Universidad Autónoma de Baja California

Instituto de Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería

Monitoring Vital Signs at Rest while Using Channel State Information of Wi-Fi signals and Artificial Intelligence Tools

Tesis para Obtener el Grado de:
Maestro en Ciencias

Presenta:

Jesús Albany Armenta García

Director de Tesis:

Dr. Félix Fernando González Navarro

Codirector de Tesis:

Dr. Jorge Eduardo Ibarra Esquer

Mexicali, B. C. Junio 7, 2022

Dedications

This thesis is dedicated to my parents, José and Tomy, and to my partner in life, Monserrat, for their endless support and encouraging for always giving my best during this investigation.

Acknowledgements

I would like to acknowledge to all the professors that provide me the required knowledge for realizing this investigation, specially to Dr. Félix Fernando González Navarro.

To Consejo Nacional de Ciencia y Tecnología (CONACyT) and the Universidad Autónoma de Baja California (UABC) for their support on my studies.

And finally to all my friends and acquaintances that volunteered to be participants in data collection.

Abstract

Breathing and heart rate are vital signs that might help identifying pathological conditions by its monitoring. This master's thesis presents a system for monitoring breathing and heart rate, which combines conventional Channel State Information sensing approaches with Machine Learning techniques to provide a reliable monitoring. Also, a new sensitive subcarrier selection method, which is an important step for processing Channel State Information data, based on Hilbert Transform is presented. Along with the system's description, this thesis provides the base theory for understanding each system's component and the task that each component does. An exhaustive analysis was also performed and presented in order to understand Channel State Information data as well as the processing of data for vital signs monitoring. Results show that a reliable breathing rate monitoring can be achieved and raise questions about heart rate monitoring which are also answered in the same chapter.

Content

Figures

Tables

1. Introduction

1.1 Problem Statement

In Mexico, 4% of men and the 2% of women of the total population present sleep apnea [1], which can lead to vehicular, work-related or domestic accidents, depressive or anxiety symptoms, memory, attention and concentrations problems and even cardiovascular diseases [2]. In addition, in 2020 of the 218,885 deaths caused by heart-related diseases, the 76.3% of these were caused by ischemic heart disease [3]. Both sleep apnea and ischemic heart are diseases that can be treated if detected in time, as well as other respiratory and heart related diseases; that is the reason why monitoring vital signs of vulnerable people is important: allowing an early detection of adverse events that compromise people's life, and by knowing the scenario of respiratory and heart diseases in Mexico, vital signs monitoring would lead to an opportune medical attention and improving the quality of life of the Mexicans.

Current monitoring systems, such as for fall detection, activity recognition and for monitoring vital signs, use wearable sensors or implement visual-based technologies, such as surveillance cameras, which might result intrusive in daily-life activities. For monitoring with wearable sensors, it is required that the subject wears the sensor, leading to situations in which the subject forgets to wear it or finds it inconvenient for realizing a certain activity. For visual-based solutions, it is required that the subject is in the line-of-sight of the device as well as good lightning and the installation of infrastructure, also they might affect the privacy of the subject [4]. In this investigation Wi-Fi is used for developing a wireless sensing system for monitoring both breathing and heart rate, overcoming the mentioned limitations. Also an exhaustive analysis with signal and data analysis tools is provided for a complete understanding of how Wi-Fi can be used for these vital signs monitoring and for finding the best configuration of the system, considering that current Wi-Fi based vital signs monitoring systems mostly relies on pure signal processing without involving Data Mining or Artificial Intelligence tools and techniques.

1.2 Objectives and Goals

1.2.1 General Objective

Design and develop a system for monitoring breathing and heart rate in controlled environments for people at rest using the Channel State Information of a Wi-Fi signal following a Data Mining methodology for a complete data understanding and definition of system's components.

1.2.2 Specific Objectives

- Define the signal processing steps that help extracting the breathing and heartbeat signals, along with the characteristic features that provide reliable classification results.
- Create machine learning models that are able to classify new observations regardless of people's height and weight.
- Once the models are trained, find the one with the best performance based on different performance metrics.
- Provide an exhaustive analysis for understanding Channel State Information data for breathing and heart rate estimation with signal and data analysis tools.
- Develop a modular system for monitoring both breathing and heart rate.

1.2.3 Goals

- A system for vital sign monitoring available on request.
- Prove the possibility of monitoring vital signs with no need of wearable devices.
- Publish the investigation results in journals and conferences.
- Create a dataset for breathing and heart rate classification available on request.

2. State of the Art

2.1 Overview

Wireless sensing technology has been a trend in recent years. The reason is that it does not require sensor devices to be installed or attached to the target object, that is why it is also known as a device-free sensing technology [5]. Indoor localization, motion detection, activity and gesture recognition, human identification, and even health monitoring are some of the fields of interest of this emerging technology.

One of the most common approaches to wireless sensing are Received Signal Strength Indicator (RSSI)-based mechanisms. The RSSI is an indication of the power level being received by the receiver (usually an antenna) [6], but these mechanisms are more commonly used in human localization and human motion detection. RSSI mechanisms can be very limited to simple environments because of its dramatic performance degradation in complex situations due to multipath fading and temporal dynamics [7], in short terms, RSSI is affected by object movement leading to uncertain detections.

In order to embrace new fields and to surpass the RSSI limitations, a new wireless sensing technology appeared. Channel State Information (CSI) measurements capture how wireless signals travel through the environment in time, frequency, and spatial domains, so it can be used for numerous sensing applications. Wi-Fi uses Multiple-Input Multiple-Output (MIMO) technology and uses Orthogonal Frequency-Division Multiplexing (OFDM) signal modulation technique, obtaining measurements per subcarrier generated by OFDM from each packet, while RSSI is measured by a single value per packet [8, 9].

CSI characterizes how wireless signals propagate from the transmitter to the receiver at certain carrier frequencies. Its amplitude and phase are affected by multi-path effects including amplitude attenuation and phase shift and if assisted with mathematical modeling and machine learning techniques, it can be used for different sensing applications [8].

In this chapter, related work exploring wireless sensing using CSI of a Wi-Fi signal will be exposed categorized by application, identifying the technology used, experiment methodology, Data Mining techniques implemented and the result of each investigation.

2.2 CSI for Indoor Localization

Indoor Localization is the process of obtaining a device or user location in an indoor environment. For this section, some CSI based approaches for Indoor Localization are listed, having the advantage to be device free from a user perspective.

Fingerprinting-based localization is an Indoor Localization method that consists of two basic phases: 1) Offline phase, also called training phase, where the database for facilitating real-time position estimation is constructed based on reference points and 2) Online phase, where the position of the device or user position is being estimated by searching each reference point to find the most closely matched spot as the target location [10].

Wu et al. [11] designed FILA, the first to use CSI to build a propagation model to improve Indoor Localization performance compared to RSSI-based approach. For fingerprinting, they leverage the CSI values including different amplitudes and phases at multiple propagation paths, known as the frequency diversity, to uniquely manifest a location. As experimental results, they used commercial 802.11n NICs and conducted extensive experiments in typical indoor environments, obtaining that FILA significantly improves the localization accuracy and speed of distance calculation as compared to traditional RSSI-based approach.

Wang et al. [10] proposed a deep-learning-based fingerprinting scheme named DeepFi based on CSI values collected from three antennas of an Intel 5300 NIC. For their experiments, they used a TP-Link router as an Access Point (AP) and as the mobile device a Dell laptop with the Intel 5300 NIC and tested their scheme in two typical Indoor Localization environments, a living room in a house and a computer laboratory. In both environments they implemented other three CSI-based fingerprinting methods to compare their performance with DeepFi. As result, they obtained that DeepFi achieved higher accuracy even in a non-LoS (Line-of-sight) environment, outperforming the other CSI-based methods.

Zhou et al. [12] treat the wireless localization as a regression problem, establishing relations between locations and CSI fingerprints using empirical data, proposing to apply Support Vector Machines (SVM) for classification to achieve human presence detection and SVM for regression to achieve localization. For CSI data preprocessing they apply Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Principal Component Analysis (PCA) to reduce the computing complexity by extracting the most contributing features from the CSI data, and normalization to enhance the precision and convergence before human presence detection. For the experimentation process, they employed a Netgear N300 wireless router and a Cisco RV180W wireless router as transmitters, with two computers with Intel 5300 NIC for their bigger environment and only both computers for a smaller environment. The evaluations prove the effectiveness of SVM and data preprocessing in human presence and localization applications, achieving a localization accuracy of 1.22m (mean error distance) for the bigger environment and 1.39m for the smaller one, and presence detection precision higher than 97% for both scenarios.

Sanam and Godrich [13] proposed a novel algorithm that adopts a threshold based subcarrier selection scheme from each MIMO link, selecting only the ones which the decrease in CSI amplitude is larger than the threshold based on the fact that these selected subcarriers conform to the diffraction fading model. Applying a preprocessing phase and PCA with SVM supervised learning model for classification on CSI data collected in a research laboratory, equipped with typical office facilities, with help of a TL-WR940N wireless router as an AP, a mobile device with Intel 5300 NIC as data collector and a host PC that serves as the centralized server for location estimation, they achieved a localization accuracy that can be extended to 98.27%.

Hoang et al. [14] proposed a combined structure between Convolutional Neural Network (CNN) and long-short term memory (LSTM) for Indoor Localization, taking information from previous time steps in user's trajectory to determine the actual location. They collected CSI data with a laptop with Intel 5300 NIC and a Nexus 5 smartphone with only one single AP and with an extensive preprocessing phase and hundreds of testing locations, they demonstrated that the CNN-LSTM structure achieves an average localization error of 2.5m with 80% of the errors under 4m, which outperforms other Indoor Localization algorithms.

2.3 CSI for Human Activity and Gesture Recognition

Human activity recognition (HAR) is of great importance for healthcare services, context awareness and even for building control systems to provide a comfortable indoor environment with high energy efficiency [15]. Conventional HAR systems require that the user to be analyzed carries a specialized device that may intrude in his daily activities. So this is where CSI of a Wi-Fi signal comes in handy.

It is proved that human activities between Wi-Fi receivers and transmitters influence in signal characteristics. This influences can be seen analyzing the CSI data from the Wi-Fi signal and can be characterized in order to do activity recognition.

Therefore, investigations employing CSI-based approaches have been done.

Wang et al. [16] developed a monitoring framework, E-eyes, that compares amplitude profiles against those from known to recognize stationary activities like cooking, sleeping or watching television where only small body movements are involved. A walking activity causes significant pattern changes of the CSI amplitude over time, so a moving variance thresholding technique is used to discriminate stationary activities from walking activity. To construct activity profiles, they used a semi-supervised approach, where a clustering-based method is implemented to identify multiple similar instances of an activity without a matching profile and then the user is the one who labels the resulting clusters. This technique can be also used to detect and update activity profiles when a change in the environment occurs. For the experimental setup, a Linksys E2500 was used as an AP and three laptops equipped with Intel 5300 NIC as the monitoring points. As result, E-eyes proved that it has potential to support emerging applications such as elder care due to its high accuracy to distinguish between stationary activities and walking when a single occupant is at home.

Tan and Yang [17] developed WiFinger, a gesture recognition system which runs on a single WiFi device connected to one AP. WiFinger is capable of identifying typical finger gestures including zoom in/out, circle left/right, swipe left/right, and flip up/down being robust to environment changes like moving furniture or someone walking and individual diversity. This is achieved by employing Multipath Mitigation and Wavelet Based Denoising. After the environmental noise removal, WiFinger recognizes gestures by calculating the similarity between the extracted CSI pattern and the constructed gesture profiles by utilizing a supervised or semisupervised approach. By employing these methods, WiFinger achieves an overall recognition accuracy over 93% even when there are environmental changes.

Wang, Wu et al. [18] proposed a second version of a system named WiFall [19], which is a passive device-free fall detection system that uses CSI amplitude for activity classification (walking, sitting down, standing up) in order to detect falls. WiFall system consists of three phases: sensing phase, where the transmitter propagates wireless signals and the receiver collects the CSI information, learning phase, where data processing, profile construction and activity decision occurs along with the classification algorithms, SVM and Random Forest, and the alerting phase, where an emergency alarm is triggered when a fall is detected. The WiFall system implements two TP-LINK TL-WDR7500 wireless routers as AP, and two desktops equipped with Intel 5300 NIC as monitoring points, which are also the servers that conduct data processing, profile construction and activity decision. As a result, they obtained that the new version of WiFall can achieve better performance with SVM and Random Forest classifier with Singular Value Decomposition (SVD) matrix factorization reaching a precision of 98% for falling detection rate.

A key limitation of prior Wi-Fi based activity or gesture recognition systems is that they can only recognize a user's activity if only a single user moves in the environment. Venkatnarayan et al. [20] designed a Wi-Fi based gesture recognition system, WiMU, which can recognize gestures of multiple users even when they perform them simultaneously with a novel method that generated virtual samples for any desired combination of gestures using a real sample of each gesture in that combination. WiMu demonstrated that it can recognize 2,3,4,5 and 6 simultaneously performed gestures (open and close door, circular arm movement, push and pull armn, sit down and stand up, kicking, and brushing teeth) with average accuracies of 95, 94.6, 93.6, 92.6, and 90.9% respectively.

Narui et al. [21] developed a system for human fall detection combining CSI and machine learning using a 1-dimensional CNN with domain adaptation to overcome covariate shift (situation where input data distribution is different between the training and the test data, but conditional distribution is the same), marking in their dataset "Fall" as an anomaly activity and "Bed", "Walk", "Run", "Stand up", "Pick up" and "Sit down" as normal. All experiments used 1 antenna transmitter and 3 antenna receivers which are equipped with Intel NIC 5300 and as result, they obtained a precision and recall of 100 for anomaly detection, with a better match in class distribution compared to other domain adaptation techniques.

Chen et al. [15] proposed an attention based bi-directional long short-term memory (ABLSTM) approach for automatic feature learning and selection in the task of HAR with Wi-Fi CSI measurements which can identify if the person is lying down, falling, walking, running, sat down, or standing up in two different environments, an activity room and a meeting room. They used a commercial Wi-Fi router as a transmitter and a laptop with Intel 5300 NIC for the experimentation process and compared their results with some other CSI benchmark approaches. They used a 10 fold cross-validation for evaluation, randomly dividing all the data into 10 folds, one fold for testing and the remaining for training. As result, ABLSTM demonstrate a superior performance than the other approaches in their experiments.

Damodaran et al. [4] implemented two different algorithms in their investigation. The first one combines discrete wavelet transform (DWT), PCA, power spectral density (PSD) and frequency of center of energy and Haart wavelet analysis to extract the lower frequency bins using a SVM as classifier. For the second algorithm they only used LSTM that operates on raw data and use only denoising via DWT. For their experimentation phase, they implemented both algorithms in HAR, specifically the activities of walk, run, sit, stand and empty, collecting the CSI data with two laptops equipped with Intel 5300 NIC with the help of

Linux 802.11n CSI Tool. As result, they obtained that both algorithms have problems differentiating between similar activities (like stand and sit, where there is no movement) and besides all the preprocessing phase done in the first algorithm it was only slightly better than the second one. As an extension, they added Fall as an activity without any additional hardware but a second SVM was implemented in order to obtain the *Post Fall* information (if the person stills lying or he stood up). They noticed that during the sit down activity, the person sits down and keeps sitting, which makes it similar to fall and keep lying so the second SVM used to give false positives.

2.4 CSI for Human Identification

Human Identification is one of the main problems that device-free sensing technologies are trying to solve, due to its important applications in smart homes, indoor intrusion detection, monitoring, and other security-related applications. CSI based approaches have proved a decent performance with no invasive methods and with Wi-Fi common devices, however, this approaches are still under an investigation phase in order to improve even more their performance, searching to reach the highest identification accuracy possible with different Data Mining techniques and preprocessing methods.

Zeng et al. [22] developed WiWho, a framework to identify a person based on a combination of step and walk analysis in small places. Step analysis requires step cycle construction and calculation of time domain features for each step, while walk analysis focuses in the overall walking behavior. Features from both analysis are used in a decision tree-based classifier, which outputs the person's identity. Some of the limitations of WiWho are that it assumes that person's walking path is a straight line and that it is designed and evaluated for a single person in the room at any given time. For performance evaluation, they used an Asus RT-AC66U 802.11n Wi-Fi router as an AP and a laptop equipped with an Intel 5300 NIC as a client. They observed that if more people is introduced in person identification, chances are increased of people having similar gait, resulting in an accuracy decrease when the group size increases, achieving an identification accuracy of 92% to 80% for a group size of 2 to 6 respectively, where only a 2-3 meters walking length is necessary.

Xin et al. [23] proposed FreeSense to identify human indoors based on CSI in their line-of-sight (LoS) path crossing moments between the transmitter and receiver using k-nearest neighbor (K-NN) classifier, where the recognition is based on the difference of personal movement influence to the Wi-Fi signal by adopting the Dynamic Time Warping (DTW) as the distance metric between the learning signal samples and the one to identify. Using a laptop equipped with Intel 5300

NIC and a TP-LINK TLWR1043ND Wi-Fi router as an AP, FreeSense achieved more than 88% accuracy with 6 different subjects. As same as WiWho, FreeSense is limited to identify only if walking across the LoS path in straight line and with only one person in the room.

Nipu et al. [24] used Boosted Decision Tree and Random Forest supervised classifiers for human identification based only in each person's gait pattern, utilizing skewness, mean, maximum, kurtosis, median, energy and highest fast Fourier transformation peaks as features; they obtained an accuracy of 84% for Boosted Decision Tree classifier and 78% with Random Forest classifier in a group of 5 people. A desktop PC equipped Intel 5300 NIC was used for their experimentation as the receiver and a TP-LINK TL-WR740N as the transmitter.

Wang, Zhao et al. [25] proposed a framework that decomposes raw noisy CSI measurements into intrinsic mode functions (IMF) and extracts multi-domain intrinsic features from them, also it uses PCA to acquire the principal components of the CSI measurements. This framework works in two identification systems, one using breathing pattern and the other one gait biometrics, both utilize softmax regression algorithm [26] to solve the identification problem. For both systems, the transmitter and receiver are PCs equipped with Intel 5300 NIC running CSI Tool on Ubuntu, and as result of their experimentation, the breathing pattern based system obtained an identification accuracy of 97.5% in no changing environments and the gait system 90.4%. For changing environments, the identification accuracy was of 91% and 90% respectively.

Wang, Han et al. [27] proposed CSI-Net, a deep learning framework based on CNNs. To achieve human identification, they trained CSI-Net with 2 separate task stages, one for biometrics estimations: body fat rate, muscle rate, water rate and bone rate; and another for human identification based on the same inputs for biometric estimation. Therefore, based on the internal composition obtained for biometrics estimation, CSI-Net identifies the person. As result, they obtained that CSI-Net can estimate biometrics accurately for every subject, achieving almost 100% accuracy in person recognition on most of the subjects.

Ding et al. [28] proposed Wihi, a human identity identification system that uses a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) blocks to identify different people by learning the gait features extracted: channel power distribution in time domain, time-frequency analysis, energy distribution in different frequency bands, maximum, minimum, mean, variance, standard deviation, median, energy, and entropy. Discrete wavelet transform (DWT) was used to eliminate the influence of the random noise in raw CSI. For their experimentation, they used a commercial WiFi device with one antenna as the transmitter and a laptop equipped with an Intel 5300 NIC as the receiver, obtaining an accuracy of 85% and 73% with group sizes of 2 and 8 respectively with 4 unrelated people performing random actions at the same room. Also they analyzed the impact of different walking paths, number of hidden nodes and walking speed on the resulting identification accuracy.

2.5 CSI for Vital Sign Monitoring

Breathing and heart rate are vital signs that indicate the current state of a human body, its health or even help to identify serious diseases. Recent approaches use CSI of a Wi-Fi signal to monitor these vital signs, proving that even minute movements, like chest movements caused by inhalation and exhalation process or heart systole and diastole, can generate some influence in the CSI without the need of using specialized instruments for sensing.

Liu et al. [29] proposed a system that estimates breathing and heart rate of one or two persons in a bed scenario, taking as input CSI amplitude measurements during people's sleep, being able to detect and classify also sleep events like turnovers. A Hampel filter and a moving average filter are used to remove environmental noise, therefore CSI amplitude can clearly reflect the periodic up-and-down chest movements generated by breathing. To select the subcarriers that are going to be used to estimate breathing rate, variance of CSI amplitude is used to quantify each subcarrier's sensitivity to minute movements, then breathing rate is estimated finding the peaks in each subcarrier that represent the breathing movements by applying a peak identification algorithm that reduce the number of fake peaks, constructing a breathing cycle and crosses all estimated breathing cycles found in the selected subcarriers to enhance accuracy. Heart rate is estimated by locating the maximum power in the average power spectral density (PSD) of all subcarriers in the normal heart rate range. For experimentation, they used a laptop equipped with an Intel 5300 NIC connected to a commercial wireless AP, obtaining that the system can achieve comparable or even better performance as compared to existing dedicated sensor based approaches. Same authors in [30] extend the system to also identify sleep postures by using time domain features like mean, maximum, minimum, variance, skewness, range, mode, median and kurtosis on each subcarrier, having a total of 270 features, therefore Fisher scores were calculated in order to only use the features having Fisher scores higher than a threshold. Also, PCA was applied in order to reduce the computational cost, converting the selected features into 20 linearly uncorrelated principal components. For posture identification, four sleep profiles where constructed from CSI features and then, they were used to train a machine learning classifiers: discriminant analysis, K-NN, SVM and Random Forest, obtaining accuracies over 90% for sleep posture identification with K-NN, SVM and Random Forest.

Wang, Zhang et al. [31] introduced the Fresnel zone model in a CSI breathing rate monitor in order to analyze the impact of body location in different Fresnel zones and how body orientation affects the breathing rate detection. Deploying a prototype system which consists of a mini PC equipped with a Intel 5300 NIC as a monitoring point and a TP-Link router as an AP, they obtained that CSI data when a subject is inside the 2_{nd}. Fresnel zone is consistent and sharp enough to be used for detection, but it starts fading when the subject moves towards the boundary. Result shows that the system is not able to detect the subject's breathing at 2 meters beyond the LoS along its perpendicular bisector. It was also observed that the body orientation of a subject while he remains at the same location affects the signal pattern, obtaining that chest movements are best captured when the subject is facing the LoS.

Shang and Wu [32] designed WiHealth, a vital signs monitoring system which uses median filter and a low-pass filter for denoising raw CSI waveform. For extracting breathing and heart rate signals, they did a spectrum analysis applying two band-pass filters with frequency bands of 0-1Hz for breathing and 1Hz-2Hz for heart beat, and by using a threshold based method with the mean squared error with multiple peaks in the spectrum, they eliminated false peaks that can lead to inaccurate estimations. For breathing and heart rate estimation, a polynomial filter was applied to smooth the CSI waveform and then a traditional peak finding algorithm was used to get the count. Using a laptop equipped with an Intel 5300 NIC with 3 antennas as the receiver and a TP-Link TL-WR1043ND router as the transmitter in a convetional meeting room, they obtained that WiHealth can estimate breathing and heart rate accurately enough with average estimation error under 0.6 breaths per minute and 6 beats per minute respectively.

Wang, Yang and Mao [33] developed PhaseBeat, a sensing system that uses CSI phase difference data for monitoring respiration and heartbeat. PhaseBeat is able to estimate breathing and heart rate when a subject is in a stationary state (sitting, standing, or sleeping). By removing environmental noise, such as the direct current component with Hampel Filter and downsampling, PhaseBeat obtains a sinusoidal-like periodic signal from each subcarrier, selecting one for further processing according to its sensitivity. Once selected, DWT is applied to the subcarrier to remove high frequency noises from the collected CSI phase difference data; the coefficients obtained from DWT are used to detect the breathing rate and the heart rate. Their experimental results obtained by using two desktop computers equipped with Intel 5300 NIC showed that an accuracy of 98% can be obtained for breathing rate estimation as well as an accuracy of 95% for heart rate estimation with a sampling rate of 400 Hz. For breathing rate, 90% of the test data have an estimated error under 0.5 breaths per minute while for heart rate an 80% of the test data have an estimated error under 2.5 beats per minute. In 2020, the same authors developed Resilient realtime breathing Beat

(ResBeat) system [34] which exploits bi-modal CSI, amplitude and phase difference for realtime breathing rate monitoring. This time, for obtaining the environment component and breathing component, they used Exponentially Weighted Moving Average (EWMA) method since it does not required a large data buffer and is more sensitive to more recent sample data being suitable for realtime monitoring. ResBeat uses peak detection with a sliding window based method to mitigate the effect of fake peaks for breathing rate monitoring in real time and also, it incorporates an adaptive signal selection method to always select the most sensitive data group. As result, they obtained that ResBeat outperformed PhaseBeat due to bi-modal CSI data approach and the adaptive signal selection method.

