Universidade de Aveiro
2023

Tiago
Couto

Deteção e classificação de movimentos captados por radar para aplicações em casas inteligentes

Detection and classification of movements captured by radar for Smart Homes applications

Universidade de Aveiro
2023

Tiago Couto

Deteção e classificação de movimentos captados por radar para aplicações em casas inteligentes

Detection and classification of movements captured by radar for Smart Homes applications

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Pedro Renato Tavares Pinho, Professor associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Daniel Albuquerque, Professor Adjunto da Escola Superior de Tecnologia e Gestão de Águeda da Universidade de Aveiro.

.

Dedico este trabalho aos meus Pais, Irmã, Namorada, Tia Rosa e Madrinha.

o júri / the jury

presidente / president

Prof. Doutor António Guilherme Rocha Campos
professor auxiliar da Universidade de Aveiro

vogais / examiners committee

Prof. Doutora Matilde Pós-De-Mina Pato
professora adjunta do Instituto Superior de Engenharia de Lisboa

Prof. Doutor Pedro Renato Tavares de Pinho
professor associado da Universidade de Aveiro

agradecimentos / acknowledgements

Gostaria de expressar a minha mais profunda gratidão a todos aqueles que foram fundamentais para a realização desta tese.

Sou abençoado por ter o apoio inesquecível da minha família. Aos meus pais, que têm sido os meus pilares de força ao longo deste percurso, e à minha irmã, que me inspirou e motivou constantemente. A vossa crença em mim alimentou a minha determinação em ser bem sucedido.

À minha namorada, o teu encorajamento, paciência e compreensão durante os altos e baixos deste percurso foram cruciais. Agradeço o teu amor e apoio, mais do que as palavras podem exprimir.

Um agradecimento especial à minha tia e madrinha, pelo encorajamento e assistência durante a fase experimental desta investigação e em todo o percurso académico. O vosso contributo foi inestimável.

Aos meus amigos, um agradecimento pela amizade e ajuda neste percurso. Estou grato a todos os que desempenharam um papel neste percurso académico. O vosso apoio e a vossa presença na minha vida fizeram uma diferença significativa e, sem vocês, não teria conseguido alcançar este objetivo.

Estou imensamente grato aos meus orientadores, Pedro Pinho e Daniel Albuquerque e ao inestimável apoio prestado pela professora Susana Brás, Carolina Gouveia e pelo Engenheiro da Bosch, Rodrigo Almeida. As orientações e empenho em garantir o sucesso desta tese foram verdadeiramente excepcionais.

Um agradecimento especial ao Instituto de Telecomunicações de Aveiro pelo apoio financeiro na cedência do radar AWR1642BOOST durante toda a execução do projeto de dissertação, bem como pela disponibilidade de espaços para estudo e reuniões com os orientadores a fim de discutir os vários aspetos que foram surgindo ao longo da execução deste projeto.

Obrigado a todos, do fundo do meu coração.

Palavras Chave

Aplicações em Smart-Homes, Frequency Modulated Continuous Wave Radar, Deteção e classificação de movimentos, Efeito Doppler, Machine Learning.

Resumo

As casas inteligentes são residências equipadas com sistemas avançados de automação e controlo, englobando uma variedade de funções e dispositivos, como iluminação, segurança, entretenimento e regulação de temperatura, com o propósito de oferecer maior conforto aos seus ocupantes. Num contexto tão vasto de aplicações, a deteção e classificação de movimento assumem uma relevância notável, permitindo, por exemplo, a adaptação da temperatura com base no tipo de movimento efetuado por uma pessoa. Nesta dissertação é apresentado um sistema para deteção e classificação de movimentos captados por radar para aplicações em casas inteligentes. Para o efeito é usado um Radar de onda contínua com frequência modulada, a operar na banda de 77 - 81 GHz, onde foi possível obter os valores da localização e da velocidade de movimento de uma pessoa onde a pessoa foi detetada. A classificação dos movimentos foi feita atráves de algoritmos de Machine Learning, tais como K - Nearest Neighbors, Support Vector Machine, Linear Discriminant Analysis and Random Forest. Para tal foi feita a recolha de dados de 10 pessoas diferentes de maneira a obter-se um extenso conjunto de dados. O posicionamento do Radar na divisão foi também estudado de maneira a ter-se uma boa recolha de dados. Com recurso ao Matlab, foi feito o processamento dos dados, onde foram criadas características para diferenciar os movimentos a classificar, por exemplo, os valores de Doppler ou velocidade que uma pessoa tem em diferentes tipos de movimento. A performance de cada algoritmo de Machine Learning utilizado é estudado.

Keywords

Smart-Homes applications, Frequency Modulated Continuous Wave Radar, Movement Detection and Classification, Doppler Effect, Machine Learning.

Abstract

Smart homes are homes equipped with advanced automation and control systems, encompassing a variety of functions and devices, such as lighting, security, entertainment and temperature regulation, with the aim of offering greater comfort to their occupants. In such a wide range of applications, the detection and classification of movement is of remarkable importance, allowing, for example, the adaptation of temperature based on the type of movement made by a person. This dissertation presents a system for detection and classification movements captured by radar for Smart Home applications. For this purpose, a frequency modulated continuous wave radar operating in the 77 - 81 GHz band is used, where it was possible to obtain the person location and movement velocity where the person was detected. The movements were classified using Machine Learning algorithms such as K - Nearest Neighbours, Support Vector Machine, Linear Discriminant Analysis and Random Forest. To this end, data was collected from 10 different people in order to obtain an extensive data set. The positioning of the radar in the division was also studied in order to have good data collection. The data was processed using Matlab, where characteristics were created to differentiate the movements to be classified, for example, the Doppler values or speed that a person has in different types of movement. The performance of each Machine Learning algorithm used was studied.

Contents

List of Figures

List of Tables

Acronyms

2D Two Dimensional

3D Three Dimensional

ADC Analog-to-Digital Converter

AOA Angle of Arrival

CNN Convolutional Neural Network

CW Continuous Wave Radar

FFT Fast Fourier Transform

FMCW Frequency Modulated Continuous Wave Radar

GUI Graphical User Interface

IF Intermediate Frequency

KNN K-nearest neighbors

LDA Linear Discriminant Analysis

MAD Median Absolute Deviations

MATLAB MATrix LABoratory

ML Machine Learning

RCS Radar Cross Section

RF Random Forest

Rx Reception

SNR Signal-to-Noise Ratio

SVM Support vector machine

TLV Type-length-value

Tx Transmission

UART Universal asynchronous receiver/transmitter

UWB Ultra-Wideband Radar

TI Texas Instruments

CHAPTER 1

Introduction

1.1 Motivation

In the beginning, Smart Home technology was primarily utilized for the control of lighting and heating systems. Nevertheless, as time has progressed, developments in smart technology have broadened its application to encompass a wide range of electrical components within a household [1]. The appearance of Smart Homes has significantly improved the overall quality of life by introducing automated and assistive services that offer enhanced convenience and comfort to residents. Illustrative examples of such services include automatic lighting systems that activate upon detecting a person presence in a room and the ability to remotely control curtains to adjust natural light levels and ensure privacy [2],[3].

Some studies ([4], [5], [6]) employ video cameras, infrared sensors, vibration sensors, or photoelectric sensors to categorize an individual movements. However, it is crucial to note that this approach often increases the overall system cost, as seen in the work [5], which involves nearly forty sensors.

On the other hand, the use of video cameras can deter some individuals due to privacy concerns. A survey conducted in [7], involving 165 participants aged 17 to 95, revealed that most participants had apprehensions about privacy and data security. Nevertheless, it is worth noting that many would accept the use of cameras if it were deemed necessary for the care of individuals with illnesses or the elderly.

Radar technology is a widely accepted choice in various environments, and its potential application in indoor settings offers several advantages, particularly in addressing the privacy issues highlighted in [7]. In contrast to cameras, which can raise privacy concerns, radar technology can yield valuable data without capturing images, making it a promising technology for indoor applications, especially when combined with the Doppler effect, which allows for subject tracking and movement detection.

Therefore, there is a compelling need for a cost-effective system that can replace numerous sensors while safeguarding privacy, a challenge not easily addressed with cameras. This system should also be designed to be efficient, eliminating the necessity for multiple sensors.

Challenged by Bosch[1], the objective is to differentiate between various movements that a person may perform in indoor environments. This distinction is aimed at automating processes such as adjusting room environments based on the activities the resident is engaged in. In practical terms, this work applies to scenarios like being standing still, walking or home workouts. Consequently, a Smart Home system can autonomously make adjustments, like moderating room temperature through an air conditioning unit, enhancing the comfort of the workout environment. This demonstrates the practical utility of movement tracking and detection within a Smart Home context.

Figure (1.1) provides a straightforward visualization of how this concept can be implemented, potentially for integration into Bosch solutions or products. In this scenario, after detecting a person, their movement is classified. For instance, if the person is seated, the system can increase the temperature to create a cozier environment for the resident. If the person is walking, the temperature remains stable, ensuring it is not too cold or too hot. On the other hand, if the person is working out, the system can lower the temperature to provide a cooler atmosphere for the workout session.

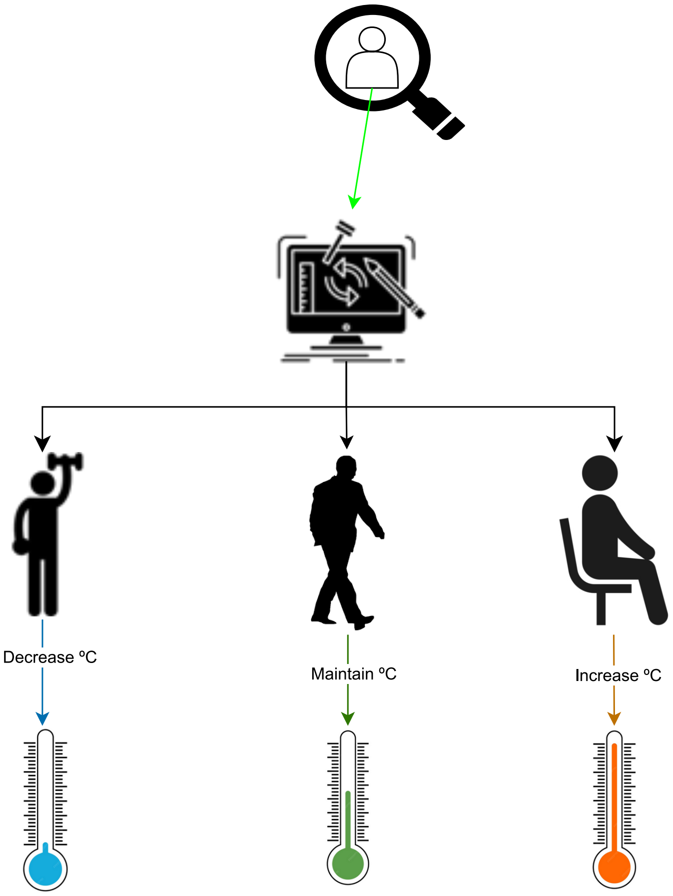


Figure 1.1: Example of a Smart Homes application.

[1] Robert Bosch GmbH, multinational engineering and technology corporation headquartered in Stuttgart, Germany, boasts a significant presence in Cacia, Aveiro, known as Bosch Termotecnologia. This particular division played a pivotal role in providing support for the research conducted in this thesis

1.2 Objectives

This dissertation aims to detect and classify the movements captured by radar for Smart Homes applications. So, the main objectives of this dissertation are:

- Select the most appropriated radar to detect movements in indoor environments;
- Define a data collection protocol;
- Study the radar spatial configuration;
- Analyse and understand the output data collected;
- Catalogue the data;
- Understand the operation of each Machine Learning algorithm;
- Extract and identify of relevant features in movement evaluation;
- Develop a machine learning model for movement evaluation and identification, using four different Machine Learning Algorithms;
- Analysis of the Machine Learning algorithms performance;
- Proof of concept of real-time detection and movement classification.

1.3 Content Organization

This section recaps the six chapters of this dissertation with a respective overview of each one. The structure is the following:

- **Chapter 1 -** The present chapter reveals the motivation for this study and work. The objectives of this dissertation are shown in this chapter;
- **Chapter 2 -** This chapter summarises the Radar technology and the relevant concepts around it. An overview of the type of the current radars is presented to select the type of Radar that is more suited for this work, with a general overview of studies, providing context and background to the research. Once the radar type is chosen, an in-depth exploration is conducted to understand the relevant calculations and principles associated with that specific radar technology;
- **Chapter 3 -** This chapter delves into the concept of Machine Learning (ML) and the specific algorithms employed in the research;
- **Chapter 4 -** This chapter introduces the software and hardware used. Moreover, describes the signal design of the transmitted signal from the radar, as well as its parameters influence on the environment, such as the maximum distance the target can be to the signal get him, for example;
- **Chapter 5 -** This chapter investigates the optimal radar placement to achieve the best results in data collection.

  Also delves into the label selection process used for the Machine Learning Algorithms, the feature extraction process, and the accompanying statistical analysis. It concludes with the testing and evaluation of the Machine Learning Algorithms, taking into account their performance with and without feature selection.

  A real-time detection and movement classification proof of concept is presented.
- **Chapter 6 -** This is the last chapter that reveals the conclusions of the work presented.

  Additionally, this chapter explores opportunities for improvement and outlines potential directions for future research in this field.

CHAPTER 2

People tracking with Radar

In this chapter, key radar concepts will be introduced to establish a foundational understanding of its operation.
Additionally, a comprehensive examination of different radar system types will be conducted to elucidate their respective operational principles. This comparative analysis will serve to distinguish the strengths and weaknesses of each radar type, facilitating the selection of the most appropriate radar technology for the intended application.

2.1 Relevant concepts of the radar technology

A basic radar system, consisting of Transmission (Tx) antenna and Reception (Rx) antenna, is illustrated in Figure 2.1.

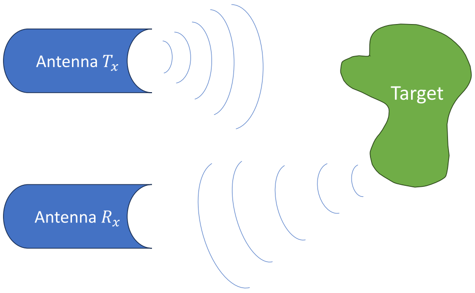


Figure 2.1: Basic Radar System.

As depicted in Figure 2.1, the radar system operates by emitting an electromagnetic waveform from the transmitter, which subsequently interacts with various objects within its detection range. The resulting reflected waves, referred to as echoes, are then captured by the radar is receiver.

Subsequently, the receiver demodulates the signal and processes the information gleaned from these echoes. However, the strength of the returned signal is significantly reduced compared to the transmitted signal due to signal dispersion and losses in the surrounding environment before reaching the receiving antenna.

Additionally, it is crucial to consider the nature of the reflection, as the power density of the echo can vary from object to object. This variability is described by the Radar Cross Section (RCS) σ_{RCS}, which plays a fundamental role in estimating the power reflected by the target.

In this way, the received power by the antenna, denoted as P_r, can be calculated by Equation (2.1), knowing the transmitted power P_t with the power gain G_t of the antenna, as documented in [8].

$$P_r = \frac{P_t G_t G_r \sigma_{RCS} \lambda^2}{(4\pi)^3 d^4} \tag{2.1}$$

In Equation (2.1), it is evident that the strength of the returned signal decreases as the distance d between the radar and the object increases.

Due to the fact that electromagnetic waves used in radar travel at the speed of light in free space, it becomes feasible to estimate the distance of a target relative to the radar, as described in [8].

2.2 Type of Radars

This topic delves into the various types of radar systems that are available. The significance of understanding these different radar types lies in the ability to comprehend how each radar works. This understanding serves as a foundation for selecting the most suitable radar.

2.2.1 Pulsed Radar

Pulsed Radar operates by transmitting pulses with a specific frequency and then waiting for these pulses to be reflected off objects within the radar range. This approach enables the determination of the range d of a target based on the elapsed time Δt, between the transmission of the pulse and the reception of its corresponding echo at the antenna, as expressed in Equation (2.2) in function of the speed of light c.

$$d = \frac{c\Delta t}{2} \tag{2.2}$$

In order to ensure that the echo has sufficient power to reach the receiving antenna, it is essential for the transmitted pulse to be of high power. It is important to highlight that the power of the reflected echo is invariably less than the power of the initially transmitted pulse.

