Vital Sign Monitoring and Movement Tracking using Commodity-Off-The-Shelf Channel State Information

Fabian Putterer
fabian.putterer@tum.de
Technical University of Munich
Garching, Germany

ABSTRACT

This report describes the results and implementations of my research project on using Channel State Information (CSI) obtained from Commodity-Off-The-Shelf devices for vital sign monitoring and movement tracking as well as activity detection. It describes how it is possible to obtain CSI from different types of devices and investigates their respective properties, advantages, and disadvantages. Additionally, ground truth on respiration (chest acceleration) and heart rate (ECG) is obtained for verification. Different approaches for activity, respiratory, and heart rate detection with a focus on heart rate variability using the provided CSI are discussed, implemented, and evaluated. Afterward, possible approaches for increasing the reliability of the detection systems are described. Combining amplitude and phase information using the complex conjugate product representation seems promising and processing algorithms for sanitizing and preparing it for further analysis are developed. Afterward, state and movement classification on the obtained data is performed using machine learning techniques. The framework for obtaining, collecting, recording, processing, and replaying CSI and other types of data in real-time that was developed as part of this project is presented and released publicly.

KEYWORDS

mobile sensing, channel state information, vital sign detection, movement detection

1 OBTAINING CSI

This research project is a continuation of my bachelor thesis[12]. Further insight into how to get the Atheros CSI Tool running and obtain CSI from it can be found there.

As part of this project, I developed a framework for obtaining, processing, recording, and replaying 802.11n[1] channel state information from multiple different devices in real-time as well as offline. It supports obtaining CSI from devices containing an Atheros WiFi chip, using the tool by Xie et al.[22], from Intel-based devices, using the work of Halperin et al.[5] as well as from the ESP microcontroller[7]. Data from multiple devices from different manufactures can be collected and previewed on a central device at once. Additionally, the collection of ground truth data for correlation is possible. Currently, acceleration data from an Android-based phone for respiratory detection as well as ECG data using an ESP microcontroller for heart rate detection are available. The capabilities of the framework as well as its components are described in more detail in Section 15.

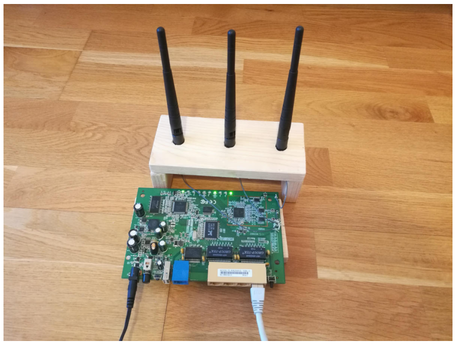


Figure 1: A WR2543ND router with the antenna array

1.1 Obtaining CSI from Atheros Devices

Obtaining CSI using Atheros-based chips is possible using the Atheros CSI Tool by Xie et al.[22]. As described in my thesis, only older devices are supported by the tool, as it relies on the transmit beamforming capabilities of the 802.11n standard[1], which are not mandatory and therefore might have been dropped in newer versions. In this work, the used Atheros-based devices were mostly the TP-Link WR2543ND as well as the WDR4300 which both contain three external antennas when using the 5 GHz band. In some cases, the AR9462, a common WiFi chip present in quite a lot of COTS laptops, was used.

A lot of research on CSI has focused on trying to correct the phase shift introduced by the hardware when using commodity-off-the-shelf (COTS) devices, which is necessary for some more complicated applications like indoor localization. This has been disregarded in this work, as it is not strictly necessary for vital sign detection. Instead, this report mostly focuses on investigating the difference in phase between individual antennas as well as the raw amplitude data to monitor the target parameters. As the phase offset is common to all receive chains of the device and therefore the same across all antennas, it can be eliminated that way without affecting the sensing capabilities of the system.

During my thesis, I was trying to apply the MUSIC algorithm[13] to obtain the angle of arrival as well as the estimated time of flight of the incoming signal. If the spacing of the individual antennas doesn't equal half a wave-length, $\frac{12.302}{2}cm = 6.151cm$ at $2437MHz$, the MUSIC spectrum will contain artifacts[15]. I therefore built and used an antenna array adhering exactly to that spacing. This can be seen in Figure 1.

The orientation of the antenna array is key to the sensitivity of the entire system. In case the RX and TX antennas are oriented parallel to each other and have a direct line of sight (LOS), there

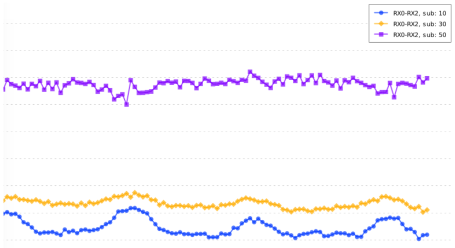


Figure 2: The evolution of the unwrapped phase difference between antenna 1 and 3 over time for three different subcarriers containg a visible respiratory pattern

won't be as many indirect, non line of sight (NLOS) paths and therefore sensing will be limited. Orientation of the transmitting antennas parallel to the ground is preferable, as this creates more NLOS paths and therefore improves the reliability of the system in locations farther away.

2 RESPIRATORY DETECTION USING RSSI

In their work, Abdelnasser et al.[2] present the UbiBreathe system. It uses the received signal strength indicator (RSSI) for monitoring the respiratory rate of the target by fixing the transmitting device to their chest. They claim to use standard WiFI APs, laptops, and cell phones[2, p. 7] and seem to obtain an RSSI value with a resolution of 0.25 which indicates the user's breathing pattern. I was only able to obtain RSSI with a resolution of 1.0, the Atheros hardware itself only reports 8 bits of precision to the driver, the Android API also reports using the lower precision. With this data, I wasn't able to detect any indication of respiratory action and therefore couldn't reproduce their results.

3 ACTIVITY DETECTION USING ATHEROS DEVICES

This section is based on data obtained from the TP-Link WDR4300 using an AR9580 chip in the 5 GHz band.

Most vital sign tracking research so far has used amplitude information from Intel-based devices for monitoring activity and respiratory rate. On Atheros-based devices, the amplitude seems quite unstable and random and is unusable for tracking those characteristics without further processing. The phase data obtained on the other hand seems to be quite stable and suited for further investigation. It contains a random phase offset due to the measurement process which can be removed by taking the difference in phase between two antennas of the same WiFi chip.

Afterward, activity, as well as the breathing pattern of a user, can be spotted in some subcarriers as can be seen in Figure 2. This is still highly dependent on the user's location within the room and might therefore not be present when no LOS or NLOS path is interrupted or influenced by the movement.

Additionally, the information is only present in some of the subcarriers, therefore a method for selecting the best subcarrier for the observation needs to be developed. To achieve this, the average variance across subcarriers can be used on the phase difference data as described by Zhu et al.[25].

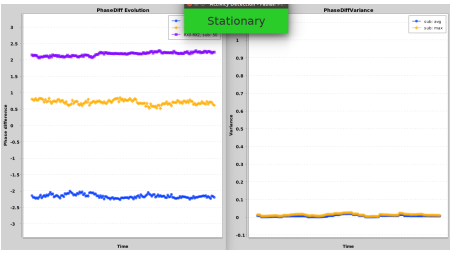


Figure 3: The evolution of the phase difference variance in a stationary scenario

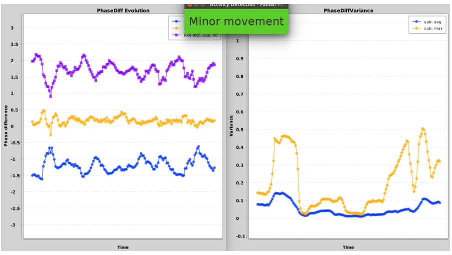


Figure 4: The evolution of the phase difference variance in a non stationary scenario

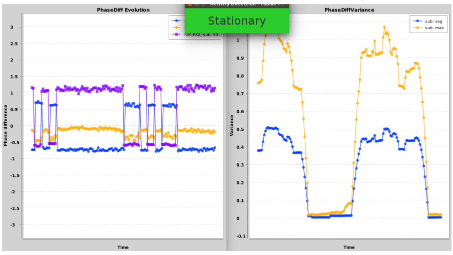


Figure 5: The evolution of the phase difference variance in a "jumping" scenario

Afterward, this criteria doesn't just allow for determining a subcarrier that responds sensitively to movement for further processing but also directly gives a metric for tracking the current activity in the room. This can be seen in Figure 3 and Figure 4.