Zhang et al. [35] analyzed the capacity of detecting breathing inside the First Fresnel Zone (FFZ) considering not only the front-side displacement of the chest caused by breathing but also the back-side displacement, applying the diffraction theory inside the FFZ to provide general theoretical foundation for CSI sensing. By using two mini PC equipped with Intel 5300 NIC as transceivers to validate their theory for RF-based human breathing sensing they obtained an accuracy for breathing rate estimation above 95% with no need of signal denoising or processing techniques while the person is lying between the two pair of transceivers and in the FFZ. Also they carried out experiments to observe the impact of subject's position inside FFZ, breathing depth, body thickness, subject's distance to transceivers and body orientation on signal variations that are used for breathing rate estimation.

Khamis et al. [36] developed CardioFi, a heart rate monitor which uses a subcarrier ranking scheme based on Spectral Stability score. For preprocessing, a Hampel filter is applied to each subcarrier in order to remove outliers, after that, they proposed to apply a de-trending method called Dynamic Window, which determines the window size adaptively. After de-trending, another Hampel filter is applied. For heart rate estimation, they fuse data from the most informative subcarriers by calculating the mean power spectral density spectrum across the subcarriers; the frequency component with the largest magnitude is the instantaneous heart rate. Using two laptops equipped with Intel 5300 NIC they obtained a median error of 1.14 bpm and 90% of the errors are below 5.1 bpm.

Zhang, Hu et al. [37] developed BreathTrack, a breath tracking system that utilizes both hardware and software approaches to enhance breathing rate estimation. BreathTrack exploits phase variations of the CSI to track human breath and proposes a joint angle of arrival-time of flight sparse recovery method to eliminate the multipath effect and extract the information of the dominant path. By using two desktop computers equipped with Intel 5300 NIC, they obtained an accuracy for breathing rate estimation above 99% for both LoS and NLoS scenarios.

3. Theoretical Framework

3.1 Channel State Information

Wi-Fi uses Multiple-Input Multiple-Output (MIMO) technology for meeting the demands of current networks, increasing a system's throughput by transmitting multiple data streams by exploiting or combating multi-path fading effects of a wireless channel. This can not be done without CSI, as it allows to adapt transmissions according to the current channel conditions [38]. For constructing CSI, the transmitter sends pre-defined symbols, known as Long Training Symbols (LTFs) for each subcarrier generated by Orthogonal Frequency-Division Multiplexing (OFDM) [39], which is the modulation technique used for Wi-Fi standards 802.11a, 802.11n, and 802.11ac, in each packet preamble [8]. This allows the receiver to estimate channel characteristics, an operation known as channel estimation, which results in the obtainment of CSI.

A simple way of defining CSI is as a collection H of $M \times N$ matrices, where M is the number of receiver antennas and N is the number of transmitter antennas, in which each matrix describes how the signal propagates in the channel by combining the effects of amplitude attenuation and phase shift for a single subcarrier [40].

A graphical representation of a CSI collection with $M = 3$, $N = 3$ and k subcarriers given a packet index i is shown in 3.1.

Each CSI entry H_i for a subcarrier k represents the Channel Frequency Response (CFR), a complex value defined as:

$$H_{\mathrm{i,k}} = |H_{\mathrm{i}}| e^{j\angle H_{\mathrm{i,k}}} \tag{3.1}$$

where $|H_{\mathrm{i}}|$ is the amplitude response and $\angle H_{\mathrm{i,k}}$ represents the phase shift response. These effects can be seen as a way of describing how the signal behaves in the presence of surrounding objects and even of a person. Therefore if a person moves inside the range of the signal, an specific signal behavior is expected according to the characteristics of the movement. By applying signal processing

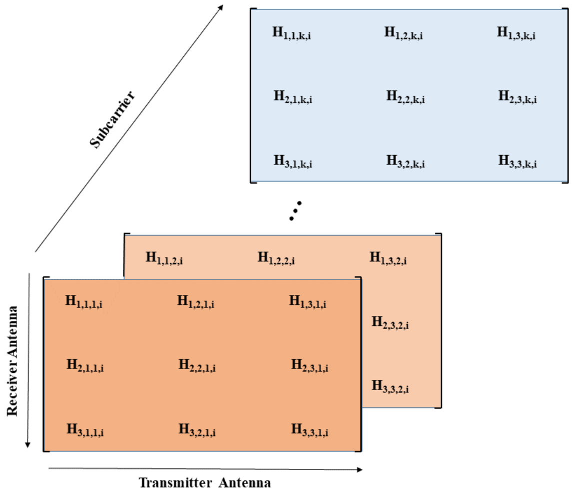


Figure 3.1: CSI collection representation

techniques, Data Mining, and Artificial Intelligence tools and techniques to these signal behaviors, wireless sensing can be achieved.

3.2 CSI Monitoring Tools

Although CSI is included since IEEE 802.11n [8], it is not shown to the user by all Wi-Fi devices. Therefore a CSI collecting tool is needed, each one with its own equipment requirement.

A) Linux 802.11n CSI Tool

Linux 802.11n CSI Tool is probably the most known CSI monitoring tool for device free sensing applications. Developed by Halperin et al. [41], this CSI tool is built on the Intel 5300 NIC and uses a custom firmware and open source Linux wireless drivers. Each CSI matrix entry has a signed 8-bit resolution where amplitude and phase shift are located. This firmware enables an Intel debug mode that records CSI from up to 30 subcarrier groups for each correctly received 802.11n packet and sends it up to the kernel driver to finally be passed to a user-space program for processing.

B) Atheros CSI Tool

Atheros CSI Tool is an open source 802.11n monitor developed by Xie and Li [42] with support to all types of Atheros 802.11 Wi-Fi chipsets. Unlike Linux 802.11n CSI Tool, it does not apply any modification to the firmware, all functionalities are implemented in software, the CSI matrix entry has a 10-bit resolution each for amplitude and phase shift. The Atheros CSI Tool is able to obtained CSI from up to 56 subcarriers for a 20 MHz channel bandwidth and 114 for a 40 MHz channel bandwidth.

C) ESP32 CSI Toolkit

Applications using the previous mentioned monitoring tools have obtained satisfactory results, proving their effectiveness, but they have two notable limitations: 1) They need one or more laptop or desktop computer equipped with an Intel 5300 NIC for Linux 802.11n CSI Tool or a compatible Atheros NIC for Atheros CSI Tool to act as receivers. This resource limitation may impact on the deployment of a large-scale system where it is needed to use low cost equipment or space is a limitation; 2) For Intel 5300 NIC, only 30 out of 52 data subcarriers can be obtained[41, 43].

ESP32 CSI Toolkit, the monitoring tool developed by Hernandez [44] uses ESP32 microcontroller units (MCU) [45] which can act as both an AP or a station under TCP and 802.11 b/g/n/e/i Wireless Local Area Network (WLAN) Medium Access Control (MAC) protocol and is able to connect to most Wi-Fi routers. In ESP32 CSI Toolkit there are two ways to obtain CSI data: 1) a SD card on board can be used, the ESP32 will automatically detect the SD card and automatically output CSI data to a CSV file or 2) CSI data can be easily collected from serial port. ESP32 CSI Toolkit can obtain CSI amplitude and phase shift from 64 subcarriers and by using a MCU, ESP32 CSI Toolkit results in a flexible, low cost and easy to deploy tool in device free sensing applications.

3.3 CSI Data Calibration

In order to use CSI for any device free sensing application first it needs to be denoised due to the fact that CSI comes with inherent noise from equipment and environmental factors. For this investigation focused on breathing and heart rate estimation, noise coming from other body movements, sudden environmental changes and electromagnetic noise needs to be mitigated.

Table 3.1 shows signal processing methods implemented in related work about monitoring vital signs using CSI; as it can be seen, some of the most used are Hampel Filters for outlier removal, Bandpass Filters to remove undesired frequencies, and Moving Average Filters for signal smoothing. The step of applying these filters is commonly named Data Calibration.

Once data is calibrated, it is highly recommended to implement a subcarrier selection method to reduce computing complexity, which needs to be considered when working with embedded devices or if a real-time functioning is intended. Table 3.1 also shows common subcarrier selection methods, where the most used is based in an amplitude variance threshold.

This section will be focused in introducing the data calibration methods previously mentioned as well as the variance-based subcarrier selection method.

Table 3.1: CSI Data Calibration Methods

Reference	**Variable**		**Calibration Method**										**Subcarrier Selection**				
	BR	HR	Hampel Filter	Highpass Filter	Lowpass Filter	Bandpass Filter	Moving Average Filter	Median Filter	Savitzky-Golay Filter	Down sampling	DWT	IFFT	Max Var.	Spectral Stability	Waveform Analysis	Mean Abs. Dev.	Sparse Recovery
[29]	✓	✓	✓				✓						✓				
[31]	✓		✓				✓						✓				
[32]	✓	✓			✓	✓		✓							✓		
[33]	✓	✓	✓							✓	✓					✓	
[46]	✓								✓				✓				
[47]	✓	✓				✓						✓	✓				
[36]		✓	✓			✓								✓			
[30]	✓	✓	✓				✓						✓				
[37]	✓		✓	✓													✓
[48]	✓	✓	✓			✓							✓				

3.3.1 Digital Filters

For calibrating CSI data it is common to rely on signal processing tools such as digital filters. Digital filters work with discrete signals and have the function of applying a mathematical processing to these kind of signals. They are composed of adders, multipliers and delay elements (also known as shifters or memory). There are two different types of filters, Finite Impulse Response or FIR and Infinite Impulse Response Filters or IIR. FIR filters explanation is straightforward, for an input which is finite in duration, its response is also finite and it involves only feed-forward calculations. For IIR filters the output is used as feedback for the filter, therefore an output in a certain time depends partially on previous outputs [49]. For explaining IIR filters see Fig. 3.2, supposed it has as an input a signal $x[n] = [0, 1, 0, 0, 0]$, its output would be $y[n] = [0, 0.5, 0.5, 0.5, ...]$ and so on while x[n] = 0.

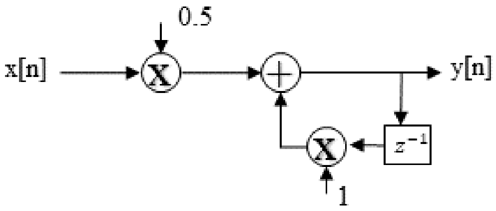


Figure 3.2: IIR Filter Example

The values 0.5 and 1 observed in Fig. 3.2 associated to an adder or multiplier are known as filter coefficients. The number of filter elements and coefficients are determined according to the desired filter function and, according to the treatment of the filter in the frequency domain of an input signal these can be classified as low-pass, high-pass, band-pass and band-stop filters [50]. For the development of this investigation only band-pass and low-pass filters were implemented.

Low-pass Filter

A low-pass filter is a type of filter that allows all frequencies until a cut-off frequency f_c. At f_c, the filter response curve presents an attenuation of -3 dB in respect to the magnitude presented in the passband region of the filter. This attenuation continues for all other frequencies above f_c [51]. In Fig. 3.3 a low-pass filter with a f_c of 4 Hz is presented, where can be seen that at this point, the magnitude of the response curve for the low-pass filter is of -3 dB.

For this investigation, two different filters explored show a low-pass filter behavior and in investigations such as in [32] used a simple low-pass filter to remove high frequency noise which can not be caused by chest movement to obtain the breathing signal.

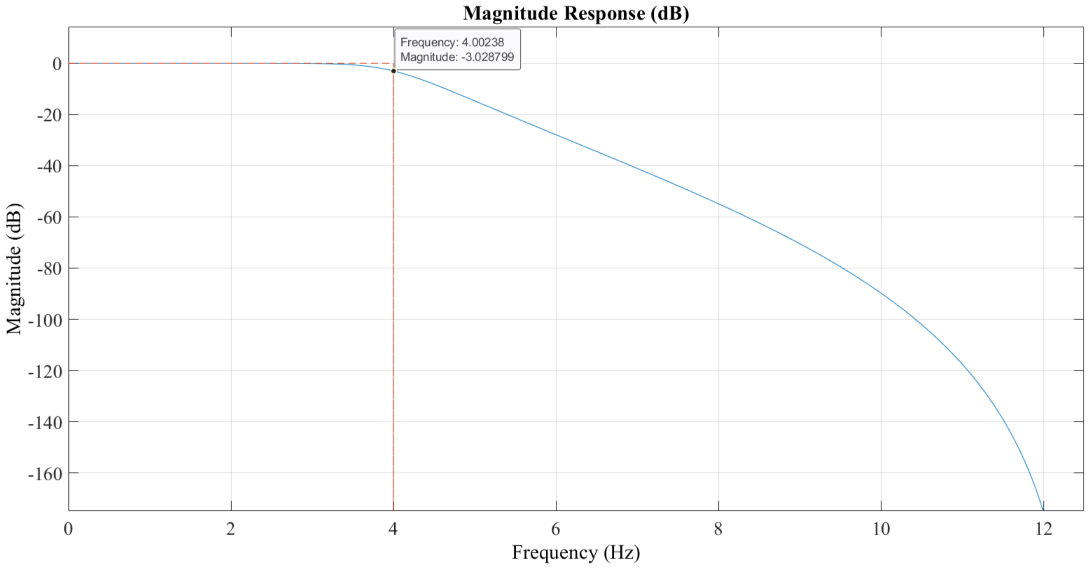


Figure 3.3: Example of frequency response of a low-pass filter (values for cut-off frequency and magnitude are approximated due to software resolution)

Band-pass Filter

A band-pass filter allows all signal frequencies between a low-cut-off frequency f_{c1} and a high-cut-off frequency f_{c2}, rejecting all other frequencies out of this interval or band. The cut-off frequencies defined are points where the response curve is at 70.7% (or -3 dB) of its maximum response [51]. Figure 3.4 shows a filter frequency response with f_{c1} and f_{c2} of 4 and 8 Hz respectively and as can be observed, the response curve at this points has a magnitude of -3 dB as stated before. The region of the filter inside the marked rectangle is called the band-width of the filter and can be defined as the difference between the cut-off frequency points.

An adult normal breathing rate at rest is between 12 and 20 breaths per minute, therefore a band-pass filter with cut-off frequencies of 0.2 Hz and 0.33 Hz would help obtaining a signal which can be assumed to represent the movement caused by breathing, this signal will be further on referred as the breathing signal. For heart rate, the normal range for an adult at rest is between 60 and 100 beats per minute, therefore a band-pass filter with cut-off frequencies of 1 Hz and 2 Hz would be able to isolate the frequency range in which heart rate can be possible found and will be further on referred in this investigation as the heartbeat signal.

Moving Average Filter

The moving average filter (MAF) can be seen as a simple low-pass filter, used for smoothing a signal reducing random noise while maintaining a sharp step response.

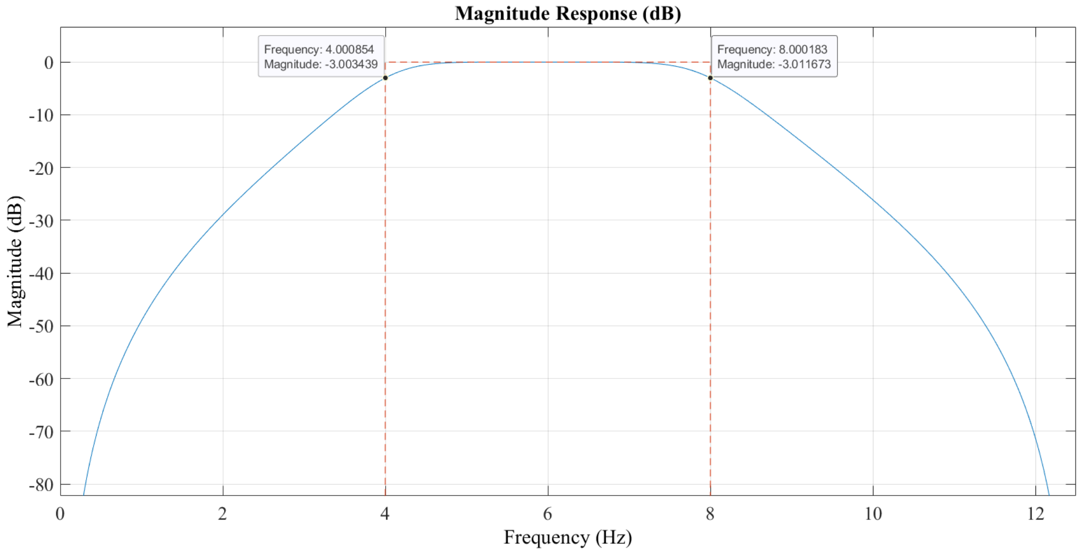


Figure 3.4: Example of frequency response of a band-pass filter (values for cut-off frequencies and magnitudes are approximated due to software resolution)

The MAF takes the current and previous $M-1$ samples, which is called a *window*, of the input signal x and calculates the average of these M samples to produce each point in the output signal $y[n]$ [52], defined as:

$$y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k] \tag{3.2}$$

Fig. 3.5 shows a sinusoidal wave signal with random noise added and the result of applying MAF. As can be seen, after filtering, the original signal can be recovered.

Besides its simple implementation, it has two principal characteristics to be considered:

1. Due the fact it requires previous values of $x[n]$ for producing $y[n]$ at a given time n, MAF accumulates a time delay which increases as the window size M increases. An alternative for reducing the time delay is using a centered moving average filter, symmetrically choosing values around $x[n]$, but this involves having values ahead of the current time n.

2. As stated before, MAF acts as a low-pass filter, but it has a poor frequency response, which can be improved increasing the window size, but this will lead to the delay mentioned before.

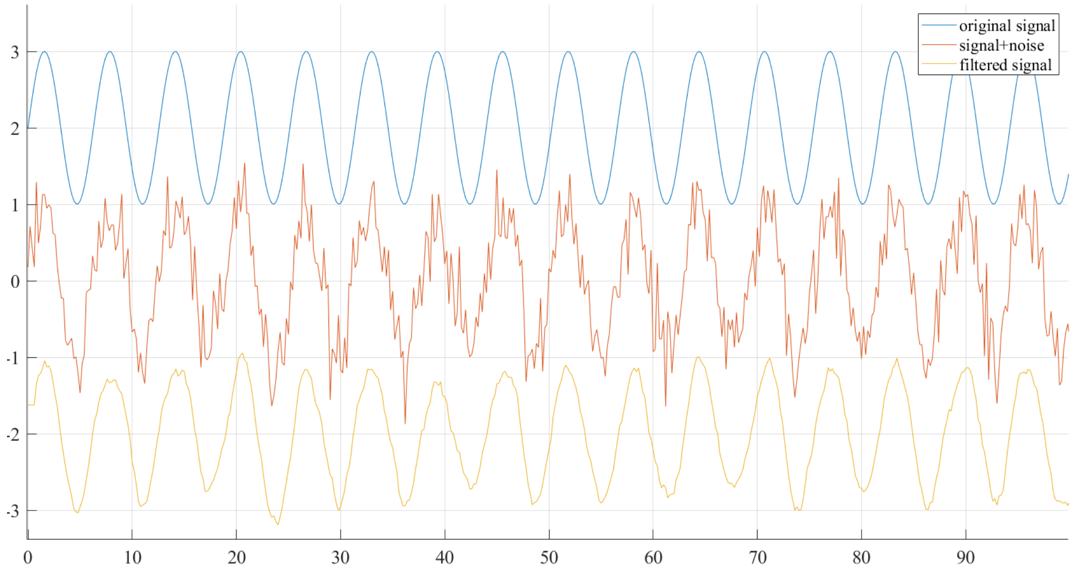


Figure 3.5: Moving average filter on a sinusoidal wave signal with random noise added

Applying MAF to CSI subcarriers for vital sign monitoring would help removing high frequency noise where breathing and heart rate are unlikely to be found, but if concerned about the time delay or with the frequency response that this filter will have, an alternative can be a Savitzky-Golay Filter.

Savitzky-Golay Filters

Savitzky-Golay (SG) filters are lowpass filters, which smooth data based on local least-squares polynomial approximation, reducing noise while maintaining the shape and height of waveform peaks. Suppose having a signal f represented as follows:

$$f(-n)..., f(-2), f(-1), f(0), f(1), f(2), ...f(n)$$

for each signal time step i, $f(i)$ is replaced by the value $\hat{f}(i)$, obtained by fitting a polynomial to f on the time steps between $i-k$ and $i+k$. This example works well for understanding the main idea of SG filters, but in practice, least-squares polynomial fitting is achieved by convolution.

The main task of this smoothing method is to find coefficients $a_0, ..., a_d$ such that the polynomial of degree d:

$$p(x) = \sum_{i=0}^{d} a_i x^i \tag{3.3}$$

minimizes the mean-squared approximation error for the group of input time steps centered on x:

$$\sum_{x=-k}^{k} (p(x) - f(x))^2 \tag{3.4}$$

being k the half width of the approximation interval (also called window length). From equation 3.4 a matrix A of dimensions $(2k+1)X(d+1)$, called the design matrix can be obtained. The matrix product Aa, where $a = [a_0, a_1, ..., a_d]^T$, gives a column vector of values for the polynomial p. Being $y = [f(-k), ..., (f(k))]^T$ the vector of input samples, the problem is reduced to find a vector a that minimizes $(Aa - y)^2$, which solution can be written as

$$a = (A^T A)^{-1} A^T y = Hy \tag{3.5}$$

The H matrix depends only on k and d and is independent of the input samples, therefore the same coefficients will be obtained from each approximation interval of input samples, and consequently, least-squares smoothing can be seen as a shift-invariant discrete convolution process [53].

Figure 3.6 shows a sinusoidal wave with random Gaussian noise added and the result of applying three different SG filters with different polynomial degree and a fixed window length. It can be seen that the SG filters success on filtering the high frequency noise, but the SG filter with $d = 3$ seems to do better than $d = 11$, this is because the polynomial degree affects the slope of the frequency response of the filter; a higher polynomial degree increases the cut-off frequency. This effect can be seen in Figure 3.7.

Another parameter that affects the frequency response of SG filters is the window length (N). Unlike polynomial degree, a higher window length decreases the cut-off frequency as seen in Figure 3.8.

Hampel Filter

As stated before, the Hampel filter is used for outlier removal. It identifies and replaces outliers based on the median and the median absolute deviation (MAD) [54].