Figure 2.2 illustrates how the power of the radar pulses operates.

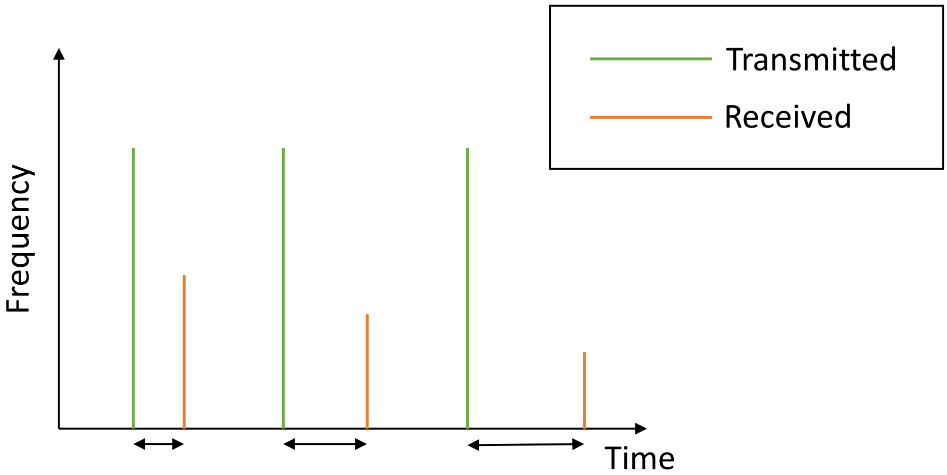


Figure 2.2: Power of transmitted/received pulses (Adapted from: [9]).

This creates an inverse relationship: as the distance from the radar increases, the power of the returning signal decreases, as outlined in [9]. Consequently, when the object is far from the radar, the echo pulse can weaken and become indistinguishable from the background noise in the environment.

Ultra-Wideband Radar (UWB) offers increased resolution, however this valuable precision comes at a higher hardware cost, as highlighted in [10]. Therefore, the decision to choose a UWB radar system or an alternative may often be influenced by cost considerations.

2.2.2 Continuous Wave Radar

Continuous Wave Radar (CW) radar transmits a continuous signal with a constant frequency, and its design primarily focuses on velocity measurement. The Doppler shift effect enables the separation of the transmitted and received signals, as noted in [11]. The interaction of the transmitted signal with an object induces a Doppler Frequency Shift, enabling the calculation of the target velocity, as explained in [12]. CW radar systems are favored for their simple architecture, ease of system integration, and low power consumption, as discussed in [13].

However, a limitation of CW radar systems is their inability to perform frequency modulation, which results in the incapacity to measure the range to a target, as detailed in [14] and [15].

2.2.3 Frequency-Modulated Continuous Wave Radar

Therefore, the study of FMCW radar, as explored in [16], becomes particularly relevant, as FMCW systems offer range measurement capabilities in addition to Doppler shift measurements.

An FMCW radar transmits a synthesised signal with frequency increasing gradually, called chirp, by an antenna T_x, that will be reflected on the object.

When the reflection reaches the receiving antenna R_x, both signals of antennas T_x and R_x will be multiplied by the mixer, resulting in an *IF* signal, Intermediate Frequency Signal [17].

This FMCW Radar can be summarised in Figure 2.3.

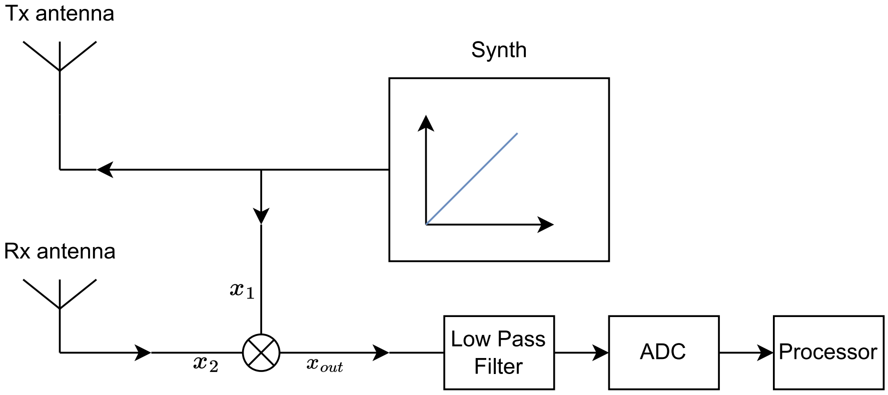


Figure 2.3: Basic model of an FMCW Radar (Adapted from: [17]).

By analyzing the frequency shifts, FMCW can measure the distance to the target and its velocity. Additionally, can estimate the Angle of Arrival (AOA) of the target by using multiple receiving antennas.

The behavior of FMCW radar is influenced by several key factors, including the bandwidth of the transmitted signal, which affects the range resolution, and the time duration of the transmitted signal, which impacts the velocity resolution.

2.3 State of the art

This section provides an overview of the current state of the art in the field of radar-based techniques for counting, tracking, and evaluating the movement of individuals in indoor environments.

The state of the art is assessed based on several factors, including the type of radar systems employed, the bandwidth used, and the dimensions of the indoor space under investigation. Additionally, it is divided in two primary approaches referenced in the existing literature:

- People counting - Where it is important to know how many people can be detected and how it detects.
- Movement classification - Capacity to identify different types of movements and the accuracy of that differentiation of movement. This topic is the most interesting to the work.

2.3.1 People Counting

This overview helps to understand key factors, including radar placement, radar types and frequencies, room dimensions, and the ML algorithms employed to detect subjects. The information from works is summarized in Table (2.1).

Table 2.1: Overview of People Counting Literature.

Work	Radar Type	Radar Placement	Room Dimensions	Accuracy	Operational Frequency
[18]	FMCW *IWR6843ISK*	1.7m (above entrance)	6m x 8m	83% (up to 10 persons)	60 GHz, 4 GHz BW
[19]	FMCW *BGT60TR24B*	Not specified	6m x 8m	95.5% (up to 3 persons)	60 GHz, 4 GHz BW
[20]	FMCW *BGT60TR13C*	Above entrance, lateral, opposite wall	3.8m x 6.5m	Not specified	60 GHz, 2 GHz BW
[21]	UWB	Not specified	Not specified	99.9% (up to 3 persons)	6.8 GHz, 1.5 GHz BW
[22]	UWB	Not specified	Not specified	98.8% (static), 95.5% (moving)	7.29 GHz, 1.5 GHz BW
[23]	UWB	Not specified	Not specified	94.7% (up to 10 persons)	7.29 GHz, 1.5 GHz BW

In [18], the purpose was to count and track people in a restaurant using an FMCW radar, *IWR6843ISK*. In a room with dimensions of 6×8 meters, and the radar positioned at a height of 1.7 meters, the conducted procedure was tested for its ability to detect people. The results indicated an accuracy in detecting up to 10 persons at 83%.

In [19], a similar objective was pursued, involving the use of an FMCW radar (specifically, the *BGT60TR24B* model) in a restaurant setting. However, in [19], the accuracy in distinguishing and detecting up to three persons reached 95.5%.

Across all the mentioned works, [18], [19], and [20], a consistent observation was made: the accuracy of people counting decreases as the expected number of individuals to be measured increases. This implies that higher crowd densities lead to reduced detection accuracy.

In a room measuring 3.8×6.5 meters, the authors in [20] explored the optimal radar placement for counting people in an indoor environment. The radar was positioned in three different configurations:

- Above the entrance;
- In the lateral of the room;
- In the opposite wall of the entrance.

In evaluating the results, it was determined that the FMCW *BGT60TR13C* radar achieved better results when placed on the opposite entrance wall.

The radar frequency used in [18]–[20] is 60 GHz, with a bandwidth of 4 GHz, except for [20], which operates with a bandwidth of 2 GHz. In contrast, [21]–[23] utilize UWB radar technology, where [21] operates at 6.8 GHz and [22], [23] at 7.29 GHz, with a bandwidth of 1.5 GHz.

The work [21] presents results of counting up to three persons with 99.9% of accuracy, where the ML used to detect it was a Convolutional Neural Network (CNN). Similarly, [22] employs a CNN and SVM for identifying a specific number of persons, achieving high accuracy of 98.8% for static people and 95.5% for moving people.

The use of UWB radar in [21]–[23] demonstrates superior accuracy in counting the number of people in a room compared to the FMCW works [18]–[20]. Even when tasked with counting a higher number of individuals, [23] attests to the effectiveness of UWB radar, with an accuracy of 94.7% for counting up to 10 persons.

2.3.2 Movement Classification

Table (2.2) describe a concise summary of the key points literature on movement detection and classification.

Table 2.2: Overview of Movement Detection and Classification Literature.

Work	**Radar Type**	**Movements Studied**	**Accuracy**	**Frequency / Bandwidth**
[24]	FMCW	Falling forward, Falling backwards, Falling vertically, Sitting, Jumping, Squatting, Turn around	94.2%	5.8 GHz / 320MHz BW
[25]	FMCW	Walking, Falling, Jumping, Squatting, Jogging, Stepping	95.5%	5.8 GHz / 320MHz BW
[26]	UWB *P440 Radio Module*	Walking, Sitting, Falling, Standing	99%	[3.1 4.8] GHz BW
[27]	UWB	Jumping, Crawling, Running	95.21% for Running	4.3 GHz Frequency
[28]	FMCW	Walking Faster, Walking Slower	>72%	Not specified
[29]	FMCW	Standing, Walking, Running, Crawling	95%	61 GHz / 6 GHz BW

It is possible to understand the type of movements that the person performs inside a room resorting to ML. Activities like walking, sitting, falling and jumping indicate a particular pattern because the movement flow is almost the same whenever it is performed.

In research [24], the radar were positioned at distances of 1 meter and 2.5 meters from the subjects. The study examined the following movements:

- Falling forward
- Falling backwards
- Falling vertically

- Sitting
- Jumping
- Squatting
- Turn around

Using a 5.8 GHz radar with a bandwidth of 320MHz, the study in [24] successfully classified movements with an accuracy of 94.2% using as ML algorithm the KNN, using the FMCW radar signals.

In [25], the focus was on detecting Walking, Falling, Jumping, Squatting and Jogging, and the objective was achieved. The results, obtained with an FMCW radar using the same frequency and bandwidth as [24], exhibited an accuracy of 95.5% in distinguishing between walking and falling.

Work [26], a UWB radar was employed to test movements like jumping, sitting, falling, and standing. The radar achieved a 99% accuracy in detecting these movements in a room measuring $4 \times 5 \times 3$ meters, with the radar placed at the opposite wall of the entrance at a height of 1 meter. The KNN algorithm was employed as ML algorithm and a bandwidth of 3.1-4.8 GHz was used.

A similar scenario is presented in [27], where jumping, crawling, and running were evaluated, resulting in a 95.21% accuracy for running using a CNN with a UWB radar.

The work in [28] accurately identified whether a person was walking faster or slower, achieving accuracies above 72%. It also detected the presence or absence of arm movements using the micro-Doppler values of each person. An FMCW radar was used, and SVM and KNN ML algorithms were applied.

Work [29] proposed a method for identifying changes in a person motion using an FMCW radar operating at 61 GHz with a 6 GHz bandwidth. Features such as skewness, kurtosis, or mean values were established as inputs for a ML algorithm, CNN, achieving an accuracy of 95%. Numerous movements were categorized, including standing, walking, running, and crawling, both as individual actions and in terms of transitions between them (e.g., standing to walking). Additionally, the study recommended the utilization of camera sensors for detecting and tracking individuals movements within indoor environments.

2.4 Radar selection

In the field of people counting and movement classification, several approaches are commonly used. Some rely on camera systems, such as the one discussed in [30], to achieve higher accuracy in these tasks. However, it is essential to note that this approach compromises one of the key advantages of radar systems, which is privacy.

Considering the performance of different radar technologies, UWB radar systems consistently exhibit better results, especially in terms of accuracy in people counting. On the other hand, FMCW radar systems, while not reaching the same level of accuracy as UWB, still offer competitive performance, with accuracy values that closely approach those of UWB systems, with works described in Tables 2.1 and 2.2.

It is worth noting that both FMCW and UWB radar systems are susceptible to multipath reflections, as mentioned in [10]. However, cost considerations play a significant role in choosing between these radar technologies.

Specifically, UWB radar systems tend to provide superior accuracy and a higher Signal-to-Noise Ratio (SNR) compared to FMCW radar systems, as demonstrated in [31].

However, this precision in UWB radar comes at a higher hardware cost, as mentioned in [10]. It is important to highlight that UWB radar systems offer better target penetration compared to FMCW radar systems. On the other hand, FMCW radar chirps have a longer duration.

Taking these factors into consideration, opting for a FMCW radar appears to be the more practical choice, primarily due to cost considerations and hardware configuration advantages. While UWB radar systems perform better in terms of maximum range, this feature is often unnecessary for indoor environments. As outlined in the state of the art section (2.3), both FMCW and UWB radar systems tend to offer similar levels of accuracy when it comes to distinguishing different movements.

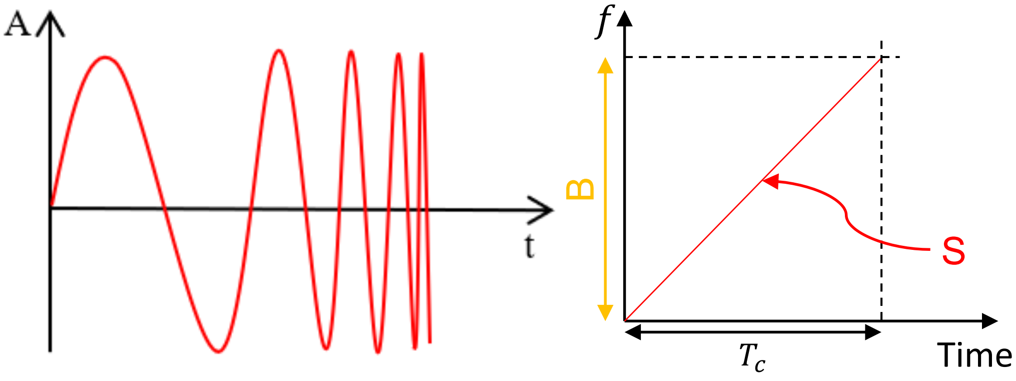
2.5 FMCW

For the reasons mentioned in the previous sections, FMCW was chosen as the most suitable radar option for the application to explore in this dissertation. Therefore, this section now aims to explain some basic concepts about FMCW, which are important for the execution of this work.

2.5.1 Chirp

A chirp is a sinusoidal signal in which the frequency f, increase gradually. Starting at f_c and finishing with a frequency, $f_c + B$, where B is the bandwidth of the chirp [17], [32].

In Figure 2.4a, a visual representation depicts a chirp signal with varying amplitude over time. In Figure 2.4b, an alternative illustration showcases the chirp signal frequency variation over time, which is referred to the chirp duration denoted as T_c.



(a) Chirp representation over time **(b)** Chirp representation over frequency

Figure 2.4: Chirp (Adapted from: [17], [32]).

The motion of a target introduces a frequency shift in the reflected signal, a phenomenon known as the Doppler effect.

When a signal is transmitted, either isotropically or directionally, some of the waves are reflected off objects and directed back toward the radar. Among these reflected waves, certain waves experience a frequency shift due to the target movement. By employing signal processing techniques, it becomes feasible to ascertain the targets velocity, direction, and range.

2.5.2 RCS – Radar Cross Section

Intuitively, larger objects have a greater surface area, which enables them to reflect a higher number of electromagnetic waves compared to smaller objects, leading to increased reflection. That the distribution and density of reflected power can vary depending on the objects shape and orientation, as highlighted in previous studies ([33], [34]).