The detection is hampered sometimes though by some jumps in the underlying phase data as can be seen in Figure 5. Removing them turns out to be quite tricky but is crucial to process the data further. A simple approach is to detect sudden changes in all subcarriers that exceed a certain threshold and to simply correct for them. This yields an issue though, as some sudden changes are caused by movement which interrupts an NLOS path and should therefore be captured. If the threshold is set low enough to remove all of the phase shifts, most of the interesting information contained in the data is lost as well. Sanitization and filtering of the input signal will be discussed in detail in Section 4.

4 RESPIRATORY DETECTION USING ATHEROS DEVICES

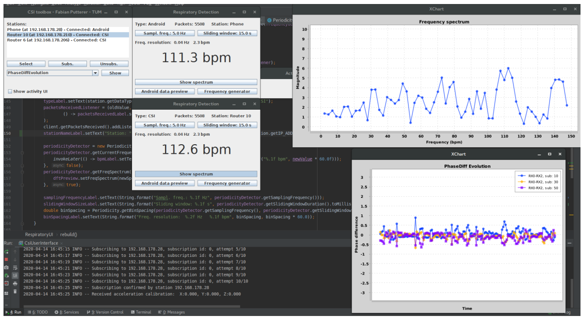
For the development and evaluation of a respiratory detection system, ground truth is required. This allows comparing the respiratory pattern detected using CSI with the actual movement occurring. In this project, an android smartphone that is strapped to the user's chest was used. An app was developed that monitors the device's internal accelerometer and transmits the collected information to the central client device which records it alongside the CSI data for further processing and analysis. As acceleration data itself isn't sensitive enough to reliably capture the breathing activity, the position/orientation data from the sensor is used instead. It combines information from the acceleration due to gravity as well as the orientation reported by the device's magnetic sensor.

One possible approach to frequency detection is to employ peak-to-peak detection by simply searching the signal for its peaks and calculating their average distance. This allows obtaining the periodicity of a repeating movement and therefore its frequency. In this case, a simple discrete Fourier transform (DFT) is used though as suggested by Abdelnasser et al.[2]. This allows detecting frequencies in more noisy data where the actually peeks can't be easily distinguished. Applying it to the ground truth data obtained from the android device directly yields a spectrum indicating the current respiratory rate without any further processing. The respiratory rate can also be seen in the phase difference of some of the monitored subcarriers, although it is distorted by noise, "jumps" and highly dependent on the user location within the room.

The best performance is achieved when the user is near the line of sight (LOS) path and the antennas are parallel to each other. When moving away from the LOS path, performance benefits significantly from the antennas only having NLOS paths due to being oriented in different directions. This can be achieved for example by tilting the transmitter antennas until they're nearly parallel to the ground. The signal therefore has to take a longer, more complex path through the room and is therefore more likely to interact with the user's chest.

To be able to apply the frequency analysis to the CSI phase difference-based data, it needs to be filtered and processed further first. Some jumps as shown in Figure 5 can be eliminated by correctly bounding the signal to the interval $[-\pi, \pi]$ and then unwrapping it. This still leaves sudden changes in the underlying data though, which can be caused when signal propagation paths are interrupted significantly. To remove them, I propose using a truncated mean on a sliding window. It represents a combination of the mean and median. It first sorts the data points in the sliding window according to their size. It then only selects a part of them and discards all non-selected, outer values as specified by the α parameter. This way, only the center values are retained and any outliers are eliminated. Finally, the mean of the remaining values is taken and returned. In this case a sliding window size of $w = 10$ at a frequency of $f = 100Hz$ with $\alpha = 0.8$ was chosen.

Applying a simple fast Fourier transform (FFT) to the resulting, filtered and smoothed signal yields a spectrum that clearly contains the actual respiratory frequency. It is desirable to obtain this frequency in real-time during monitoring to provide quick feedback to the user. The frequency estimation therefore needs to operate on a small sliding window $t \in [5, 15]s$. This reduces the resolution of the spectrum. As the respiratory rate r itself is usually located within a very thin band of frequencies, $r \in [12, 20]bpm = [0.2, 0.33]Hz$ there will only be a few bins in the resulting spectrum that can be distinguished. The precision of the obtained respiratory rate can be significantly improved by interpolation instead of just using the highest peak of the spectrum. This is achieved by fitting a quadratic function to the peak and its neighboring two data points in the spectrum. The peak of the resulting parabola then yields the estimated breathing frequency. In my experiments, this closely matched the actual respiratory rate as observed using the ground truth as can be seen in Figure 7. Figure 6 shows a test of the system when tapping periodically along to a musical beat.



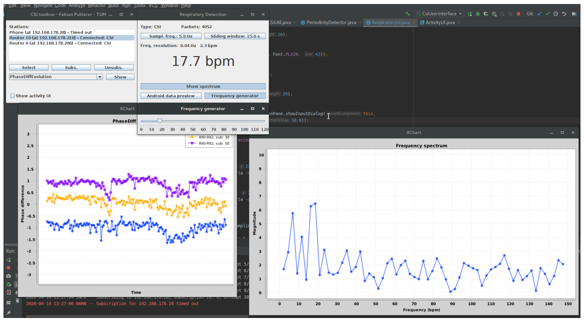
```
Stations:
Phone (at 192.168.178.28) - Connected: Android
Router 10 (at 192.168.178.210) - Connected: CSI
Router 6 (at 192.168.178.206) - Connected: CSI
PhaseDiffEvolution

Type: Android    Packets: 5508    Station: Phone
Sampl. freq.: 5.0 Hz    Sliding window: 15.0 s
Freq. resolution:  0.04 Hz   2.3 bpm

Type: CSI    Packets: 5508    Station: Router 10
Sampl. freq.: 5.0 Hz    Sliding window: 15.0 s
Freq. resolution:  0.04 Hz   2.3 bpm
```

```
2020-04-14 16:45:15 INFO -- Subscribing to 192.168.178.28, subscription id: 0, attempt 5/10
2020-04-14 16:45:17 INFO -- Subscribing to 192.168.178.28, subscription id: 0, attempt 6/10
2020-04-14 16:45:19 INFO -- Subscribing to 192.168.178.28, subscription id: 0, attempt 7/10
2020-04-14 16:45:21 INFO -- Subscribing to 192.168.178.28, subscription id: 0, attempt 8/10
2020-04-14 16:45:23 INFO -- Subscribing to 192.168.178.28, subscription id: 0, attempt 9/10
2020-04-14 16:45:25 INFO -- Subscribing to 192.168.178.28, subscription id: 0, attempt 10/10
2020-04-14 16:45:25 INFO -- Subscription confirmed by station 192.168.178.28
2020-04-14 16:45:25 INFO -- Received acceleration calibration:  X:0.000, Y:0.000, Z:0.000
```

Figure 6: The periodicity detection detecting the frequency of the movement of a foot tapping along to music. The two frequencies, one obtained using the Android-based ground truth, the other obtained using CSI closely match.



```
Stations:
Phone (at 192.168.178.28) - Timed out
Router 10 (at 192.168.178.210) - Connected: CSI
Router 6 (at 192.168.178.206) - Connected: CSI
PhaseDiffEvolution

Type: CSI    Packets: 4052
Sampl. freq.: 5.0 Hz    Sliding window: 15.0 s
Freq. resolution:  0.04 Hz   2.3 bpm
```

```
2020-04-14 13:27:06 WARN -- Subscription for 192.168.178.28 timed out
```

Figure 7: The periodicity detection detecting the respiratory rate by smoothing, FFT and interpolation.

5 SUBCARRIER SELECTION

One important problem of all CSI-based sensing techniques that use a subcarrier-based approach for detecting movement is selecting the best subcarrier to use for the analysis. As subcarriers have slightly different frequencies, they will have different propagation paths and therefore react differently to movement. The goal of subcarrier selection is to determine the subcarrier that reacts the most sensitively to the desired movement and can thereby be used for further processing.

To address this problem, a lot of research simply selects the carrier with the highest variance, assuming it is therefore the most sensitive to movement. This works fine for amplitude as strong changes in amplitude usually represent the largest movement. This yields a problem for phase-based sensing though, as large movement may only cause a small change in case it's not located near a propagation path, and small movement may cause a large change/"jump" when it for example interrupts the LOS path. Therefore variance is unsuited for subcarrier selection in the phase-based detection systems used in this project. Instead, recurrence plots as proposed by Liu et al.[9] were implemented and used. Due to lots of noise, they still were not sufficient for detecting the best subcarrier. As the sensing system proposed later in this report will use and combine data from multiple subcarriers and antennas, this is not relevant though and has not been investigated in detail. All results based on single-carrier data are therefore just tracking a few selected carriers.