Considering a configurable-width sliding window of time series data, for each window, the median x_m of the contained data x_k is calculated and the MAD scale estimator is defined as:

$$S = k * median(|x_{k1} - x_m|, |x_{k2} - x_m|, ..., |x_{kn} - x_m|) \tag{3.6}$$

where $k = 1.4826$, which makes S equal to the standard deviation σ for normally distributed data. A certain value x$_i$ will be defined as an outlier if:

$$|x_i - x_m| > tS_k \tag{3.7}$$

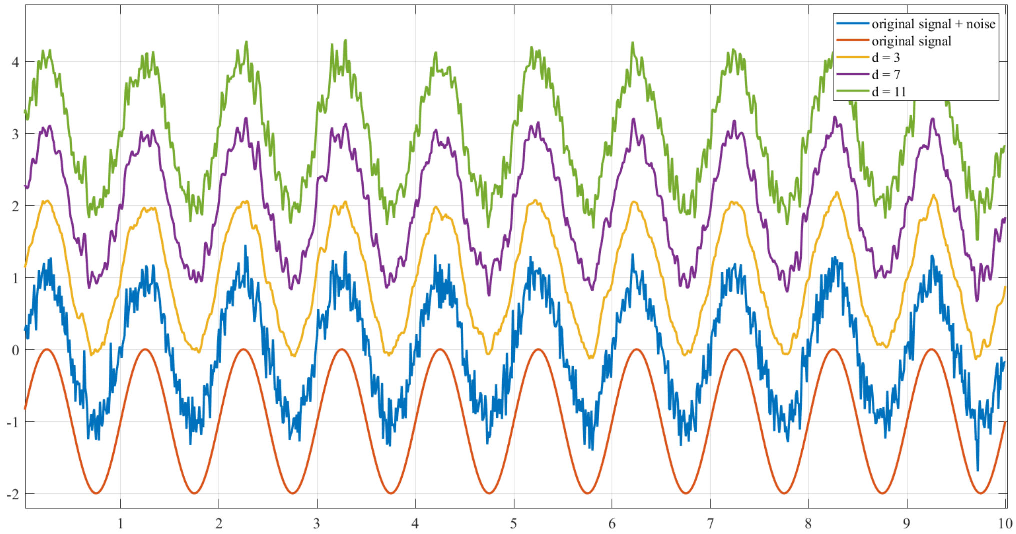


Figure 3.6: Results of applying SG filters with different polynomial degrees on a sinusoidal wave with noise

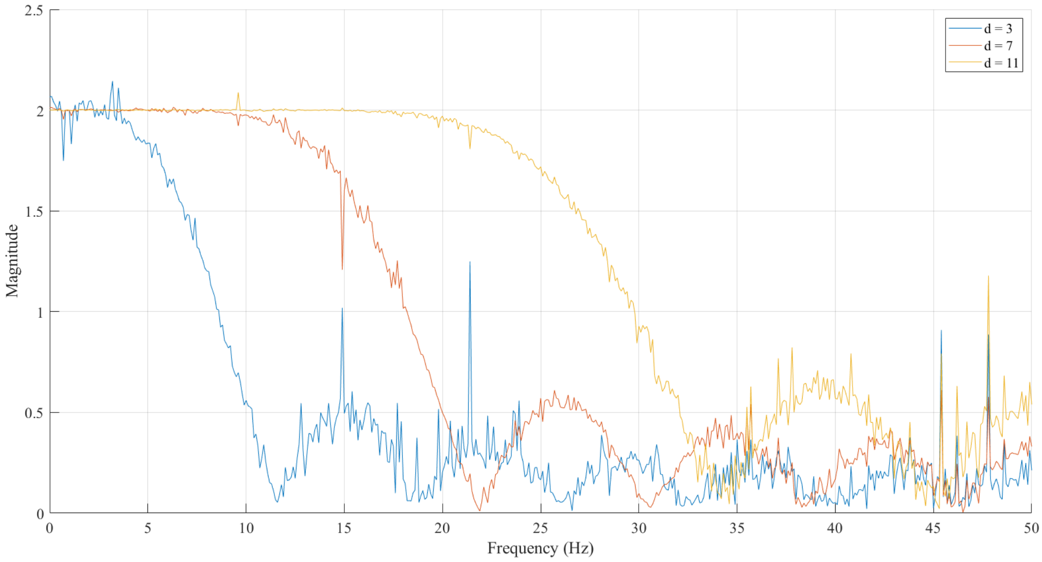


Figure 3.7: Frequency response of SG filters with different polynomial degree

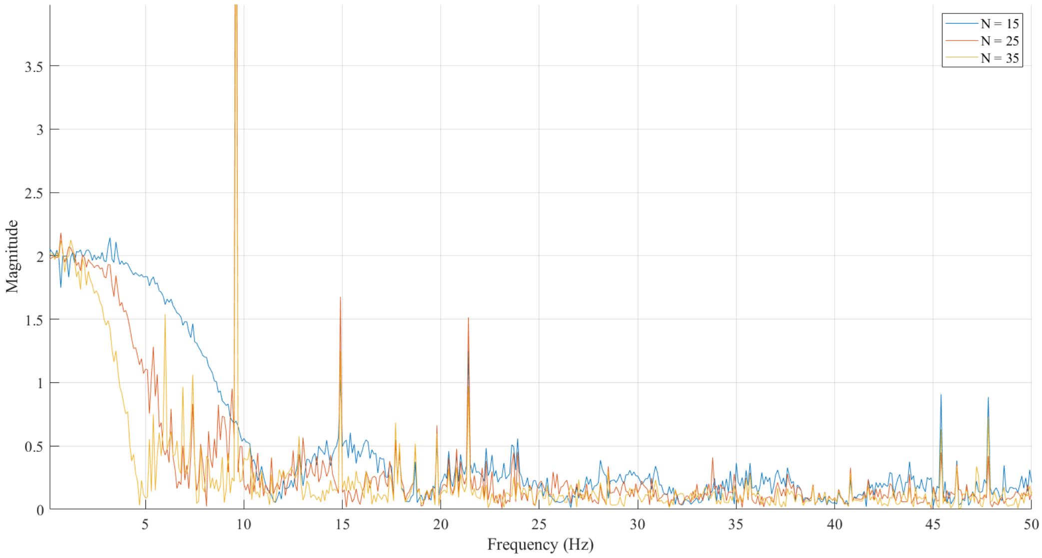


Figure 3.8: Frequency response of SG filters with different window length

where t is a configurable threshold value for outlier detection. If x_i is defined as an outlier, x_i takes the value of x_m [55].

Figure 3.9 illustrates how Hampel filter works, having a sine wave with some values added intentionally as outliers, $t = 3$ and a window size of 6, can be seen that the filter correctly identifies those added outliers and replaces them with a value closer to the waveform real value in that position.

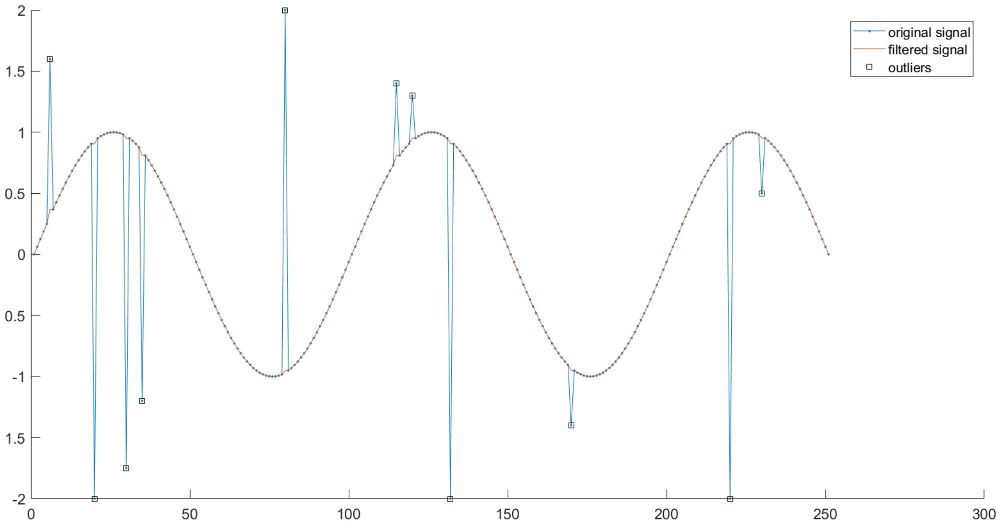


Figure 3.9: Hampel filter implementation on a sine wave

For device free sensing applications using CSI, Hampel filter is applied to each subcarrier to remove outliers which typically have significantly different values from neighboring CSI measures.

3.3.2 Subcarrier Selection

To reduce computational cost, selection of the most representative subcarriers to apply vital signs estimation needs to be done. The most common method in the literature explored [29–31, 46–48] is based on the variance of CSI amplitude of each subcarrier, selecting those with the highest variance given a time window.

It is this variance-based subcarrier selection method the one used for this investigation and it is shown in Algorithm 1, which receives as input all subcarriers' CSI amplitudes in a time window and the number of subcarriers (n) to be selected. This algorithm outputs are the indexes of the n subcarriers with highest variance of CSI amplitude.

As can be seen in Fig. 3.10, the subcarrier that presented the highest variance of CSI amplitude shows a clear sinusoidal waveform, which corresponds to a breathing signal; while in Fig. 3.11 it is presented the 50_{th} sensitive subcarrier if they were in descending order according to their variance, and, as can be seen, its waveform is clearly altered by other sinusoidal waves at different frequencies. Besides this subcarrier selection method do not focus in frequency analysis, it seems to be capable of identifying subcarriers where breathing is captured and therefore its corresponding breathing rate frequency is dominant.

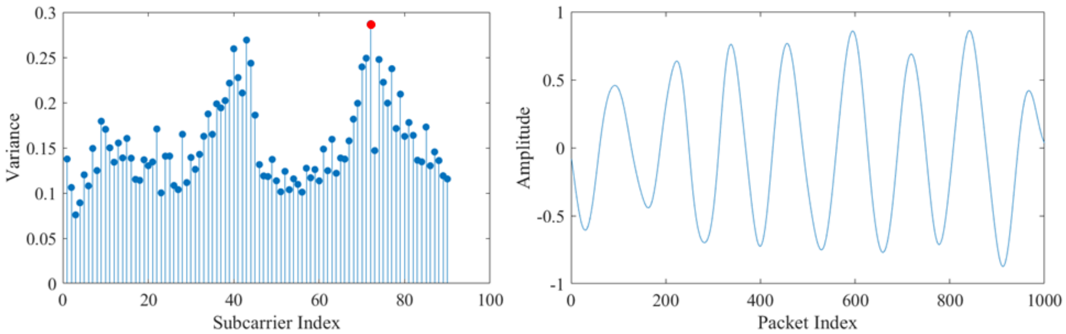


Figure 3.10: Subcarrier with highest variance

Algorithm 1: Subcarrier selection algorithm

```
Data: $subcarriersData$, $n$
Result: $n$ most sensitive subcarriers (ss)
begin
    $[sEntries, sNumber] = \text{size}(subcarriersData)$ // number of entries and subcarriers of subcarriersData

    for $i = 1$ to $sNumber$ do
        // This will be done for each subcarrier
        $\sum = 0$
        $\overline{X} = \text{mean}(subcarriersData[i])$
        $N = \text{length}(subcarriersData[i])$ // number of elements in subcarrier

        for $j = 1$ to $N$ do
            $\sum = \sum + (subcarriersData[i][j] - \overline{X})^2$
        end
        $\sigma^2[i] = \frac{\sum}{N}$ // The variance of each subcarrier is stored
    end
    // The $n$ subcarriers with the highest values of $\sigma^2$ are choosen as the most sensitive
    $ss = \max(\sigma^2, n)$
end
```

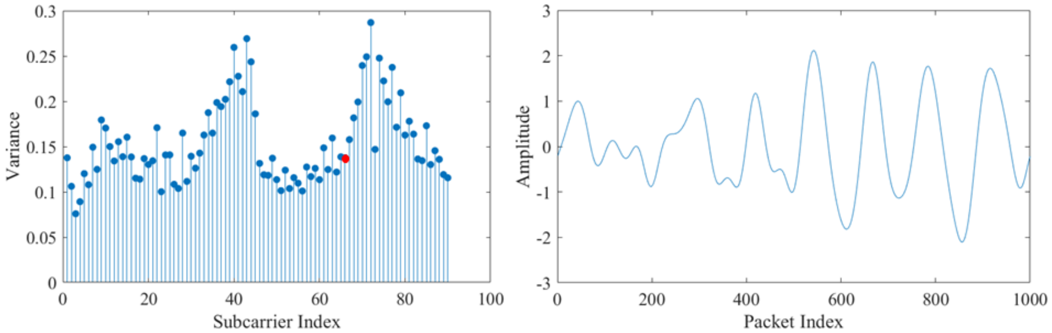


Figure 3.11: Subcarrier that placed in 50_{th} according to its variance

3.4 Data Analysis Tools

3.4.1 Discrete and Fast Fourier Transform

The Fourier Transform is a tool that allows to look at the frequency domain of a signal instead of the time domain. The Discrete Fourier Transform (DFT) is the version of the Fourier transform that focuses in working with discrete signals. The DFT of a signal $x[n]$ is defined as:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \tag{3.8}$$

which provides information of how $x[n]$ consists of complex exponentials at different frequencies [56].

Implementing directly DFT in a computer software will result inefficient due the amount of calculations that are done in order to be computed, having a complexity of $O(N^2)$. Fast Fourier Transform (FFT) is a version of DFT that solves this problem with a complexity of $O(NlogN)$ which is implemented in software such as in MATLAB and is commonly used for analyzing the frequency properties of a signal [50]. Given a signal $x[n] = cos(2\pi(10)) + 2cos(2\pi(25)) + 5cos(2\pi(40))$ by applying FFT to it in MATLAB the result obtained can be seen in Fig. 3.12 where peaks at the frequencies of 10, 25 and 40 Hz can be observed.

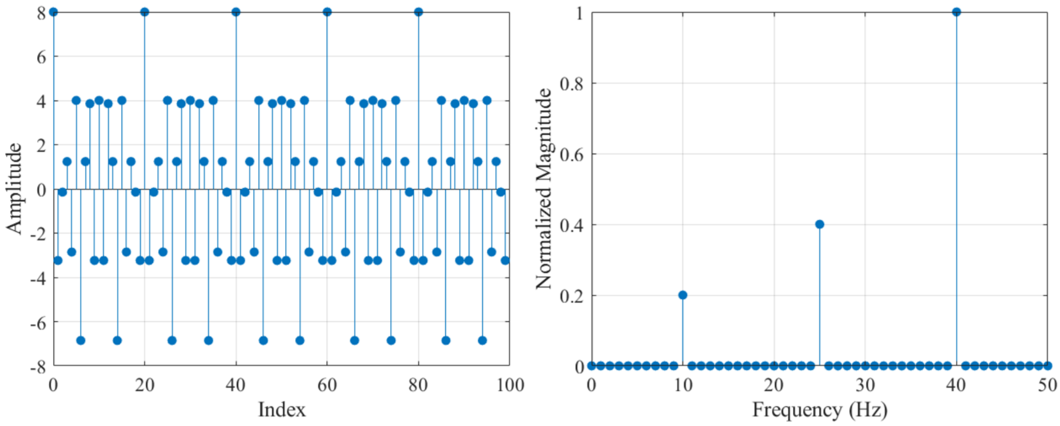


Figure 3.12: FFT demonstration. Left: $x[n]$; Right: Frequency domain obtained with FFT on $x[n]$

DFT was also applied to $x[n]$, obtaining the exactly same result, being the execution time of the function the only difference between both of them with 0.53 milliseconds for FFT and 2.22 milliseconds for DFT.

In a CSI-based sensing application, FFT is used for analyzing the frequency domain of each subcarrier and for extracting frequency domain features of the signal.

3.4.2 Hilbert Transform and Variational Mode Decomposition

The Hilbert transform of a signal $x(t)$ is defined as a convolution between the Hilbert transformer $\frac{1}{\pi t}$ and the signal $x(t)$:

$$\mathcal{H}(x(t)) = x(t) * \frac{1}{\pi t} = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau \tag{3.9}$$

but because convolution with Hilbert transformer is a improper integral, $\mathcal{H}(x(t))$ of a signal $x(t)$ is obtained as the Cauchy principal value $\mathcal{P}$ of the convolution integral:

$$\mathcal{H}(x(t)) = \frac{1}{\pi} pv \int_{\mathbb{R}} \frac{f(\tau)}{t - \tau} d\tau \tag{3.10}$$

Together, $x(t)$ and $\mathcal{H}(x(t))$ create an analytic signal with an amplitude and a phase, where the derivative of the phase can be identified as the instantaneous frequency [57], for any given time, there is only one frequency value (a scalar). This analytic signal is defined as:

$$f_A(t) = f(t) + j\mathcal{H}f(t) = A(t)e^{j\phi(t)} \tag{3.11}$$

The instantaneous frequency should be interpreted as a localized frequency with a narrow band. A standard bandwidth measure v is given by:

$$v^2 = \pi^2(N_1^2 - N_0^2) \tag{3.12}$$

where N_0 is the expected number of zero crossings per unit of time:

$$N_0 = \frac{1}{\pi} \left(\frac{m_2}{m_0} \right)^{\frac{1}{2}} \tag{3.13}$$

and N_1 the expected number of extrema per unit of time:

$$N_1 = \frac{1}{\pi} \left(\frac{m_4}{m_2} \right)^{\frac{1}{2}} \tag{3.14}$$

in which m_i is the i_{th} moment of the spectrum. For a narrow band signal $v = 0$, the expected numbers of extrema and zero crossings have to be equal. Parting from this concept, a function with the same number of extrema and zero crossings or with a difference of at most by one will be called an intrinsic mode function (IMF) or simply mode, from which a meaningful local instantaneous frequency can be identified [58].

A variational model that determines the relevant bands adaptively and that estimates the corresponding modes u_k concurrently in a way that they reconstruct the given input signal optimally adressing the presence of noise, while being band-limited about a center frequency estimated is the Variational Mode Decomposition (VMD). Each mode is defined as:

$$u_k(t) = A_k(t)cos(\phi_k(t)); \tag{3.15}$$

where each mode has an instantaneous frequency $\phi'_k(t)$ that is nondecreasing, varies slowly and is concentrated around a central value f_k. The process of VMD consists of finding a set of $u_k(t)$ and $f_k(t)$ that minimizes the constrained variational problem. For obtaining the K-IMF initialized at 0, as well as the central frequencies f_k and the Lagrange multiplier $\hat{\lambda}$, VMD iteratively update the IMF until converging with a defined tolerance ϵ:

$$\sum_k ||u_k^{n+1}(t) - u_k^n(t)||_2^2/||u_k^n||_2^2 < \epsilon \tag{3.16}$$

For the $(n+1)_{th}$ iteration, VMD iterates over the K modes and obtains the frequency-domain waveforms for each mode:

$$u_k^{n+1}(f) = \frac{X(f) - \sum_{i<k} u_i^{n+1}(f) - \sum_{i>k} u_i^n(f) + \frac{\hat{\lambda}^n(f)}{2}}{1 + 2\alpha(2\pi(f - f_k))^2} \tag{3.17}$$

and the k_{th} central frequency:

$$f_k^{(n+1)} = \frac{\int_0^\infty f|u_k^{n+1}(f)|^2 df}{\int_0^\infty |u_k^{n+1}(f)|^2 df} \tag{3.18}$$

and finally the Lagrange multiplier:

$$\hat{\lambda}^{n+1}(f) = \hat{\lambda}^n(f) + \tau\left(X(f) - \sum_k u_k^{n+1}(f)\right) \tag{3.19}$$

where α is a penalty factor and τ is the update rate of the Lagrange multiplier [59].

Hilbert Transform and VMD will be used for analyzing CSI and for designing a new subcarrier selection algorithm based on frequency stability in a time window due to the fact that Hilbert Transform can provide the instantaneous frequency for each time unit of the window.

3.4.3 Visualization of High Dimensional Data

Numerous time and frequency domain features can be extracted from the breathing and heartbeat signal, which values may be specific for a certain breathing or heart rate. As a preview of what will be presented in further chapters, these signals will be split in segments, and from each segment, a series of time and frequency domain features will be extracted, obtaining what will be referred as a multi-dimensional observation. A common way to analyze or even make use of this kind of data is by using Data Dimensionality Reduction Techniques [60].

Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset by finding new variables, which are linear functions of the original variables of the dataset, that maximize variance and that are uncorrelated with each other [61].

Given a dataset of dimensions $n \times m$, where n is the number of entries or observations and m the number of variables or better known as features, can be also seen as a data matrix X. The main goal is to find a linear combination of the X columns given by:

$$Xa = \sum_{j=1}^{m} a_j x_j \tag{3.20}$$

where a is a vector of length m. The variance of any linear combination is given by:

$$a^T Sa = \frac{\sum_{i=1}^{n} (x_i \bar{x})^2}{n} \tag{3.21}$$

where S is the covariance matrix of the data matrix and a^T is the transpose of a. Therefore, in order to find the linear combination that maximizes the variance it is needed to find the a vector that maximizes $a^T Sa$. Vector a is an eigenvector with its corresponding eigenvalue λ_k of the covariance matrix S where the largest eigenvalue and its respective eigenvector need to be found. S matrix, as other symmetric matrix, contains n eigenvalues with their respective eigenvectors, which are the solution to find k linear combinations:

$$X_{ak} = \sum_{j=1}^{m} a_{kj} x_j \tag{3.22}$$

where the linear combinations X_{ak} are called *principal components.*

For obtaining the first principal component of the X matrix, this can be seen as followed:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & ... & x_{1,m} \\ x_{2,1} & x_{2,2} & ... & x_{2,m} \\ ... & ... & ... & ... \\ x_{n,1} & x_{n,2} & ... & x_{n,m} \end{bmatrix} \qquad a_{k=1} = \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_m \end{bmatrix}$$

$$X_{i,1:m} * a = \begin{bmatrix} x_{i,1} & x_{i,2} & ...x_{i,m} \end{bmatrix} * \begin{bmatrix} a_1 \\ a_2 \\ ... \\ a_m \end{bmatrix} = \sum_{j=1}^{m} x_{i,j} a_j = Y_i$$

$$Y = Xa_{k=1} = \begin{bmatrix} Y_1 \\ Y_2 \\ ... \\ Y_n \end{bmatrix}$$

where Y is the first principal component ($k = 1$) which can be seen as a new representation of X in one dimension.

T-distributed Stochastic Neighbor Embedding

T-distributed Stochastic Neighbor Embedding, also known as t-SNE, is a dimensionality reduction technique able to capture the global structure of data while revealing the global structure and the presence of clusters by modeling dissimilar datapoints by means of large pairwise distances and modeling similar datapoints by means of small pairwise distances based on a t-Student distribution [62].

Given a dataset X, the similarity between a datapoint x_j and x_i is given by the joint probability p_{ij} defined as:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{3.23}$$

where as for example, conditional probability $p_{j|i}$ is calculated as:

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/\sigma_i^2)} \tag{3.24}$$

where $||x_i - x_j||$ can be replaced by a distance metric to measure distance between x_i and x_j, and σ_i is obtained by doing a binary search given a perplexity value usually between 5 and 50. Being Y a low-dimensional representation of X, it is posible to obtain a joint probability q_{ij} of the counterparts x_i and x_j, y_i and y_j:

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{3.25}$$

If y_i and y_j are faithful representations of x_i and x_j, p_{ij} and q_{ij} will be equal. Therefore, with t-SNE the objective is to find the Y representation that minimizes the difference between both probabilities. In order to measure the faithfulness in which q_{ij} represents p_{ij}, Kullback-Leibler divergence is used, minimizing the sum in all datapoints given a gradient descent method:

$$\frac{\delta C}{\delta y_i} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{3.26}$$

Y is obtained iteratively, given a iteration t, $Y(t)$ is defined as:

$$Y(t) = Y(t-1) + \eta\frac{\delta C}{\delta Y} + \alpha(t)(Y(t-1) - Y(t-2)) \tag{3.27}$$

where η is a learning rate and $\alpha(t)$ represents the momentum at iteration t, both being inputs defined by the user.

3.5 Classification Methods

As can be seen in Chapter 2, machine learning techniques, such as classification methods, have been used for device-free sensing with CSI applications, e.g. Support Vector Machines and K-Nearest Neighbors. In order to understand how these classification methods work, as well as a third classification method, Quadratic Discriminant Classifier, a brief description will be presented.

3.5.1 Support Vector Machines

Support Vector Machines model was developed to solve classification and regression problems in machine learning. The goal is to find a model from the input feature space i.e. mapping of the original data into a high dimensional feature space through a kernel function, which predicts a target variable [63]. In the experiments, a 2-degree polynomial kernel function was selected as follows:

$$K(X_i, X_j) = (X_i' X_j + 1)^2 \tag{3.28}$$

where X_i and X_j are input vectors. Classification is performed by evaluating the following formula:

$$y' = sgn(\mathbf{w} \cdot \mathbf{X}' + b) \tag{3.29}$$

where **w** and **b** are obtained by solving a quadratic optimization problem; and X' is a test sample.

3.5.2 K - Nearest Neighbors

K-Nearest Neighbor applies the concept of similarity among samples to assign class labels according to the assumption that similar samples belong to the same object class [63]. The decision rule for a nearest neighbors algorithm using a single neighbor for comparison would be:

$$\hat{y} = \operatorname*{argmin}_{X_i \in D_n, \theta_i} dist(X, X_i) \tag{3.30}$$

where D_n is the training set and θ_i is the class or label associated to X_i. This algorithm has a free-parameter named k which sets the number of neighbors to be considered in the *voting* step to assess which will be the label to assign to the test sample X according to a distance metric, being the euclidean distance one of the most common, defined as:

$$D = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \tag{3.31}$$

where D is the distance between two samples (observations), X_i and Y_i the values of their predictor variables.

3.5.3 Quadratic Discriminant Classifier

Quadratic Discriminant Classifier consists of discriminant functions that numerically associate a test sample X_i to a class C_i i.e. the one with maximum value computed as follows:

$$\begin{gathered} D_i(X) = -\frac{1}{2}X\Sigma_i^{-1}X' + \mu_i\Sigma_i^{-1}X' - \\ \frac{1}{2}\mu_i\Sigma_i^{-1}\mu' - \frac{1}{2}ln|\Sigma_i| + ln[P(C_i)] \end{gathered} \tag{3.32}$$

where Σ_i is the covariance matrix, μ_i the mean or centroid, and $P(C_i)$ is the *apriori* probability, all computed for each class C_i [63].