RCS, is a measure of the effective area that the target presents to a radar system. Represents the theoretical amount of energy that, when scattered uniformly, can create an echo detected by a radar, in other words, RCS means how easily the target can be detected by a radar [33]–[35].

In Figures 2.5 and 2.6, we can understand how the objects shape affects the power density of the reflected signal.

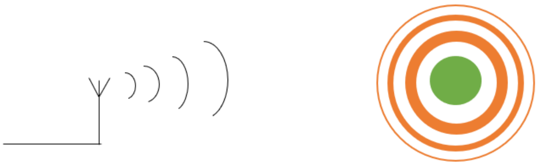


Figure 2.5: Isotropic reflections.

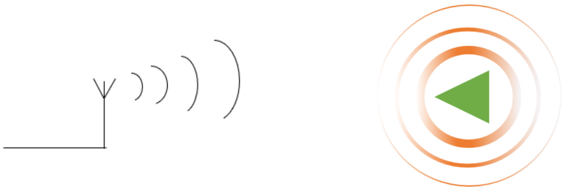


Figure 2.6: Reflections on a heterogeneous object.

The reflection, as depicted in figure 2.5, occurs isotropically. Yet, it is important to note that as the signal travels greater distances, its power will naturally decrease.

On the other hand, in figure 2.6, occurs the same situation, although the power density of the reflected signal is denser in certain regions caused by the shape of the object [36].

2.5.3 Range estimation

Within the mixer, the multiplication process occurs between the signal $x_1 = \sin(\omega_1 t + \phi_1)$, transmitted by antenna T_x, and the signal $x_2 = \sin(\omega_2 t + \phi_2)$, received by antenna R_x. It is important to emphasize that both of these signals are characterized by their respective frequencies and phases. Following this multiplication process, the resulting signal x_{out}, as outlined in Equation (2.3), demonstrates the following two properties [17]:

1. The instantaneous frequency of the output signal x_{out}, is equal to the difference between the instantaneous frequency of the transmitted signal ω_1, and the instantaneous frequency of the received signal ω_2 after a low pass filter.

2. The initial phase of the output signal equals the difference between the initial phase of the transmitted signal ϕ_1, and the initial phase of the received signal ϕ_2.

$$x_{out} = \sin[((\omega_1 - \omega_2)t) + (\phi_1 - \phi_2)] \tag{2.3}$$

So, assuming only one object in front of the radar, the following figures, 2.7 and 2.8, can picture what happens in terms of frequency and time. Additionally, τ signifies the time interval between when the chirp signal is transmitted and when the corresponding echo signal is received by the radar antenna, represented in equation 2.4 [37] in function of the distance to the detected object d and the speed of light c.

The Intermediate Frequency (IF) signal consist of a tone with constant frequency, once the difference between the two chirps is fixed. This signal is only valid during the time interval where both the Tx and Rx chirps overlap [37].

$$\tau = \frac{2d}{c} \tag{2.4}$$

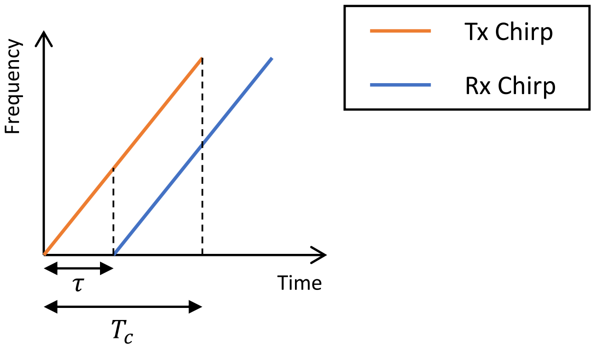


Figure 2.7: Representation of the transmitted and reflected chirp (Adapted from: [17]).

As it was explained, the frequency of the resulted signal of the mixer follows the first property, which means that the difference of the chirp T_x and R_x respectively represented in figure 2.7, can be resumed in figure 2.8 [17].

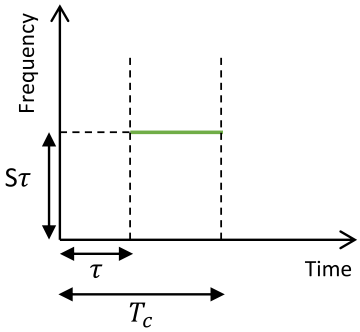


Figure 2.8: Output signal of the mixer (Adapted from:[17]).

So, the output signal of the mixer is an IF signal with a fixed frequency with $S\tau$ value [37]. It is worth noting that S signifies the slope of the frequency. The frequency can be measured with the equation (2.5).

$$f_0 = \frac{S2d}{c} \tag{2.5}$$

In this context, d represents the distance of the target relative to the radar, and c denotes the speed of light.

When multiple objects are present in front of the radar, it becomes evident that the number of received chirps is directly proportional to the number of objects. In simpler terms, if three targets are within the radar range at different distances, three chirps will be received by the antenna. This occurs because each target reflects its chirp, and these reflected signals arrive at the receiver antenna at different times.

Subsequently, the output of the mixer generates distinct signals with various frequencies corresponding to each of the targets, as described in [17]. This process is illustrated in Figure 2.9.

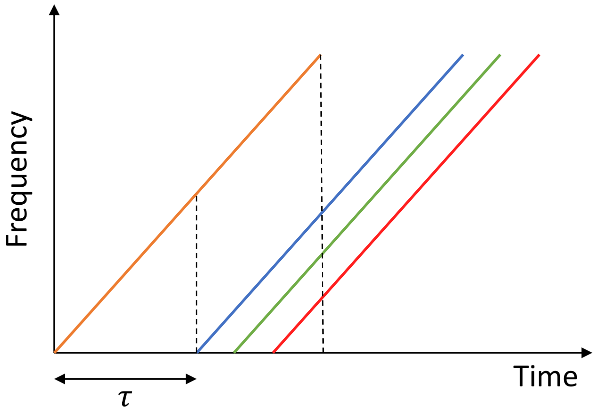


Figure 2.9: Multiple targets in front of the radar.

Thus, it is advantageous to handle the output signals of the mixer in the frequency domain rather than the time domain. This approach involves utilizing a FFT, which can convert a signal from the time domain to the frequency domain. The outcome is the creation of peaks in the frequency domain, with each peak corresponding to a specific target, as depicted in Figure 2.10.

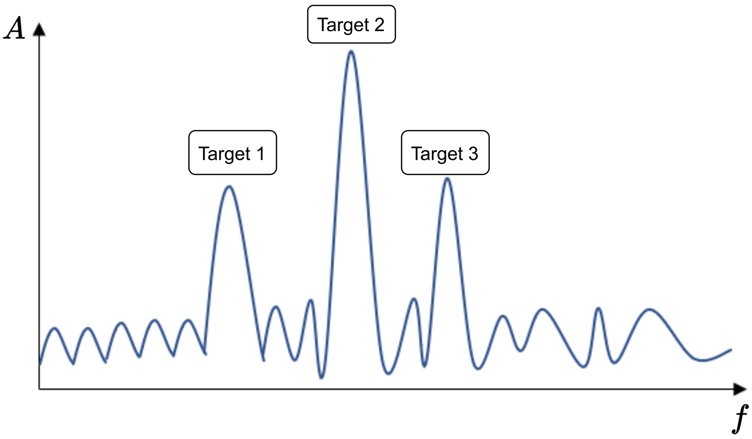


Figure 2.10: Range-FFT for three persons targeted.

In the frequency domain, the number of peaks observed in the FFT corresponds to the number of targets in the radar range. The first peak designates the closest object to the radar, the second peak signifies the second closest, and so forth. This particular FFT operation is commonly referred to as the range-FFT.

2.5.4 Range resolution

When two targets are very close to each other, they may appear as a single peak in the range-FFT, potentially causing the radar to perceive them as a single entity or person. This scenario can introduce errors in detection and tracking.

As mentioned in [37], range resolution is determined by the bandwidth of the signal, as demonstrated in Equation (2.6). This means, that increasing the bandwidth results in improved range resolution.

$$d_{res} = \frac{c}{2B} \tag{2.6}$$

2.5.5 Velocity estimation and resolution

Using an FMCW radar, the measurement of a target velocity can be achieved by transmitting two chirps separated by a time T_c and performing a range-FFT on these two chirps.

In this scenario, it may appear impossible to estimate the velocity because the FFT result will exhibit two peaks with the same frequency, corresponding to each of the transmitted chirps.

However, it is important to recognize that even when dealing with identical frequencies, each chirp possesses a unique phase, represented in Figure 2.11.

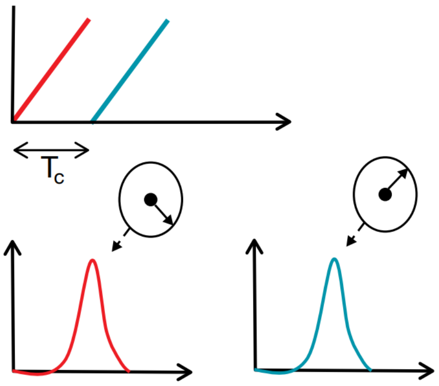


Figure 2.11: Velocity measurement (Adapted from: [37]).

By calculating the phase difference, represented as θ, between these two chirps, it becomes possible to determine the target velocity, as detailed in Equation (2.7) [37]. Moreover, it is crucial to understand how to measure the distinct velocities of two targets located at the same distance from the radar and for that it is necessary to transmit more than two chirps, as detailed in [37].

$$\phi = \frac{4\pi T_c v}{\lambda} \Leftrightarrow v = \frac{\lambda\phi}{4\pi T_c} \tag{2.7}$$

Subsequently, a second FFT operation known as the Doppler-FFT will generate two peaks within the FFT, one for each target, localized by their distinct phases, represented in figure 2.12

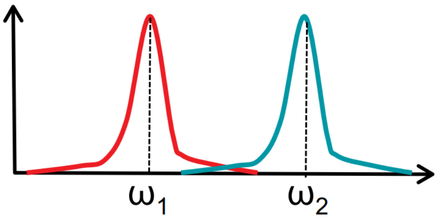


Figure 2.12: Doppler-FFT (Adapted from: [37]).

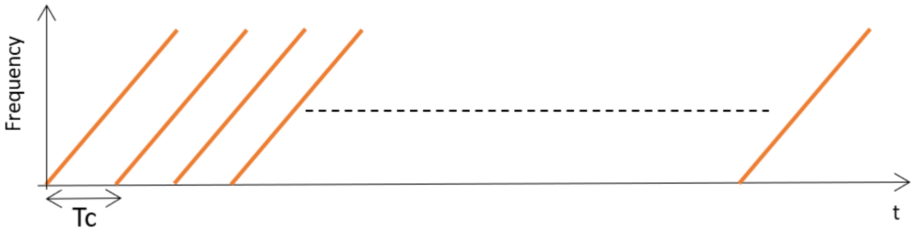


Figure 2.13: Representation of a chirps frame, typically used to detect and determine the velocity of multiple targets.

This set of multiple chirps transmitted is referred to as a frame, where N chirps, evenly spaced, are transmitted, as described in [37]. An example of such a frame is illustrated in Figure 2.13.

N represents the number of chirps. With this understanding, in conjunction with the frequency variations described in Equation (2.7), it is possible to see the velocity resolution (2.8).

$$\Delta v = \frac{\lambda}{2NT_c} \Leftrightarrow v_{res} = \frac{\lambda}{2T_f} \tag{2.8}$$

As evident in Equation (2.8), NT_c represents the total time duration of the frame, denoted as T_f.

2.5.6 Angle of Arrival estimation and resolution

A limitation arises when two individuals, for instance, share the same range and velocity, resulting in only one point being displayed. This can create the illusion that there is only one person present.

To address this issue and differentiate between multiple targets, it becomes essential to determine the angle of arrival for each target. This involves calculating the angle between the receiver antenna and the respective target.

By employing a setup with one transmitter antenna and two receiver antennas, the process from transmission to the reflections received by the receiver antennas, is depicted in Figure 2.14.

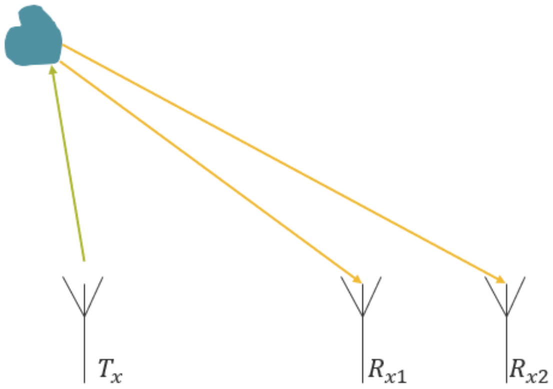


Figure 2.14: Procedure for estimating the angle of arrival using one transmitter antenna and two receivers antennas (Adapted from: [38]).

It is critical to acknowledge that the receiver antennas are in close proximity to each other, which results in the reflected signals being very close and essentially parallel.

With the two receiver antennas separated by a distance $d_{antennas}$ represent as d in Figure 2.15, basic trigonometric principles apply. The reflection signal arriving at the farthest antenna must travel a greater distance compared to the signal reaching the closer antenna.

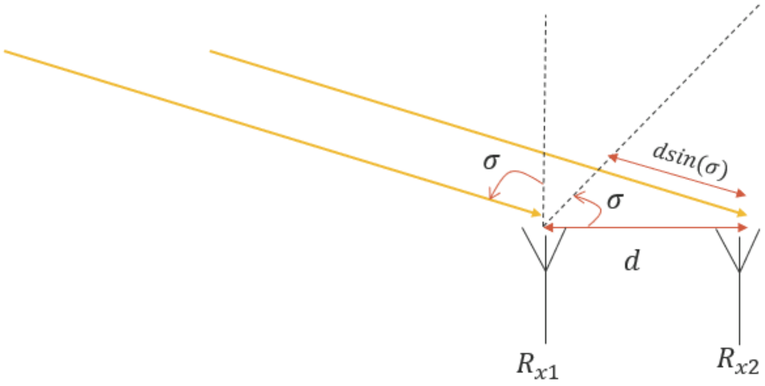


Figure 2.15: Close vision to estimate the additional distance to the farthest antenna (Adapted from: [38]).

This additional distance can be succinctly expressed in Equation (2.9).

$$d_{add} = d_{antennas} \sin(\theta) \tag{2.9}$$

This calculation becomes more intuitive and easier to comprehend when observing the zoomed-in view provided in Figure 2.15.

Indeed, as depicted in Figure 2.15, considering the additional distance traveled by the reflected signals and the phase difference between each reflected signal within the frame, it is possible to determine the AOA of the target.

The Two Dimensional (2D)-FFT is associated with each receiver antenna, and with the phase difference of the received signals, a subsequent FFT, known as the angle-FFT, can be employed to determine the angle of arrival.

However, it is crucial to ascertain how small the angle can be and still distinguish the two objects, ensuring they appear as two distinct peaks in the angle-FFT. The phase difference between the two signals arriving at the antennas can be succinctly expressed in Equation (2.10).

$$\theta_{res} = \frac{\lambda}{N_{antennas} d_{antennas} \cos(\theta)} \tag{2.10}$$

Where $N_{antennas}$ is the number of Rx antennas.

With Equation (2.10) providing the angle resolution, it becomes possible to determine how small the angle between two targets can be while still appearing as two distinct peaks, each corresponding to an individual target, in the angle-FFT, as discussed in [17].

CHAPTER 3

Machine Learning

This chapter describes the basics of ML and the functionality mode of each algorithm tested in the work.

3.1 Basics of Machine Learning

With the rise of electronic computers, the requirement for processing large datasets became unavoidable. To meet this demand, it was imperative to develop algorithms capable of efficiently handling substantial volumes of data and generating outputs through learning processes.

This process is known as Machine Learning (ML). The literature reports several algorithms based on different principles regarding data processing or strategies for establishing relationships between variables, as mentioned in [39].

Machine Learning is fundamentally about creating value through algorithms capable of discovering patterns and identifying relationships between concepts.

Machine Learning algorithms can be categorized into subgroups based on their underlying principles or methods, including:

- Supervised Learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

Machine Learning algorithms can be broadly classified into two primary groups: Supervised and Unsupervised algorithms, with these two categories being the main focus [40].