6 OBTAINING CSI FROM INTEL DEVICES

As most of the research on leveraging CSI from COTS devices so far has used data obtained from Intel-based devices, obtaining data from the IWL5300 chip has been implemented in this research project as well. It is based on the work by Halperin et al.[5] and uses their CSI Tool for obtaining, previewing, storing, and processing CSI in real-time as part of the framework presented in Section 15. This enables capturing CSI from different types of devices at the same time and comparing and combining them.

The only chip supported by the tool is the Intel WiFi Link 5300 (IWL5300), as the device's firmware has to be modified. It is a client/consumer WiFi chip with 2 or 3 antennas available as half-size or full-size PCIe card. It was usually used in consumer laptops as the main WiFi device and therefore requires an x86-based device to operate.

As part of this project, using an Atheros-based OpenWRT router running the injector tool as the transmitter was attempted. As receiver, a laptop running Ubuntu 14.04 with the IWL5300 was used. This requires cross-compiling the injector as well as its dependency, the lorcon[21] library, for the MIPS architecture used by the router. To be picked up by the receiver and calculate CSI for, the transmitted frame needs to fulfill certain criteria. It needs to be transmitted using a high throughput rate, e.g. `HT20`, and, as monitor mode is being used, needs to be sent from and to the MAC address `00:16:ea:12:34:56`[4].

Certain modifications of the tool are required to achieve this on an Atheros-based device. A deeper investigation into the frames actually sent using Wireshark reveals that the library wrongly packs the header of the frame, therefore causing the receiver to mistake parts of the fragment sequence field for the source MAC address and vice versa. This has been fixed by now (see this issue). Obtaining CSI using the IWL5300 chip still wasn't possible in this work though.

Using a second IWL5300 chip, mounted in another x86 device as transmitter worked without any issues. This suggests two IWL5300 are required for successfully capturing CSI which increases the resource constraints in deployment scenarios, as two x86 devices will be needed as well. In this case, a laptop and an APU Board were used, both running Ubuntu 14.04.

According to Wang and Mao[18], using the 2.4 GHz band is not recommended as random noise might be encountered, therefore capturing CSI using the 5 GHz band on channel 36 was chosen. Only a few other channels can be used, as the IWL5300 is designed as a client chip and may therefore not initiate radiation on most others. During testing, when using any channel other than 36, other issues were encountered.

7 RESPIRATORY DETECTION USING INTEL DEVICES

Respiratory detection using CSI data from Intel devices has been studied in depth[9, 10, 16, 17, 19, 20, 23, 24]. Most previous research has been focusing on amplitude data obtained from a single subcarrier. In my experiments, the phase data obtained from the IWL5300 was quite chaotic, contained a lot of noise and was not usable for sensing.

On the other hand, initially, the amplitude information seems to be quite stable, especially compared to Atheros-based chips, where the inverse situation applies. It contains some noise though, which usually manifests as a sinusoidal offset that is only present in some frames. Some part of the time, the signal is stable and reacts sensitively to movement in the room. During other instances, sudden changes in amplitude can be observed that seem to represent the same shape as before, just with a sinus wave added to all carriers. This can be seen in the Figures 8, 9, and 10. All three frames shown were captured consecutively at a frequency of $100Hz$.

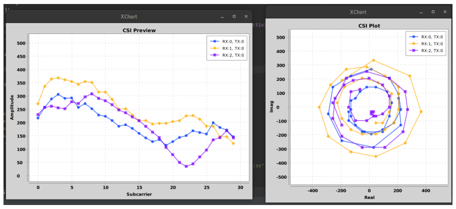


Figure 8: Stable IWL5300 CSI amplitude across all subcarriers

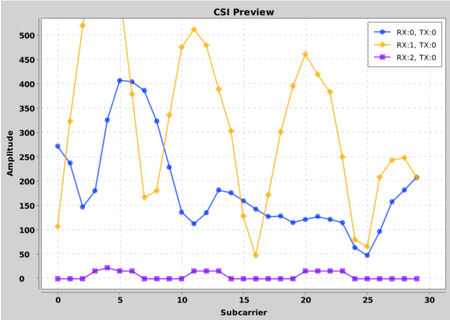


Figure 9: Unstable IWL5300 CSI with a sinusoidal offset

The offset effect seems to reduce when disconnecting an antenna. It is likely related to some internal processing of the chip which

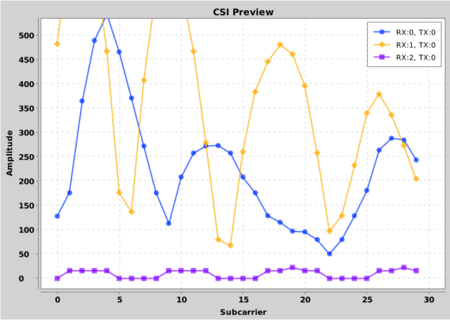


Figure 10: Unstable IWL5300 CSI with a different sinusoidal offset

is trying to improve the link quality and thereby throughput. It vanishes completely when all but one antenna are disconnected. It can therefore and has been mitigated in this project by running most experiments on Intel devices with just a single antenna connected. This does not cause any further issues as amplitude compared to phase is stable and there is no offset due to sampling that would have to be corrected, for example by taking the difference between two connected antennas.

Detecting respiratory action and obtaining the breathing frequency is trivial using amplitude from Intel-based devices. It is present in almost all subcarriers and can be easily extracted, either using peak-to-peak detection or the Fourier transform. This can be seen in Figure 11.

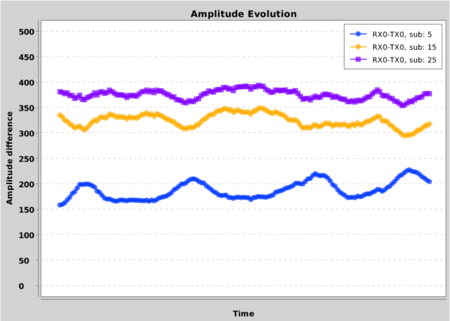


Figure 11: IWL5300 CSI amplitude clearly shows the respiratory pattern in all subcarriers with just one antenna connected.

At first, the phase data from Intel devices seems quite random and noisy. Further investigation shows, though, that there is a constant offset of 8π linearly being applied over 30 subcarriers, as can be seen in Figure 12. Removing this offset by determining each subcarrier's new phase $\hat{\phi_s}$ as

$$\hat{\phi_s} = \phi_s - \frac{8\pi}{s}$$

yields usable and stable phase information as can be seen in Figure 13.

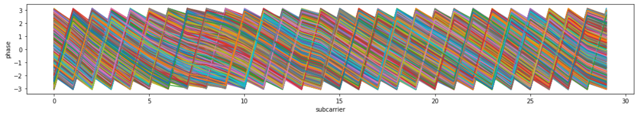


Figure 12: Raw phase information for ten frames obtained using the IWL5300

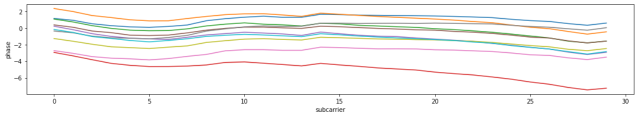


Figure 13: Corrected phase information for ten frames obtained using the IWL5300

8 HEART RATE DETECTION USING INTEL DEVICES

Obtaining heart rate from CSI is even more challenging than respiratory rate. This is due to the small movement caused by the heartbeat and therefore its minor influence on the WiFi signal. Therefore, before being able to extract the heart rate, more significant components like the breathing rate need to be detected and removed from the signal. This can be achieved using a Fourier transform.

Additionally to the pure frequency of the heart rate, obtain additional characteristics about the heartbeat is also desirable. One of them is the so-called heart rate variability (HRV) which measures the average offset and therefore the offset of the individual heartbeats from a purely periodic signal. This can serve as an important indicator of well-being as it allows correlations with the stress and relaxation levels of the subject[11]. Due to this, obtaining HRV has been one of the main foci of this project.