3.5.4 Performance Metrics

Performance metrics provide information of how well a classification or regression model is by giving a quantitative and objective measurement. Focusing on classification, these metrics allow to compare the performance of different classifiers doing the same classification task and select the one that best resolves the classification problem as will be seen in further chapters.

The metrics that will be used for evaluating the performance of the classifiers that will be presented in further chapters are:

Confusion Matrix

It is a table that presents the ground-truth labels (Targets) vs. the predictions of the classifier (Outputs). It works as a basis for obtaining other performance metrics. Each cell of the matrix adopts values for True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) organized as in Fig. 3.13 for a two-class classification problem.

		Outputs	
	Class	1	0
Targets	1	TP	FN
	0	FP	TN

Figure 3.13: Confusion Matrix example

Considering *Class 1* as positive and *Class 0* as negative, Confusion Matrix values can be explained as:

- TP: predictions that are labeled as positive and they are truly positive.
- TN: predictions that are labeled as negative and they are truly negative.
- FP: predictions that are labeled as positive but they are actually negative.
- FN: predictions that are labeled as negative but they are actually positive.

Accuracy

It is the number of correct predictions over all predictions made, defined as

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{3.33}$$

It is important to notice that this metric is only meaningful if we have an equal number of observations belonging to each class.

Precision

Proportion of positive predictions that were actually correct.

$$Precision = \frac{TP}{TP + FP} \tag{3.34}$$

Recall

Proportion of positive predictions that were identified among all of the observations that should have been identified as positive.

$$Recall = \frac{TP}{TP + FN} \tag{3.35}$$

Specificity

Being the opposite of Recall, it is the proportion of negative predictions that were identified among all of the observations that should have been identified as negative.

$$Specificity = \frac{TN}{TN + FP} \tag{3.36}$$

F-Score

Single measurement that can be used for representing both Precision and Recall. It is obtained by calculating the Harmonic Mean between these two metrics.

$$\textit{F-Score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.37}$$

ROC Curves

Receiver Operating Characteristics (ROC) curves are generated by plotting the Recall (also referred as true positive rate) on the y-axis of a graph and the false positive rate ($1 - Specificity$) on the x-axis for different values of a continuous test, resulting in a list of test values and their corresponding recall and false positive rate. A ROC curve that follows $y = x$ produces false positive results at the same rate as true positive results, which is not desirable [64].

A measurement that can be obtained from ROC Curves is the Area Under the ROC Curve (AUC), which as its name states, is the area underneath the ROC Curve. AUC measures how much the model is capable of distinguish between classes, considering that an AUC with a value of 1 is a model with a perfect capability of distinguishing between classes.

3.6 Fresnel Zone Model

According to Fresnel theory, for establishing an optimal connection for communication between a receiver antenna and a transmitter antenna, the LoS between them should be clear of obstructions, and objects inside the Fresnel zones can cause fading effects. Fresnel zones are a series of concentric ellipsoids drawn around the communication terminals and, when an object appears in one of the Fresnel zones, two paths are created in which the radio signals travel, one being the LoS path and the other one is created by the reflection with the object. The signals that travel in both paths are combined in the receiver antenna, creating a superimposed signal. When the object moves while the LoS remains the same, the reflected signal from the object changes over time and as well as the superimposed signal [65, 66].

The radius for a Fresnel zone n can be calculated following the next equation:

$$r_n = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}} \tag{3.38}$$

where λ is the wavelength of the signal transmitted, d_1 the distance in meters between one antenna and a point in the LoS and d_2 the distance between the second antenna and the same point [67]. A representation of these parameters is shown in Fig. 3.14.

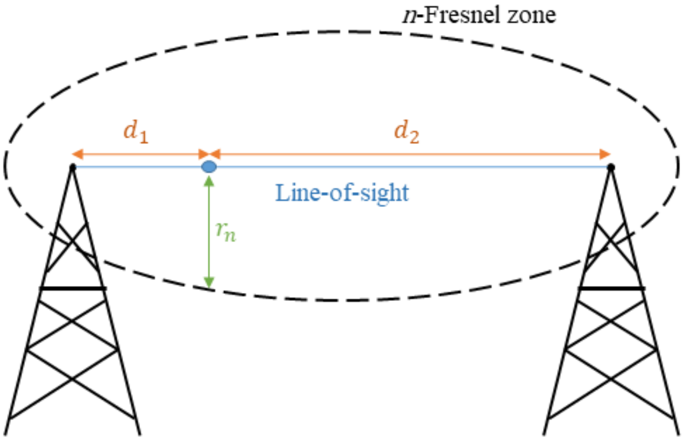


Figure 3.14: $n-$Fresnel zone representation

For CSI sensing applications Fresnel zone model has been used to provide theoretical foundation for the physical distribution of CSI-based systems, such as in [35] where the authors conduct all the experiments inside the first Fresnel zone because more than 70% of the energy is located in this zone, therefore if an object moves in this zone, the amplitude and phase of the received signal can be greatly affected, obtaining that results for a lying subject are better if subject is nearer the LoS inside the first Fresnel Zone, and results get worse if subject is near the boundary for monitoring breathing rate. In [31] Fresnel zone model was also used and authors found out that if the subject is at the middle of the 2_{nd} and 3_{rd} Fresnel zone the signal peaks and valleys representing the breathing of the subject are more defined than if at a boundary of these Fresnel zones.

4. Proposed System

4.1 System Overview

The vital signs monitor proposed in this work consists of four main components as shown in Fig. 4.1. At first, network packets need to be collected in order to obtain CSI amplitudes and phase shifts. Only the amplitudes are later on calibrated by a series of filters, and, once calibrated, time and frequency domain features are extracted for further breathing and heart rate classification, displaying the classification result on screen with a desktop application.

A detailed description of each system's main component will be presented in the following subsections.

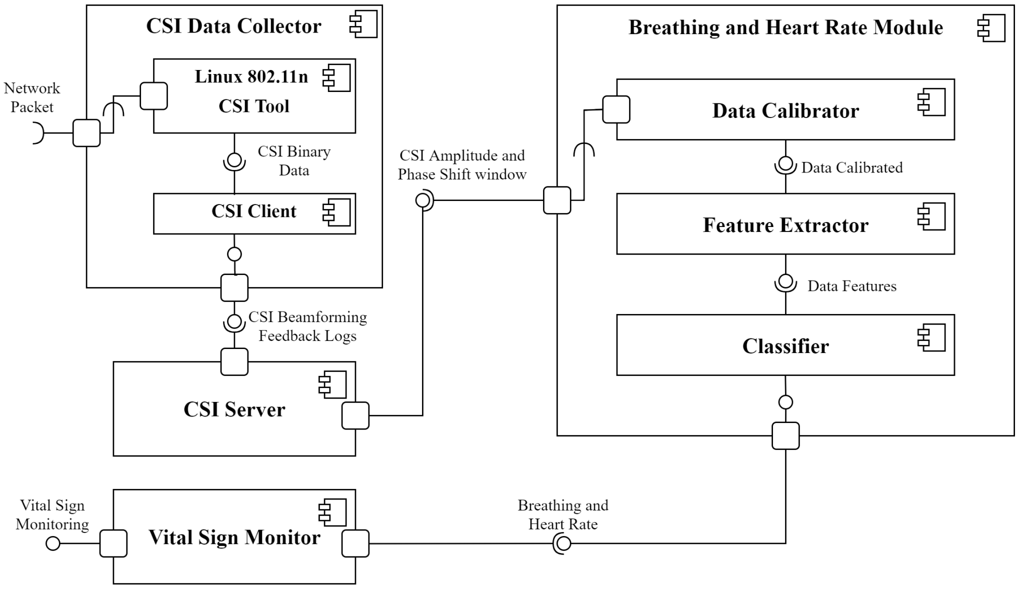


Figure 4.1: System's Component Diagram

A) CSI Data Collector

CSI data is collected using a plugin developed by Bingxian et al. [68] for Linux 802.11n CSI Tool [41]. This plugin logs the CSI data collected in the receiver to a server being executed in MATLAB, which allows real time CSI data processing in another computer requiring only a TCP connection with the receiver.

This component provides a stream of bytes, which contains CSI data, via socket to the CSI Server. The *CSI Server* works as a bridge between the data collector and the *Breathing and Heart Rate Module*, providing the CSI amplitude and phase shift of 30 subcarriers.

B) Breathing and Heart Rate Module

The Breathing and Heart Rate Module is in charge of processing the CSI amplitude in order to give as result a breathing range and pattern classification, as well as a heart rate range classification, addressing the estimation of these two vital signs as a classification problem. To achieve this task, the Breathing and Heart Rate Module is divided into three subcomponents: the Data Calibrator, Feature Extractor, and the Classifier components. These three subcomponents were developed using MATLAB R2020b and run in the same platform:

- *Data Calibrator:* Prepares the CSI amplitude in order to obtain the breathing signal for further processing. At first, it waits for a 40 seconds time window of CSI amplitudes. Once collected, a Hampel identifier is applied to replace outliers with the median of the time window. Once outliers from every subcarrier are replaced, a Savitzky-Golay (SG) filter with a window length of 51 is applied to smooth every subcarrier. SG filter acts as a lowpass filter, therefore the frequency response is a thing to keep in mind, which slope is affected by the polynomial degree as stated before. Fig. 4.2 shows the effect of different polynomial degrees in time and frequency domain with a fixed window length of 51; a normal adult breathing rate at rest is between 12 and 16 breaths per minute, while heart rate is between 60 and 100 heart beats per minute, which can be translated to a frequency range of 0.2 and 0.26 Hz and 1 to 1.7 Hz respectively; hence analyzing the frequency response for different polynomial degrees, a degree of 3 seems to filter frequencies above 0.5 Hz, but at the same time it arises the interval between 1 to 1.5 Hz, where heart rate is more likely to be found. Therefore, SG filter degree was set to 3 and window length of 51. After Savitzky-Golay filter is applied, a bandpass filter with cut-off frequencies of 0.1 Hz and 0.4 Hz is applied to every subcarrier, which is expected to extract the periodic movements caused by breathing at rest, obtaining the breathing signal. Once subcarriers are completely filtered, the ten most sensitive subcarriers are selected based on a commonly used variance method [30, 31, 46–48] in order to reduce

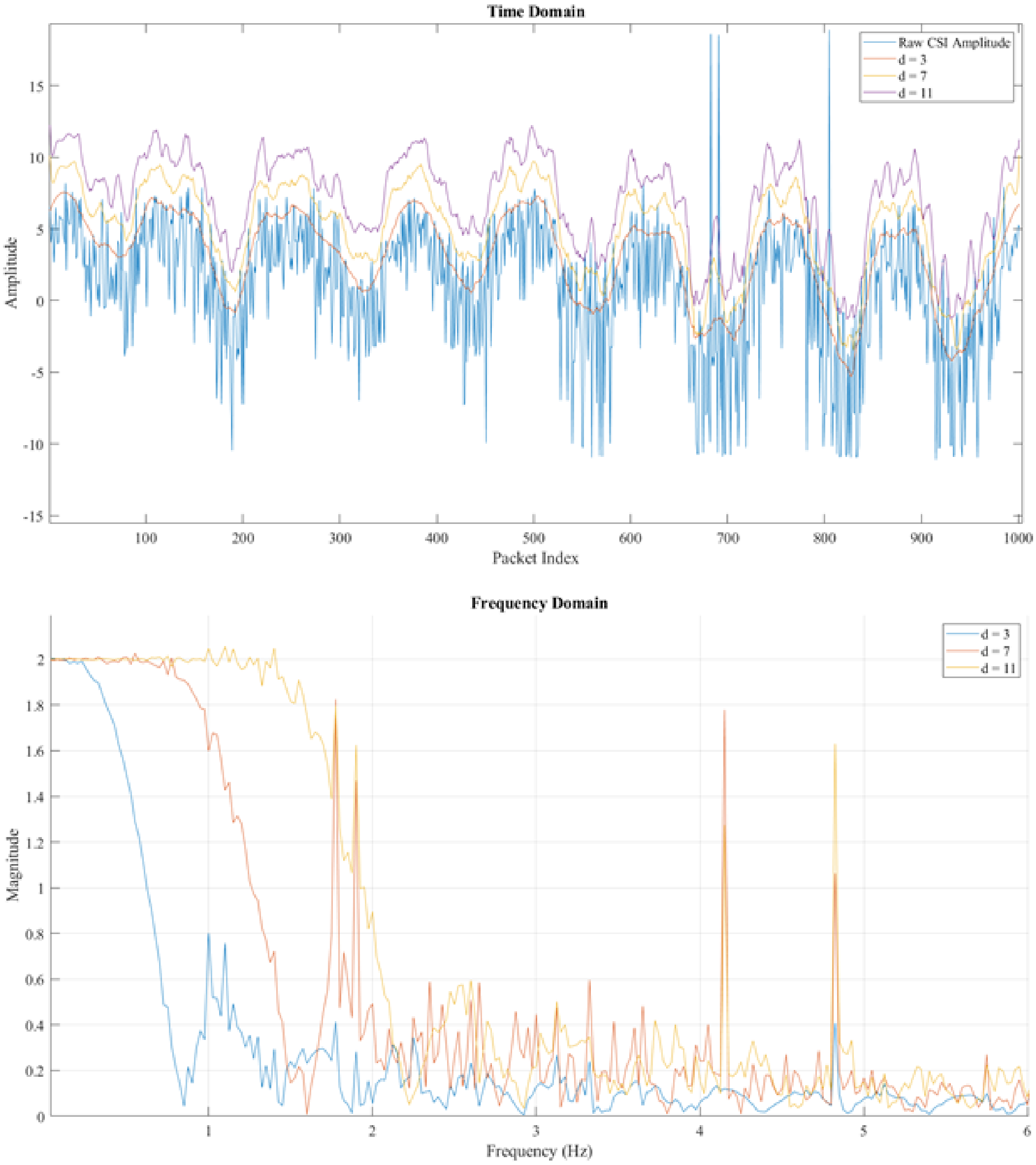


Figure 4.2: Results obtained by applying Savitzky-Golay filter with different polynomial degrees on CSI amplitude

Table 4.1: CSI Extracted Time Domain Features

TDF	Formula
Mean	$\frac{1}{T}\sum_{t=1}^{T} x_t$
Variance	$\frac{1}{N-1}\sum_{t=1}^{T}(x_t-\mu)^2$
Skewness	$\frac{1}{N}\sum_{t=1}^{T}(x_t-\mu)^3/\left(\sqrt{\frac{1}{N}\sum_{t=1}^{T}(x_t-\mu)^2}\right)^3$
Kurtosis	$\frac{1}{N}\sum_{t=1}^{T}(x_t-\mu)^4/\left(\frac{1}{N}\sum_{t=1}^{T}(x_t-\mu)^2\right)^2$

computational complexity as seen in Algorithm 1. This subcarrier selection algorithm is only executed at the moment when the first sample window is calibrated; for the next sample windows, calibration is only applied to the subcarriers labeled as the most sensitive. For obtaining the heartbeat signal, we substract the breathing signal obtained to the subcarriers filtered with the Savitzky-Golay filter in order to remove the breathing components from our sensitive subcarriers. Once this is done, only the first five sensitive subcarriers are used for further heart rate processing and classification. A bandpass filter with cut-off frequencies of 1 Hz and 2 Hz is applied to these five sensitive subcarriers, obtaining what is expected to be the heartbeat signal.

- *Feature Extractor:* This subcomponent extracts time and frequency domain features from breathing and heartbeat signal and pass them to the next subcomponent as two different observations. From time domain, features such as mean, variance, skewness and kurtosis [69], which can be seen in Table 4.1 along with the formula used to calculate them, are extracted by this subcomponent from every sensitive subcarrier calibrated. For extracting features of the frequency domain, Fast Fourier Transform (FFT) is applied to every subcarrier, obtaining features such as frequency with highest amplitude and standard deviation. Another pair of features obtained are the first breathing and heart rate estimation, which is done by calculating the mean of each point in the frequency spectrum between the subcarriers, obtaining what we called the *subcarriers' mean spectrum*. From this spectrum, the vital sign is estimated by obtaining the frequency with highest amplitude, which can be seen as the number of breaths or beats per second, therefore, by multiplying it by 60 seconds will give the number of breaths per minute (BrPM) for breathing rate and beats per minute (BPM) for heart rate. This can be summarized as follows:

$$FirstEstimation = argmax(meanSpectrum) * 60 \tag{4.1}$$

  Due to the fact that body movements done by the subject or changes in the environment (artifacts) may have a negative impact on this first estimation,

Breathing Rate Range	**Class label**
Below 10 BrPM	1
11 - 13 BrPM	2
14 - 16 BrPM	3
17 - 19 BrPM	4
Above 20 BrPM	5

Table 4.2: Labels assigned to breathing rate ranges

Breathing Rate Pattern	**Class label**
Bradypnea (below 12 BrPM)	1
Normal Breathing Rate (12 - 20 BrPM)	2
Tachypnea (above 20 BrPM)	3

Table 4.3: Labels assigned according to heart rate range

this is averaged with the last three estimations in order to reduce the error that these artifacts might produce at a certain time. The last features are obtained by applying the Discrete Wavelet Transform (DWT) [70] of level 4 to the time domain series. For the generated detail and approximation coefficients from each subcarrier, mean and variance are calculated. At last, the features extracted are passed to the *Classifier* as two different observations, one for breathing rate classification and the other one for heart rate classification.

- *Classifier:* This subcomponent receives both observations and classify the observations in labels according to breathing rate ranges as shown in Table 4.2 and according to breathing rate patterns in Table 4.3. For heart rate classification, heart rate ranges are labeled as shown in Table 4.4. These classification tasks are accomplished with three different classification methods: Support Vector Machines (SVM), K-Nearest Neighbor (K-NN) and Quadratic Discriminant Classifier (QDC).

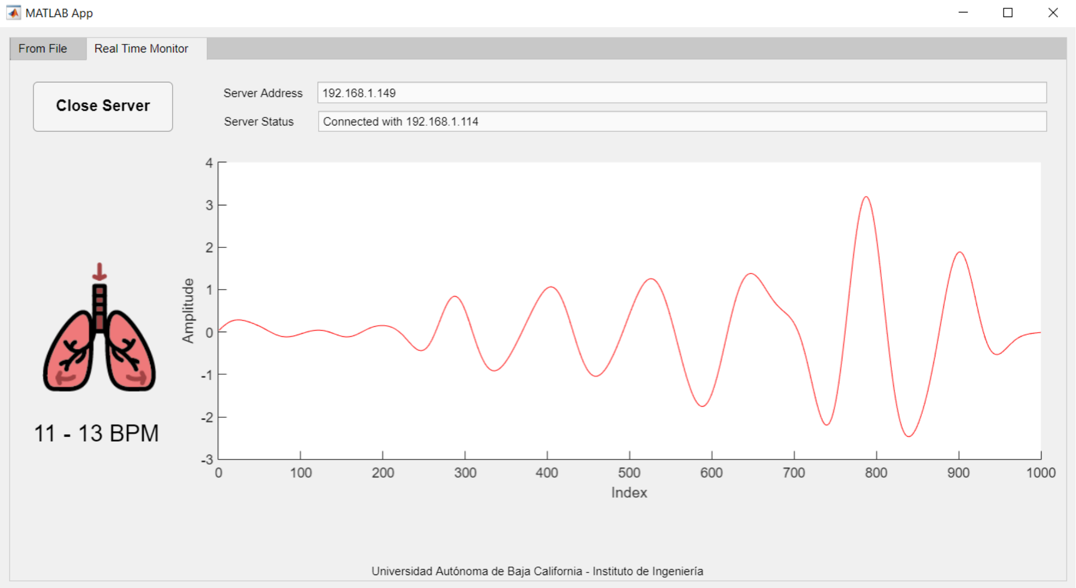
D) CSI Vital Sign Monitor

This component is a desktop application developed using MATLAB R2020b App Designer and it is responsible for displaying the breathing rate classification results in real time, along with a graph that shows the last 40 seconds of CSI calibrated amplitudes received as shown in Fig. 4.3. For heart rate classification results, this component comes with a second function which allows to calibrate and classify

Heart Rate Range	**Class label**
Below 60 BPM	1
61 - 80 BPM	2
Above 80 BPM	3

Table 4.4: Labels assigned according to breathing rate patterns

observations from a file with dat or csv extension, allowing to change filter parameters as well as the classification model, a feature that makes possible to process both breathing and heart rate files. The reason why heart rate classification was discarded for real time monitoring with this application is due to experimental results that will be presented in the following chapters.



Server Address 192.168.1.149

Server Status Connected with 192.168.1.114

11 - 13 BPM

Figure 4.3: Real-time functioning of breathing rate monitoring using the developed system

4.2 Deployment

Linux 802.11n CSI Tool shell scripts for collecting CSI are executed on two Intel NUCs equipped with Intel 5300 NIC, both with Ubuntu 14.04 LTS, one working as the transmitter (TX) sending packets, while the other one works as the receiver (RX). RX establishes a TCP connection using sockets in order to send the CSI data: amplitude and phase shift, to a MSI GS65 laptop executing the MATLAB script that sets a server that listens to the socket from where CSI data is arriving, processes the CSI data and executes the CSI Vital Sign Monitor in Windows 10 OS –see Appendix A Fig. 8.3.

5. Methodology

In this chapter an explanation of the steps followed for collecting CSI data and extracting time and frequency domain features in order to create the training and test data sets for the classifiers is given. The training data set was used for training three different classifiers per classification task: breathing rate range classification, breathing pattern classification and heart rate range classification. Results from this training step are also presented in this chapter. Additionally, an extra step of data visualization is included for analyzing if clusters are formed for each class according to the classification task.

5.1 Collecting CSI Data

Before executing the system in a real-time environment, experiments were conducted in the Laboratory of Data Science and Engineering of *Universidad Autónoma de Baja California* with 17 participants whose information is presented in Table 5.1. CSI data was collected using the Linux 802.11n CSI Tool [41] in order to set up each module that compose the system proposed and test its performance, processing the data collected as stated in the previous chapter simulating a real-time functioning.

Participants were asked to lay on their back on a bed. Two mini-PC Intel NUC D54250WYKH, both equipped with Intel 5300 NIC and running Ubuntu 14.04, were placed one on each side of the bed with a separation of 1.5 m, keeping the participant's chest in LoS or at least inside the first two Fresnel zones as in [31]. The bed was placed at the center as seen in Fig. 5.1. One mini-PC acts as a transmitter (TX) sending packets to the other mini-PC, which is the receiver (RX), at 25 packets/s in the 5 GHz band. Considering the center frequency of the 5 GHz band channel selected is of 5340 MHz with a $\lambda = 0.05614\ m$ and the stated distance between antennas, according to Equation 3.38, the radius of the 1_{st} and 2_{nd} Fresnel zone would be of 0.1451 and 0.2053 meters respectively.

Table 5.1: Participants Information

Tag	**Weight (kg)**	**Height (m)**	**Sex**	**Age**	**Disease**
S1	79	1.76	M	23	None
S2	47	1.58	F	23	None
S3	110	1.83	M	23	None
S4	80	1.69	M	24	None
S5	47	1.52	F	22	None
S6	53	1.55	F	24	None
S7	71	1.74	M	23	None
S8	72	1.68	M	23	None
S9	76	1.64	F	27	None
S10	69	1.6	F	64	None
S11	95	1.7	M	68	None
S12	78	1.56	F	46	None
S13	90	1.65	M	46	None
S14	80	1.6	F	48	None
S15	70	1.67	M	34	None
S16	73	1.68	M	44	None
S17	115	1.75	M	23	None

For collecting data for breathing rate classification, participants were asked to control their breathing rate by following the beats of a metronome. Five sets of data were collected from each participant, increasing the metronome rhythm from 18 to 42 in steps of 6 beats per minute (which corresponds to breathing rates from 9 to 21 BrPM), having each rhythm a duration of 5 minutes. Additionally a triaxial accelerometer was placed on each participant's chest in order to obtain a breathing rate ground-truth based on spectral analysis. Meanwhile, for heart rate classification participants were asked to maintain the same position as for breathing rate, but now wearing a Polar H9 sensor [71] for obtaining a heart rate ground-truth; a single set with a duration of 5 minutes was obtained from each participant.

5.2 CSI Calibration and Feature Extraction

Once data from each participant were collected, only CSI amplitude was passed to the Breathing and Heart Rate Module in order to be processed for further feature extraction. Fig. 5.2 shows the resultant signal from each calibration step of a single sensitive subcarrier. As can be seen, the final result of the calibration process for breathing signal extraction is a periodic signal.