3.2 Machine Learning Algorithms

3.2.1 Supervised learning algorithms

Supervised learning, as described in [41], involves predicting outcomes based on historical data. In this approach, the algorithm generates a function that maps input data to the desired output. The primary task of the Machine Learning algorithm is to identify patterns in the data and construct a mathematical model that can make accurate predictions.

Classifiers

Classifiers can determine where the data fits better in a certain label. There are two types of models:

- Classification
- Regression

Classification model

In supervised learning, the algorithm maps input data (data samples) to predefined classes, as stated in [40]. The goal, as illustrated in Figure 3.1, is to derive a decision boundary (e.g., a line) that can effectively distinguish between the two datasets, as explained in [42]. This decision boundary allows the algorithm to classify new, unseen data points into one of the predefined classes.

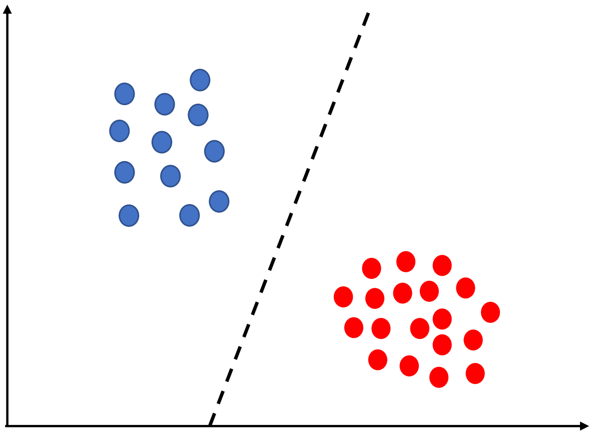


Figure 3.1: Classification Model (Adapted from: [42]).

Regression model

In supervised learning, the algorithm maps the input data to a real-time domain, as mentioned in [40]. Figure 3.2 illustrates the use of this model, where the objective is to determine a line that can approximate the data behavior, as described in [42]. This line helps in making predictions about the continuous evolution of the data.

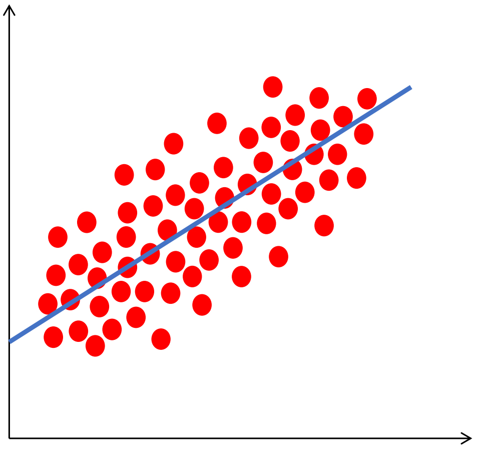


Figure 3.2: Regression Model (Adapted from: [42]).

The process of learning can be divided into **training** and **testing**, and follows the diagram on Figure (3.3).

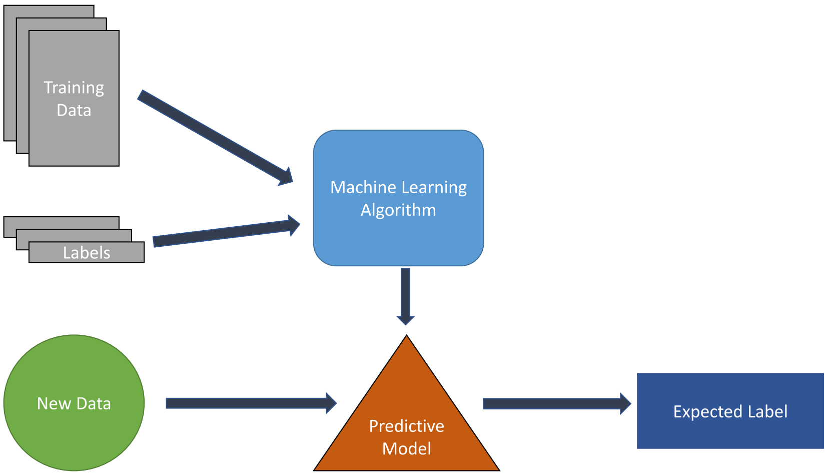


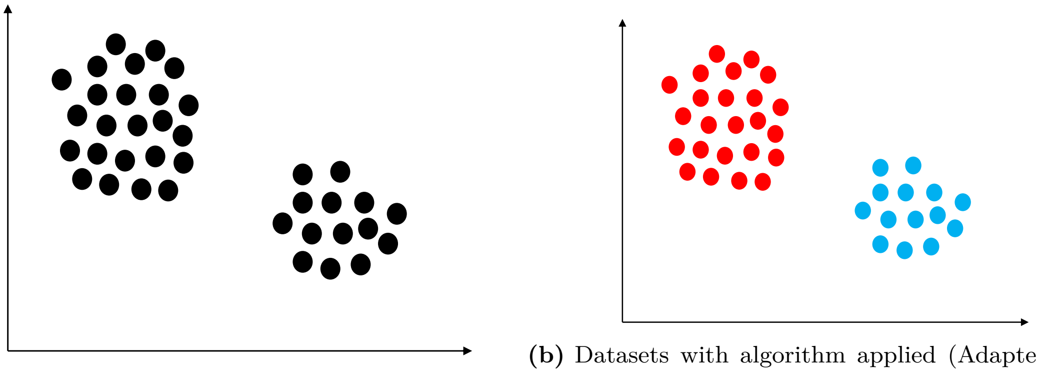
Figure 3.3: Supervised Learning Model (Adapted from: [39], [40]).

During the training stage of a supervised learning algorithm, data samples are provided as input to the algorithm. These samples are labelled and used for learning, with the goal of creating a predictive model, as outlined in [40].

In the testing phase, the capabilities of the learning model are evaluated. This is done by using the trained model to predict outcomes for test data, as mentioned in [40]. The model performance is assessed based on how accurately it can predict the outcomes for this new, unseen data.

3.2.2 Unsupervised learning algorithm

Unsupervised learning operates without predefined background labels. Instead, the algorithm primary objective is to autonomously generate these labels, as stated in [40]. It achieves this by identifying similarities among data points from different datasets and organizing them into groups, known as clusters [40]. The operational concept of the algorithm is visually represented in Figure 3.4.



(a) Datasets without algorithm (Adapted from: [42]).

(b) Datasets with algorithm applied (Adapted from: [42]).

Figure 3.4: Unsupervised Learning Algorithm.

3.3 Supervised learning Algorithms

In this work, supervised learning was applied to implement four different algorithms, each of which underwent testing to assess its accuracy. The primary objective was to determine the most appropriate algorithms for the given context. These classes were then utilized for training the data and subsequently for predicting or classifying the test data.

3.3.1 K - Nearest Neighbors

In the KNN algorithm, classes are defined, and new data is compared to the K nearest neighbors data values. The predominant class among the neighbors determines the class of the new data [43].

This algorithm can be better understood by examining Figure 3.5.

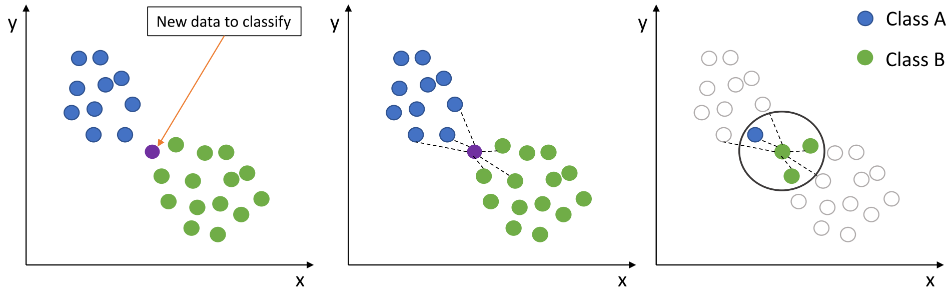


Figure 3.5: KNN ML algorithm (Adapted from: [44]).

Figure 3.5 illustrates the KNN algorithm functioning. It shows two different classes, class A (blue) and class B (green), along with a new data point (purple) that needs to be classified. The algorithm works by comparing this new data point to its closest neighbors.

In this example, the algorithm uses ($K = 3$), meaning it looks at the three closest neighbors of the new data point. Analyzing the figure, two of the nearest neighbors belong to class B, and only one belongs to class A. As a result, the algorithm assigns the new data point to class B because it is the predominant class among its closest neighbors.

3.3.2 Support Vector Machine

SVM is a ML algorithm that excels at classifying two classes. The primary function of the SVM algorithm is to identify a line or a hyperplane, that effectively separates all the classes. It is important to note that this line or hyperplane does not have to be linear, it must, however, be adept at discriminating between the various classes most effectively [43].

The operation of this algorithm can be illustrated using a simple example with two classes, as shown in Figure 3.6. This demonstrates how the SVM algorithm works to find the optimal boundary that best distinguishes between different classes.

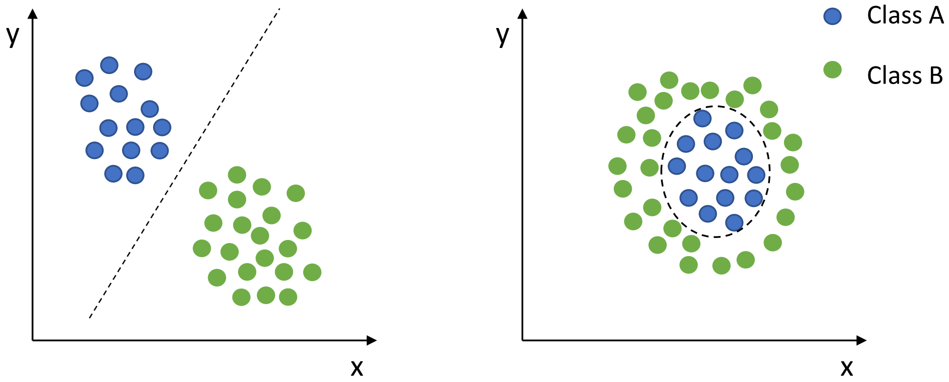


Figure 3.6: SVM ML algorithm (Adapted from: [45]).

In Figure 3.6, the line is drawn to separate the two classes, and when a new data point is introduced, its classification is determined based on which "side" of the line or boundary it falls on.

3.3.3 Linear Discriminant Analysis

LDA algorithm, much like the SVM algorithm, aims to differentiate between different classes. However, LDA takes a different approach by projecting the data into a lower-dimensional space in order to maximize the separation between classes.

Figure 3.7 provides an example of the LDA algorithm in action with two classes, illustrating how it transforms the data to enhance the distinction between them [46].

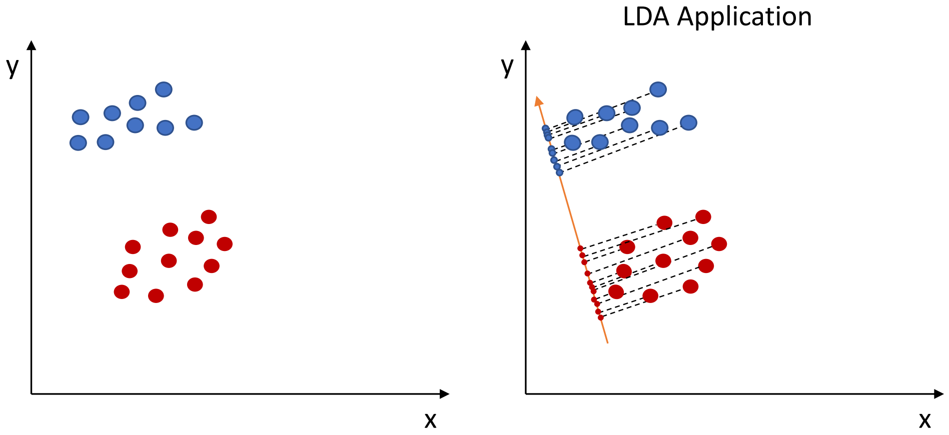


Figure 3.7: LDA Application (Adapted from: [47]).

In figure 3.7, the data begins in two dimensions, but with the application of the LDA algorithm, the data is transformed into a one-dimensional representation. This transformation enhances the separation between classes, as clearly visible in the graph. The final result of this process can be seen in Figure 3.8.

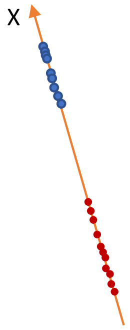


Figure 3.8: LDA Application (Adapted from: [47]).

3.3.4 Decision Trees

The Decision Trees algorithm is structured in a way where each node represents a conditional test based on the values of certain features, which helps in narrowing down the possible classes for a given data point. This process continues until the correct class is determined. Figure 3.9 provides a visual representation of how the Decision Trees algorithm works, illustrating the hierarchical decision-making process that leads to classifying data points into different categories [43].

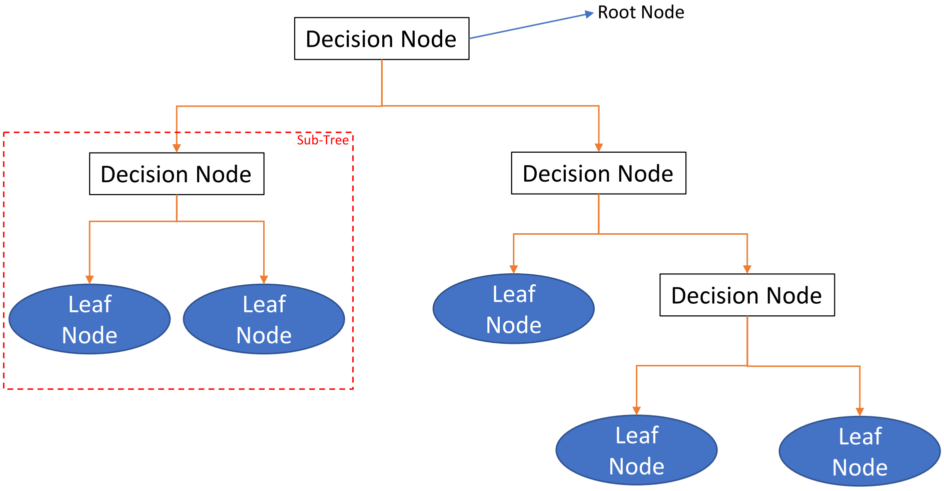


Figure 3.9: Decision Tree ML algorithm (Adapted from: [43]).

A simple decision tree can be created using a single feature, such as the person velocity. Figure 3.10 illustrates how this decision tree could be constructed to distinguish between whether a person is stopped, walking, or running based on their velocity. This type of decision tree serves as a basic example of how the Decision Trees algorithm can be employed for classification tasks.

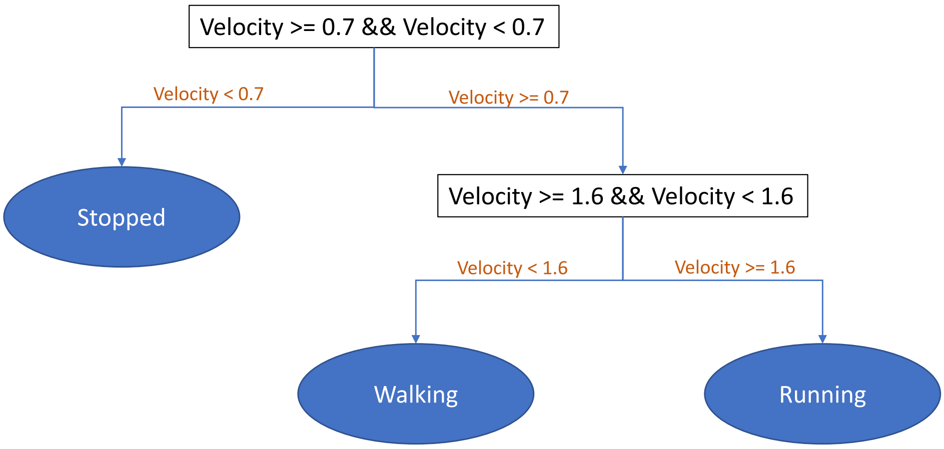


Figure 3.10: Decision Tree ML algorithm example.