HRV detection relies on being able to extract the heart rate and measure the distance and thereby the variability/non-periodicity of its peaks. Applying a forward and inverse Fourier transform to remove the respiratory effects would cause any non-periodic effect to vanish. Therefore, a more sophisticated approach is required, that preserves the locality of the individual frequency components. As suggested by Wang et al.[19] and German-Sallo[3], one solution to this problem is using the discrete wavelet transform (DWT), a spatially localized extension of the Fourier transform. This allows decomposing the signal step by step, each time removing the upper half of the frequency spectrum from the signal.

This decomposition can be seen in figure 14. The left side represents the approximation coefficients. The contained frequencies are halved during each step, in the end, only the respiratory rate remains. The right side shows the detail coefficients, which represent the information contained in the upper half of frequencies during each removal. The detail coefficients representing the frequencies heart rate is usually contained in can afterward be combined to obtain a localized representation of target frequencies. Afterward, further processing, e.g. using a normal DFT or peak-to-peak detection can be applied.

As the heart rate only causes a minor change in signal, a directional antenna was used to improve sensitivity as suggested by Wang et al.[19]. Like with respiratory detection, ground truth

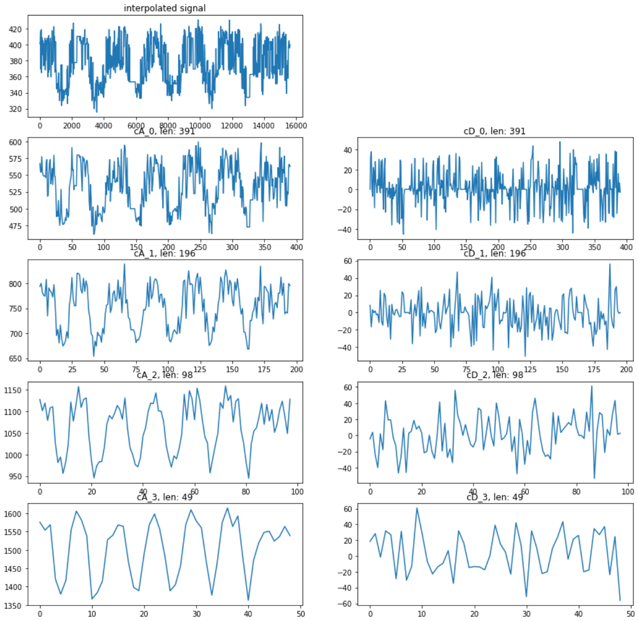


Figure 14: The wavelet transform being applied. In the lowest approximation coefficient, the regular respiratory pattern can be seen.

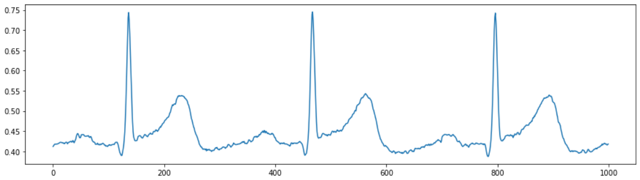


Figure 15: Raw data from the ECG sensor

data was obtained for analysis and verification. This was achieved by using a simple electrocardiogram (ECG) sensor in combination with an ESP microcontroller. One of the main issues encountered with this approach is caused by the microcontroller and therefore sensor being connected to the recording device directly for data transmission. This causes the influence of the power grid's operating frequency to be represented within the measured heart rate data. To eliminate this factor, the entire setup was designed using an access point (AP) for data transmission between the recording device (laptop) and the data sources. The Laptop uses its internal battery, which removes any trace of the power grid's operating frequency from the signal. The obtained heart rate signal can be seen in Figure 15.

There are multiple representations for HRV, one of them being the root mean square of successive differences (rMSSD). This method measures the peak-to-peak distance between the individual R-peaks of the QRS complexes. This yields a HRV value of $29.5ms$ for the heartbeat signal shown in figure 15. The individual R-R intervals can then also be visualized in a so-called Poincare Plot.

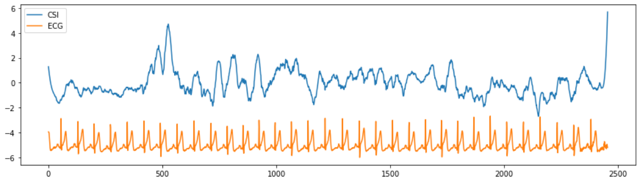


Figure 16: Raw CSI amplitude and ECG data

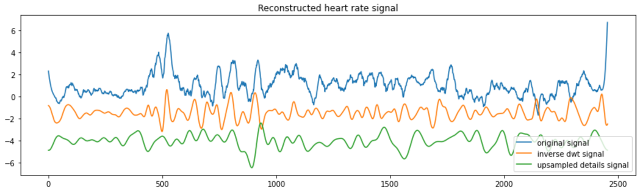


Figure 17: The reconstructed heart rate signal. The upsampled signal is the result of simply upsampling all of the detail coefficients to the same resolution and summing them.

Using the entire setup, ECG heart rate data, as well as Intel and Atheros CSI data, was now captured for different scenarios. Different distances (2m, 4m), antennas (omnidirectional, directional) and locations (inpath, nearpath, behindpath) were evaluated. An in-depth investigation of the obtained data was performed.

The raw CSI amplitude data and the ECG signal can be seen in Figure 16. No direct correlation can be seen. Using the wavelet transform, setting all out of band coefficients to 0 and reversing the transform, we can now obtain all frequency information in the desired band, in this case $[0.78125, 3.125]Hz \approx [47, 188]bpm$. The reconstructed heart rate signal resulting from this can be seen in Figure 17.

No clear indication of the actual heart rate could be seen here either. Running a simple DFT on the reconstructed heart rate signal yields no indication of the actual underlying heart rate as can be clearly seen in Figure 18 compared with the ECG ground truth data.

During this project, extracting heart rate from WiFi CSI wasn't possible despite using directional antennas, multiple different scenarios, different wavelets, and different devices. The same analysis was also performed on phase data, as can be seen in figure 19.

9 OBTAINING CSI FROM THE ESP MICROCONTROLLER

Obtaining CSI from Intel and Atheros-based devices requires either a router or an x86 device. They are quite large devices and in the case of routers even require a stationary power supply. Using a mobile device with its own power source which can be easily transported and installed is therefore desirable. This can be achieved using an ESP microcontroller.

One of the main advantages of this solution, additionally to it being mobile, is that, unlike with the other tools, no workaround is used for obtaining CSI but the functionality is built-in with the device. Due to this, a station-to-access point connection isn't strictly required and the device also supports passive listening to frames sent by other networks. On the other hand, the ESP only has one

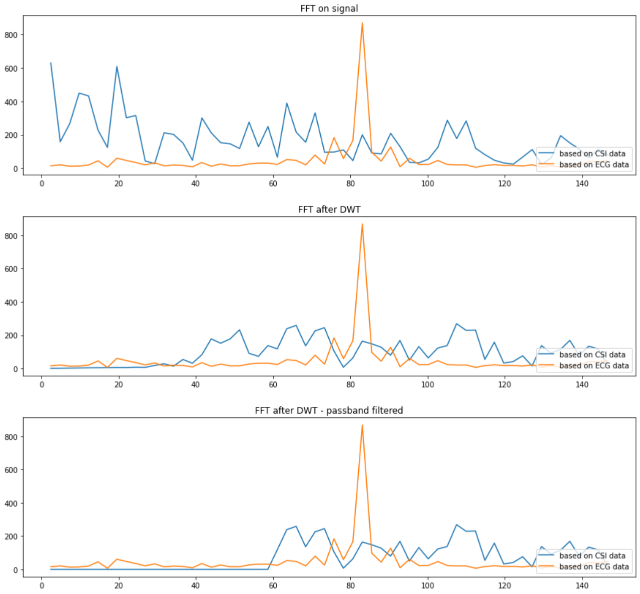


Figure 18: The frequency spectrum of the original signal and reconstructed heart rate signal. They don't show any indication of the frequency representing the current heart rate clearly shown by the ECG ground truth frequency spectrum.

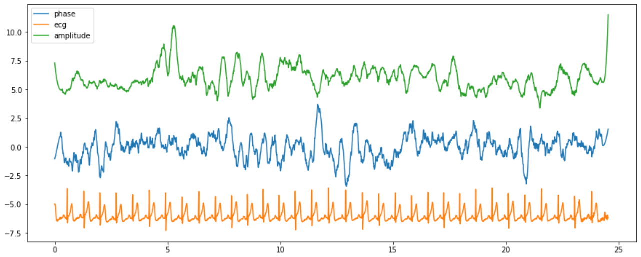


Figure 19: Comparison of the reconstructed heart rate signal based on amplitude and phase.

antenna, implying that phase data can not be used, as there is no way to calculate the difference between multiple antennas.