The result of processing CSI amplitude for obtaining the heartbeat signal is presented in Fig. 5.3. The obtained heartbeat signal remains with noise. This situation, as well as its effect on classification, will be discussed in a next chapter.

Once breathing and heartbeat signals are extracted from CSI amplitude an observation with a total of 160 features is formed for breathing rate classification

Figure 5.1: Experimental scenario for collecting CSI

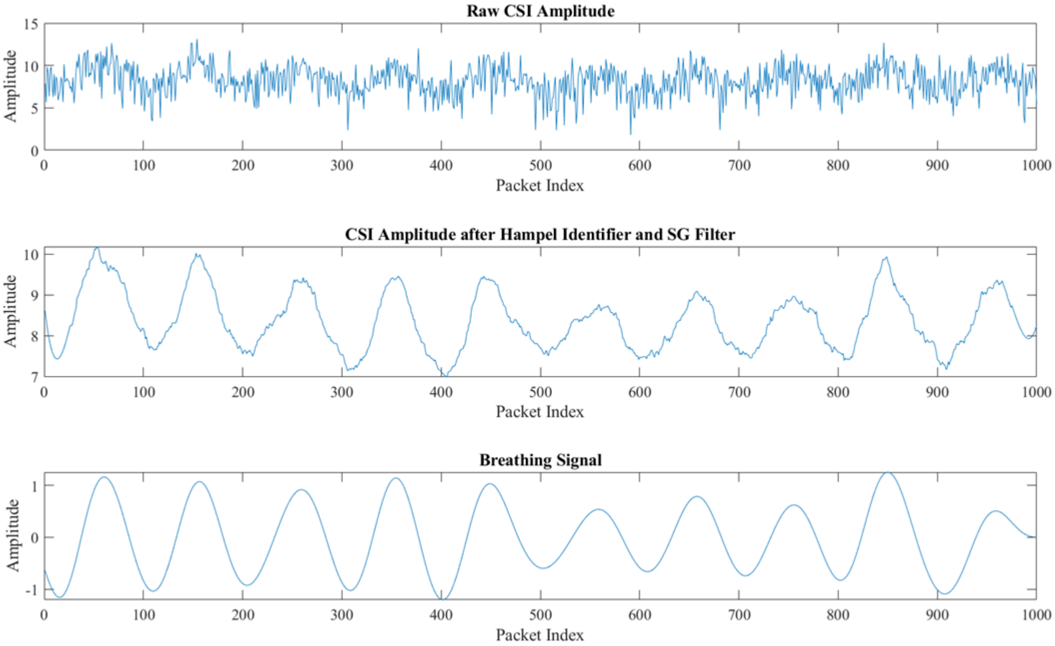


Figure 5.2: Result from each calibration step for breathing signal extraction

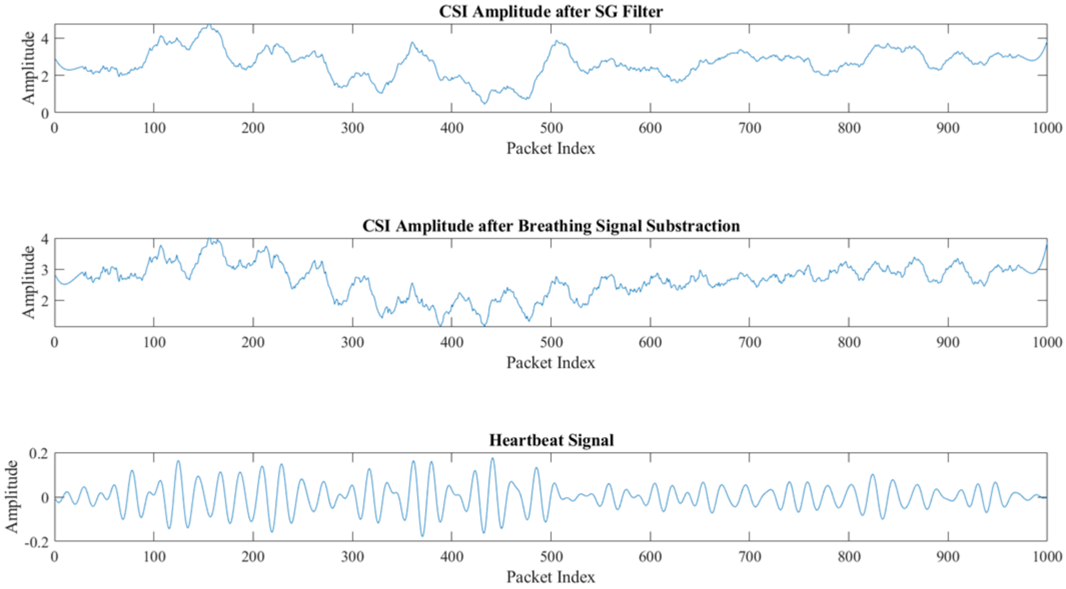


Figure 5.3: Result from each calibration step for heartbeat signal extraction

while an observation of 80 features is formed for heart rate classification. This two observations are obtained for each time window and are later used to build a training and test data set. The training data set contains observations of 13 participants, leaving the observations of 4 participants for the test data set.

Before training the classifiers, a data visualization step was done to analyze if based on the features extracted, observations from each data set are grouped into clusters according to their label.

5.3 Data Visualization

t-SNE with euclidean distance metric was implemented with the purpose of visualizing in two dimensions if observations are clustered according to their labels and therefore expect promising results from the classification algorithms. As shown in Fig. 5.4 (top-left), the observations are grouped into clusters according to the breathing rate range. Although it was expected to form only five well divided clusters, it can be seen that observations labeled as “1” are separated into two clusters primarily divided by the observations labeled as “5”. This situation indicates that it might be possible to divide the observations with a breathing rate below 10 BrPM into two different labels, but, as these clusters have a well defined region in the plane, this situation is not expected to have a negative effect in further classification, so label division will not be done.

Fig. 5.4 (top-right) shows the observations grouped into clusters by using the breathing pattern as label. As can be seen, the three clusters have their own well-defined region in the plane, being easily differentiated from each other. Therefore, the breathing rate pattern classifier is expected to have a promising performance.

At last, t-SNE was also implemented in the heart rate data set, where heart rate range is the class label. As can be seen in Fig. 5.4 (bottom) observations are not grouped as for breathing data set.

In the next chapter will be observed that the K-NN for breathing rate range and pattern classification provided a better result with Hamming distance for breathing rate range classification and with City-block distance for breathing rate pattern classification, therefore a t-SNE visualization with both distance metrics was made, obtaining the projections observed in Fig. 5.5. It can be observed that both distances achieve the task of grouping data with the same label and, for breathing rate range data, the two groups separated labeled as class "1" with euclidean distance now are next to each other.

From this visualization experiment, we could expect to have promising results for both breathing rate range and pattern classification, but for heart rate range classification two questions arise, *does the signal obtained really represents the heartbeats? are the features extracted for heart rate range classification representative?* The answer for both questions will be given in the next chapter.

5.4 Training the Classifiers

Training data set is made up of 13 participants of different ages, sex and body types, therefore it is expected to build a generalized classifier. This set was used to train three different classifiers for each classification task: K-NN, SVM and QDC, which hyperparameter values were found using Bayesian Optimization [72] in order to find the configuration of the model which minimizes the cross-validation loss. For K-NN, the hyperparameters to tune were k, the distance metric, distance weight and data standardization, while for SVM, the hyperparameters were the box constraint, kernel function, kernel scale, polynomial order (in case of a polynomial kernel function) and data standardization. For QDC, hyperparameter tuning was not needed.

Once hyperparameters were tuned using Bayesian Optimization, each classifier was evaluated using a 10-Fold Cross-Validation with the performance metrics of accuracy, recall, precision, specificity and F-score, along with an averaged confusion matrix between folds for each classifier.

K-NN Classifier Tuning and Evaluation

The K-NN classifier hyperparameters values found with Bayesian Optimization for breathing rate range classifications were a $k = 1$, hamming distance as distance

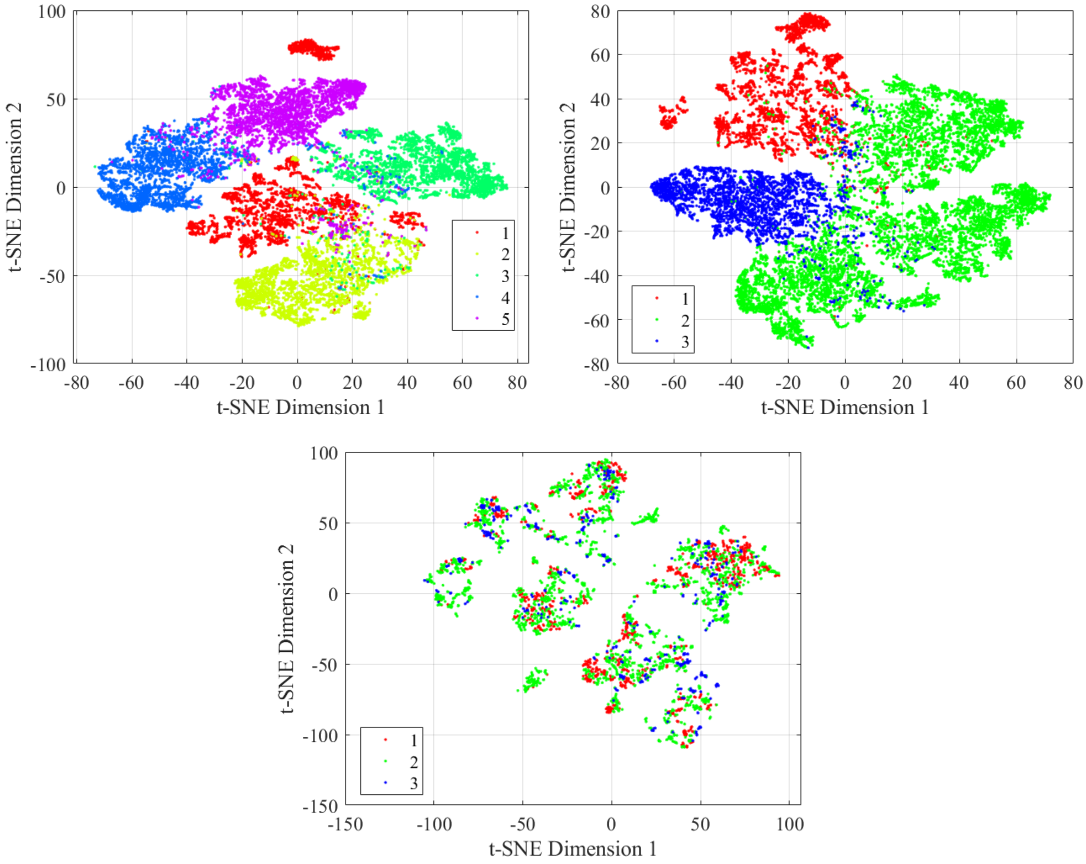


Figure 5.4: Observations grouped in clusters according to different class label. Top-left: Breathing rate range; Top-right: Breathing pattern; Bottom: Heart rate range

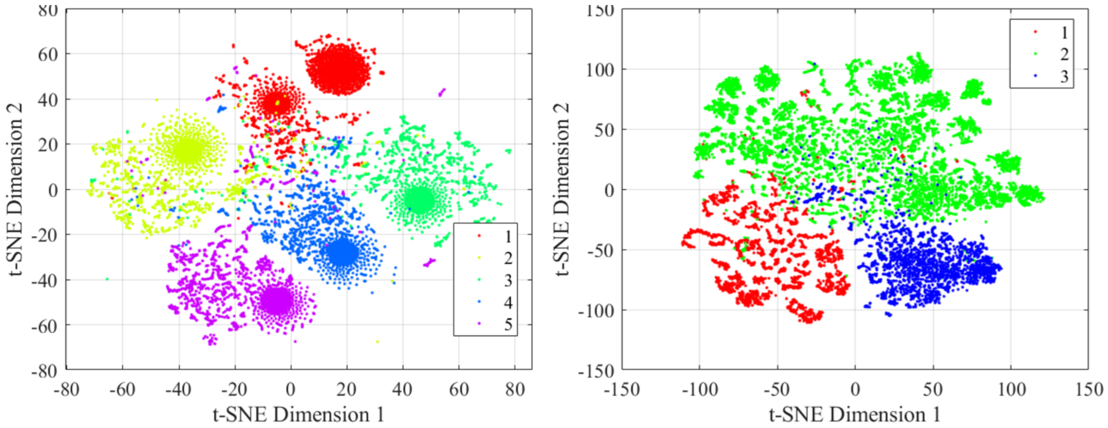


Figure 5.5: t-SNE with different distance metric. Left: Hamming distance in breathing rate range label; Right: City-block distance in breathing rate pattern label

metric, an inverse distance weight and applying data standardization.

Evaluation results obtained by this first classifier were an average accuracy, recall, precision, specificity and F-score between classes of 99.03%, 97.60%, 97.64%, 99.39%, and 97.62% respectively, while a breakdown of these metrics, as well as the confusion matrix averaged across folds are presented in Table 5.2.

Table 5.2: K-NN Evaluation Results for Breathing Rate Range Classification. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	349 ± 0.91	6.3 ± 0.84	0.1 ± 0.1	0.5 ± 0.22	0 ± 0
	2	6 ± 0.54	334 ± 0.60	4.3 ± 0.42	1 ± 0.30	0.1 ± 0.1
	3	1.7 ± 0.30	7.1 ± 0.92	315.4 ± 0.90	2.4 ± 0.40	0.3 ± 0.15
	4	0.8 ± 0.33	1.1 ± 0.35	2.6 ± 0.50	288.5 ± 0.75	1.3 ± 0.26
	5	0.6 ± 0.22	0.7 ± 0.26	0.8 ± 0.30	1.9 ± 0.41	315.1 ± 0.50

Class	Accuracy	Recall	Precision	Specificity	F-score
1	99.03%	98.06%	97.47%	99.29%	97.76%
2	98.38%	96.70%	95.66%	98.83%	96.17%
3	98.82%	96.48%	97.59%	99.41%	97.03%
4	99.29%	98.03%	98.03%	99.57%	98.03%
5	99.65%	98.75%	99.46%	99.87%	99.10%

For breathing pattern classification, hyperparameter tuning resulted in the same values for k and distance metric as for the breathing rate range classifier. The distance weight used for this model was squared-inverted and data standardization was not applied. Table 5.3 shows the evaluation results obtained by this classifier, having an average accuracy, recall, precision, specificity and F-score achieved between classes were of 99%, 98.39%, 98.37%, 99.09%, and 98.38% respectively.

Table 5.3: K-NN Evaluation Results for Breathing Pattern Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	357.1 ± 0.78	8.5 ± 0.76	0 ± 0
	2	9.3 ± 1.20	954.5 ± 1.79	3 ± 0.76
	3	0.7 ± 0.26	3.1 ± 0.55	305.4 ± 0.58

Class	Accuracy	Recall	Precision	Specificity	F-score
1	98.87%	97.68%	97.28%	99.22%	97.48%
2	98.54%	98.73%	98.80%	98.28%	98.76%
3	99.59%	98.77%	99.03%	99.77%	98.90%

At last, K-NN hyperparameters tuning for heart rate classification resulted in a k equals to 6, a city-block distance metric, an inverse distance weight and application of data standardization. Results are presented in Table 5.4, evaluation results show an average accuracy, recall, precision, specificity and F-score between classes of 95.4%, 89%, 93.77%, 95.04%, and 91.07% respectively. These results seem to show that it is possible to achieve heart rate classification, but for ensuring this statement, this classifier needs to be tested using the test data set of the four remaining participants.

Table 5.4: K-NN Evaluation Results for Heart Rate Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	81.1 ± 0.9	8.7 ± 0.97	0.4 ± 0.22
	2	4.2 ± 0.57	199.6 ± 0.6	1.5 ± 0.43
	3	1.3 ± 0.59	7.2 ± 0.73	33.8 ± 0.70

Class	Accuracy	Recall	Precision	Specificity	F-score
1	95.68%	89.92%	93.77%	97.78%	91.74%
2	93.61%	97.22%	92.63%	88.00%	94.87%
3	96.92%	79.93%	94.91%	99.36%	86.62%

QDC Classifier

For breathing rate range classification, QDC presented an average accuracy, recall, precision, specificity and F-score of 97.24%, 93.13%, 93.15%, 98.29%, and 93.02% were obtained between classes respectively. The breakdown of performance metrics and the confusion matrix averaged across folds are presented in Table 5.5.

Compared to K-NN for breathing rate range classification, QDC presents lower values in the five performance metrics used.

Table 5.5: QDC Evaluation Results for Breathing Rate Range Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	332.7 ± 1.89	6.7 ± 0.76	2.1 ± 0.81	6.5 ± 0.85	7.9 ± 1.09
	2	2.2 ± 0.74	316.7 ± 1.65	9.1 ± 0.81	8.6 ± 1.33	8.8 ± 1.15
	3	0.8 ± 0.42	1.8 ± 0.29	297 ± 1.75	17.9 ± 0.9	9.4 ± 1.01
	4	0.1 ± 0.1	0 ± 0	6 ± 1.02	272.5 ± 2.32	16 ± 1
	5	0 ± 0	0 ± 0	0 ± 0	9.1 ± 0.97	310 ± 0.92

Class	Accuracy	Recall	Precision	Specificity	F-score
1	98.40%	93.48%	99.09%	99.76%	96.19%
2	97.73%	91.69%	97.39%	99.34%	94.45%
3	97.13%	90.85%	94.55%	98.69%	92.65%
4	96.09%	92.49%	86.64%	96.88%	89.45%
5	96.88%	97.15%	88.08%	96.82%	92.38%

QDC for breathing pattern classification obtained an average accuracy, recall, precision, specificity and F-score between classes of 95.51%, 92.08%, 92.9%, 95.72%, and 92.43% respectively, which values are lower compared to the ones obtained by K-NN for this classification task. Table 5.6 presents the confusion matrix averaged across folds and the performance metrics obtained for each class for this classifier.

Table 5.6: QDC Evaluation Results for Breathing Pattern Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	325 ± 1.25	36.7 ± 1.1	3.9 ± 0.48
	2	14.3 ± 1.48	921.7 ± 2.56	30.8 ± 1.79
	3	0 ± 0	24.7 ± 1.86	284.5 ± 1.91

Class	Accuracy	Recall	Precision	Specificity	F-score
1	96.66%	88.89%	95.80%	98.88%	92.21%
2	93.51%	95.34%	93.76%	90.90%	94.54%
3	96.38%	92.01%	89.14%	97.40%	90.55%

For heart rate classification, QDC obtained an average accuracy, recall, precision, specificity and F-score of between classes of 86.69%, 80.25%, 83.70%, 90.51%, and 81.24% respectively. Table 5.7 shows the confusion matrix averaged across folds as well as the breakdown of performance metrics per class.

Table 5.7: QDC Evaluation Results for Heart Rate Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	79.3 ± 0.9	10.5 ± 0.87	0.4 ± 0.22
	2	22.7 ± 1.62	178.4 ± 1.55	4.2 ± 0.68
	3	3 ± 0.49	11.4 ± 0.98	27.9 ± 0.98

Class	Accuracy	Recall	Precision	Specificity	F-score
1	89.16%	87.92%	75.71%	89.62%	81.29%
2	85.55%	86.90%	89.10%	83.47%	87.96%
3	94.38%	65.95%	86.31%	98.44%	74.48%

SVM Classifier Tuning and Evaluation

Hyperparameters tuning of SVM for breathing rate range classification gave as outcome that the values of hyperparameters that result in the best model according to our criteria (minimum loss) are a box constraint of 996.07, a polynomial kernel function of 2^{nd} order with an scale of 1 and applying data standardization.

It was expected that SVM for breathing rate range classification to surpass both K-NN and QDC, but it was not the case, only surpassing QDC with an average accuracy, recall, precision, specificity and F-score of 98.20%, 95.46%, 95.49%, 98.88%, and 95.46% between classes for breathing rate range classification. A breakdown of these metrics for breathing rate range as label is presented in Table 5.8.

For breathing rate pattern classification, the hyperparameters values obtained were a box constraint of 0.001 with a linear kernel function with a scale of 1 and with data standardization. The SVM with this hyperparameter configuration presented an average accuracy, recall, precision, specificity and F-score of 96.19%, 92.06%, 95.13%, 95.89%, and 93.5% between classes, while a breakdown of performance metrics for each class is presented in Table 5.9, having a slightly better performance if compared to QDC for breathing pattern classification in almost every performance metric for the three possible classes.

The hyperparameters defined for SVM for heart rate classification were a box constraint of 0.001 with a polynomial kernel function of 2^{nd} order, a kernel scale of 1 and with data standardization. As with the two previous classification methods used for heart rate range classification, this classifier presents a lower performance if compared to breathing classification tasks, but a higher performance if compared to the training results obtained by QDC for the heart rate classification. Training evaluation results present an average accuracy, recall, precision, specificity and F-

Table 5.8: SVM Evaluation Results for Breathing Rate Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

				Outputs		
	Class	1	2	3	4	5
	1	344.8 ± 1.03	7.4 ± 0.58	1.9 ± 0.46	0.8 ± 0.25	1 ± 0.33
	2	6.2 ± 0.47	330.6 ± 1.13	6.2 ± 0.8	1.4 ± 0.27	1 ± 0.26
Targets	3	1.5 ± 0.43	6.5 ± 0.58	312.6 ± 1.03	4.8 ± 0.47	1.5 ± 0.40
	4	0.3 ± 0.15	0.7 ± 0.21	12 ± 1.03	278.2 ± 0.95	3.1 ± 0.60
	5	0.2 ± 0.2	0.7 ± 0.26	2 ± 0.6	14.4 ± 1.07	301.8 ± 1.19

Class	Accuracy	Recall	Precision	Specificity	F-score
1	98.82%	96.88%	97.68%	99.36%	97.28%
2	98.17%	95.72%	95.58%	98.82%	95.64%
3	97.78%	95.63%	93.44%	98.32%	94.50%
4	97.72%	94.53%	92.88%	98.41%	93.69%
5	98.54%	94.58%	97.87%	99.50%	96.19%

Table 5.9: SVM Evaluation Results for Breathing Pattern Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

			Outputs	
	Class	1	2	3
	1	331.9 ± 2.01	33.7 ± 1.96	0 ± 0
Targets	2	6.5 ± 0.97	944.6 ± 1.72	15.7 ± 1.43
	3	0.1 ± 0.1	37.9 ± 2.84	271.2 ± 2.8

Class	Accuracy	Recall	Precision	Specificity	F-score
1	97.55%	90.78%	98.06%	99.48%	94.27%
2	94.29%	97.70%	92.96%	89.39%	95.27%
3	96.73%	87.71%	94.56%	98.82%	90.97%

score of 90.29%, 80.4%, 84.55%, 90.12%, and 82.14% between classes. A complete breakdown of performance metric for each class is presented in Table 5.10.

Observing the results from the three different models used to build the classifiers, K-NN seems to outperform SVM and QDC in every classification task, while SVM slightly outperforms QDC. With this first evaluation, we expect to get promising results from the trained classifiers while working with the test data set, which will give us an idea of how our experimental system will behave if deployed in a real-time scenario.

Table 5.10: SVM Evaluation Results for Heart Rate Classification.
Top: Confusion Matrix; Bottom: Performance Metrics

			Outputs	
	Class	1	2	3
Targets	1	70.6 ± 1.14	18.4 ± 1.29	1.2 ± 0.36
	2	12.7 ± 1.39	187.8 ± 1.13	4.7 ± 0.68
	3	1.6 ± 0.37	10.5 ± 1.28	30.2 ± 1.17

Class	Accuracy	Recall	Precision	Specificity	F-score
1	89.94%	78.28%	83.23%	94.18%	80.61%
2	86.26%	91.48%	86.69%	78.19%	89.01%
3	94.67%	71.44%	83.75%	98.00%	76.82%

6. Results

In this section, we present the results obtained from the classifiers using a test data set. This set consists of 5180 observations for breathing rate classification and 1040 for heart rate classification collected from participants that were not in the training group and were collected at a different date. We will also give answer to the previously enunciated questions along with a new one, *is it viable to apply a dimensionality reduction algorithm for reducing the amount of features to be processed by the classifiers?*.

As we intend to deploy this experimental system in a real-time scenario, the response time of the classifiers is a thing to keep in mind. Therefore, in this section we introduce the response time as a new performance metric for the classifier.