So, if new data has a velocity of 1.7 m/s, the Decision Tree shown in Figure 3.10 would classify this data point as "Running". This showcases how a Decision Tree can be a simple yet effective tool for classifying data based on specific features or criteria.

3.3.5 Random Forest

The Random Forest algorithm is built on the foundation of Decision Trees in Machine Learning. It leverages a specific number of Decision Trees, each employing different features for the nodes, while considering all available features.

Once the Decision Trees generate their classification results, the Random Forest algorithm combines these results to determine the majority vote, which is then attributed to new test data, as outlined in [48]. A visual representation of this algorithm can be seen in Figure 3.11.

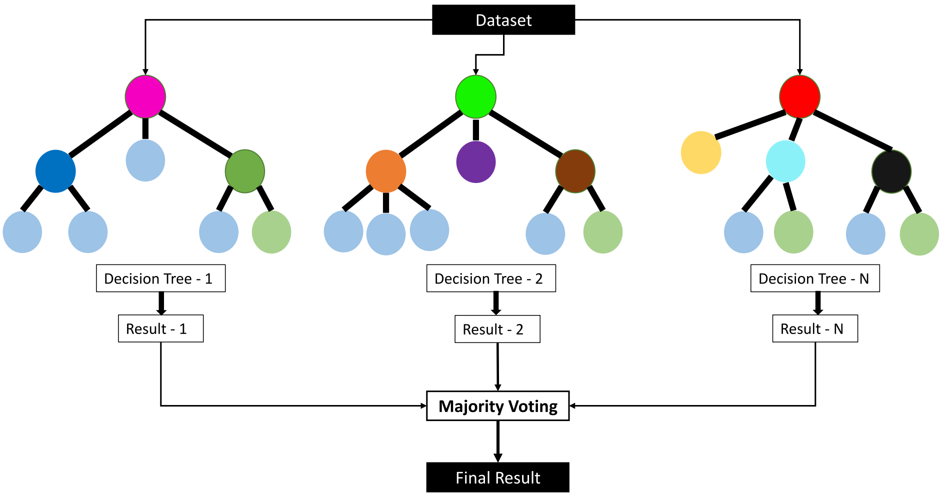


Figure 3.11: Random Forest ML algorithm (Adapted from: [49]).

CHAPTER 4

Output Data and Chirp Design

This chapter outlines the methodology for processing the data acquired from the radar. It delves into the utilization of these data values for subsequent processing, while considering the structure of each data frame. It also addresses how modifications in the parameters of the chirps within each frame can impact the analysis of the environmental characteristics. This approach is critical for extracting meaningful information from radar data and making informed decisions based on the radar observations.

4.1 Hardware and Software tools

This section presents the hardware and software tools used in the project dissertation.

4.1.1 Board Texas Instruments AWR1642BOOST-EVM

The AWR1642BOOST-EVM is a 77 GHz mmWave evaluation board developed by Texas Instruments (TI). It is designed to connect directly to a microcontroller unit. The AWR1642 chip integrated on this board operates in the 76 GHz to 81 GHz range and is primarily used for automotive radar applications, but it can also be employed in indoor environments, as mentioned in [50].

Control, configuration, and data visualization for this board can be achieved using a Universal asynchronous receiver/transmitter (UART)-to-USB interface. For more detailed specifications regarding the board, you can refer to [51].

4.1.2 MATLAB 2021b

The AWR1642BOOST board functionality can be observed in real-time with the assistance of MATrix LABoratory (MATLAB) software. The MATLAB runtime allows the execution of the mmWave Visualizer, enabling configuration of operational parameters such as frequency, range resolution, and other settings to visualize point cloud objects, moving targets, Doppler-range plots, and more, as discussed in [52].

4.1.3 Uniflash

CCS Uniflash is a standalone software tool designed for transferring data to the flash memory of a target board, as mentioned in [53]. This tool is often used for programming and configuring flash memory in embedded systems and microcontroller applications.

4.1.4 mmWave SDK Demo

This software product is utilized to configure TI mmWave devices, as mentioned in [54]. The accompanying software development kit (SDK) includes demonstrations that illustrate various control and data processing aspects of mmWave applications. These demos may also offer data visualization on a personal computer, as indicated in [54].

4.1.5 mmWave Sensing Estimator - Texas Instruments Software

mmWave Sensing Estimator is a real-time calculator used to assess the potential performance of radar systems. It allows users to determine various radar characteristics, such as maximum detection distance and range resolution. This software achieves this by modifying chirp parameters and considering scene-specific arguments in accordance with radar theory and equations.

Moreover, mmWave Sensing Estimator is equipped with a capability to provide warnings when changes to parameters could potentially result in suboptimal radar functionality. This ensures that users can make informed decisions about radar configurations and settings, as mentioned in [55].

4.1.6 mmWave Visualizer - Texas Instruments Software

mmWave Visualizer Demo is a software tool provided by Texas Instruments (TI). It is intended to be used in conjunction with the mmWave SDK demo. This software runs on mmWave devices, including the AWR1642BOOST-EVM board, and it enables communication with the radar system, allowing users to visualize real-time data from radar acquisitions.

Once communication with the radar is established, users have the capability to adjust various parameters such as the frequency band, range resolution, and maximum radial velocity, among others.

With this software, users can view and store information in various formats, including a scatter plot of detected objects, Range-Doppler plots, and Range and noise profiles, as described in [52]. This tool is valuable for visualizing and analyzing radar data in real-time.

4.1.7 People Counting - Texas Instruments Software

The People Counting software developed by TI is designed for the detection and tracking of one or more individuals using radar technology. It also has the capability to count the number of people within the radar field of view.

Through communication with the radar, users can modify the signal configuration transmitted by the radar, adapting it to the specific requirements of the room or environment.

One of the key features of this software is its ability to store a data stream of tracked and detected points. It distinguishes between static objects and individuals, providing data such as point cloud information, range, Doppler measurements, velocity, acceleration, and SNR values. This data is valuable for understanding and analyzing the movements and behavior of people within the radar coverage area.

4.2 Output information sent to host

In the mmWave Visualizer, the output data can be visualized in real-time. This data includes information such as point cloud objects, Range Doppler Heatmap, Range and Noise Profile, as well as SNR and Noise values for each detected point. These values are generated by the mmWave SDK Demo, as detailed in [52].

Users have the option to save this data in a binary file for a specified duration. This duration can be set by the user through a specific feature in the mmWave Visualizer, which is initiated using the "Record Start" button. More comprehensive information about the mmWave Visualizer Demo can be accessed in [56].

Remembering that the frame is a set of N chirps transmitted, output packets that contain detection information, are sent out every frame through the UART, conforms to the format illustrated in Figure 4.1. It is important to note that the total packet size is a multiple of 32 Bytes, in line with the information provided in [57]. This structured data is essential for further analysis and processing of radar observations in the context of this dissertation.

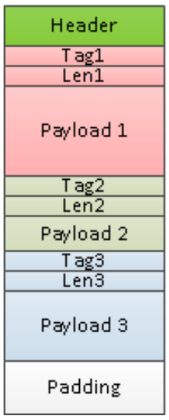


Figure 4.1: Output packet structure returned by the radar [57].

Each packet consists in the frame header, and the number of TLV items. Each TLV item consists of type (identifier), length and payload information.

The length of the packet is variable and depends on the number of detected objects, so, the packet length can change from frame to frame. However, to maintain consistency and alignment, the packet is always padded so that the total packet length is a multiple of 32 bytes. It is worth noting that if there are no detected objects in a particular frame, the present TLV is not sent, as indicated in [58].

4.2.1 Frame Header

The frame header consistently comprises 44 bytes, with each byte serving specific purposes. These bytes are categorized and organized as shown in table (4.1) in the order that facilitates byte-by-byte processing and interpretation of the radar data. The Frame Header serves as the beginning of each packet, and contains identifier known as the "Magic Word". This Magic Word is used to locate the start of each packet within the data stream, as detailed in [59].

Table 4.1: Frame Header.

Value	Type	Number of Bytes	Description
Magic Word	uint16_t	8	Word where it is initialized ({0x0102,0x0304,0x0506,0x0708})
Version	unint32_t	4	SDK Version
Total Packet Length	unint32_t	4	Packet Length including frame header length in Bytes
Platform	unint32_t	4	Device in use
Frame Number	unint32_t	4	Number of frame on read
Time [CPU cycles]	unint32_t	4	Time when the message was created
Number of detected objects	unint32_t	4	Number of detected objects of the respective frame
Number of TLV	unint32_t	4	Number of TLV of the respective frame
Subframe Number	unint32_t	4	Number of subframes

4.2.2 TLV Header

After processing the last 4 bytes related to the "Subframe Number", the TLV Header can be read, and its structure is outlined in Table (4.2). This header contains essential information about the identifiers and their corresponding lengths. Each identifier is associated with a specific number that facilitates visualization in the mmWave Visualizer. The details of each identifier and their corresponding values are provided in Table (4.3), as elaborated in [59].

Table 4.2: TLV Header.

Value	Type	Number of Bytes	Description
Identifier (Tag)	unint32_t	4	Type of message in the payload
Length [in Bytes]	unint32_t	4	Length of the payload

Table 4.3: Description of each identifier.

Value	Description	Details
1	Detected Points	Number of points detected in the payload
2	Range Profile	Value of received power (dB) that the radar captured in every distance
3	Noise Profile	Value of noise (dB) in every distance
4	Azimuth Static Heatmap	Azimuth Static Heatmap
5	Range-Doppler Heatmap	Range Doppler Heatmap
6	Statistics (Performance)	Statistic of performance about CPU load
7	Side Info for Detected Points	SNR (dB) and Noise (dB) values for each point
8	Azimuth/Elevation Static Heatmap	Azimuth/Elevation Static Heatmap
9	Temperature Statistics	Statistics Information

Subsequently to the explanation of the identifiers and their corresponding lengths in Table (4.3), each identifier value corresponds to a specific length. The data from the mmWave Visualizer output file is initially in binary format. For analysis, the data needs to be converted from binary to hexadecimal. The subsequent reading process is executed byte by byte, following the order detailed in Tables (4.1) and (4.2).

The identification of an identifier value, such as '7', signifies that the following bytes correspond to "Side Info for Detected Points". This section has a specific length, calculated as (4 Bytes x Number of Detected Objects), with the number of detected objects defined in the frame header [59].

However, if the identifier value was '4', the length would be entirely different. This demonstrates the significance of Table (4.3) as it provides the necessary information about the length of each identifier. Details regarding the length values for each identifier can be found in [59], with data processing carried out using MATLAB 2021b.

This process is repeated for each TLV, involving the identification of the identifier and its corresponding length. The entire data recorded in the mmWave Visualizer is read using this method, with the process repeated for each frame until the maximum file length is reached.

4.3 Chirp Design on TI Radar

Understanding how the radar behaves under different chirp parameters is essential. Radar systems offer a multitude of configuration possibilities, and the main objective is to identify the most suitable chirp scheme to optimize radar performance. Experimenting with various chirp parameters allows radar operators to fine-tune the system according to their specific needs and improve its overall effectiveness.

4.3.1 Chirp Design

The mmWave Visualizer provides the capability to load a configuration that can be applied to the radar. It is important to note that this configuration encompasses a wide range of parameters, extending beyond just chirp design. It also includes aspects related to frame configuration and other parameters, which are detailed in [54]. This resource explains the impact of each parameter and the possible values they can take.

However, it is worth emphasizing that the radar behavior is primarily influenced by the chirp configuration, as illustrated in Figure 4.2.

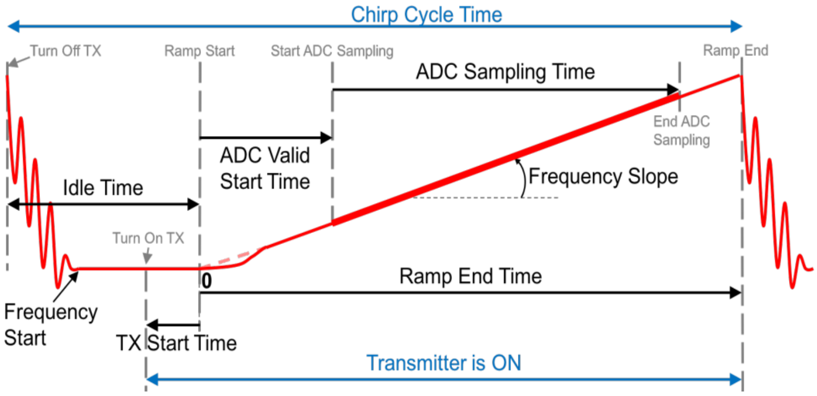


Figure 4.2: Chirp Design [54].

The impact of each parameter has been thoroughly evaluated through a series of simulations. These simulations involved applying various configurations to the radar using the mmWave Visualizer that will be described hereinafter.

To create a comprehensive configuration, it is essential to assign values to each configuration command as described in [54]. However, to configure the Chirp Cycle Time and make changes to the chirp design, the *profileCfg* configuration command is the key element that allows for these adjustments.

4.3.2 Programming Chirp Parameters

The room used in the present work has dimensions of 3.8 × 3 meters. Consequently, when configuring the chirp parameters, it is essential to take into account the entire dimensions of the room, enabling the radar to cover the area in which targets can be detected.

First and foremost, the maximum distance at which a person can be detected is a critical factor. By employing Equation (4.1), it is possible to calculate this distance, where F_s represents the Sampling Rate, and S is the slope of the chirp. This equation takes into consideration the speed of light c.

$$Range_{max} = \frac{F_s c}{2S} \tag{4.1}$$

At the same time, the slope follows the equation (4.2).

$$S = \frac{B}{T_c} \tag{4.2}$$

In Equation (4.2), B represents the bandwidth, and T_c is the chirp duration. Depending on the value of the bandwidth, it is possible to calculate the range resolution using Equation (2.6).

On the other hand, another key parameter to consider is the maximum velocity that can be captured. If this value is known, is possible to use Equation (4.3) to calculate the chirp duration T_c and the velocity resolution (2.8).

$$v_{max} = \frac{\lambda}{4T_c} = \frac{c}{4T_c f} \tag{4.3}$$

N represents the number of chirps, although the number of chirps can be treated as "loops" within the frame. In this context, "loops" refer to the number of chirps transmitted in a single frame. In the context of radar signal processing, a "frame" refers to a specific sequence or combination of chirp signals. When a frame is configured with two types of chirps, it means that the frame consists of a specific number of repetitions (loops) of the first chirp signal followed by the same number of repetitions of the second chirp signal.

For example, if a frame is configured with two chirp types, it means that the radar will transmit the first chirp N times in succession and then transmit the second chirp N times in succession within the same frame. This approach allows radar systems to gather data from both chirp types within a single frame.

Increasing the number of chirps in a frame can lead to improved velocity resolution. The frame parameters can be automatically adjusted, allowing for changes in values like velocity or range. This adjustment process can be facilitated using a TI tool known as the "mmWave Sensing Estimator", as mentioned in [55].

4.3.3 Simulations

The mmWave Visualizer offers a configuration panel that allows users to select various parameters, including the platform in use and the SDK version. However, the primary parameter that users can modify is the bandwidth, which can be set to values within the range of 76-77 GHz and 77-81 GHz. Additionally, users can select the real-time data plots they want to visualize, such as Point Cloud and Range Profile.

