monitoring for Future Healthcare - Emerging Trends, Challenges, and Opportunities." *IEEE Reviews in Biomedical Engineering* 16:171–191. <https://doi.org/10.1109/RBME.2022.3156810>.

- Gringoli, Francesco, Marco Cominelli, Alejandro Blanco, and Joerg Widmer. 2021. “AX-CSI: Enabling CSI Extraction on Commercial 802.11ax Wi-Fi Platforms.” In *Proceedings of the 15th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & CCharacterization*, 46–53. WiNTECH’21. <https://doi.org/10.1145/3477086.3480833>.
- Gringoli, Francesco, Matthias Schulz, Jakob Link, and Matthias Hollick. 2019. “Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets.” In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 21–28. WiNTECH ’19. <https://doi.org/10.1145/3349623.3355477>.
- Gu, Yu, Xiang Zhang, Zhi Liu, and Fuji Ren. 2019. “WiFi-Based Real-Time Breathing and Heart Rate Monitoring during Sleep.” *2019 IEEE Global Communications Conference (GLOBECOM)* (December). <https://doi.org/10.1109/globecom38437.2019.9014297>.
- Gu, Yu, Yifan Zhang, Jie Li, Yusheng Ji, Xin An, and Fuji Ren. 2020. “Sleepy: Wireless Channel Data Driven Sleep Monitoring via Commodity WiFi Devices.” *IEEE Transactions on Big Data* 6 (2): 258–268. <https://doi.org/10.1109/TB DATA.2018.2851201>.
- Halperin, Daniel, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. “Tool Release: Gathering 802.11n Traces with Channel State Information.” *ACM SIGCOMM CCR* 41, no. 1 (January): 53.
- Hernandez, Steven M., and Eyuphan Bulut. 2020. “Lightweight and Standalone IoT Based WiFi Sensing for Active Repositioning and Mobility.” In *21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)* (WoWMoM 2020). June.
- . 2022a. “Online Stream Sampling for Low-Memory On-Device Edge Training for WiFi Sensing.” In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, 9–14. May. <https://doi.org/10.1145/3522783.3529521>.
- . 2022b. “WiFederated: Scalable WiFi Sensing Using Edge-Based Federated Learning.” *IEEE Internet of Things Journal* 9, no. 14 (July): 12628–12640. <https://doi.org/10.1109/JIOT.2021.3137793>.
- . 2023. “WiFi Sensing on the Edge: Signal Processing Techniques and Challenges for Real-World Systems.” *IEEE Communications Surveys & Tutorials* 25 (1): 46–76. <https://doi.org/10.1109/COMST.2022.3209144>.

- Hernandez, Steven M., Deniz Erdag, and Eyuphan Bulut. 2021. "Towards Dense and Scalable Soil Sensing Through Low-Cost WiFi Sensing Networks." In *2021 IEEE 46th Conference on Local Computer Networks (LCN)*, 549–556. <https://doi.org/10.1109/LCN52139.2021.9525003>.
- Hernandez, Steven M., Md Touhiduzzaman, Peter E. Pidcoe, and Eyuphan Bulut. 2022. "Wi-PT: Wireless Sensing based Low-cost Physical Rehabilitation Tracking." In *2022 IEEE International Conference on E-health Networking, Application & Services (HealthCom)*, 113–118. <https://doi.org/10.1109/HealthCom54947.2022.9982743>.
- Hsu, Chen-Yu, Aayush Ahuja, Shichao Yue, Rumen Hristov, Zachary Kabelac, and Dina Katabi. 2017. "Zero-Effort In-Home Sleep and Insomnia Monitoring Using Radio Signals." *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, no. 3 (September). <https://doi.org/10.1145/3130924>.