6.1 Breathing Rate Range Classification

Table 6.1 shows the classification results obtained from K-NN classifier with the test data set. As can be seen, it presents an exceptional performance with a mean accuracy, recall, precision, specificity and F-score between classes of 99.18%, 97.95%, 97.96%, 99.49%, and 97.95% respectively, proving that this classifier gives a reliable response even with new observations for breathing rate range classification. The response time from the prediction of the 5180 observations was of 2.696 s.

Results obtained from QDC show a lower performance if compared to K-NN, specifically for class 4 and 5 which recall, precision and F-score values are considerably lower. The mean accuracy, recall, precision, specificity and F-score obtained between classes were 91.35%, 78.55%, 78.81%, 94.62%, and 78.43% respectively, while a complete breakdown of the performance metrics can be seen in Table 6.2. The only advantage that this classifier presents when compared to K-NN is the response time, which is of 0.070 s for all test data set observations, almost 40 times faster than K-NN.

Table 6.1: KNN for Breathing Rate Range Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	1007	4	0	8	0
	2	22	1038	0	0	0
	3	1	7	1029	4	0
	4	9	0	1	1028	11
	5	16	14	3	6	972

Class	Accuracy	Recall	Precision	Specificity	F-score
1	98.84%	98.82%	95.45%	98.85%	97.11%
2	99.09%	97.92%	97.65%	99.39%	97.79%
3	99.69%	98.85%	99.61%	99.90%	99.23%
4	99.25%	98.00%	98.28%	99.56%	98.14%
5	99.03%	96.14%	98.80%	99.74%	97.49%

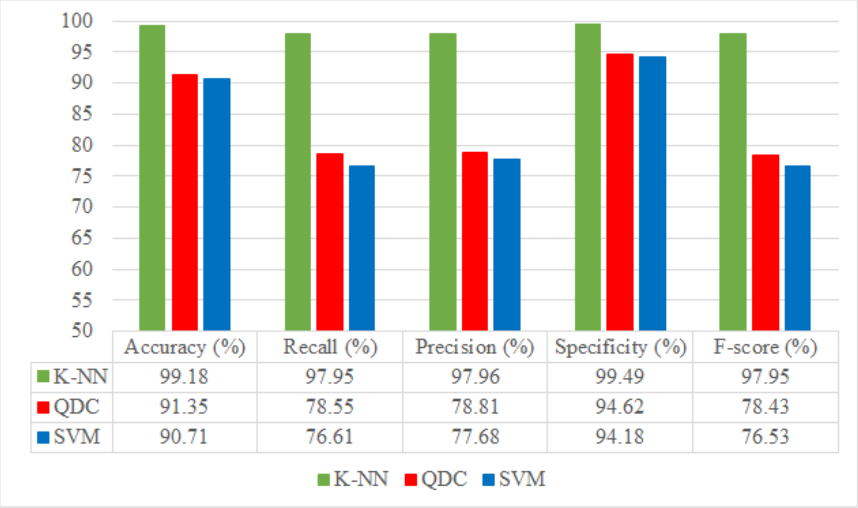
Table 6.2: QDC for Breathing Rate Range Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	842	83	31	48	15
	2	77	841	120	11	11
	3	1	99	913	15	13
	4	0	0	103	646	300
	5	0	0	0	187	824

Class	Accuracy	Recall	Precision	Specificity	F-score
1	95.08%	82.63%	91.52%	98.13%	86.85%
2	92.26%	79.34%	82.21%	95.58%	80.75%
3	92.36%	87.70%	78.23%	93.86%	82.70%
4	87.18%	61.58%	71.22%	93.68%	66.05%
5	89.85%	81.50%	70.85%	91.87%	75.80%

Finally, SVM for breathing rate range classification in test data set shows a mean accuracy, recall, precision, specificity and F-score between classes of 90.71%, 76.71%, 77.68%, 94.18%, and 76.53%. Compared to QDC, SVM shows an slight improvement for class 1 and 2, but the recall and F-score for class 4 and 5 decreases considerably as can be seen in Table 6.3.In terms of response time, SVM is also outperformed by QDC, having a total response time of 1.13 s. A bar graph comparing the performance metrics of accuracy, recall, precision, specificity and F-score obtained by the three classifiers is presented in Fig. 6.1.

In addition, we compute the ROC curves for each classifier, obtaining that K-NN presents the greatest area under the curve (AUC) for each class as can be



	Accuracy (%)	Recall (%)	Precision (%)	Specificity (%)	F-score (%)
🟩 K-NN	99.18	97.95	97.96	99.49	97.95
🟥 QDC	91.35	78.55	78.81	94.62	78.43
🟦 SVM	90.71	76.61	77.68	94.18	76.53

Figure 6.1: Results for Breathing Rate Range Classification

Table 6.3: SVM for Breathing Rate Range Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	873	91	3	34	18
	2	57	928	65	6	4
	3	2	92	945	2	0
	4	11	17	316	607	98
	5	5	2	31	349	624

Class	Accuracy	Recall	Precision	Specificity	F-score
1	95.73%	85.67%	92.09%	98.20%	88.76%
2	93.55%	87.55%	82.12%	95.10%	84.75%
3	90.14%	90.78%	69.49%	89.97%	78.72%
4	83.92%	57.86%	60.82%	90.53%	59.31%
5	90.21%	61.72%	83.87%	97.12%	71.10%

seen in Fig. 6.2. Meanwhile, the ROC curve for class 4 obtained by SVM and QDC presents a smaller AUC if compared to other classes, demonstrating that both classifiers face difficulties with the class that corresponds to a breathing rate between 17 and 19 BrPM.

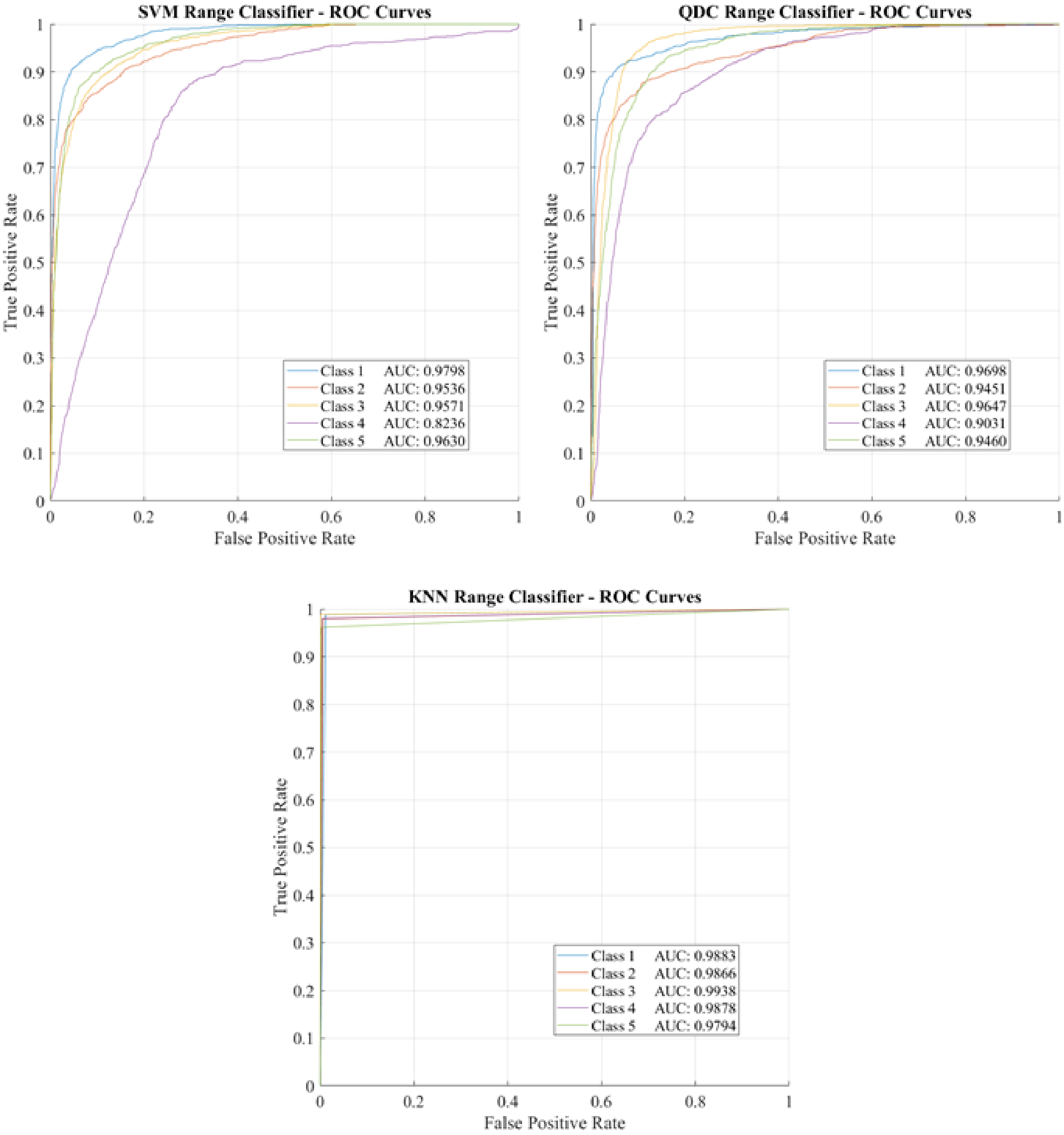


Figure 6.2: ROC Curves for the three Breathing Rate Range Classifiers

6.2 Breathing Pattern Classification

Starting with K-NN, it obtained a mean accuracy, recall, precision, specificity and F-score of 98.60%, 97.59%, 97.16%, 95.85%, and 97.37% respectively, whose performance metrics per class can be seen in Table 6.4. The response time obtained is similar to the one obtained from K-NN for breathing rate range classification, with a total response time of 2.562 s. With this response time, it is expected to be the slowest if compared to the other two classifiers.

Table 6.4: KNN for Breathing Pattern Classification with Test Data Set.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
	1	1023	11	0
Targets	2	17	3107	37
	3	22	22	941

Class	Accuracy	Recall	Precision	Specificity	F-score
1	99.03%	98.94%	96.33%	90.06%	97.61%
2	98.32%	98.29%	98.95%	98.37%	98.62%
3	98.44%	95.53%	96.22%	99.12%	95.87%

For QDC, results presented in Table 6.5 show a lower performance when compared to K-NN with a low recall for class 1 and 3, as well as a low specificity for class 2. In brief, QDC presents a mean accuracy, recall, precision, specificity and F-score between classes of 86.72%, 70.78%, 83.17%, 84.87%, and 75.16% respectively. The only advantage of QDC over K-NN, that it is also presented for breathing rate range classification, is the response time, which is of 0.06 s.

Table 6.5: QDC for Breathing Pattern Classification with Test Data Set.
Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
	1	637	396	1
Targets	2	72	2961	128
	3	0	423	562

Class	Accuracy	Recall	Precision	Specificity	F-score
1	90.95%	61.61%	89.84%	98.26%	73.09%
2	80.33%	93.67%	78.33%	59.44%	85.32%
3	89.34%	57.06%	81.33%	96.92%	67.06%

Table 6.6 presents the results for breathing pattern classification obtained by SVM, whose performance is better than QDC, but not as good as K-NN. It shows a mean accuracy, recall, precision, specificity and F-score between classes of 93.35%, 89.32%, 88.78%, 93.93%, and 88.94% respectively. The response time is of 1.228 s, faster than K-NN but not as fast as QDC.

In order to easily compare the performance of the three different classifiers for breathing pattern classification the performance metrics values of accuracy, recall, precision, specificity and F-score is shown in Fig. 6.3.

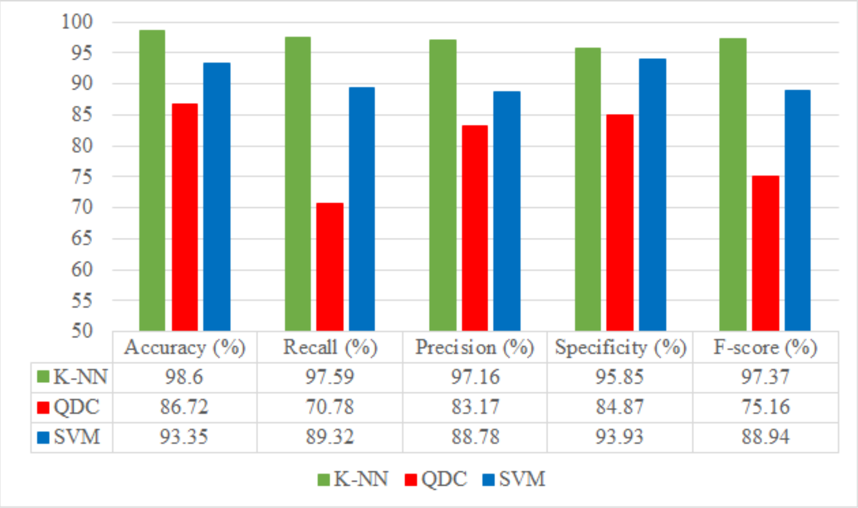


Figure 6.3: Results for Breathing Rate Pattern Classification

Table 6.6: SVM for Breathing Pattern Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	971	63	0
	2	7	2873	281
	3	0	166	819

Class	Accuracy	Recall	Precision	Specificity	F-score
1	98.65%	93.91%	99.28%	99.83%	96.52%
2	90.02%	90.89%	92.62%	88.66%	91.75%
3	91.37%	83.15%	74.45%	93.30%	78.56%

At last, ROC curves were also computed for this classification task, obtaining that K-NN is the classifier with the greatest AUC for the three different classes, followed by SVM with AUCs of 0.9966, 0.9410 and 0.9591 as seen in Fig. 6.4 which clearly outperform QDC.

6.3 Heart Rate Classification

Finally, heart rate classification did not go as expected based on the results obtained from training evaluation. Results obtained from K-NN, which is the classifier that obtained the best performance for the previous classification tasks, presented a mean accuracy, recall, precision, specificity and F-score between classes of 56.55%, 35.91%, 40.28%, 68.03%, and 30.91%. A complete breakdown of this

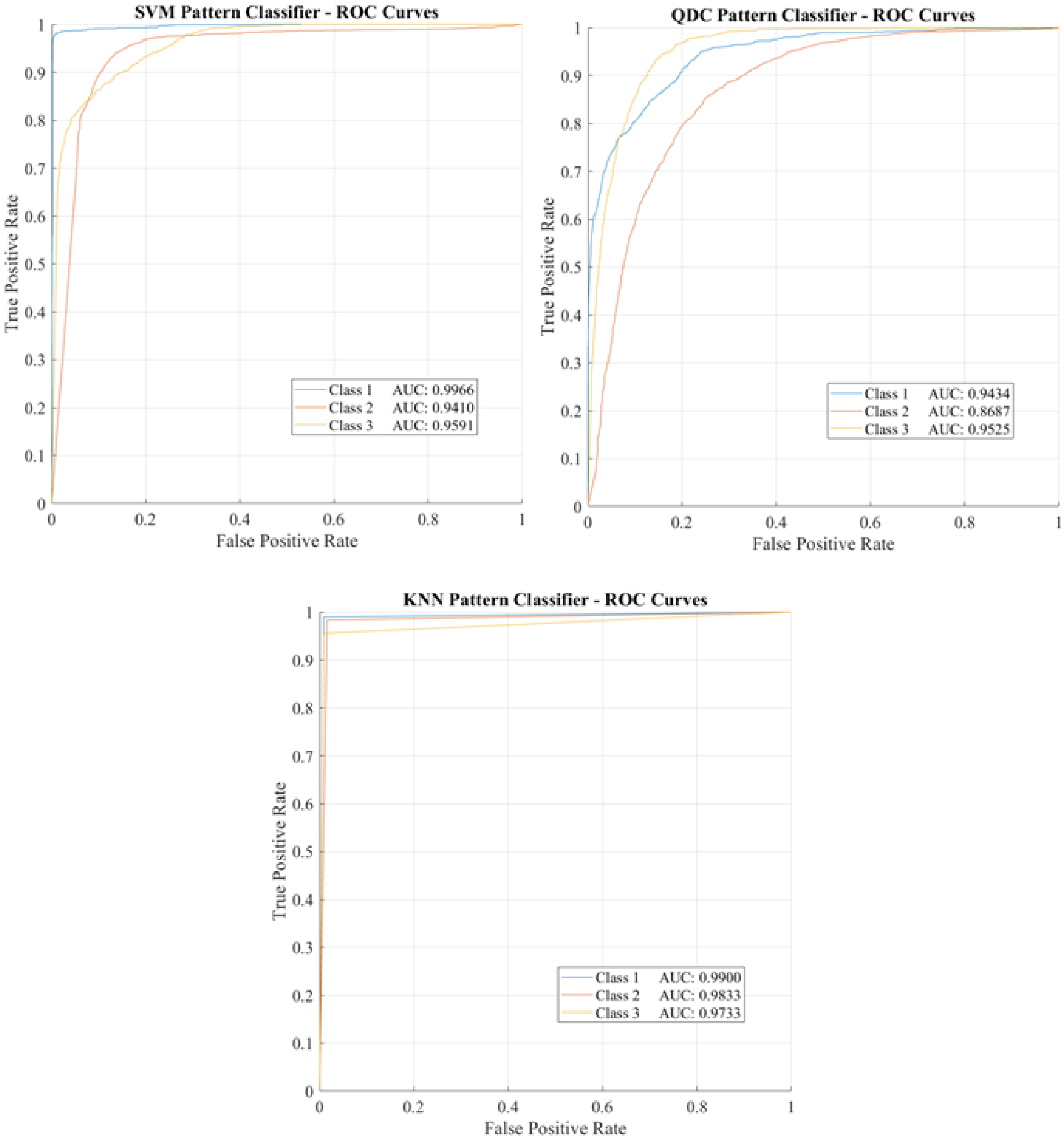


Figure 6.4: ROC Curves for the three Breathing Pattern Classifiers

performance metrics can be seen in Table 6.7. This performance shows that we can not rely on the response of this classifier, situation that will be also repeated for the next two classifiers.

Table 6.7: KNN Performance for Heart Rate Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	84	259	26
	2	66	227	22
	3	53	257	46

Class	Accuracy	Recall	Precision	Specificity	F-score
1	61.15%	22.76%	41.38%	82.27%	29.37%
2	41.92%	72.06%	30.55%	28.83%	42.91%
3	66.58%	12.92%	48.92%	92.98%	20.44%

As same as K-NN, QDC presents a poor performance for heart rate classification, with a mean accuracy, recall, precision, specificity and F-score between classes of 55%, 34.61%, 37.01%, 67.23% and 26.16% respectively while Table 6.8 shows the complete breakdown of these metrics.

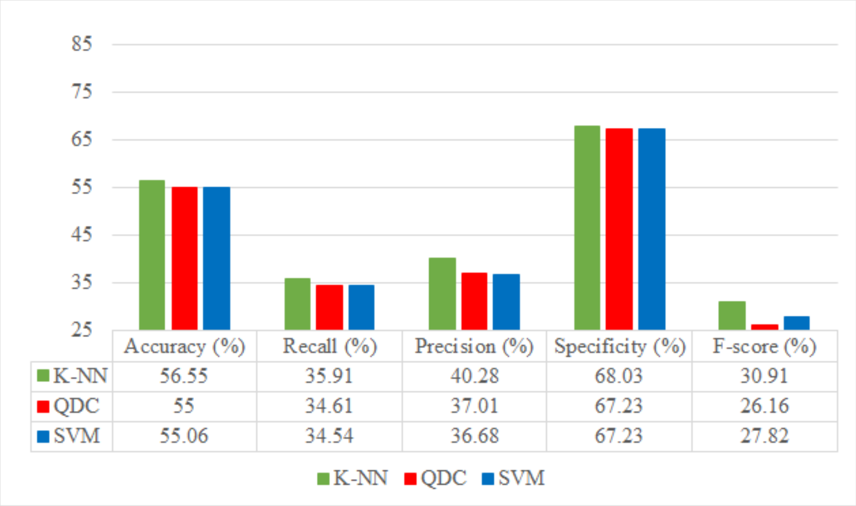
Table 6.8: QDC Performance for Heart Rate Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	53	296	20
	2	47	258	10
	3	65	264	27

Class	Accuracy	Recall	Precision	Specificity	F-score
1	58.85%	14.36%	32.12%	83.31%	19.85%
2	40.67%	81.90%	31.54%	22.76%	45.54%
3	65.48%	7.58%	47.37%	95.61%	13.08%

SVM did not present better results, as can be seen in Table 6.9, presenting a mean accuracy, recall, precision, specificity and F-score of 55.06%, 34.54%, 36.68%, 67.23% and 27.82% respectively. A bar graph comparing the performance metrics previously mentioned of the three different classifiers is shown in Fig. 6.5.

ROC curves in Fig. 6.6 show that the three classifiers tend to give a random or a constant class as response as seen in the different confusion matrices obtained (see Table 6.7,6.8 and 6.9) where the three classifiers mostly labeled the observations as class 2. This poor performance is due to two reasons, first, we can not vary



	Accuracy (%)	Recall (%)	Precision (%)	Specificity (%)	F-score (%)
🟩K-NN	56.55	35.91	40.28	68.03	30.91
🟥QDC	55	34.61	37.01	67.23	26.16
🟦SVM	55.06	34.54	36.68	67.23	27.82

Figure 6.5: Results for Heart Rate Range Classification

Table 6.9: SVM Performance for Heart Rate Classification with Test Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs		
	Class	1	2	3
Targets	1	50	288	31
	2	52	243	20
	3	56	254	46

Class	Accuracy	Recall	Precision	Specificity	F-score
1	58.94%	13.55%	31.65%	83.90%	18.98%
2	40.96%	77.14%	30.96%	25.24%	44.18%
3	65.29%	12.92%	47.42%	92.54%	20.31%

our heart rate at will at rest, which will lead to have unbalanced data for each class as can be seen in evaluation results from training phase where most of the data collected is labeled as class 2, which is a heart rate range between 60 and 79 BPM, a normal rate at rest, but less than the half are class 1 (below normal heart rate at rest) and we got even less data labeled as class 3 (above 80 BPM), which was intended to be an elevated heart rate (above 100 BPM), but we did not get data from that heart rate range from neither of our participants. We could have expanded our class 3 range to be above 70 BPM and keep our class 2 in a range of 60 to 70 BPM to have a more balanced number of observations per class, but before we decided to do that, a question arises which answer is the second reason: are we really obtaining the heartbeat signal and therefore, obtaining particular feature values from each heart rate range?