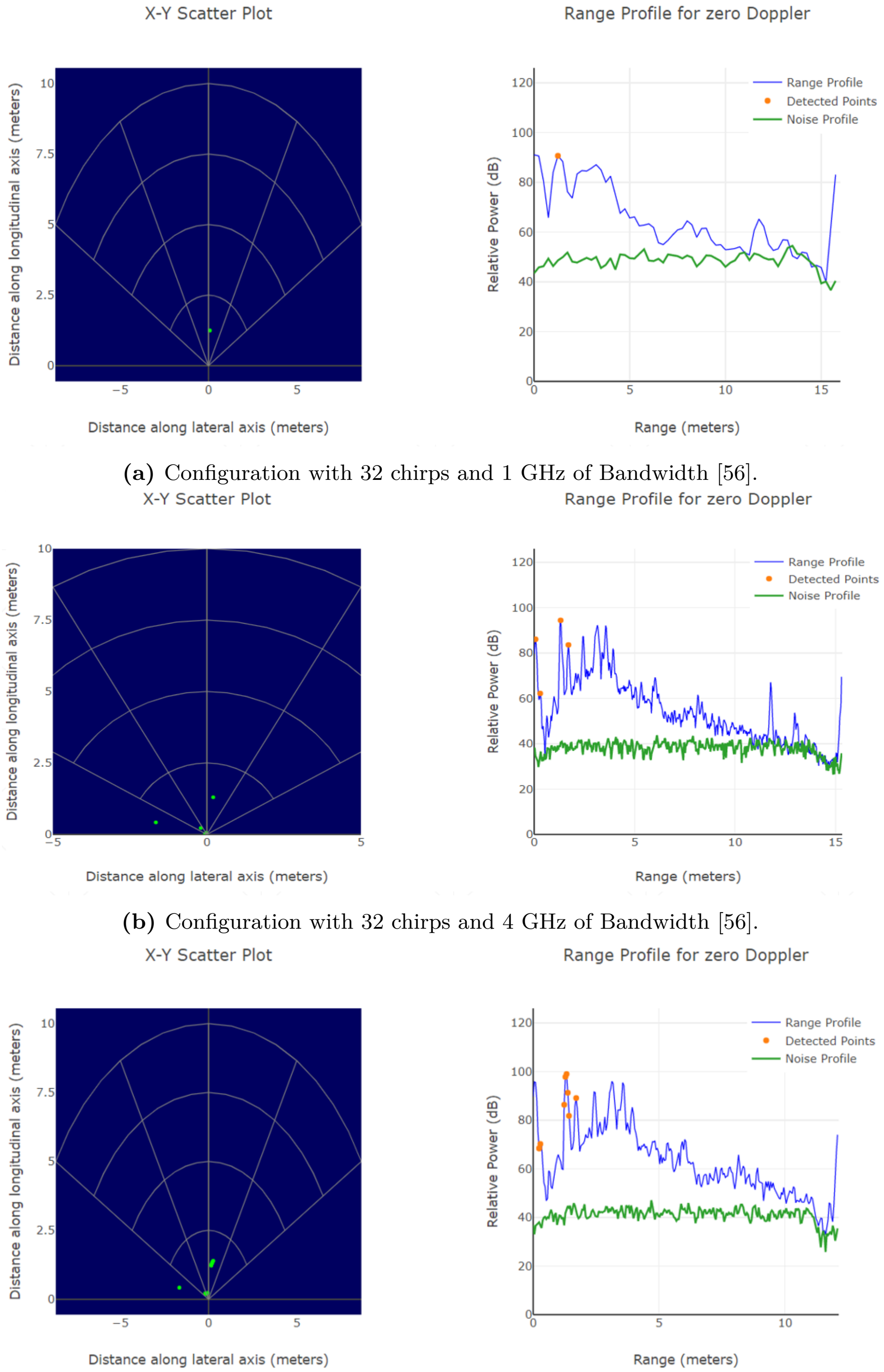
By configuring these parameters in the "Configure" panel, a setup is created for future radar applications. The ability to choose between two bandwidth options enables users to observe how the radar behaves under different configurations. While plot selection impacts the real-time visualization of processed data, it is the bandwidth of the chirps that significantly influences the radar operating mode.

Three different configurations were examined, including two with similar bandwidth settings and one with distinct values. The goal was to compare the results obtained under these configurations and determine which one performs best, shedding light on the most suitable chirp parameter values.

The simulations were conducted in two phases: initially testing the radar with a single box covered with aluminum paper and then introducing two boxes covered with aluminum paper into the setup, with dimensions (length, width, height) $21 \times 6 \times 17$ and $35 \times 12 \times 25$ cm. These boxes were positioned at specific distances from the radar. Despite efforts to control the environment, some anomalies, like unexpected points appearing in the Point Cloud plot, were observed due to environmental factors beyond control.

Simulation for One Cardboard box

In the initial simulation, a box was covered with aluminum paper and positioned at a short distance from the radar, approximately 1.5 meters away. Subsequently, three different radar configurations were tested, and the results from each configuration were compared. Visualizations of the Point Cloud, Range Profile, and Noise Profile can be observed in figures 4.3 and 4.3c for the different bandwidths tested.



(a) Configuration with 32 chirps and 1 GHz of Bandwidth [56].

(b) Configuration with 32 chirps and 4 GHz of Bandwidth [56].

(c) Configuration with 64 chirps and 4 GHz of Bandwidth [56].

Figure 4.3: Point Cloud (X-Y Scatter Plot) and Range profile and noise profile, for each tested bandwidth with 32 and 64 chirps.

When comparing the two figures in Figure 4.3, both configurations effectively detect the box placed in front of the radar. However, a difference is observed when using a 4 GHz bandwidth. In this case, the Point Cloud plot reveals that the radar captures more data points of the box compared to the configuration with a 1 GHz bandwidth. This indicates the advantage of using a 4 GHz bandwidth.

Furthermore, when examining the Range/Noise Profile, the values obtained with the 1 GHz bandwidth configuration are not as precise as those obtained with the 77-81 GHz configuration. This discrepancy arises because using a 1 GHz bandwidth restricts the number of ADC samples to only 64, which does not yield better results compared to using a 4 GHz bandwidth, where it is possible to work with 256 ADC samples, as explained in Section 4.3.2.

Subsequently, the configuration that uses 77-81 GHz is selected and compared with the third configuration that also uses 77-81 GHz.

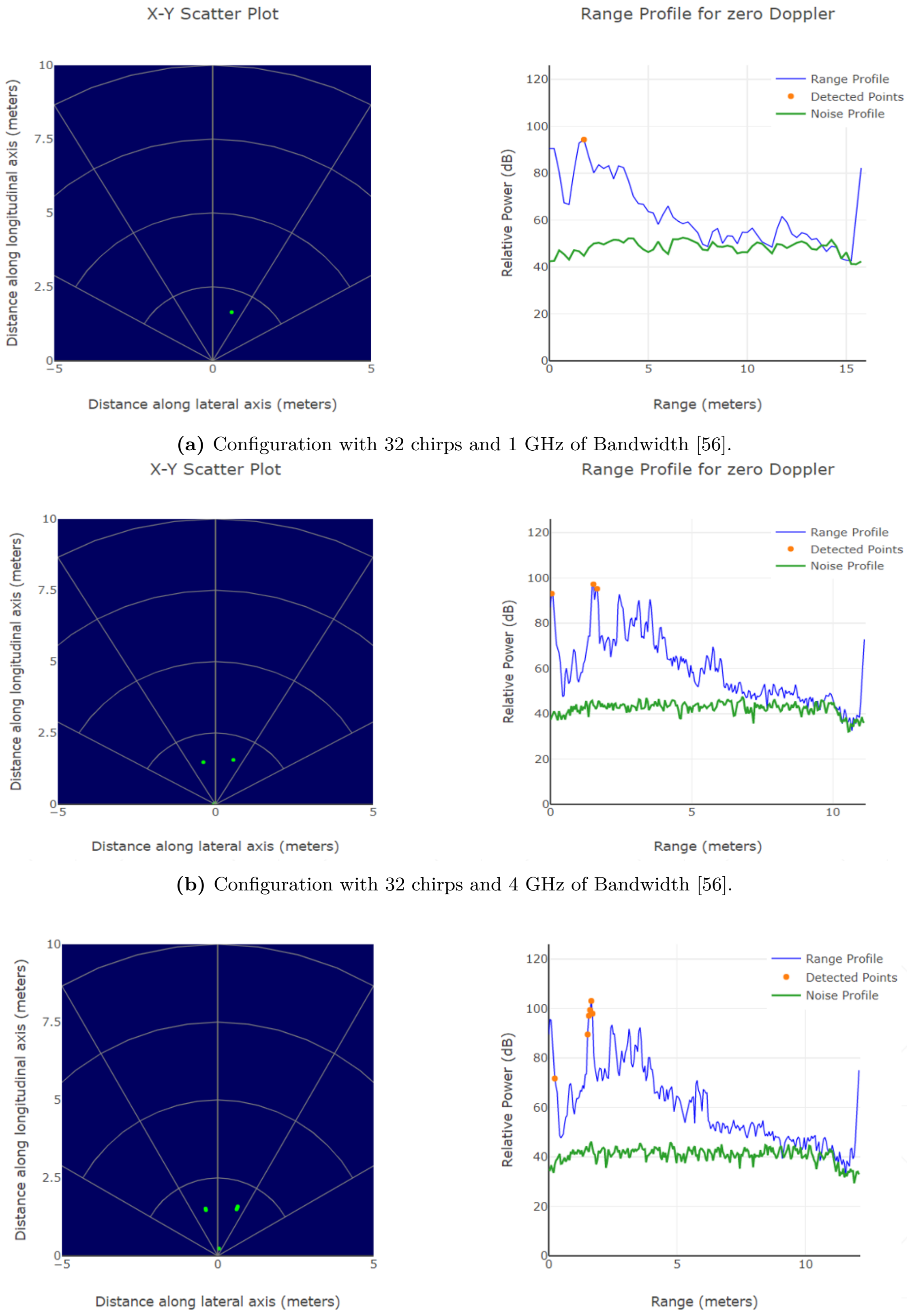
In Figure 4.3c, observes the radar behavior. When compared with Figure 4.3b, it becomes evident that the configuration with 4 GHz of bandwidth, is superior in terms of detection capability. This happens because the box is detected with more associated points, having increased range resolution.

With this simulation, the configuration with 4 GHz of bandwidth and 64 chirps, is the best choice to study and apply the parameters on the radar.

Simulation for Two boxes

The second simulation followed a procedure similar to that described in the previous case, with the exception that it involved two boxes. Both boxes were coated with aluminum paper and placed at the same distance from the radar side by side. However, one of the boxes was notably smaller in size compared to the other.

The simulation results, which showcase the different configurations observed through the mmWave Visualizer, are displayed in Figure 4.4.



(a) Configuration with 32 chirps and 1 GHz of Bandwidth [56].

(b) Configuration with 32 chirps and 4 GHz of Bandwidth [56].

(c) Configuration with 64 chirps and 4 GHz of Bandwidth [56].

Figure 4.4: Point Cloud (X-Y Scatter Plot) and Range profile and noise profile, for each tested bandwidth with 32 and 64 chirps.

When operating with a 1 GHz bandwidth, the radar encounters difficulties in detecting the smaller box. As a result, utilizing this configuration proves to be disadvantageous. In contrast, when the radar is set to operate with a 4 GHz bandwidth, as illustrated in Figure 4.4b, it successfully detects both boxes.

Subsequently, the same scenario was retested with the configuration with 4 GHz of bandwidth and 64 chirps. Upon comparing the two sets of plots, namely (4.4b) and (4.4c), it becomes evident that only a similar situation to section (4.3.3) occurs. This means that more data points are assigned to each box. However, upon analyzing the real-time data, the detection of the small box with the configurations with 32 chirps is not permanent, something that does not happen with a 64 chirps configuration.

Range resolution is indeed a significant factor influencing the ability to detect more data points within the boxes. The use of a 4 GHz bandwidth, as contrasted with 1 GHz, contributes to this phenomenon, as reflected in equation (2.6).

Table (4.4) represents the three chirps parameters used before.

Table 4.4: Frame parameters of the different configurations in mmWave Visualizer.

Frame Parameters (Units)	Test 1	Test 2	Test 3
Start Frequency Chirp(GHz)	76	77	77
Slope (MHz/us)	50	60	68
Frame duration (ms)	100	100	100
Sampling Rate (Msps)	5.333	2.5	5.500
Maximum Range (m)	12.799	8.930	9.706
Range Resolution (m)	0.250	0.044	0.047
Maximum velocity (m/s)	0.992	0.974	7.297
Velocity resolution (m/s)	0.124	0.122	0.456
Number of Loops	16	16	32

The configuration with 4 GHz, Test 3 in Table (4.4), of bandwidth while using 64 chirps, demonstrates the potential for delivering better results when compared to the configurations that uses 1 GHz of bandwidth or 32 chirps, leading to improved velocity resolution and an increased capacity for detection.

4.3.4 Impact of Chirp Parameters

Through various simulations conducted with three different configurations, it becomes evident that the parameters in the chirp configuration are of utmost significance. Even a minor alteration to one of these parameters can lead to a substantially different radar operation, highlighting the sensitivity and critical importance of these configuration settings.

Based on the simulations outlined in Section (4.3.3), it can be concluded that the number of chirps plays a significant role in enhancing detection performance. Additionally, bandwidth has a direct impact on improving distance resolution, in other words, a larger bandwidth results in better range resolution. This conclusion aligns with the calculations presented in Section (4.3.2).

To summarize the type of impact each parameter can have on radar behavior, Table (4.5) provides a comprehensive overview of these effects [42].

Table 4.5: Impact of the configuration parameters in the behaviour of the radar [42].

Requirement	**Resolution**	**Maximum Value**
Range	Bandwidth	Chirp Frequency Slope and Analog-to-Digital Converter (ADC) Sample Rate
Velocity	Frame Period	Duration of the Chirp

Choosing the appropriate parameters for a radar system is crucial to achieve a good balance between range and velocity resolution. A summary of the conclusions from Table (4.5) reveals the following:

- Bandwidth: The bandwidth has a significant impact on range resolution. Increasing the bandwidth improves the range resolution, while decreasing it degrades the range resolution;
- Chirp Frequency Slope and ADC Sample Rate: These parameters influence the maximum range achievable. Increasing the chirp frequency slope and ADC sampling rate extends the maximum range value;
- Frame Period: Increasing the frame period results in improved velocity resolution;
- Chirp Duration: Increasing the chirp duration leads to a lower maximum velocity value.

CHAPTER 5

People detection and Movement Classification

This chapter delves into the specifics of feature extraction, which involves selecting and computing relevant attributes from radar data to use in machine learning algorithms. It may also address the limitations of mmWave Visualizer in this regard and highlight how the People Counting software offers enhanced capabilities for feature extraction and analysis in the context of people counting and tracking. This chapter outlines the accuracy achieved by each tested ML algorithm, with the objective of determining the most suitable one for implementation in one final model.

5.1 People Counting vs mmWave Visualizer software

The mmWave Visualizer software is indeed valuable for comprehending how changes in chirp parameters impact radar behavior. However, it comes with certain limitations. For instance, the number of points in the Point Cloud must be high, even for relatively large objects. Additionally, when a person starts moving, the data can include several random points, complicating feature calculations.

TI has developed another software called "People Counting Software", capable to overcome this problems. This software is designed for tracking and counting people and offers the capability to save various data while it is actively tracking individuals. This includes data such as Doppler and range values, the position of the person at each time step, etc. The output data can be visualized using the People Counting User Guide [60]. This capability makes the People Counting software particularly well-suited for the purpose of this dissertation.

5.1.1 Radar Placement

Identifying the optimal radar placement is of utmost importance to ensure the comprehensive capture of all movements performed by the target individuals and accurate detection of their positions within the room. The radar placement should be strategically chosen to provide reliable detection across all positions that a person may pass through, ensuring consistent and accurate confirmation of their movements.

To determine the optimal placement, the radar was tested at various positions, including on the floor, at a middle height (approximately 1.5 meters) and on the ceiling of the room. The conclusions regarding radar placement effectiveness were drawn during testing using the People Counting Software. This evaluation was conducted in a room with specific dimensions, which are:

- Height - 2 meters
- Width - 3.8 meters
- Length - 3 meters

Floor

Positioning the radar on the floor can lead to inconsistent detection results. For instance, when a person is standing, the radar may only detect them if they make abrupt movements. Furthermore, even when initial detection occurs, the radar might struggle to maintain consistent tracking of the person movements.

The radar ability to maintain detection is heavily influenced by the body movements and balance. When a person is seated, their body tends to be more static, making it challenging to maintain detection without abrupt movements. Conversely, when a person is standing, the natural body balance movements, even if subtle and unconscious, can assist in maintaining detection for a brief period. This disparity in body dynamics between sitting and standing positions can affect the radar capability to consistently track a person movements.