- Hu, Jiaxing, Jianfei Yang, Jenn-Bing Ong, Dazhuo Wang, and Lihua Xie. 2022. "ResFi: WiFi-Enabled Device-Free Respiration Detection Based on Deep Learning." In *2022 IEEE 17th International Conference on Control & Automation (ICCA)*, 510–515. <https://doi.org/10.1109/ICCA54724.2022.9831898>.
- IEEE. 2021. "IEEE Standard for Information Technology—Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks—Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)*, 1–4379. <https://doi.org/10.1109/IEEESTD.2021.9363693>.
- Jiang, Zhiping, Tom H. Luan, Xincheng Ren, Dongtao Lv, Han Hao, Jing Wang, Kun Zhao, Wei Xi, Yueshen Xu, and Rui Li. 2022. "Eliminating the Barriers: Demystifying Wi-Fi Baseband Design and Introducing the PicoScenes Wi-Fi Sensing Platform." *IEEE Internet of Things Journal* 9 (6): 4476–4496. <https://doi.org/10.1109/JIOT.2021.3104666>.
- Kabir, M. Humayun, Md. Ali Hasan, and Wonjae Shin. 2022. "CSI-DeepNet: A Lightweight Deep Convolutional Neural Network Based Hand Gesture Recognition System Using Wi-Fi CSI Signal." *IEEE Access* 10 (October): 114787–114801. <https://doi.org/10.1109/access.2022.3217910>.
- Khamis, Abdelwahed, Chun Tung Chou, Branislav Kusy, and Wen Hu. 2018. "CardioFi: Enabling Heart Rate Monitoring on Unmodified COTS WiFi Devices." In *Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 97–106. MobiQuitous '18. <https://doi.org/10.1145/3286978.3287003>.

- Korany, Belal, and Yasamin Mostofi. 2022. “Nocturnal Seizure Detection Using Off-the-Shelf WiFi.” *IEEE Internet of Things Journal* 9, no. 9 (May): 6996–7008. <https://doi.org/10.1109/jiot.2021.3115505>.
- Lao, Xiang Qian, Xudong Liu, Han-Bing Deng, Ta-Chien Chan, Kin Fai Ho, Feng Wang, Roel Vermeulen, et al. 2018. “Sleep Quality, Sleep Duration, and the Risk of Coronary Heart Disease: A Prospective Cohort Study With 60,586 Adults.” *Journal of Clinical Sleep Medicine* 14 (01): 109–117. <https://doi.org/10.5664/jcsm.6894>.
- Lauteslager, Timo, Stylianos Kampakis, Adrian J. Williams, Michal Maslik, and Fares Siddiqui. 2020. “Performance Evaluation of the Circadia Contactless Breathing Monitor and Sleep Analysis Algorithm for Sleep Stage Classification.” *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, <https://doi.org/10.1109/embc44109.2020.9175419>.
- Li, Fan, Cheng Xu, Yang Liu, Yun Zhang, Zhuo Li, Kashif Sharif, and Yu Wang. 2016. “Mo-sleep: Unobtrusive sleep and movement monitoring via Wi-Fi signal.” *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)* (December). <https://doi.org/10.1109/pccc.2016.7820634>.
- Liu, Jian, Yingying Chen, Yan Wang, Xu Chen, Jerry Cheng, and Jie Yang. 2018. “Monitoring Vital Signs and Postures During Sleep Using WiFi Signals.” *IEEE Internet of Things Journal* 5, no. 3 (June): 2071–2084. <https://doi.org/10.1109/jiot.2018.2822818>.
- Liu, Jian, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. 2015. “Tracking Vital Signs During Sleep Leveraging Off-the-Shelf WiFi.” In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 267–276. MobiHoc ’15. <https://doi.org/10.1145/2746285.2746303>.
- Liu, Xuefeng, Jiannong Cao, Shaojie Tang, and Jiaqi Wen. 2014. “Wi-Sleep: Contactless Sleep Monitoring via WiFi Signals.” *2014 IEEE Real-Time Systems Symposium* (May). <https://doi.org/10.1109/rtss.2014.30>.