Gathering CSI from ESPs is possible and integrated with the framework developed during this project. The implementation is based on the work by [7].

The obtained data allows detecting respiratory action but is only available for one antenna and way less sensitive than that of the other devices. This is likely due to the small antenna being used. A comparison of data from all three devices can be seen in Figure 20.

10 IMPROVING RELIABILITY

As the ESP only has one antenna, it can easily disconnect when the orientations of the receiving and transmitting antenna are not set up correctly or there is an obstacle in the line of sight path. Due to this and the ESPs weak signal sensitivity, further experiments have focused on the larger devices using more capable antennas.

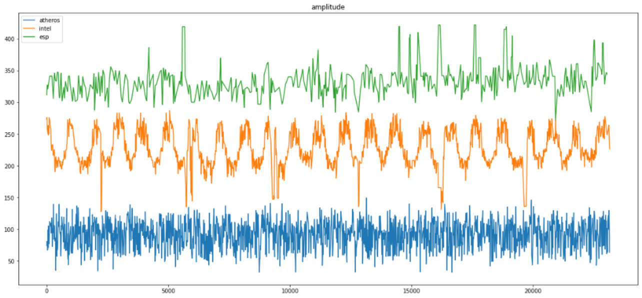


Figure 20: CSI data obtained from all three devices in parallel

Most of the approaches discussed so far rely on data from a single device for a single subcarrier. They usually only use one antenna or one difference between two different antennas. They all focus on either amplitude or phase. This allows detecting activity, movement, and respiratory action, but the reliability highly depends on the subject's position within the room. The accuracy can be improved a bit by not orienting the transmitting and receiving antennas parallel to each other, therefore not having a line of sight between the transceivers. In my experiments, the most successful orientation meant leaving the receiver antennas oriented in a direction they would be able to receive a signal from the transmitter in and tilting the transmitting antennas 30° away from being parallel to the ground. Compared to the situation before, when only movement near the LOS path was detectable, this way, coverage of the room can be improved significantly.

Additionally, instead of just looking at amplitude or phase data, which each show some lack in sensing capability when moving across the room, they can also be combined and used in unison to overcome the limitations of the more simplistic systems presented before. This will be investigated in the following sections.

11 LEVERAGING AMPLITUDE AND PHASE

Most research so far simply focused on extracting the amplitude or phase data from the most variant subcarrier and using that on its own to perform the sensing. Those systems, while usually functioning in some setups, usually lack robustness and reliability, especially against the subject moving or being located in different positions in the room. Due to this, Zeng et al.[23] proposed combining amplitude and phase information to improve sensitivity and reliability.

They model the signal's propagation paths through the room using Fresnel zones as described by Wang et al.[16]. This will cause a complementary shift in amplitude and phase representing a simple, distorted rotation in the signal space. This way, the signal will shift between constructive and destructive interference once per Fresnel zone boundary. Given a frequency of $5189MHz$, the signal has a wavelength of $l_{s_1} = 5.79cm$, with a frequency of $2437MHz$, it is $l_{s_2} = 12.3cm$. The lengths of the NLOS paths of the boundaries are therefore $2n \times \frac{l_s}{2}$ (constructive) and $(2n + 1) \times \frac{l_s}{2}$ (destructive).

Using the LOS path length f, and the NLOS paths $2a$, we can then obtain the distance between the LOS path and the individual Fresnel zone boundaries as:

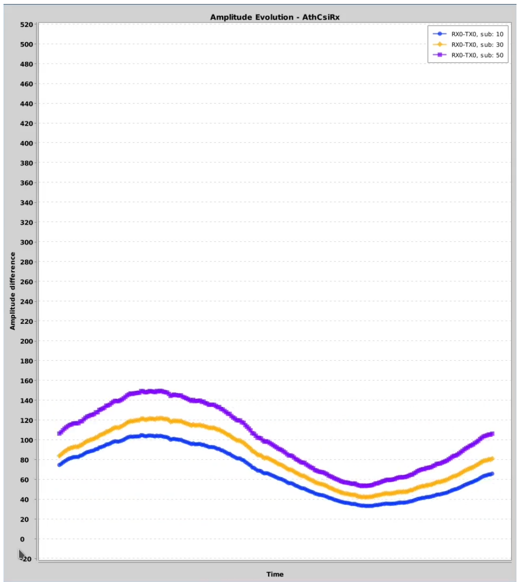


Figure 21: CSI amplitude while moving a big metal plate between Fresnel zones

$$l_m = \frac{\sqrt{(2a)^2 - f^2}}{2}$$

0.27 cm	destructive
0.39 cm	constructive
0.48 cm	destructive
0.56 cm	constructive
0.63 cm	destructive
0.69 cm	constructive
0.75 cm	destructive

Using a large metal plate, this can also be confirmed by moving it orthogonally to the LOS path along the given boundaries. The resulting change in amplitude, switching between constructive and destructive interference, can be seen in Figure 21. When the plate is within a location of a high constructive or destructive interference, its differential quotient approximates to 0. This implies, that a change/movement in that location will only cause a small deviation in the resulting amplitude measurement. But, according to the model proposed by Zeng et al.[23], in these locations, the effect on phase will be the largest possible, allowing us to sense most movement when leveraging a combination of the two.

The postulated, rotating effect can not be observed in the raw CSI data or two-dimensional CSI difference. Due to this, the subcarrier-wise complex conjugate multiplication (CM) of CSI data of two different antennas as suggested by Zeng et al.[24] was investigated. This effectively results in using an amplitude ratio combined with a phase difference between antennas, a two-dimensional plot which can be seen in Figure 22.

Comparing the phase difference evolution and CSI CM plots reveals, that depending on the scenario, a movement might be visible in the phase, e.g. if the CM plot crosses over the origin, but in other scenarios, as seen from the Fresnel zone model, the movement cannot be seen from phase alone but a clear change in the CM shape plot can be spotted.

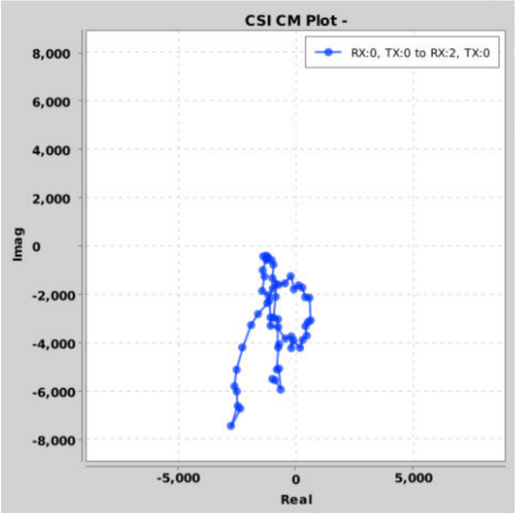


Figure 22: An unprocessed plot of the complex conjugate (CM) product between two different antennas

Using this insight, Zeng et al.[23][24] were able to improve their sensing reliability quite substantially. They argue based on the CM, but afterward, continue using amplitude and phase separately, but complimentary.

Instead of processing them independently, what if we look at and try to classify the CM data in the signal space directly? This key insight will be explored in the following sections. First of all, we will need to apply some filtering and processing to the signal as it is quite chaotic and contains a lot of noise and "jumping". All data for these experiments was obtained from Atheros-based devices.

The signal chaotically changes amplitude. This corresponds to the observations in Section 7. The amplitude data was just discarded therefore beforehand. Using this new representation, we now simply apply normalization to the CM data. This will remove any absolute amplitude changes but will keep any changes in amplitude relative to other subcarriers intact.

Additionally, Sanitization of the CM plot is performed using smoothing and rotation filtering. The "jumps" we saw before in raw phase data are now clearly visible. They manifest themselves as rotations by 120° around the origin as well as instances, where all subcarriers simply collapse to the same phase, as can be seen in Figure 23. This might be due to some internal processing or a switch in the MCS index. Given our new representation, we can now easily eliminate the "jump" by filtering all samples with the difference between all phase values over all subcarriers below a threshold. Using interpolation of all other packets, we can fill in the usually short gaps.

This still leaves sudden changes due to rotation. Those can also be eliminated now by detecting determining the average phase of the previous and current frame. We then rotate one of them and test for overlap. If they are similar enough, but the overall rotation is large, a permanent phase offset is added to all subsequent frames.