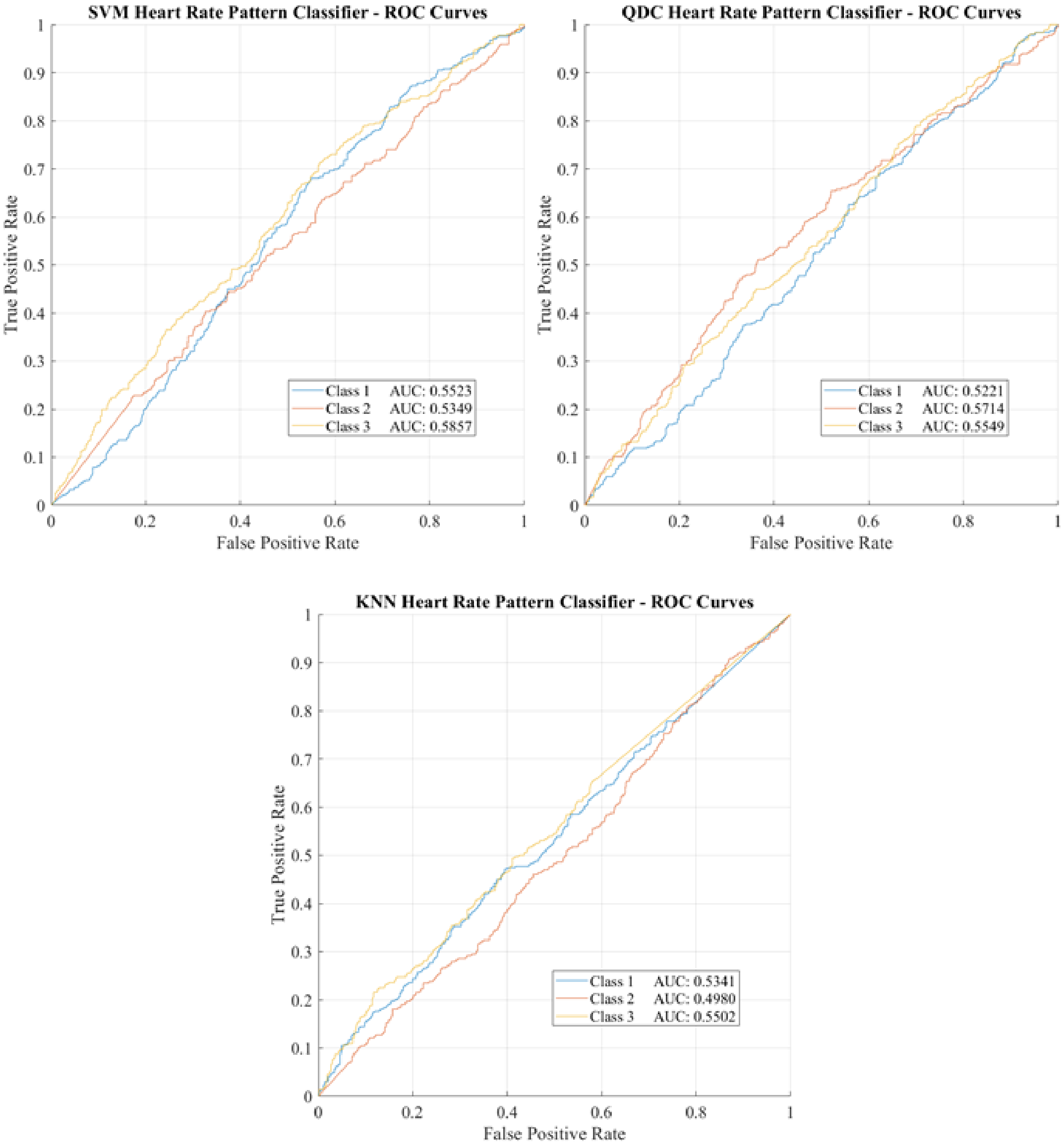


Figure 6.6: ROC Curves for the three Heart Rate Pattern Classifiers

6.4 Processing Heart Rate Data from Faraday Cage

To answer the previous stated questions, the experimental system was isolated from potential interfering RF signals from external sources. By isolating our system, we expect to discard the possibility that an interfering RF signal (from external sources or due to reflections with environment changes) has a larger magnitude than the heartbeat signal that we want to identify. To achieve this, we collected data inside a Faraday Cage to obtain what is assumed to be the heartbeat signal. A single participant was positioned at the center of the cage, and the equipment was set with the same configuration described in chapter 5 to collect data. Once sufficient data was available, the filters used for calibration were applied, and spectral analysis was performed.

A similar behavior from the spectral analyses performed in the Laboratory of Data Science and Engineering and in the Faraday Cage was observed. There were no significant peaks at the expected heart rate frequencies between 1.1 and 1.15 Hz (corresponding to heart rates between 66 and 70 BPM). Instead, we found significant peaks before 1 Hz and after 1.4 Hz for the Faraday Cage experiment as seen in Fig. 6.7, where the normalized magnitude spectrum obtained in the Laboratory of Data Science and Engineering (also shown) has the same behavior.

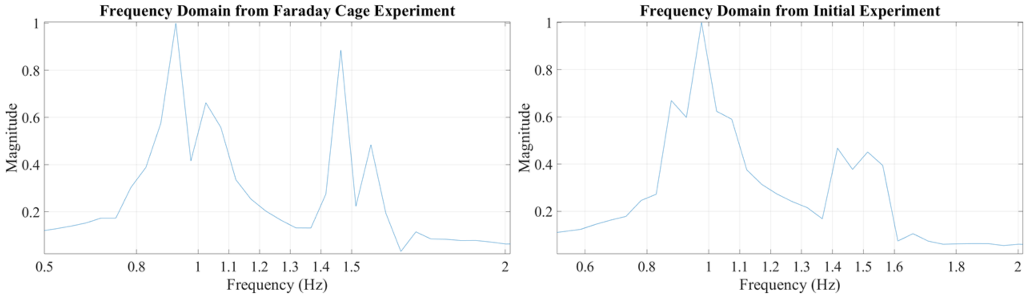


Figure 6.7: Frequency domain obtained with FFT for the two experimental scenarios. Left: Faraday Cage Experiment Scenario; Right: Initial Experiment Scenario

Based on this result, we can deduce that those frequency peaks are generated by our own equipment, hence, giving answer to the previous questions presented, we are being limited by our own equipment for obtaining a reliable heartbeat signal, and therefore the feature values obtained are really characterizing the noise. A solution for this might be using directional antennas such as in [73], where they also stated that using an antenna array will be more beneficial.

6.5 Reducing Data Dimensionality

Finally, we did a final experiment in order to enhance the response time by applying a dimensionality reduction algorithm. For this, we use Principal Component Analysis [61] for extracting the first three principal components of the training and test breathing rate range dataset. In Fig. 6.8 is presented the result obtained from applying PCA to the training dataset, where as in t-SNE, data from each label are grouped in a well-formed region of the 3-Dimensional Space. The reason of using PCA instead of t-SNE is that PCA can be easily used for new incoming data by saving the mean, standard deviation and the eigenvectors of previous entries.

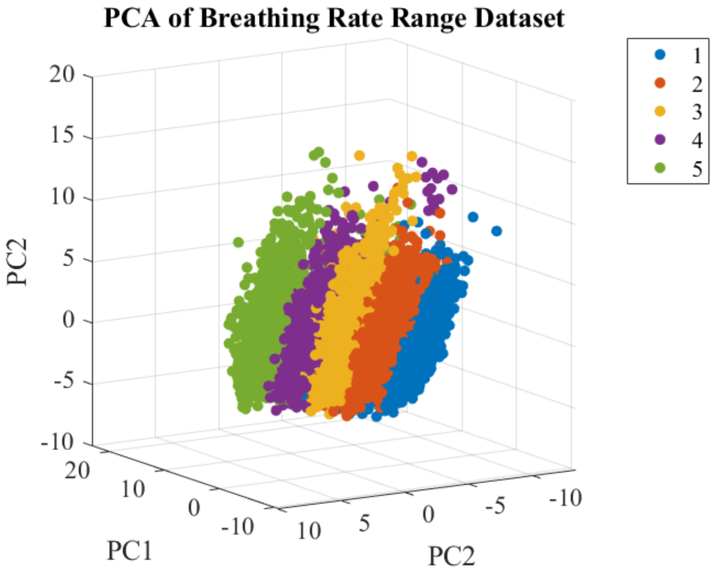


Figure 6.8: Space obtained with PCA for Breathing Rate Range Data Set

Once the three principal components of the training data set were obtained, they were used for training three classifiers whose hyperparameters were defined by Bayesian Optimization. The mean, standard deviation and eigenvectors obtained for reducing dimensionality of the training data set were used for obtaining the three principal components of the test data set.

For KNN with chebychev distance, $k = 33$, with an inverse distance weight and with data standardization, an average accuracy, recall, precision, specificity and F-score between classes of 92.35%, 80.93%, 81.07%, 95.20%, and 80.49% respectively (see Table 6.10 for a breakdown of these performance metrics) were obtained from the test data set. This is a significant lower performance compared to KNN for breathing rate range classification without PCA, but, as expected, the response time was improved with only 0.07 s for classifying the whole data set.

QDC obtained an average accuracy, recall, precision, specificity and F-score between classes of 92.81%, 82.10%, 82.56%, 95.51%, and 81.49% with a response time of 0.047s, being faster than KNN and an improvement if compared to its version without PCA, but its recall and F-score for class 4 are worse.

The best hyperparameters for SVM found with Bayesian Optimization are a third order polynomial kernel function with a box constraint of 0.1069 and with data standardization. With this configuration, the PCA version of SVM presented

Table 6.10: KNN for Breathing Rate Range Classification with PCA Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	853	162	4	0	0
	2	44	959	56	0	1
	3	1	15	940	84	1
	4	0	2	200	579	268
	5	0	0	24	128	859

Class	Accuracy	Recall	Precision	Specificity	F-score
1	95.93%	83.71%	94.99%	98.82%	88.99%
2	94.59%	90.47%	84.27%	95.66%	87.26%
3	92.57%	90.30%	76.80%	93.14%	83.00%
4	86.83%	55.20%	73.20%	94.87%	62.93%
5	91.85%	84.97%	76.06%	93.52%	80.28%

Table 6.11: QDC for Breathing Rate Range Classification with PCA Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
Targets	1	842	174	3	0	0
	2	61	940	59	0	0
	3	1	6	974	60	0
	4	0	1	150	575	323
	5	0	2	15	76	918

Class	Accuracy	Recall	Precision	Specificity	F-score
1	95.39%	82.63%	93.14%	98.51%	87.57%
2	94.15%	88.68%	83.70%	95.56%	86.12%
3	94.32%	93.56%	81.10%	94.52%	86.89%
4	88.22%	54.81%	80.87%	96.71%	65.34%
5	91.97%	90.80%	73.97%	92.25%	81.53%

an average accuracy, recall, precision, specificity and F-score between classes of 92.79%, 81.99%, 82.43%, 95.49%, and 81.45%. A breakdown of these metrics is presented in Table 6.12. This version presents a better performance than its previous version, but recall for class 4 and precision for class 5 values are slightly lower. The response time of this classifier is slower than KNN and QDC with 0.169 s for the whole PCA test data set, but considerably faster compared to its version without PCA.

ROC Curves for these classifiers version were also computed and are presented in Fig. 6.9. As can be seen, SVM and QDC with PCA are better classifiers that their versions without PCA based on the AUCs (see Fig. 6.2, but KNN presents

Table 6.12: SVM for Breathing Rate Range Classification with PCA Data Set. Top: Confusion Matrix; Bottom: Performance Metrics

		Outputs				
	Class	1	2	3	4	5
	1	831	185	3	0	0
	2	36	975	47	0	2
Targets	3	1	9	992	38	1
	4	0	1	213	581	254
	5	0	1	18	125	867

Class	Accuracy	Recall	Precision	Specificity	F-score
1	95.66%	81.55%	95.74%	99.11%	88.08%
2	94.58%	91.98%	83.26%	95.24%	87.40%
3	93.63%	95.29%	77.93%	93.21%	85.74%
4	87.82%	55.39%	78.09%	96.05%	64.81%
5	92.26%	85.76%	77.14%	93.84%	81.22%

lower values than its version without data dimensionality reduction, demonstrating that if we want the classifier with the greatest performance, the best option is KNN with the 160 features.

6.6 The Hilbert Transform-based Subcarrier Selection Algorithm

A low-pass filter with a f_c of 2 Hz and a detrend method which consisted in subtracting the mean value of each CSI amplitude subcarrier were applied in a time window of 40 seconds in order to locate if in one of the IMFs or modes, breathing can be identified along with the instantaneous frequency for each subcarrier. It is expected that if breathing signal can be obtained by this method, the Hilbert spectrum for a constant breathing rate would be a straight line or at most present small variations around the breathing rate frequency. 8 modes were extracted from a breathing sample of 15 BrPM and it can be observed that the breathing signal from a single subcarrier is located in the 7_{th} mode and its Hilbert spectrum shows small variations around 0.25 Hz (15 BrPM) with higher energy if compared to the Hilbert spectrum of the 6_{th} mode as seen in Fig. 6.10.

The same procedure was followed for a breathing sample of 21 BrPM, which results showed that for this sample, breathing signal can be also located in the 6_{th} mode as shown in Fig. 6.11, in which its Hilbert spectrum shows that the instantaneous frequency variates around 0.35 Hz (21 BrPM) and it has more energy if compared to the Hilbert spectrum of the 7_{th}. Based on these results, it can be concluded that modes which can be identified as the breathing signal have more

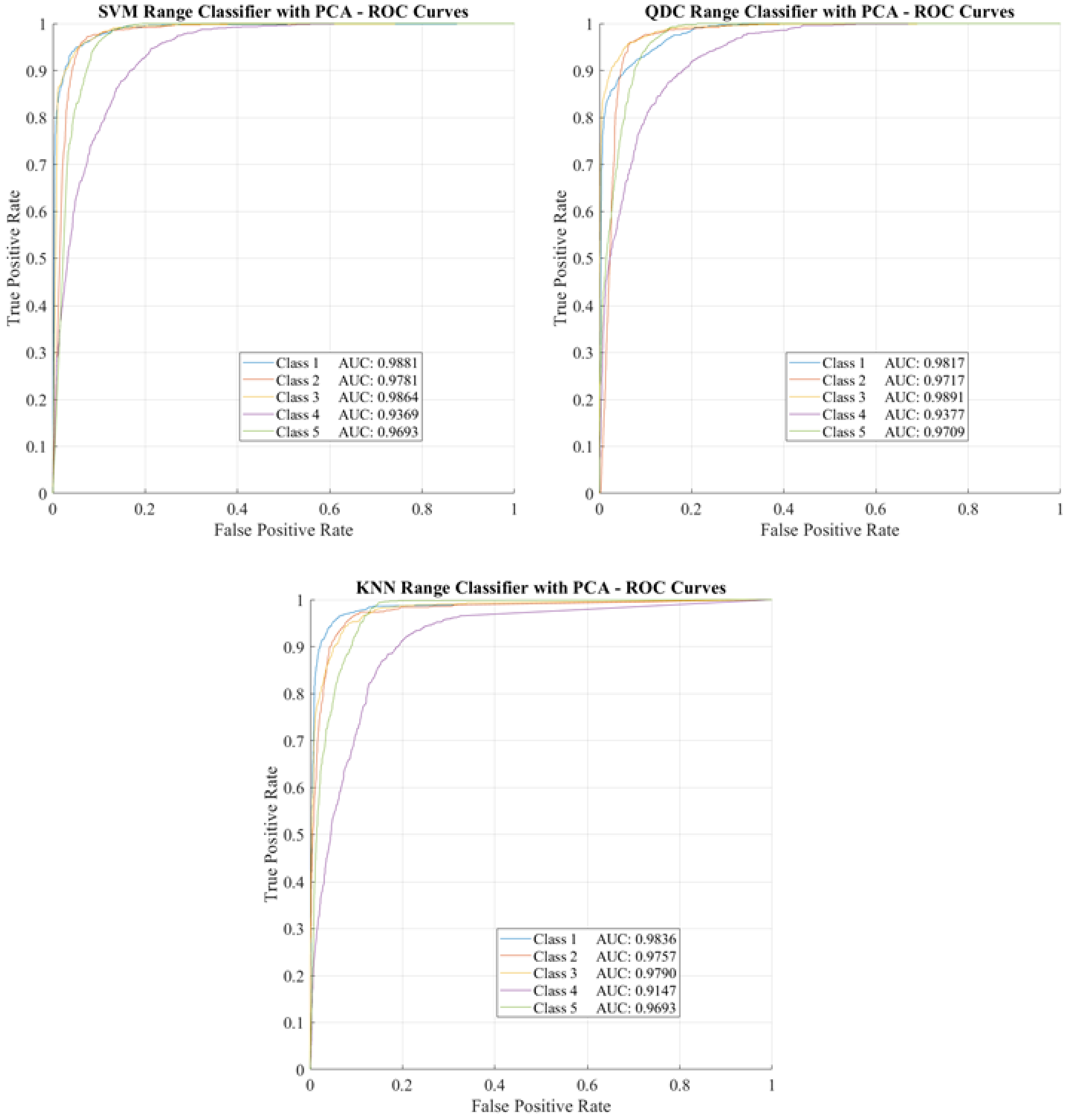


Figure 6.9: ROC Curves for the three different classifiers for PCA Dataset

energy in the Hilbert spectrum around the frequency range in which breathing rate is normally located compared to other modes in the same range.

From this experiment, the idea of a subcarrier selection method based on Hilbert spectrum arose. With the first time window of 40 seconds of CSI amplitudes, VMD is applied to every subcarrier for extracting 8 modes from which Hilbert spectrum is obtained. Next, in order to find in which mode breathing signal is located, the mode with highest energy around the normal breathing frequency range in the time window is selected. Finally, for selecting the sensitive subcar-

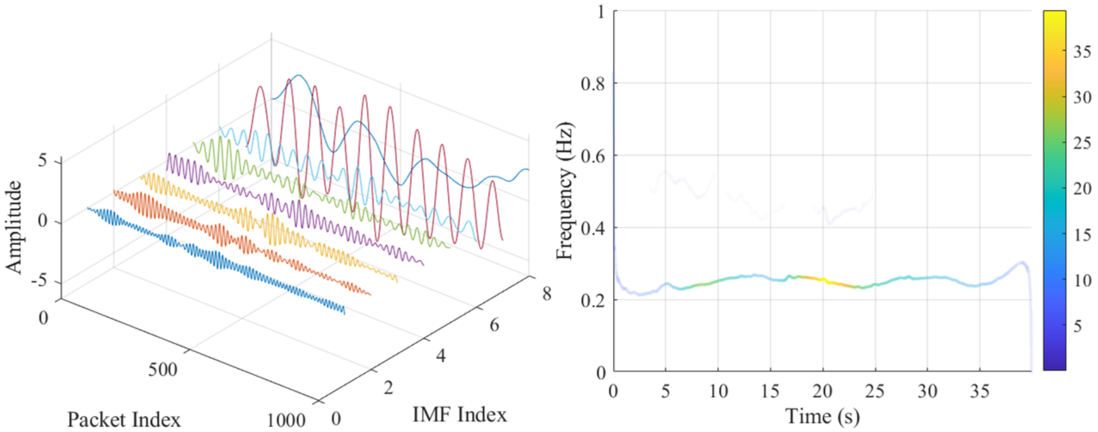


Figure 6.10: Results of applying VMD and Hilbert Transform on VMDs on a single subcarrier of a 15 BrPM sample. Left: IMFs obtained by VMD; Right: Hilbert Spectrum of the 6_{th} and 7_{th} IMF

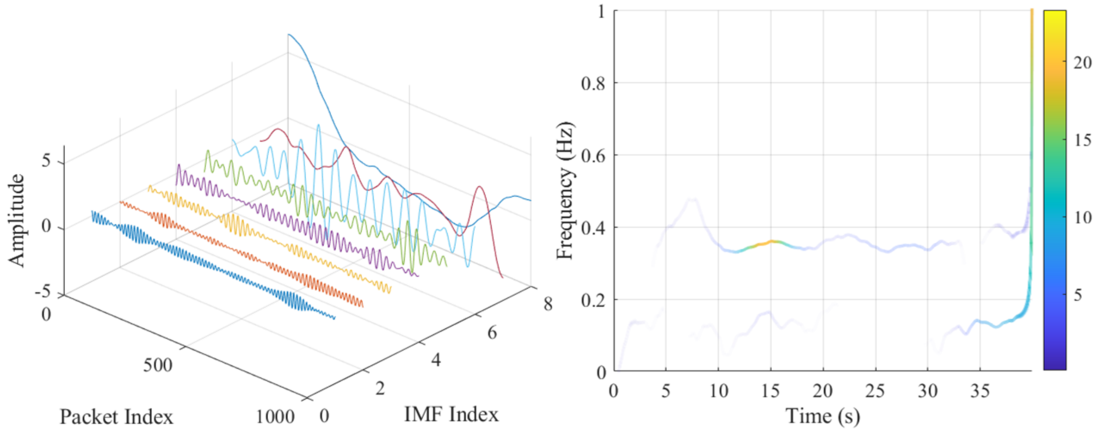


Figure 6.11: Results of applying VMD and Hilbert Transform on VMDs on a single subcarrier of a 21 BrPM sample. Left: IMFs obtained by VMD; Right: Hilbert Spectrum of the 6_{th} and 7_{th} IMF

riers, the variance of instantaneous frequencies of the previous selected modes is calculated, and knowing that breathing rate changes do not occur immediately, the sensitive subcarriers are the ones with minimum variance of instantaneous frequencies. The algorithm for this sensitive subcarrier selection method using MATLAB functions is presented in Algorithm 2.

For evaluating the performance of this proposed method, the first breathing rate estimation based on the frequency with highest amplitude obtained by using FFT was calculated on subcarriers selected by this method from the same CSI data collected from the 17 participants. Using this proposed method, the root mean square error (RMSE) was of 1.59 BrPM, while for the common variance subcarrier selection method the RMSE obtained was of 1.90 BrPM.

Algorithm 2: Algorithm for selecting sensitive subcarriers based on Hilbert Transform using MATLAB functions

```
Data: n, numIMFs, subcarriersData, fs, IMFindexes
Result: n most sensitive subcarriers(ss)
begin
    $[sEntries, sNumber]$ = size($subcarriersData$) // number of entries and subcarriers of subcarriersData

    for $i = 1$ to $sNumber$ do
        // apply VMD to each subcarrier obtaining numIMFs of IMFs
        $[imfs, \sim]$ = vmd($subcarriersData(:, i)$,'NumIMF',$numIMFs$)
        // obtain instantaneous frequency and energy using hht function
        $[insf, inse]$ = hht(imfs(:,$imfIndexes$),$fs$)
        // find the IMFs were most of the energy is found
        $[\sim, idxMaxE]$ = max(mean($inse$))
        // Obtain the instantaneous frequency variance from the IMFs with more energy
        $imfVar(i)$ = var($insf(:, idxMaxE)$)
    end
    // find the n subcarriers with less instantaneous frequency variance
    $[\sim, idx]$ = mink($imfVar$,$n$)
    $ss = subcarriersData(:, idx)$;
end
```

7. Conclusions and Future Work

7.1 Conclusions

The use of CSI of a Wi-Fi signal seems to be a promising mechanism for wireless sensing applications, but its implementation in not controlled environments is still a challenge with issues such as the interference created by other Wi-Fi devices and even from other people's movements, requiring to add more signal processing steps with different techniques for mitigating a specific interference. In this investigation it was not only presented a vital signs monitoring system along with a detailed explanation of its components, but also a detailed analysis that could be done in other CSI-based systems using Data Mining and Artificial Intelligence tools and techniques, which can be seen as the main contribution of this investigation.

Based on the results obtained during the development of this investigation, the following particular conclusions were drawn:

1. By analyzing the frequency spectrum with the Fourier and Hilbert transform of CSI data, it can be seen that sensitive subcarriers capture the chest movements caused by human breathing. Based on investigations done by other scientists, there is no doubt that the Fresnel zone model theory provides a fundamental basis for the positioning of the system's antennas and subject, enhancing the capturing of chest movements.

2. Realiable breathing rate monitoring using CSI of a Wi-Fi signal can be achieved by employing Artificial Intelligence tools such as machine learning methods along with the conventional approach. For heart rate monitoring, results show that for achieving a generalized reliable model, a more complex and expensive approach must be used, such as directional antennas or an antenna array for obtaining a clear heartbeat signal, where artifacts generated by the proper equipment do not have a higher presence than the heartbeat signal.

3. K-NN was the classification method with highest accuracy, precision, recall, specificity, F-score and AUC, but it is also the slowest due to the fact that each new observation needs to be compared with the training data. For a faster approach, PCA can be applied or QDC classifier can be used, but sacrificing recall and precision. Further performance tests for a real-time environment need to be done.
4. The selection of sensitive subcarriers proved to be a task such as an important as signal processing as well as it reduces computer complexity for further tasks. By maintaining the same calibration steps but using the VMD and Hilbert transform based subcarrier selection method, a lower error was reported compared to the one obtained by the common variance selection method. The execution time of the proposed method needs to be considered due to the fact that it is considerably higher, however, it is only executed once.

7.2 Future Work

As future work, the next tasks are considered to be addressed as well as the following lines of investigation arose:

1. As stated before, not all network devices provide CSI data and actually the most common CSI collecting tool is the Linux 802.11n CSI tool, but it requires an specific network device, the Intel 5300 NIC. This hardware limitation results unpractical for a large-scale deployment as it requires a computer equipment only for collecting and transferring CSI data. A new investigation can be the transferring of this whole system to embedded devices and using the ESP32 CSI Toolkit for collecting CSI data, which might result in a more affordable solution, comparing its performance with solutions such as the stated in this investigation. It is of our interest the use of Field Programmable Gate Arrays (FPGA) for processing CSI data.
2. Besides exploring the breathing and heart rate classification, the deployment of the system for investigation purposes only provides a breathing rate range as response. This response can be used as feedback for the Data Calibrator subcomponent for the next breathing rate estimation and therefore adjusting filter parameters for a precise estimation which can be provided to the final user as response. In short, a closed loop system.
3. The implementation of directional antennas or antenna arrays for extracting the heartbeat signal, along with the heart rate estimation and classification

of multiple people with different types of body and ages in order to ensure the functioning of the system.

4. A literature review of different sensitive subcarrier selection methods as well as a benchmark of these methods with the one proposed in this investigation.
5. The design of a Plug-and-Play network device for collecting CSI data for facilitating CSI investigations along with a desktop application that allows its configuration, such as network band or sample frequency (known also as packet rate), and basic signal processing for sensing applications.