- Liu, Xuefeng, Jiannong Cao, Shaojie Tang, Jiaqi Wen, and Peng Guo. 2016. “Contactless Respiration Monitoring Via Off-the-Shelf WiFi Devices.” *IEEE Transactions on Mobile Computing* 15, no. 10 (October): 2466–2479. <https://doi.org/10.1109/tmc.2015.2504935>.
- Lu, Wei, Michelle M. Nystrom, Parag J. Parikh, David R. Fooshee, James P. Hubenschmidt, Jeffrey D. Bradley, and Daniel A. Low. 2006. “A semi-automatic method for peak and valley detection in free-breathing respiratory waveforms.” *Medical Physics* 33 (10): 3634–3636. <https://doi.org/10.1118/1.2348764>.

- Meulen, Fokke B. van, Angela Grassi, Leonie van den Heuvel, Sebastiaan Overeem, Merel M. van Gilst, Johannes P. van Dijk, Henning Maass, Mark J. H. van Gastel, and Pedro Fonseca. 2023. "Contactless Camera-Based Sleep Staging: The HealthBed Study." *Bioengineering* 10 (1). <https://doi.org/10.3390/bioengineering10010109>.
- Mott, Nathaniel. 2019. "Linksys Aware uses mesh Wi-Fi to detect home invaders." Accessed: 2023-09-14 (October). [https://www.tomshardware.com/news/linksys-aware-mesh-wi-fi-security\\_40604.html](https://www.tomshardware.com/news/linksys-aware-mesh-wi-fi-security_40604.html).
- Nandakumar, Rajalakshmi, Shyamnath Gollakota, and Nathaniel Watson. 2015. "Contactless Sleep Apnea Detection on Smartphones." In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, 45–57. MobiSys '15. <https://doi.org/10.1145/2742647.2742674>.
- Nelson, Kathy L., Jean E. Davis, and Cynthia F. Corbett. 2022. "Sleep quality: An evolutionary concept analysis." *Nursing Forum* 57 (1): 144–151. <https://doi.org/10.1111/nuf.12659>.
- Palipana, Sameera, David Rojas, Piyush Agrawal, and Dirk Pesch. 2018. "FallDeFi: Ubiquitous Fall Detection Using Commodity Wi-Fi Devices." *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, no. 4 (January). <https://doi.org/10.1145/3161183>.
- Qian, Kun, Chenshu Wu, Zheng Yang, Yunhao Liu, and Zimu Zhou. 2014. "PADS: Passive detection of moving targets with dynamic speed using PHY layer information." In *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 1–8. <https://doi.org/10.1109/PADSW.2014.7097784>.
- Ren, Yanzhi, Chen Wang, Jie Yang, and Yingying Chen. 2015. "Fine-grained sleep monitoring: Hearing your breathing with smartphones." *2015 IEEE Conference on Computer Communications (INFOCOM)*, <https://doi.org/10.1109/infocom.2015.7218494>.
- Restuccia, Francesco. 2021. "IEEE 802.11bf: Toward Ubiquitous Wi-Fi Sensing." *CoRR* abs/2103.14918. arXiv: 2103.14918. <https://arxiv.org/abs/2103.14918>.
- Scott, Alexander J., Thomas L. Webb, Marrissa Martyn-St James, Georgina Rowse, and Scott Weich. 2021. "Improving sleep quality leads to better mental health: A meta-analysis of randomised controlled trials." *Sleep Medicine Reviews* 60:101556. <https://doi.org/10.1016/j.smrv.2021.101556>.

- Shang, Jiacheng, and Jie Wu. 2016. "Fine-Grained Vital Signs Estimation Using Commercial Wi-Fi Devices." In *Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop*, 30–32. S3. <https://doi.org/10.1145/2987354.2987360>.