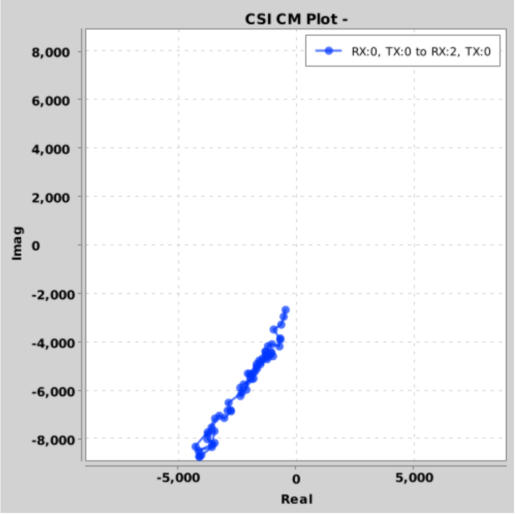


Figure 23: A CM-plot with all subcarriers aligned along the same phase value

Afterward, some additional smoothing using an average over a sliding window is performed. This leaves us with sensitive, reliable, and stable data that reacts to a lot of movement that could not be observed using just amplitude or phase. Using this data for classification will be investigated in the following sections.

12 CLASSIFICATION OF THE CM REPRESENTATION

The CM representation is sensitive to a lot of movement and activity, but there is no clear path to extracting and characterizing information from it directly. It is obvious when observed by a human, but inferring information algorithmically is a bit more tricky. Two different types of data will be investigated and classified, stationary data representing different locations and temporal data representing different movements and gestures.

To enable characterizing movement as observed in CSI data, either classically or using machine learning, we need to derive some characteristic from the CM shape that responds to and represents the measurement. The input data is two-dimensional, and in the case of movement, there are multiple samples over time. The goal is therefore to find some kind of dimensionality reduction procedure.

To eliminate noise, the system needs to be able to treat rotations equally and only take into account changes in the subcarriers relative to each other. We are therefore looking at just characterizing the shape and its change relative to itself instead of looking at the absolute position in the signal space. One approach is therefore dissecting the shape into lower-dimensional characteristics, that still fully represent it. To achieve this, we can obtain the angles and distances between neighboring subcarriers. This decomposes our two-dimensional shape into two one-dimensional data sets that we can now use for further classification, without losing any of the relative components of the data. To create a continuous function, the data is generated using absolute instead of relative angle values as can be seen in Figure 24.

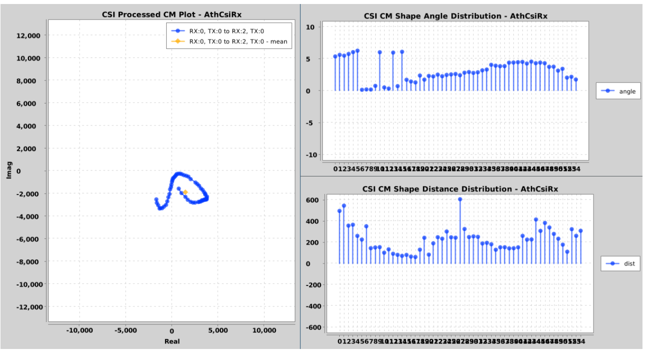


Figure 24: A CSI CM shape with the corresponding one-dimensional angle and distance distributions

For further processing, it might be interesting to fix the starting position of the resulting distributions, by subtracting the angle of the first value from the entire function. This removes information about the global rotation though. As the sanitization algorithm for the CM shape stabilizes sudden phase rotations though, this is not necessary anymore. Maintaining the global phase is therefore desirable. While being useless for static classification due to global offsets, it is still useful for the classification of movements that gradually change the total phase.

Another issue with the angle data is that it represents an angle. This means there are discontinuous locations within the function where the angle wraps around. This can be removed by unwrapping, but then the resulting function is theoretically unbounded, making further processing like machine learning more complicated. Instead, a classification method, that treats the given data as a function that wraps around at 2π is needed.

Additionally, sometimes there is quite a sudden change in angle between two subcarriers, but the shape stays about the same. This can be seen in Figure 24 in the angle distribution, where there are three discontinuous values. During movement, even if the shape stays the same, those may shift one or two subcarriers to the left or right. We therefore need a representation, that is invariant to small shifts between subcarriers as well as considers the wrap around from 0π to 2π to be equivalent.

One approach to solving this problem is employing a kernel-like convolutional filter as is the case for convolutional neural networks (CNNs). To accommodate the circular nature of the angle wrap around, we need to modify the filter to wrap around the top and bottom of the data as is done in a circular convolutional neural network (CCNN) as proposed by Schubert et al.[14]. Afterward, we can then actually just use a CNN for the classification of states or feed the output of the CNN into some kind of recursive neural network (RNN) like a Long short-term Memory (LSTM) network to classify movement.

13 CLASSIFYING STATIONARY STATES

Using the sanitized CM representation of the CSI data we can now perform classification. The approach for classifying stationary states will build on a classical CNN architecture. As moving the transmitter and receiver or restarting the respective devices causes

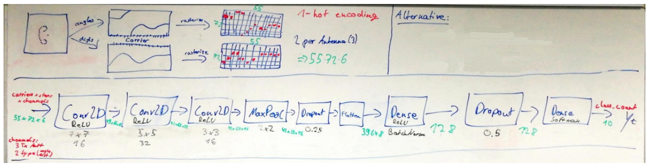


Figure 25: The processing / learning architecture

an unknown offset to be introduced to the CSI data, stationary classification may be limited to a few states where propagation paths are distinctively interrupted but do not change too much when moving a little bit. Nonetheless, this serves as a buildup to approaching movement classification, which requires deriving some characteristic indicating movement. While the stationary CNN approach might not be capable of reliably classifying all states, it can be trained in unison with an RNN appended to its end, allowing the RNN to detect a change in position, even though it would not be capable of reproducibly detecting the same state with slightly different positions.

As states, different locations in the room were chosen and marked. At those positions, CSI "images" were taken without any movement occurring. The framework supports saving the current two-dimensional CSI CM shape as well as the distance and angle distributions along with the class of the current position, comparable to taking an image with a camera.

To ensure that movement between training and test of the models is sufficient and the network can not just similarly classify "images" taken right after each other, the training, validation, and test data sets are captured independently. The entire processing architecture can be seen in Figure 25. First, the two-dimensional CM shape is decomposed into one-dimensional angle and distance distributions. Those are converted to two-dimensional 1-hot encoded arrays so that they can be processed by the CNN, thereby causing nearby values in the encoding to be treated similarly. Using a circular kernel will then allow us to use a wrap-around compared to just encoding a single value. As different transmitter antennas have different propagation paths that react differently to movement, multiple transmit antennas are used, always deriving the complex conjugate product from the same two receiver antennas. This improves reliability and sensitivity. The result are 2 (angle and distance) arrays for all 3 TX antennas, all containing 55x72 values (55 subcarriers), therefore an individual sample has a size of 55x72x6.

This data can now be used to train the model, which consists of 3 convolutional layers with 16 to 32 filters, followed by pooling and two dense layers. For regularization, batch normalization and dropout are used.

After some tweaking of hyperparameters and using angle as well as distance data from multiple antennas, the model achieves an accuracy of $45 - 50\%$ for classifying seven different locations. This is a significant improvement over pure guessing which would yield an accuracy of $\frac{1}{7} \approx 14.2\%$ but far from reliable classification. An instance of the training process can be seen in Figure 26.

At some point, the model fails at generalizing, even when increasing regularization. This is likely due to the classification of pure states not being possible reliably. The classification for a few states always succeeds, but some other states or only slightly better than guessing.