When a person is walking in the room, the radar primarily focuses on the target feet. This approach may not be ideal since it could lead to issues with detecting movements where the person is at a height of 30 cm from the ground, such as running on a treadmill.

Given these considerations, placing the radar on the floor does not appear to be the best decision for consistent and accurate movement detection.

Ceiling

The radar positioning and its line of sight are critical factors that can affect its performance in various scenarios.

For activities like abdominal exercises or other floor-based movements, where the person is lying down, the radar might not be able to detect the individual due to the radar angle and the obstructed view.

In contrast, activities like walking or running on a treadmill, where the person is in an upright position, are more likely to be accurately detected by the radar.

Ceiling-mounted radar may not be the best option for scenarios that involve a lot of floor-based exercises or activities where the target is positioned close to the ground, this means that do not generally cover all possible movements, this means that do not generally cover all possible movements.

Middle Height

Placing the radar at a middle height, around 150 cm from the floor, seems to have resolved many of the detection issues mentioned above and allows the radar to consistently detect individuals in various types of movements. This positioning strikes a balance between floor-based and ceiling-based radar placements, making it a practical choice.

However, it is important to note that radar systems may still face challenges when it comes to detecting individuals in certain corners of the room. To address this, the radar should be strategically positioned in the room to ensure complete coverage of the entire room area.

Middle Height Lateral

Initially, the radar is placed at a height of 150 cm from the floor, positioned in the center of the room wall. In this configuration, there are limitations in detecting targets that are directly alongside the radar. Consequently, if a person is located in one of the closest corners of the room, the radar may not detect them.

Figure 5.1 provides a visual representation of this scenario, showing that the radar field of view is restricted, and targets in corners 1 and 2 (highlighted in red boxes) may be difficult to detect.

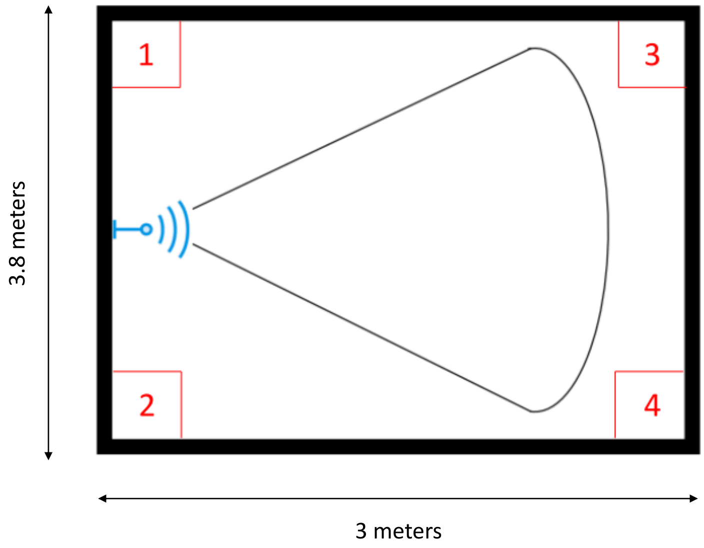


Figure 5.1: Radar Placed in the middle of the lateral wall of the room.

Middle Height Corner

When the radar is positioned in one corner of the room, it becomes possible to reach even the closest corners, addressing the previous limitation. However, the opposite corner remains a challenge for detection because it is the furthest point from the radar.

To overcome this challenge, it is essential to set the maximum detection distance to reach the furthest corner effectively. In Figure 5.2, the line of sight with the radar located in a room corner is illustrated. By determining the appropriate maximum distance and placing the radar in one corner of the room, it is clear that this configuration offers optimal coverage.

Positioning the radar in a corner results in a broader and more comprehensive surveillance area within the room.

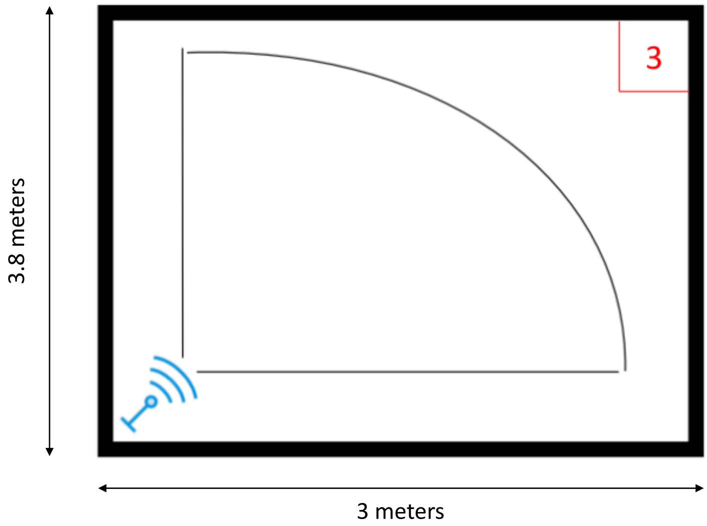


Figure 5.2: Radar Placed in the corner of the room.

Appropriate Radar placement

The placement of the radar on the ceiling or the floor has limitations in capturing all possible movements of a person. When a person is lying down, a ceiling-mounted radar cannot detect them. Conversely, if a person is running on a treadmill and the radar is placed on the floor, it may not detect the person either.

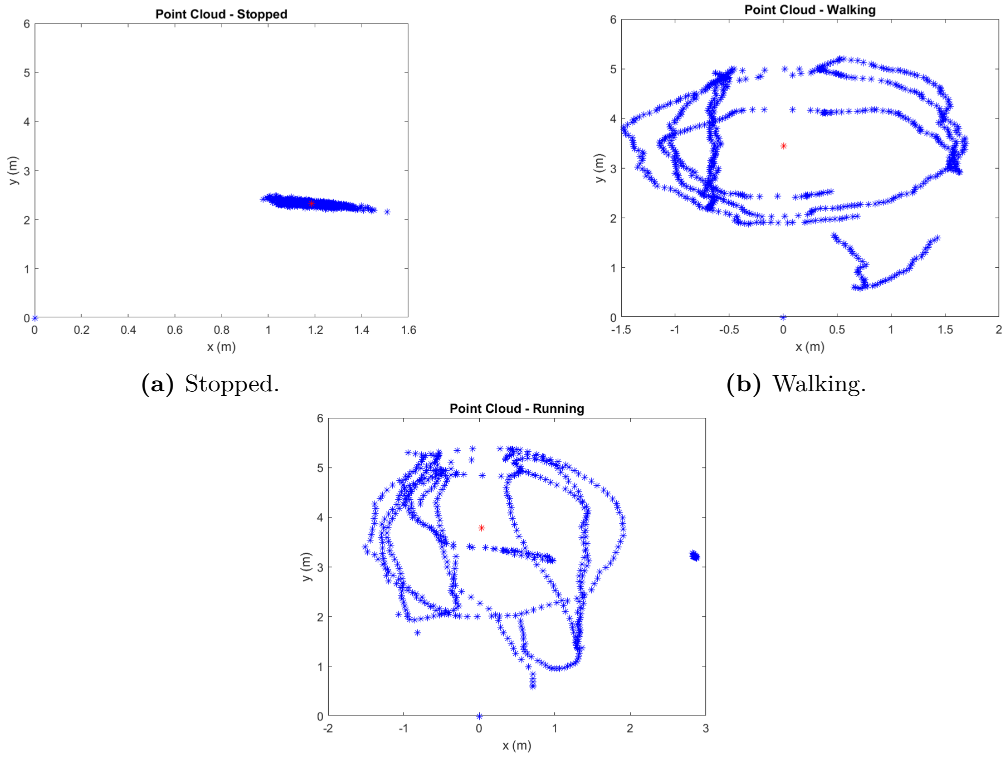
The optimal radar position is at a middle height, where it can capture all movements effectively. However, if the radar is positioned in the center of a wall at middle height, its line of sight may not cover the closest corners to the radar. This means that if a person is in one of the closest corners, the radar may not detect their movements. To address these limitations, the best positions are the corners of the room. Although, it is necessary to adjust the radar signal parameters to maximize the coverage range and ensure that it can effectively monitor the entire room area.

5.1.2 Point Cloud

Three different types of movements were tested for further comparison between them. Those movements are:

- Stopped
- Walking
- Running

Initially, a comparison was made by creating Point Cloud representations of each movement. This was achieved by using resources in MATLAB to generate the Point Cloud, as illustrated in Figure 5.3, for each activity, allowing for the identification of differences between them.



(a) Stopped.

(b) Walking.

(c) Running.

Figure 5.3: Point Cloud of each movement.

In each of the figures contained in Figure 5.3, the blue point located at coordinates (0,0) represents the position of the radar. The y-axis denotes the distance in front of the radar, the x-axis represents the lateral distance to the radar, and the red point signifies the centroid of each Point Cloud.

Figures 5.3b and 5.3c illustrate the trajectory of a person (depicted as blue points). However, solely analyzing the Point Cloud, it is challenging to distinguish between running and walking movements.

On the other hand, one can determine if a person is in motion by comparing the Point Cloud between Figure 5.3a and Figures 5.3b and 5.3c.

It is reasonable to expect that the mean distance from all points to the centroid of the cloud would be lower when a person is stationary compared to when they are walking or running. However, this distance measurement alone may not be sufficient to differentiate between walking and running.

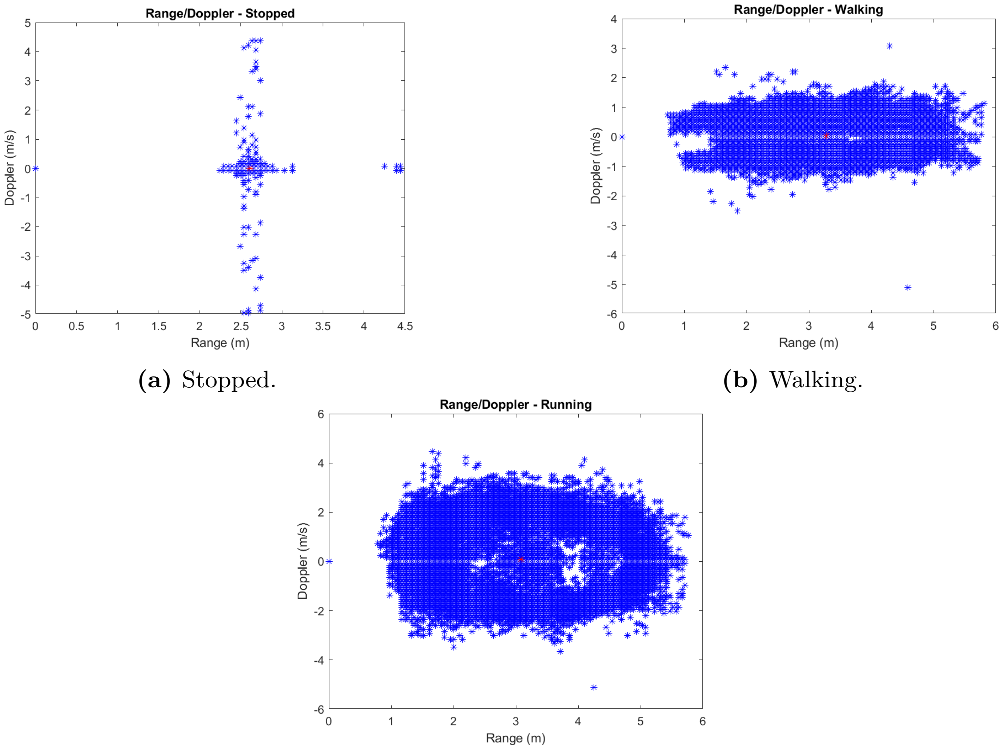
This limitation arises because the centroid reflects the average values of the distance to the radar in both the frontal (y) and lateral (x) directions. If a person runs within a confined area, the value of this feature will be small. Conversely, if a person walks in a more spacious area, the feature will have a greater value. This implies that the feature value can vary not only based on the type of movement but also on the physical space in which the movement occurs. Therefore, it is essential to consider additional features or patterns to accurately distinguish between walking and running.

5.1.3 Range and Doppler values

Using the People Counting software, two parameters retrieved are:

- Range
- Doppler

Figure 5.4 represents a graph of Doppler over Range for different movements, illustrating how Doppler values vary with distance from the radar. This graph can be essential for analyzing and differentiating various types of movements, as each type of movement may exhibit distinct Doppler characteristics at different ranges.



(a) Stopped.

(b) Walking.

(c) Running.

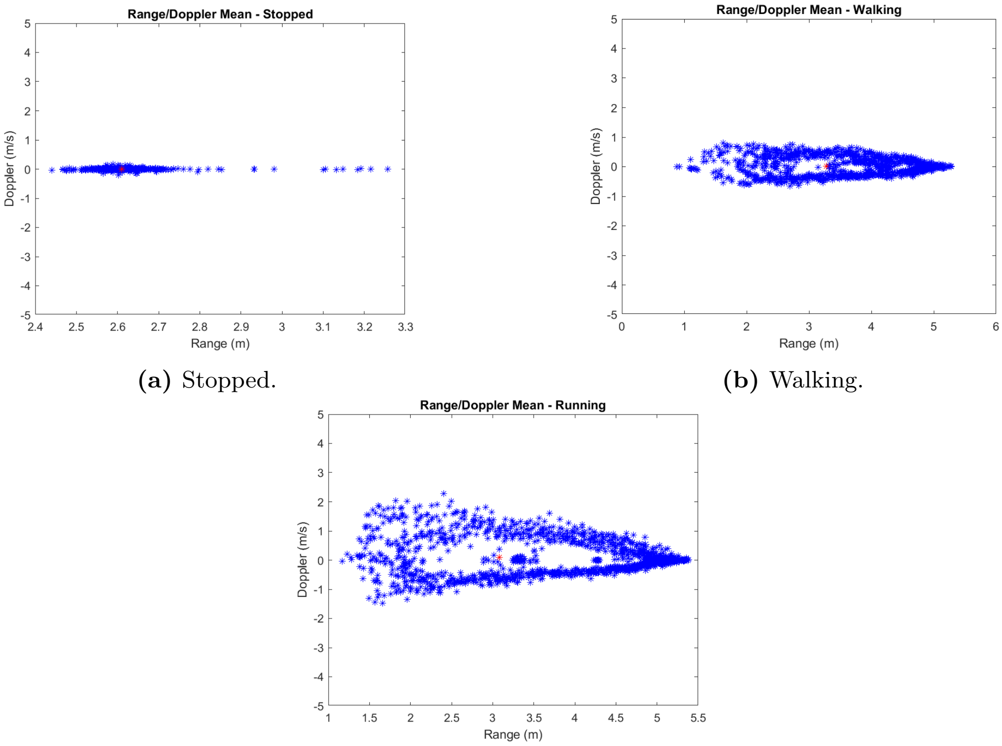
Figure 5.4: Range-Doppler graph for each movement.

In Figure 5.4, the data points represent information collected from every frame during the recording. However, some of these data points can be misleading due to clustering or radar non-detections, which can lead to incorrect values.

To address this issue, a mean value for each range and Doppler parameter in each frame is calculated. This approach helps filter out values that do not align with the greater density of points in the data volume.

For example, in Figure 5.4a, when a person is stationary, the Doppler values should not be close to 5 m/s, as there is little to no movement. However, the data may show some Doppler values, close to 5 m/s, that do not match this expectation. Similarly, some range values, the ones that are close to 4.5 meters, might deviate from the expected values, approximately 2.5 meters.

By calculating the mean of each Doppler and range value in each frame, the results can be observed in Figure 5.5. These calculated means provide a more coherent representation of the data, making it easier to interpret and analyze compared to the raw data in Figure 5.4.