- Sharma, Pragya, Xiaonan Hui, Jianlin Zhou, Thomas B. Conroy, and Edwin C. Kan. 2020. "Wearable radio-frequency sensing of respiratory rate, respiratory volume, and heart rate." *npj Digital Medicine* 3 (July). <https://doi.org/10.1038/s41746-020-0307-6>.
- Shirakami, Itsuki, and Takashi Sato. 2021. "Heart Rate Variability Extraction using Commodity Wi-Fi Devices via Time Domain Signal Processing." In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, 1–4. <https://doi.org/10.1109/BHI50953.2021.9508523>.
- Soltanaghaei, Elahe, Rahul Anand Sharma, Zehao Wang, Adarsh Chittilappilly, Anh Luong, Eric Giler, Katie Hall, Steve Elias, and Anthony Rowe. 2020. "Robust and Practical WiFi Human Sensing Using On-Device Learning with a Domain Adaptive Model." In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, 150–159. November. <https://doi.org/10.1145/3408308.3427983>.
- Song, Hang, Bo Wei, Qun Yu, Xia Xiao, and Takamaro Kikkawa. 2020. "WiEps: Measurement of Dielectric Property With Commodity WiFi Device—An Application to Ethanol/Water Mixture." *IEEE Internet of Things Journal* 7 (12): 11667–11677. <https://doi.org/10.1109/JIOT.2020.2999210>.
- Soto, Julio C.H., Iandra Galdino, Egberto Caballero, Vinicius Ferreira, Débora Muchaluat-Saade, and Célio Albuquerque. 2022. "A survey on vital signs monitoring based on Wi-Fi CSI data." *Computer Communications* 195:99–110. <https://doi.org/https://doi.org/10.1016/j.comcom.2022.08.004>.
- The Beacon. 2023. "Wi-Fi® by the numbers: Technology momentum in 2023," May 15, 2023. Accessed July 8, 2023. <https://www.wi-fi.org/beacon/the-beacon/wi-fi-by-the-numbers-technology-momentum-in-2023>.
- Tsubota, Kazuya, Takashi Ohhira, and Hideki Hashimoto. 2023. "Heart Rate Variability and Body-Movement Extractions using Wi-Fi Channel State Information." In *2023 IEEE/SICE International Symposium on System Integration (SII)*, 1–6. <https://doi.org/10.1109/SII55687.2023.10039288>.

- Walid, Brahim, Jianhua Ma, Muxin Ma, Alex Qi, Yunlong Luo, and Yihong Qi. 2021. “Recent Advances in Radar-Based Sleep Monitoring — A Review.” In *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 759–766. <https://doi.org/10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00125>.
- Wang, Wei, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. “Understanding and Modeling of WiFi Signal Based Human Activity Recognition.” In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 65–76. MobiCom ’15. <https://doi.org/10.1145/2789168.2790093>.
- Wang, Xuanzhi, Kai Niu, Jie Xiong, Bochong Qian, Zhiyun Yao, Tairong Lou, and Daqing Zhang. 2022. “Placement Matters: Understanding the Effects of Device Placement for WiFi Sensing.” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, no. 1 (March). <https://doi.org/10.1145/3517237>.
- Wang, Xuyu, Chao Yang, and Shiwen Mao. 2017a. “PhaseBeat: Exploiting CSI Phase Data for Vital Sign Monitoring with Commodity WiFi Devices.” In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 1230–1239. <https://doi.org/10.1109/ICDCS.2017.206>.
- . 2017b. “ResBeat: Resilient Breathing Beats Monitoring with Realtime Bimodal CSI Data.” In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 1–6. <https://doi.org/10.1109/GLOCOM.2017.8255021>.
- Wang, Yongchuan, Song Yang, Fan Li, Yue Wu, and Yu Wang. 2021. “FallViewer: A Fine-Grained Indoor Fall Detection System With Ubiquitous Wi-Fi Devices.” *IEEE Internet of Things Journal* 8 (15): 12455–12466. <https://doi.org/10.1109/JIOT.2021.3063531>.