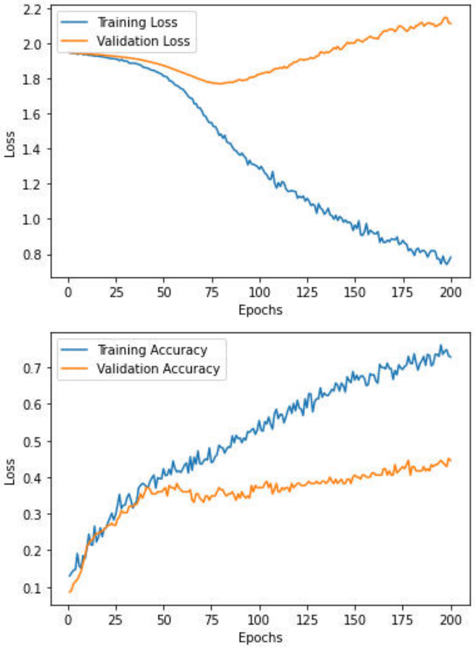


Figure 26: The training curves during model fitting for stationary samples

During experimentation, it can be observed, that for some instances, for example when the subject is near a propagation path, a small movement will yield a lot of change in the CSI data. This usually means better detection, but might also cause the phase to complete an entire rotation and therefore end up in the same configuration as before. On the other hand, in some locations, a large movement might not influence a propagation path, as there is none nearby, thereby causing no change in the observed signal and therefore not being sensitive at all.

Additionally supplying the two-dimensional, raw CM shape, was also attempted but did not improve the model's performance significantly. This does not hinder further progress though, as state classification depends on multiple external factors and offsets, that aren't relevant to movement detection. Therefore, movement classification will be attempted in Section 14 based on the state characteristic extraction pipeline developed for state classification.

14 CLASSIFYING MOVEMENT

Based on the stationary classifier developed in Section 14, a movement classifier can now be developed.

The framework developed during this project supports recording temporal data of the CM shape as well as angle and distance distributions, comparable to a video captured by a camera. Each "video" spans a time interval of $t = 3s$ recorded at a frequency of 20Hz.

The previous, CNN-based pipeline is time distributed and executed for each of the 60 frames individually. This gives the RNN

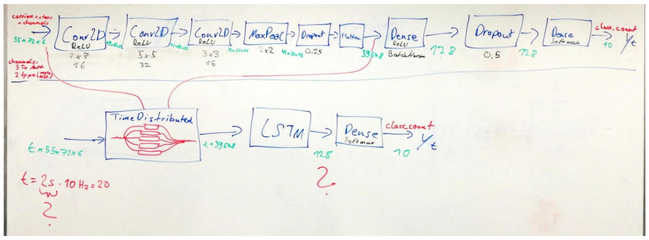


Figure 27: The processing / learning architecture for movement samples

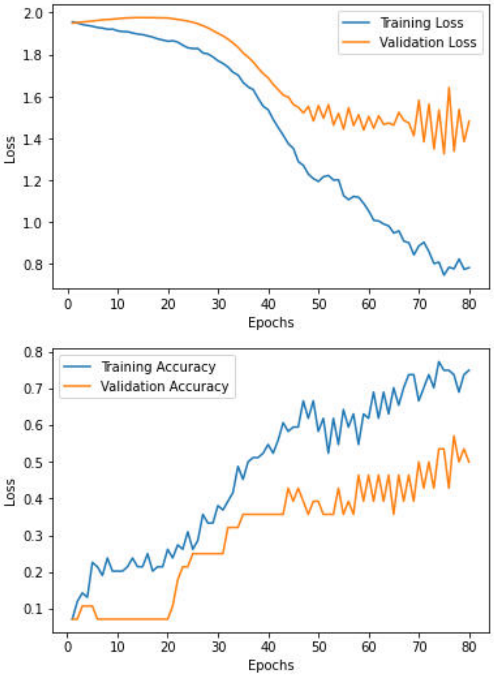
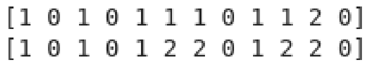


Figure 28: The training process for all seven classes



```
[1 0 1 0 1 1 1 0 1 1 2 0]
[1 0 1 0 1 2 2 0 1 2 2 0]
```

Figure 29: The predicition for standup(0), sitdown(1), and stationary(2), the upper line is prediction, the lower one label

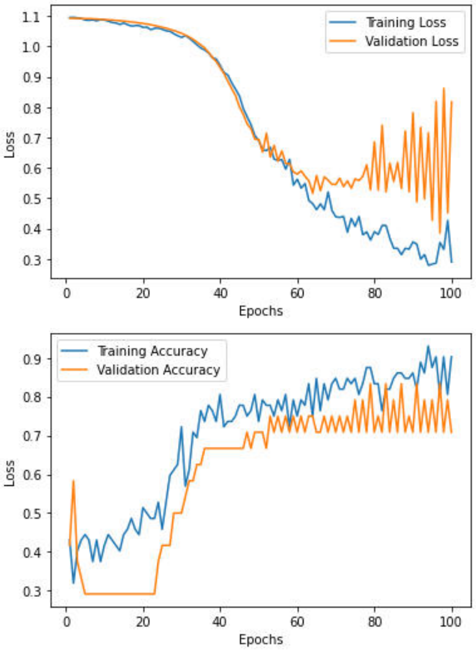


Figure 30: The training process for just three classes

some characteristic, describing the current state, to perform the classification on. The input samples therefore have a size of 55x72x7x60. The entire architecture can be seen in Figure 27.

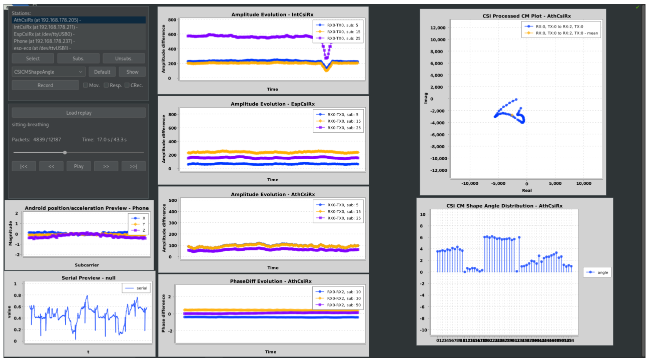
Seven different movement types have been selected for evaluation. Standing up, sitting down, remaining stationary, waving, nodding, clapping, and walking. They were recorded in different locations and orientations, therefore not requiring separation by train, validation, and testing datasets, as they can be randomly picked from the entire dataset. There are 40 samples per class, yielding 2400 frames overall.

Training the model on all seven classes (guessing accuracy is 14.2%) yields an accuracy of $\approx$ 55% as can be seen in Figure 28. Looking at the actual predictions on the test set, it can be seen that the model is pretty confident on some classes, while constantly mistaking the same other classes for each other. In Figure 29, this can be seen for just the states standing up(0), sitting down(1), and being stationary(2) where the model constantly classifies standing up correctly but keeps confusing sitting down and being stationary.

Reducing model expressiveness, the train and validation gap can be closed further, but no better performance on the validation/test data can be observed. Training just on three classes, standing up, sitting down, and being stationary yields a test performance of 80% (guessing: 33%) as seen in Figure 30.

The inaccuracy for some pairs of classes is likely due to similarities in the movement, for example in the case of waving and nodding, which both contain periodic movement at the same frequency. Distinguishing them using CSI data might be possible but will require further research.

Another approach could be to just look at a single, one-dimensional characteristic over time, for example the variance of the phase, amplitude, or CM representation, and use a simple convolution to compare it to some learned function for each class. As each movement class usually has a characteristic intensity pattern, this could allow determining the movement performed. Further research will be needed to investigate whether it is possible to classify movement this way.



Stations:
AthCsiRx (at 192.168.178.205) -
IntCsiRx (at 192.168.178.211) -
EspCsiRx (at /dev/ttyUSB0) -
Phone (at 192.168.178.237) -
esp-ecg (at /dev/ttyUSB1) -

CSICMShapeAngle

sitting-breathing

Packets: 4839 / 12187 Time: 17.0 s / 43.3 s

Figure 31: A screenshot of the client application

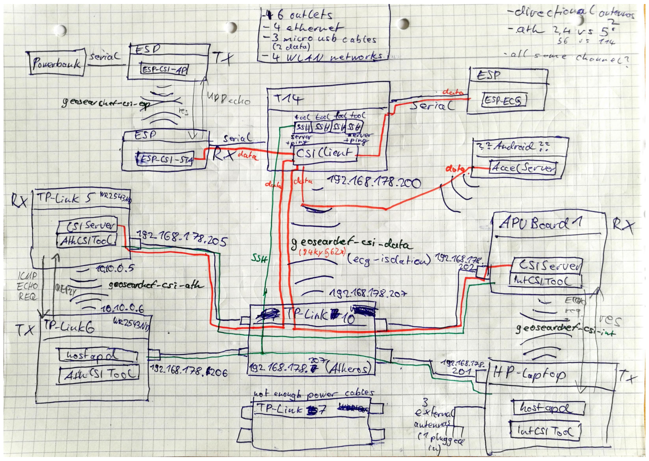


Figure 32: The network architecture used for gathering data

15 FRAMEWORK

As part of this project, a framework was developed, that allows gathering, previewing, processing, recording and replaying CSI as well as ECG and acceleration data in real-time as well as offline. For each device type, a server is provided. A central client application can then subscribe to each server, thereby getting notified about new data generated by the device.