8. Appendix A

Software Diagrams and Captures

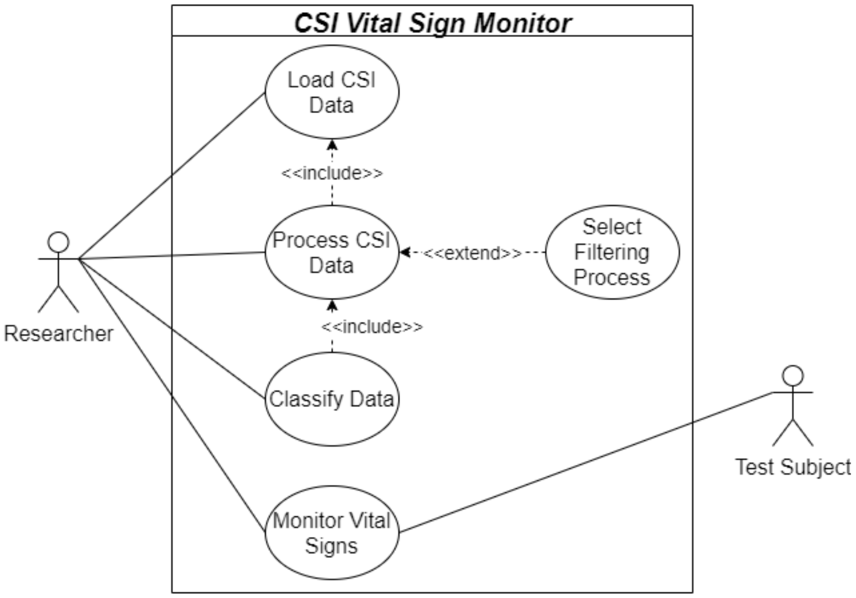


Figure 8.1: Use-case Diagram

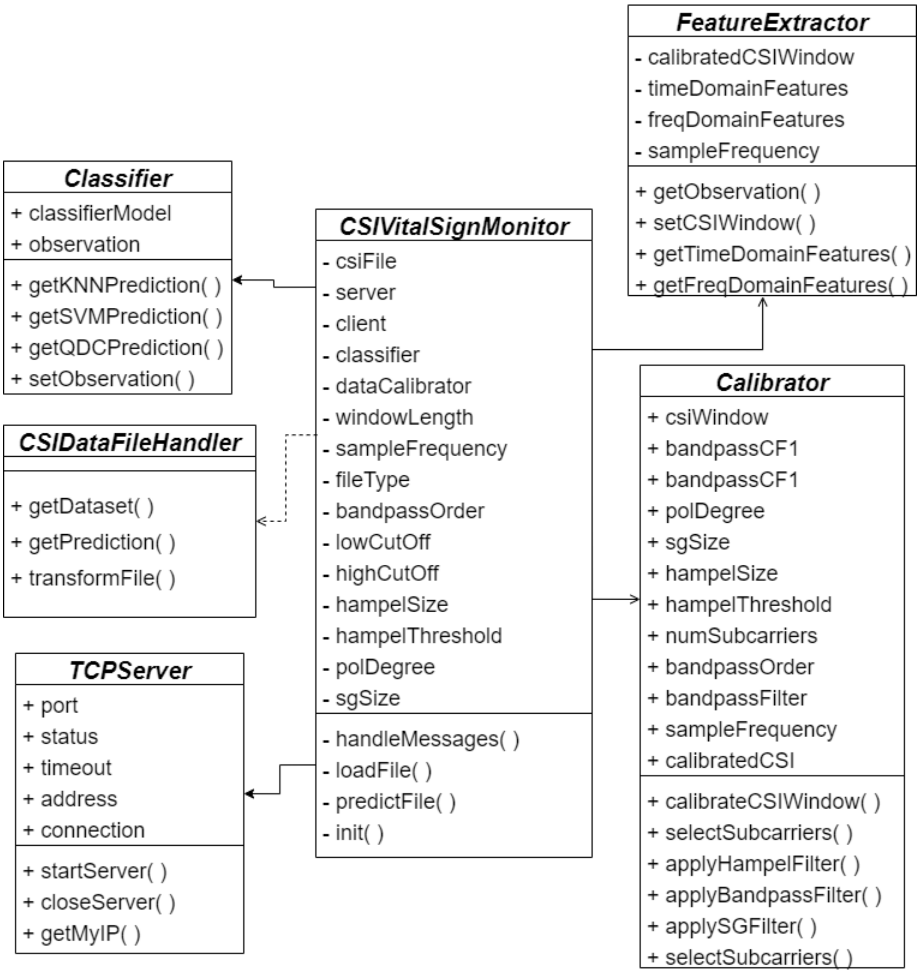


Figure 8.2: Class Diagram

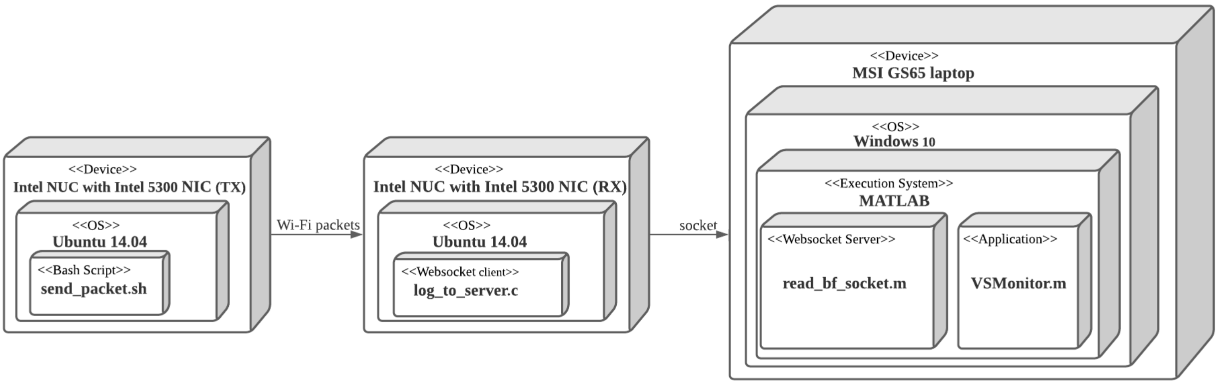
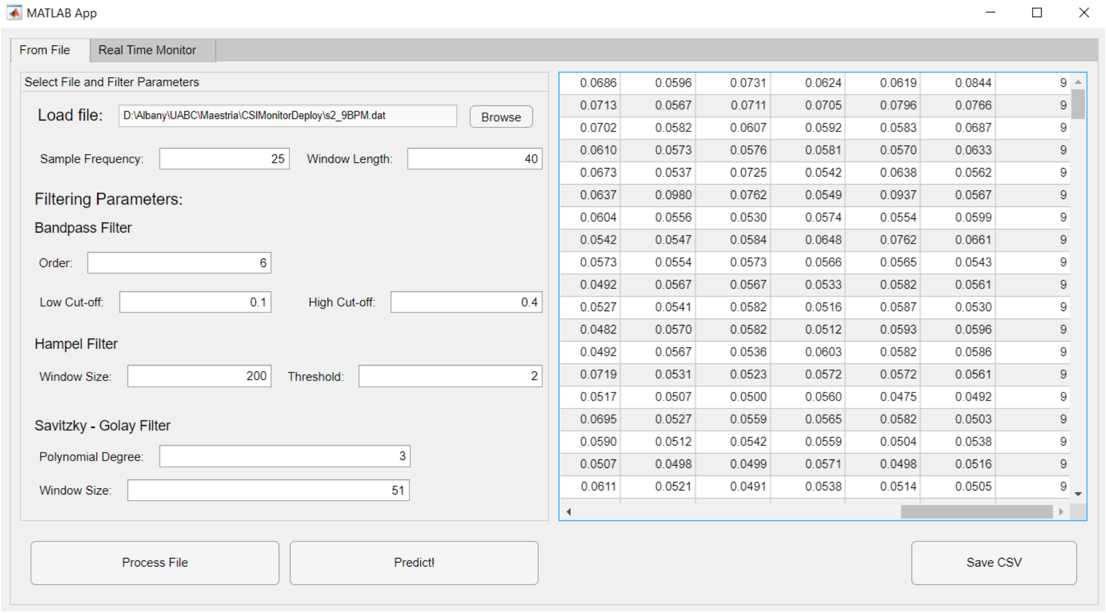


Figure 8.3: Deployment Diagram



Load file: D:\Albany\UABC\Maestria\CSIMonitorDeploy\s2_9BPM.dat

Sample Frequency: 25

Window Length: 40

Filtering Parameters:

Bandpass Filter

Order: 6

Low Cut-off: 0.1

High Cut-off: 0.4

Hampel Filter

Window Size: 200

Threshold: 2

Savitzky - Golay Filter

Polynomial Degree: 3

Window Size: 51

0.0686	0.0596	0.0731	0.0624	0.0619	0.0844	9
0.0713	0.0567	0.0711	0.0705	0.0796	0.0766	9
0.0702	0.0582	0.0607	0.0592	0.0583	0.0687	9
0.0610	0.0573	0.0576	0.0581	0.0570	0.0633	9
0.0673	0.0537	0.0725	0.0542	0.0638	0.0562	9
0.0637	0.0980	0.0762	0.0549	0.0937	0.0567	9
0.0604	0.0556	0.0530	0.0574	0.0554	0.0599	9
0.0542	0.0547	0.0584	0.0648	0.0762	0.0661	9
0.0573	0.0554	0.0573	0.0566	0.0565	0.0543	9
0.0492	0.0567	0.0567	0.0533	0.0582	0.0561	9
0.0527	0.0541	0.0582	0.0516	0.0587	0.0530	9
0.0482	0.0570	0.0582	0.0512	0.0593	0.0596	9
0.0492	0.0567	0.0536	0.0603	0.0582	0.0586	9
0.0719	0.0531	0.0523	0.0572	0.0572	0.0561	9
0.0517	0.0507	0.0500	0.0560	0.0475	0.0492	9
0.0695	0.0527	0.0559	0.0565	0.0582	0.0503	9
0.0590	0.0512	0.0542	0.0559	0.0504	0.0538	9
0.0507	0.0498	0.0499	0.0571	0.0498	0.0516	9
0.0611	0.0521	0.0491	0.0538	0.0514	0.0505	9

Figure 8.4: Processing a dat file with the developed system

9. Appendix B

Products of this Thesis

Conferences

1. A. Armenta-Garcia, F. F. Gonzalez-Navarro, J. Caro-Gutierrez, B. L. Flores-Rios and J. E. Ibarra-Esquer, “BReML: A Breathing Rate Estimator Using Wi-Fi Channel State Information and Machine Learning”, 2021 Mexican International Conference on Computer Science (ENC), 2021, pp. 1-8, doi: 10.1109/ENC53357.2021.9534797.

Contests

1. First place in “Concurso de Innovación en Ciencia y Tecnología RGMX Japón 2021” that took place in the “Simposio Virtual RGMX Japón 2021” organized by “Red Global MX Capítulo Japón”.
2. Participation in “Encuentro Estatal de Jóvenes Investigadores BC 2021” organized by “Universidad Autónoma de Baja California”

In Process

1. Database register in INDAUTOR titled “Base de Datos Generada a Partir del Procesamiento de la Información de Estado del Canal Generada por Dispositivos de Red ante Distintas Frecuencias Respiratorias de Sujetos en Reposo”.
2. Software register in INDAUTOR titled “CSIVSMonitor: Software para Estimación de Frecuencia Respiratoria a partir de CSI utilizando Procesamiento Digital de Señales y Aprendizaje Automático”.

3. A. Armenta-Garcia, F. F. Gonzalez-Navarro, J. Caro-Gutierrez, G. Galaviz-Yanez, J. E. Ibarra-Esquer, W. Flores-Fuentes, “Mining Wi-Fi Channel State Information for Breathing and Heart Rate Classification”, Pervasive and Mobile Computing, 2022 (in review).

References

[1] D. P. White, "Sleep apnea," *Proceedings of the American Thoracic Society*, vol. 3, no. 1, pp. 124–128, 2006, pMID: 16493160. [Online]. Available: https://www.atsjournals.org/doi/abs/10.1513/pats.200510-116JH 1

[2] Secretaría de Salud, "En méxico, cuatro por ciento de hombres y dos por ciento de mujeres sufren apnea del sueño," 2016. [Online]. Available: https://www.gob.mx/salud/articulos/en-mexico-cuatro-por-ciento-de-hombres-y-dos-por-ciento-de-mujeres-sufren-apnea-del-sueno 1

[3] INEGI, "Características de las defunciones registradas en méxico durante 2020, preliminar," 2021. [Online]. Available: https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/EstSociodemo/DefuncionesRegistradas2020_Pre_07.pdf 1

[4] N. Damodaran, E. Haruni, M. Kokhkharova, and J. Schäfer, "Device free human activity and fall recognition using WiFi channel state information (CSI)," *CCF Transactions on Pervasive Computing and Interaction*, vol. 2, no. 1, pp. 1–17, 2020. [Online]. Available: https://doi.org/10.1007/s42486-020-00027-1 1, 7

[5] M. A. A. Al-Qaness, M. A. Elaziz, S. Kim, A. A. Ewees, A. A. Abbasi, Y. A. Alhaj, and A. Hawbani, "Channel state information from pure communication to sense and track human motion: A survey," *Sensors*, vol. 19, no. 15, p. 3329, 2019. 3

[6] K. Heurtefeux and F. Valois, "Is RSSI a good choice for localization in wireless sensor network?" *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, pp. 732–739, 2012. 3

[7] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor Localization via Channel Response," *ACM Comput. Surv.*, vol. 46, no. 2, dec 2013. [Online]. Available: https://doi.org/10.1145/2543581.2543592 3

[8] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, Jun. 2019. 3, 13, 14

[9] F. Li, M. A. A. Al-Qaness, Y. Zhang, B. Zhao, and X. Luan, "A Robust and Device-Free System for the Recognition and Classification of Elderly Activities." *Sensors (Basel, Switzerland)*, vol. 16, no. 12, dec 2016. 3

[10] X. Wang, L. Gao, S. Mao, and S. Pandey, "Csi-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2017. 4

[11] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "Csi-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013. 4

[12] R. Zhou, X. Lu, P. Zhao, and J. Chen, "Device-free presence detection and localization with svm and csi fingerprinting," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7990–7999, 2017. 4

[13] T. F. Sanam and H. Godrich, "An improved csi based device free indoor localization using machine learning based classification approach," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 2390–2394. 5

[14] M. T. Hoang, B. Yuen, K. Ren, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "A cnn-lstm quantifier for single access point csi indoor localization," 2020. 5

[15] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "Wifi csi based passive human activity recognition using attention based blstm," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2714–2724, 2019. 5, 7

[16] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-Eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. Association for Computing Machinery, 2014, pp. 617–628. [Online]. Available: https://doi.org/10.1145/2639108.2639143 6

[17] S. Tan and J. Yang, "WiFinger: Leveraging Commodity WiFi for Fine-Grained Finger Gesture Recognition," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '16. Association for Computing Machinery, 2016, pp. 201–210. [Online]. Available: https://doi.org/10.1145/2942358.2942393 6

[18] Y. Wang, K. Wu, and L. M. Ni, “WiFall: Device-Free Fall Detection by Wireless Networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, feb 2017. 6

[19] C. Han, K. Wu, Y. Wang, and L. M. Ni, “WiFall: Device-free fall detection by wireless networks,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, apr 2014, pp. 271–279. 6

[20] R. H. Venkatnarayan, G. Page, and M. Shahzad, “Multi-User Gesture Recognition Using WiFi,” in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys ’18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 401–413. [Online]. Available: https://doi.org/10.1145/3210240.3210335 7

[21] H. Narui, R. Shu, F. F. González-Navarro, and S. Ermon, “Domain adaptation for human fall detection using wifi channel state information,” in *Precision Health and Medicine - A Digital Revolution in Healthcare*, ser. Studies in Computational Intelligence, A. Shaban-Nejad and M. Michalowski, Eds. Springer, 2020, vol. 843, pp. 177–181. [Online]. Available: https://doi.org/10.1007/978-3-030-24409-5_17 7

[22] Y. Zeng, P. H. Pathak, and P. Mohapatra, “Wiwho: Wifi-based person identification in smart spaces,” in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2016, pp. 1–12. 8

[23] T. Xin, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, “Freesense: Indoor human identification with wi-fi signals,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–7. 8

[24] M. N. Alam Nipu, S. Talukder, M. S. Islam, and A. Chakrabarty, “Human identification using wifi signal,” in *2018 Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, 2018, pp. 300–304. 9

[25] J. Wang, Y. Zhao, X. Fan, Q. Gao, X. Ma, and H. Wang, “Device-free identification using intrinsic csi features,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8571–8581, 2018. 9

[26] Y. Ren, P. Zhao, Y. Sheng, D. Yao, and Z. Xu, “Robust softmax regression for multi-class classification with self-paced learning,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, ser. IJCAI’17. AAAI Press, 2017, p. 2641–2647. 9

[27] F. Wang, J. Han, S. Zhang, X. He, and D. Huang, “Csi-net: Unified human body characterization and pose recognition,” 2019. 9

[28] J. Ding, Y. Wang, and X. Fu, “Wihi: Wifi based human identity identification using deep learning,” *IEEE Access*, vol. 8, pp. 129 246–129 262, 2020. 9

[29] J. Liu, Y. Wang, Y. Chen, J. Yang, X. Chen, and J. Cheng, “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*, ser. MobiHoc ’15. Association for Computing Machinery, 2015, p. 267–276. 10, 17, 25

[30] J. Liu, Y. Chen, Y. Wang, X. Chen, J. Cheng, and J. Yang, “Monitoring vital signs and postures during sleep using wifi signals,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2071–2084, 2018. 10, 17, 38

[31] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie, “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*, ser. UbiComp ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 25–36. [Online]. Available: https://doi.org/10.1145/2971648.2971744 10, 17, 25, 36, 38, 43

[32] J. Shang and J. Wu, “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*, ser. S3. New York, NY, USA: Association for Computing Machinery, 2016, p. 30–32. [Online]. Available: https://doi.org/10.1145/2987354.2987360 11, 17, 18

[33] X. Wang, C. Yang, and S. Mao, “Phasebeat: Exploiting csi phase data for vital sign monitoring with commodity wifi devices,” in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 1230–1239. 11, 17

[34] ——, “Resilient respiration rate monitoring with realtime bimodal csi data,” *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 187–10 198, 2020. 12

[35] F. Zhang, D. Zhang, J. Xiong, H. Wang, K. Niu, B. Jin, and Y. Wang, “From fresnel diffraction model to fine-grained human respiration sensing with commodity wi-fi devices,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, Mar. 2018. [Online]. Available: https://doi.org/10.1145/3191785 12, 36

[36] A. Khamis, C. T. Chou, B. Kusy, and W. Hu, “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*, ser. MobiQuitous ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 97–106. [Online]. Available: https://doi.org/10.1145/3286978.3287003 12, 17

[37] D. Zhang, Y. Hu, Y. Chen, and B. Zeng, “Breathtrack: Tracking indoor human breath status via commodity wifi,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3899–3911, 2019. 12, 17

[38] K. Ghavami, “Channel estimation and symbol detection in massive mimo systems using expectation propagation,” Ph.D. dissertation, LSU, 2017. 13

[39] Y. Wu and W. Zou, “Orthogonal frequency division multiplexing: a multicarrier modulation scheme,” *IEEE Transactions on Consumer Electronics*, vol. 41, no. 3, pp. 392–399, 1995. 13

[40] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Predictable 802.11 packet delivery from wireless channel measurements,” *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 159–170, aug 2010. [Online]. Available: https://doi.org/10.1145/1851275.1851203 13

[41] ——, “Tool Release: Gathering 802.11n Traces with Channel State Information,” *ACM SIGCOMM CCR*, vol. 41, no. 1, p. 53, 2011. 14, 15, 38, 43

[42] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity wifi,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’15. New York, NY, USA: ACM, 2015, p. 53–64. [Online]. Available: http://doi.acm.org/10.1145/2789168.2790124 14

[43] M. Atif, S. Muralidharan, H. Ko, and B. Yoo, “Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS),” *Journal of Computational Design and Engineering*, vol. 7, no. 5, pp. 644–656, 05 2020. [Online]. Available: https://doi.org/10.1093/jcde/qwaa048 15

[44] S. M. Hernandez and E. Bulut, “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)*, Cork, Ireland, Jun. 2020. 15

[45] Espressif Systems, "ESP32 Series - Datasheet," 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32{_}datasheet{_}en.pdf 15

[46] Y. Zeng, D. Wu, R. Gao, T. Gu, and D. Zhang, "Fullbreathe: Full human respiration detection exploiting complementarity of csi phase and amplitude of wifi signals," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 3, Sep. 2018. [Online]. Available: https://doi.org/10.1145/3264958 17, 25, 38

[47] S. Lee, Y. Park, Y. Suh, and S. Jeon, "Design and implementation of monitoring system for breathing and heart rate pattern using wifi signals," in *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2018, pp. 1–7. 17

[48] Y. Gu, X. Zhang, Z. Liu, and F. Ren, "Wifi-based real-time breathing and heart rate monitoring during sleep," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019. 17, 25, 38

[49] V. K. Ingle and J. G. Proakis, *Digital signal processing using MATLAB.* Cengage Learning, 2012. 18

[50] M. Weeks, *Digital signal processing using MATLAB and wavelets.* Infinity Science Press, 2007. 18, 27

[51] T. L. Floyd, *Dispositivos electrónicos*, 8th ed. México: Pearson Education, 2008, pp. 756–760. 18, 19

[52] H. C. Chen and S. W. Chen, "A moving average based filtering system with its application to real-time qrs detection," in *Computers in Cardiology, 2003*, 2003, pp. 585–588. 20

[53] R. W. Schafer, "What is a savitzky-golay filter? [lecture notes]," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011. 22

[54] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, 1974. 22

[55] R. K. Pearson, "Data cleaning for dynamic modeling and control," in *1999 European Control Conference (ECC)*, 1999, pp. 2584–2589. 24

[56] A. V. Oppenheim, S. H. Nawab, and A. S. Willsky, *Senales y sistemas.* Prentice Hall Hispanoamericana, 1998. 27

[57] M. Johansson, "The hilbert transform," *Mathematics Master's Thesis. Växjö University, Sweden*, vol. 19, 1999. 28

[58] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998. 28

[59] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014. 29

[60] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54 776–54 788, 2020. 30

[61] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2015.0202 30, 65

[62] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008. 31

[63] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000. [Online]. Available: https://doi.org/10.1109/34.824819 32, 33

[64] Z. H. Hoo, J. Candlish, and D. Teare, "What is an roc curve?" *Emergency Medicine Journal*, vol. 34, no. 6, pp. 357–359, 2017. [Online]. Available: https://emj.bmj.com/content/34/6/357 35

[65] D. Zhang, H. Wang, and D. Wu, "Toward centimeter-scale human activity sensing with wi-fi signals," *Computer*, vol. 50, no. 1, pp. 48–57, 2017. 36

[66] D. Wu, D. Zhang, C. Xu, H. Wang, and X. Li, "Device-free wifi human sensing: From pattern-based to model-based approaches," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 91–97, 2017. 36

[67] G. E. Athanasiadou, "Incorporating the fresnel zone theory in ray tracing for propagation modelling of fixed wireless access channels," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007, pp. 1–5. 36

[68] B. Lu, Z. Zeng, L. Wang, B. Peck, D. Qiao, and M. Segal, "Confining wi-fi coverage: A crowdsourced method using physical layer information," in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2016, pp. 1–9. 38

[69] B. R. Nayana and P. Geethanjali, "Analysis of statistical time-domain features effectiveness in identification of bearing faults from vibration signal," *IEEE Sensors Journal*, vol. 17, no. 17, pp. 5618–5625, 2017. 40

[70] M. J. Shensa *et al.*, "The discrete wavelet transform: wedding the a trous and mallat algorithms," *IEEE Transactions on signal processing*, vol. 40, no. 10, pp. 2464–2482, 1992. 41

[71] Polar, "Polar H9 - User Manual," 2020. [Online]. Available: https://support.polar.com/e{_}manuals/h9-heart-rate-sensor/polar-h9-user-manual-english/manual.pdf 44

[72] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," 2012. 47

[73] X. Wang, C. Yang, and S. Mao, "On csi-based vital sign monitoring using commodity wifi," *ACM Trans. Comput. Healthcare*, vol. 1, no. 3, may 2020. [Online]. Available: https://doi.org/10.1145/3377165 65