- “Wi-Fi Driver - Wi-Fi Channel State Information - ESP32 - — ESP-IDF Programming Guide v5.1 documentation.” n.d. Accessed August 19, 2023. <https://docs.espressif.com/projects/esp-idf/en/v5.1/esp32/api-guides/wifi.html#wi-fi-channel-state-information>.
- Xie, Yaxiong, Zhenjiang Li, and Mo Li. 2015. “Precise Power Delay Profiling with Commodity WiFi.” In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 53–64. MobiCom ’15. <https://doi.org/10.1145/2789168.2790124>.

- Yang, Jianfei, Xinyan Chen, Han Zou, Dazhuo Wang, Qianwen Xu, and Lihua Xie. 2022. "EfficientFi: Toward Large-Scale Lightweight WiFi Sensing via CSI Compression." *IEEE Internet of Things Journal* 9, no. 15 (August): 13086–13095. <https://doi.org/10.1109/JIOT.2021.3139958>.
- Yang, Jianfei, Han Zou, Hao Jiang, and Lihua Xie. 2018. "Device-Free Occupant Activity Sensing Using WiFi-Enabled IoT Devices for Smart Homes." *IEEE Internet of Things Journal* 5, no. 5 (October): 3991–4002. <https://doi.org/10.1109/jiot.2018.2849655>.
- Yang, Zheng, Kun Qian, Chenshu Wu, and Yi Zhang. 2021. "Understanding of Channel State Information." In *Smart Wireless Sensing: From IoT to AIoT*, 11–21. [https://doi.org/10.1007/978-981-16-5658-3\\_2](https://doi.org/10.1007/978-981-16-5658-3_2).
- Yu, Bohan, Yuxiang Wang, Kai Niu, Youwei Zeng, Tao Gu, Leye Wang, Cuntai Guan, and Daqing Zhang. 2021. "WiFi-Sleep: Sleep Stage Monitoring Using Commodity Wi-Fi Devices." *IEEE Internet of Things Journal* 8, no. 18 (September): 13900–13913. <https://doi.org/10.1109/jiot.2021.3068798>.
- Yu, Meng-Chieh, Huan Wu, Jia-Ling Liou, Ming-Sui Lee, and Yi-Ping Hung. 2013. "Multiparameter Sleep Monitoring Using a Depth Camera." In *Biomedical Engineering Systems and Technologies*, 311–325.
- Zaffaroni, Alberto, Sam Coffey, Stephen Dodd, Hannah Kilroy, Graeme Lyon, Damien O'Rourke, Katharina Lederer, Ingo Fietze, and Thomas Penzel. 2019. "Sleep staging monitoring based on sonar smartphone technology." *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, <https://doi.org/10.1109/embc.2019.8857033>.
- Zeng, Youwei, Dan Wu, Jie Xiong, Enze Yi, Ruiyang Gao, and Daqing Zhang. 2019. "FarSense: Pushing the Range Limit of WiFi-Based Respiration Sensing with CSI Ratio of Two Antennas." *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, no. 3 (September). <https://doi.org/10.1145/3351279>.
- Zeng, Yunze, Parth H. Pathak, Chao Xu, and Prasant Mohapatra. 2014. "Your AP Knows How You Move: Fine-Grained Device Motion Recognition through WiFi." In *Proceedings of the 1st ACM Workshop on Hot Topics in Wireless*, 49–54. HotWireless '14. <https://doi.org/10.1145/2643614.2643620>.
- Zhang, Feng, Chenshu Wu, Beibei Wang, Min Wu, Daniel Bugos, Hangfang Zhang, and K. J. Ray Liu. 2021. "SMARS: Sleep Monitoring via Ambient Radio Signals." *IEEE Transactions on Mobile Computing* 20, no. 1 (January): 217–231. <https://doi.org/10.1109/tmc.2019.2939791>.