The framework and further documentation on how to build it can be found at:

https://github.com/putterer/csi-client-app

An example of the client application running and replaying pre-recorded CSI data from all device types as well as ECG and acceleration data can be seen in Figure 31.

15.1 Network Architecture

A sketch of the network architecture used for recording data from all possible sources can be seen in Figure 32. All Atheros and Intel-based routers are connected to a central AP using ethernet cables, which connects to the client application using a WiFi data network. This is done as the client also records ECG data as ground truth and therefore needs to be separated from any grid-based power supply to avoid noise generated by the grid's operating frequency.

15.2 Previews

Different types of previews are supported, including showing raw as well as heavily processed and filtered data.

- **Raw CSI:** Signal space plot of the obtained CSI
- **Raw Amplitude:** Per subcarrier amplitude as well as evolution of multiple subcarriers
- **Raw Phase Diff:** Per subcarrier phase difference between two antennas as well as evolution of multiple subcarriers
- **CM Plot:** Plot of the complex conjugate product between two antennas, raw as well as heavily processed, smoothed, and filtered
- **CM Distributions:** Angle as well as distance distributions of the CM shape
- **Android Evolution:** Evolution of acceleration data obtained from an android application
- **Serial Evolution:** Evolution of raw serial data, e.g. ECG data

Previews can be used by selecting the respective station and using the Default and Show buttons. Additional settings will be prompted using the Show button, else the defaults will be assumed.

15.3 Recording

All incoming data from all stations can be recorded in real-time by pressing the Record button. The data will be compressed and written to the selected folder.

15.4 Replaying

Recorded data can later be loaded and replayed, applying all preview and processing algorithms as if the data were being generated in real-time. This can be achieved using the Load replay button and will restart the application, therefore closing all previews. The recording can then be controlled using the provided interface.

15.5 Processing

Using the Movement and Respiratory checkboxes, the respective components can be activated for the selected station. This allows tweaking settings for and showing results for activity detection and respiratory detection algorithms, including a preview of the frequency analysis of the current signal using the Fourier transform.

15.6 Complex Conjugate Multiplication Sample Recording

Using the CRecording checkbox, the CM recording dialog can be opened. This allows recording samples of the current heavily processed CM shape as well as angle and distance distributions. These allow for applying machine learning techniques later on. The recording can be switched between temporal and non-temporal mode, thereby recording either single samples or batches of 60 samples (at 20 Hz for 3 seconds). A sample is recorded by pressing a button or keyboard key for the desired sample class, which is also stored and can be used as an indicator for further processing.

REFERENCES

[1] 2009. IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput. *IEEE Std 802.11n-2009* (Oct 2009). https://doi.org/10.1109/IEEESTD.2009.5307322

[2] Heba Abdelnasser, Khaled A. Harras, and Moustafa Youssef. 2015. UbiBreathe: A Ubiquitous Non-Invasive WiFi-Based Breathing Estimator. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Hangzhou, China) *(MobiHoc '15)*. Association for Computing Machinery, New York, NY, USA, 277–286. https://doi.org/10.1145/2746285.2755969

[3] Zoltan German-Sallo. 2014. Wavelet Transform based HRV Analysis. *Procedia Technology* 12 (2014), 105–111. https://doi.org/10.1016/j.protcy.2013.12.462 The 7th International Conference Interdisciplinarity in Engineering, INTER-ENG 2013, 10-11 October 2013, Petru Maior University of Tirgu Mures, Romania.

[4] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Linux 802.11n CSI Tool - FAQ. https://dhalperi.github.io/linux-80211n-csitool/faq.html.

[5] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11N Traces with Channel State Information. *SIGCOMM Comput. Commun. Rev.* 41, 1 (Jan. 2011), 53–53. https://doi.org/10.1145/1925861.1925870

[6] Ren-You Huang and Lan-Rong Dung. 2016. Measurement of heart rate variability using off-the-shelf smart phones. *Biomedical engineering online* 15, 1 (2016), 1–16.

[7] jonathanmuller. 2021. Gather CSI (Channel State Information) frames with the use of an ESP32 WiFi chip. https://github.com/jonathanmuller/ESP32-gather-channel-state-information-CSI-.

[8] Jian Liu, 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* (Hangzhou, China) *(MobiHoc '15)*. Association for Computing Machinery, New York, NY, USA, 267–276. https://doi.org/10.1145/2746285.2746303

[9] Xuefeng Liu, Jiannong Cao, Shaojie Tang, and Jiaqi Wen. 2014. Wi-Sleep: Contactless Sleep Monitoring via WiFi Signals. In *2014 IEEE Real-Time Systems Symposium*. 346–355. https://doi.org/10.1109/RTSS.2014.30

[10] Xuefeng Liu, 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, 10 (2016), 2466–2479. https://doi.org/10.1109/TMC.2015.2504935

[11] Peter Nickel and Friedhelm Nachreiner. 2003. Sensitivity and Diagnosticity of the 0.1-Hz Component of Heart Rate Variability as an Indicator of Mental Workload. *Human Factors* 45, 4 (2003), 575–590. https://doi.org/10.1518/hfes.45.4.575.27094 PMID: 15055455.

[12] Fabian Putterer. 2019. *Indoor Localization Using Commercial Off-The-Shelf WiFi Devices.* Bachelor's Thesis. https://doi.org/10.13140/RG.2.2.29124.24967

[13] R. Schmidt. 1986. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation* 34, 3 (March 1986), 276–280. https://doi.org/10.1109/TAP.1986.1143830

[14] Stefan Schubert, Peer Neubert, Johannes Pöschmann, and Peter Protzel. 2019. Circular Convolutional Neural Networks for Panoramic Images and Laser Data. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. 653–660. https://doi.org/10.1109/IVS.2019.8813862

[15] Honghao Tang. 2014. *DOA estimation based on MUSIC algorithm.* Bachelor's Thesis. Linnæus University, Kalmar Växjö, Sweden.

[16] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. 2016. Human Respiration Detection with Commodity Wifi Devices: Do User Location and Body Orientation Matter?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 25–36. https://doi.org/10.1145/2971648.2971744

[17] Pei Wang, Bin Guo, Tong Xin, Zhu Wang, and Zhiwen Yu. 2017. TinySense: Multi-user respiration detection using Wi-Fi CSI signals. In *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. 1–6. https://doi.org/10.1109/HealthCom.2017.8210837

[18] Xuyu Wang and Shiwen Mao. 2019. Deep Learning for Indoor Localization based on Bi-modal CSI Data. *Appl. Mach. Learn. Wirel. Commun* 81 (2019), 343.

[19] Xuyu Wang, Chao Yang, and Shiwen Mao. 2017. 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

[20] Xuyu Wang, Chao Yang, and Shiwen Mao. 2017. TensorBeat: Tensor Decomposition for Monitoring Multiperson Breathing Beats with Commodity WiFi. *ACM Trans. Intell. Syst. Technol.* 9, 1, Article 8 (sep 2017), 27 pages. https://doi.org/10.1145/3078855

[21] Joshua Wright and dragorn. 2009. Library - lorcon - Loss Of Radio CONtrol. https://github.com/kismetwireless/lorcon.

[22] Yaxiong Xie, 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* (Paris, France) *(MobiCom '15)*. ACM, New York, NY, USA, 53–64. https://doi.org/10.1145/2789168.2790124

[23] Youwei Zeng, Dan Wu, Ruiyang Gao, Tao Gu, and Daqing Zhang. 2018. Full-Breathe: Full Human Respiration Detection Exploiting Complementarity of CSI Phase and Amplitude of WiFi Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 3, Article 148 (sep 2018), 19 pages. https://doi.org/10.1145/3264958

[24] Youwei Zeng, 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, 3, Article 121 (sep 2019), 26 pages. https://doi.org/10.1145/3351279

[25] Yanzi Zhu, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y. Zhao, and Heather Zheng. 2020. Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors. *Proceedings 2020 Network and Distributed System Security Symposium* (2020). https://doi.org/10.14722/ndss.2020.23053