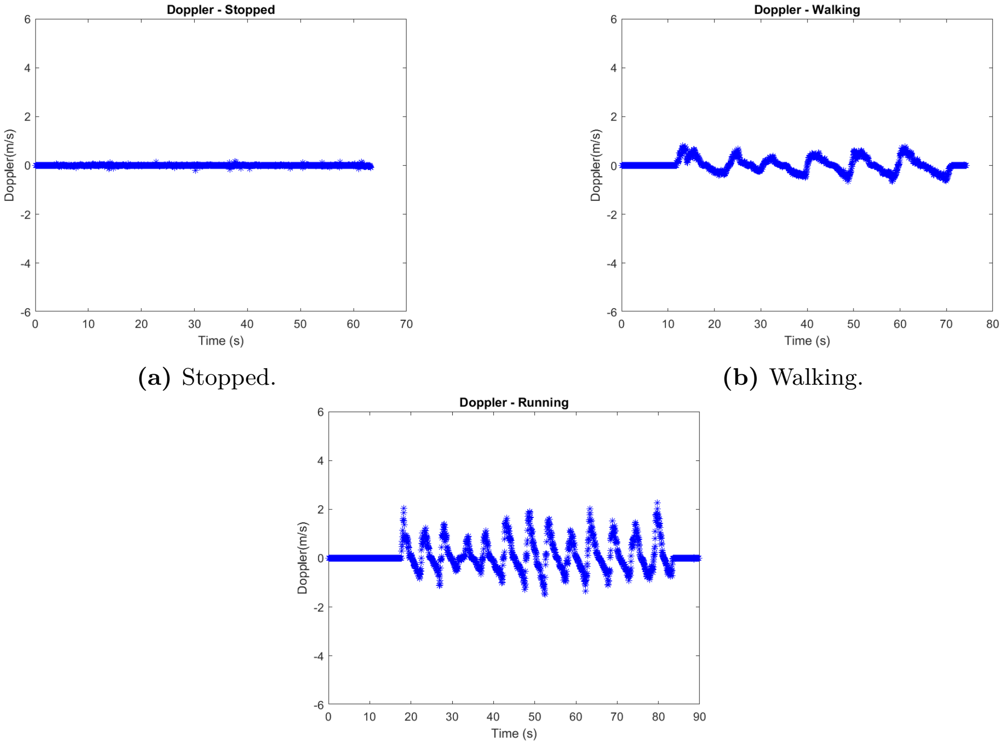
(a) Stopped.

(b) Walking.

(c) Running.

Figure 5.5: Range-Doppler Mean graph for each movement.

Figure 5.6 shows the representation of the Doppler values variation in time. Doppler values are expected to be more pronounced when a person is running (Figure 5.6c) because the velocity achieved during running is generally higher. Consequently, the Doppler value when a person is stopped will be close to zero (Figure 5.6a) due to the absence of movement.



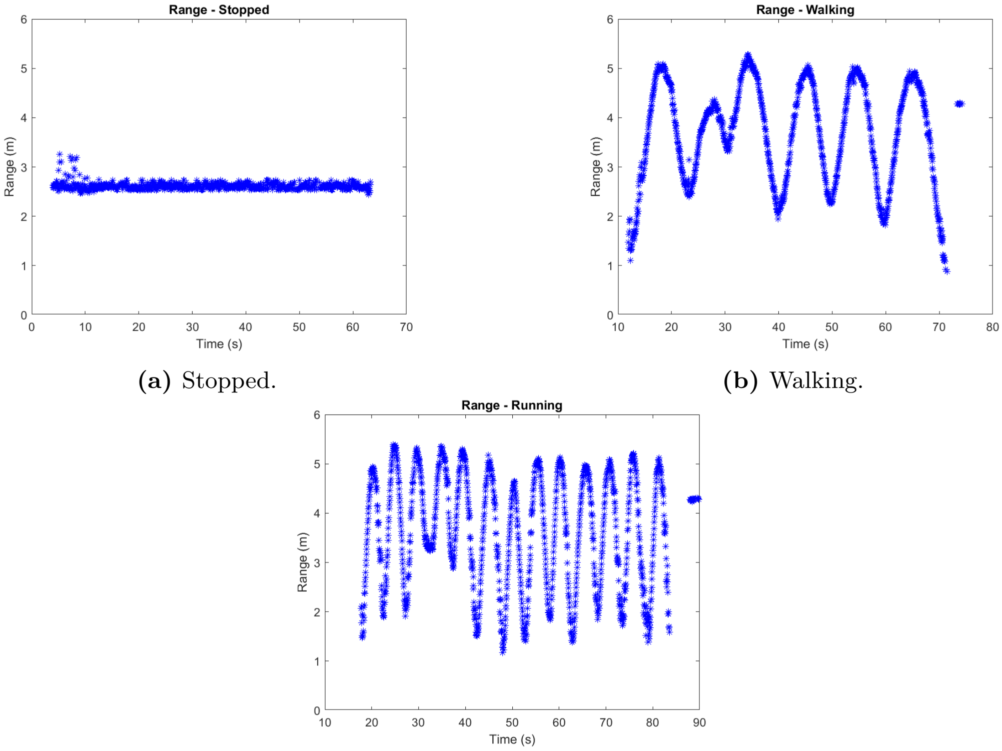
(a) Stopped.

(b) Walking.

(c) Running.

Figure 5.6: Doppler along time for each movement.

When a person is walking, the Doppler value will be lower than when running, reflecting the slower velocity associated with walking compared to running, as shown in Figure 5.6b. These distinctions in Doppler values based on different activities are essential for accurately classifying and understanding human movement in radar data.



(a) Stopped.

(b) Walking.

(c) Running.

Figure 5.7: Range along time for each movement.

Figure 5.7 shows the representation of the Range values variation in time. In Figure 5.7b and 5.7c, it is evident that the distance covered when running is greater than the distance covered when walking, within approximately the same time window. The frequency of fluctuations in the range values, in the same time window, with both increases and decreases over time, is much higher when running compared to walking, although the values presented in Figure 5.7 depends in the trajectory that the subject has made.

5.2 Label Selection

In the context of the study scenario, it is essential to identify probable movements in an indoor environment, as this is crucial for building a practical model. Activities like running are less common in indoor environments compared to activities like walking or standing. Running typically occurs sporadically, often when someone is in a hurry or late for an appointment, or during exercise routines. The focus is on detecting and classifying the more common and fundamental movements, such as walking and stopped or working out, which are typical activities in indoor.

In this context, the relevant movements for classification include:

- Stopped
- Moving
- Workout

These movements serve as global labels because a person can be either lying down or sitting and still be considered as "stopped". Therefore, the labels, along with their respective sub-labels, can be summarized as depicted in Figure 5.8.

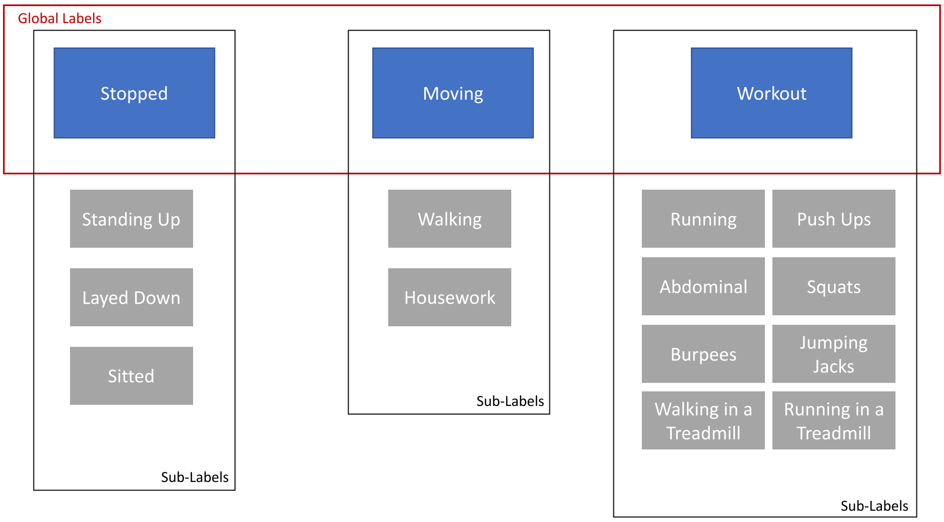


Figure 5.8: Label Selection.

Emphasizing that the global labels are the ones that will be distinguished and considered for classification.

5.3 Data Acquisition

To enhance the accuracy of ML algorithms, having access to a large and diverse dataset of training examples is crucial. A substantial and varied collection of data samples, covering a wide spectrum of feature values, enables a more comprehensive understanding of how different movements or patterns can be accurately classified.

In this particular case, the data collection involved ten different individuals, with each person performing ten movements for each global label. This results in a total of 100 data collections for each label and 300 observations in total. Each movement had a duration of approximately one minute.

5.4 Feature Extraction

When using the People Counting software for data collection, information regarding various activities, such as a person walking in a room, is recorded and stored in a .mat file.

In MATLAB, this data is analyzed and organized with the aim of creating features that are best suited for classifying labels. The values used in the study, which were extracted from the .mat file, are presented in table (5.1). This data likely serves as the basis for further analysis and implementation of machine learning algorithms for activity classification.

Table 5.1: Signal variables obtained from People Counting Software.

	Value in study
1	Range
2	Doppler
3	SNR
4	Positions on axis x
5	Positions on axis y

The signal variables presented in Table (5.1) can be used to extract various features for the purpose of distinguishing different movements.

5.4.1 Velocity and Acceleration calculation

Velocity is undeniably a critical feature for distinguishing between the three movements, as it imparts valuable information. When a person is stationary, the velocity is approximately zero. When running, the velocity is higher than when walking. Comparable distinctions can also be made if acceleration values are taken into consideration. These parameters provide essential insights for effectively classifying movements.

Hence, various methods for measuring velocity are considered, and the one that aligns best with the intended purpose is selected.

Coordinates Method

The People Counting software output data provides velocity and acceleration values in coordinates, but these values require correction. For each movement, the velocity calculation is based on the coordinate values of velocity. This calculation can be observed in Figure 5.9.

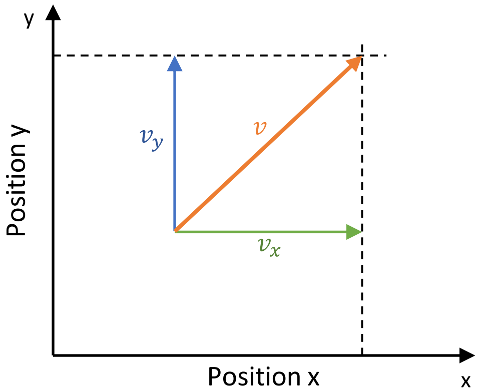


Figure 5.9: Velocity coordinate calculation.

Referring to Figure 5.9, the velocity value when given the velocity coordinates can be summarized by the formula (5.1).

$$velocity = \sqrt{(v_x^2) + (v_y^2)} \tag{5.1}$$

Using the results provided by the People Counting software, the mean velocity values obtained are presented in Table (5.2).

Table 5.2: Velocity values for the coordinates method.

Movement	Final Velocity (m/s)
Stopped	0.5291
Walking	0.6710
Running	1.2756

The velocities presented in Table (5.2) are expressed in meters per second (m/s). Hence, when scrutinizing these values, it becomes apparent that it would be highly unlikely for a person to have nearly the same velocity when they are stationary compared to when they are walking. The difference in velocity between these two states should be easily distinguishable.

Outlier Removal with MAD Method

The next step involves determining the actual velocity values, and for this purpose, the position values in the Point Cloud were employed. The formula in (5.2) can be utilized to estimate the velocity value in a uniform motion.

$$velocity = \frac{x_2 - x_1}{t_2 - t_1} \tag{5.2}$$

In the previous formula, x_2 represents the last position of the target and x_1 is the initial position, with the respective times of last position t_2 and initial position t_1.

This principle is applied to the data presented in figures 5.3, where the positions are represented in 2D coordinates. The time between each point corresponds to the frame time in the configuration.

Figure 5.10 offers a visual representation of a target trajectory. Each blue point, separated by time frame t_f seconds, on the graph denotes a specific position that the target has occupied, giving a clear visual representation of the path it has followed.

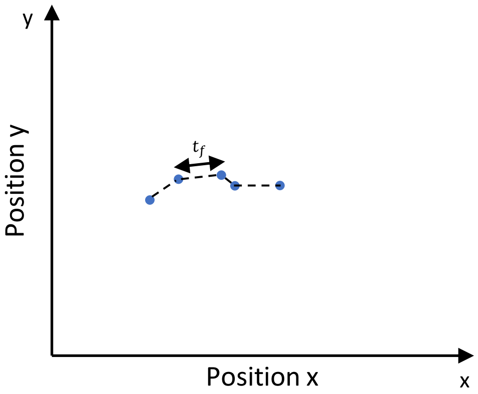


Figure 5.10: Velocity calculation with positions coordinates.

The equation (5.3) shows how to calculate the distance between each point, taking into account the time intervals between frames. This calculation helps determine the distance covered between each frame, which is essential for tracking and understanding a target movement.

$$velocity = \frac{\sqrt{(x_2, y_2)^2 - (x_1, y_1)^2}}{t_f} \tag{5.3}$$

In Figure 5.10, some points are more distant from each other than others, yet the time interval between them remains consistent. This pattern is where the velocity value can either increase or not. When a person is stationary, the points tend to be much closer to each other within each time frame.

Conversely, in the case of running, it is expected that with each successive time frame, the points will be farther apart from each other. Table (5.3) provides the mean velocity values for each type of movement, offering insight into the differences in movement characteristics.

Table 5.3: Velocity values for the Outliers Remotion with MAD method.

Movement	Final Velocity (m/s)
Stopped	0.5803
Walking	0.7564
Running	0.8191

Table (5.3) effectively distinguishes the three movements and provides corresponding values. It indicates that the velocity for stationary movement is the lowest, for running it is the highest, and for walking, it falls in between.

However, these velocity values may not always align with practical expectations. For instance, the recorded running speed of 2.9488 km/h (0.8191 m/s) might seem low, as people can run at much higher speeds, even within a confined area.

This discrepancy can be attributed to a limitation in the radar performance. When the radar is in operation, it may not continuously detect a person, resulting in inaccurate velocity values. Figure 5.11 illustrates this point, where a person is consistently running, but there are periods when the radar fails to detect the target, leading to recorded velocities of zero during those intervals. Figure 5.11 is created with resource to MATLAB by using the values of positions in axes x and y, provided by the outcome data of the people counting software, and the t_f time frame value defined as 50 ms.

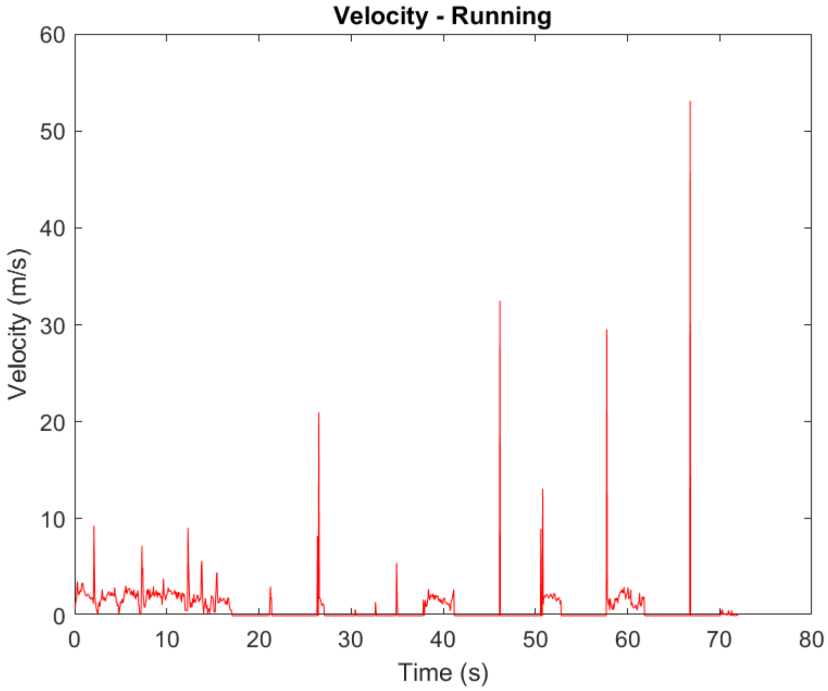


Figure 5.11: Velocity values with outliers.

Another underlying issue is the velocity value recorded when a person is detected again after a period of no detection. In the data processing, the assumption is that the person has covered a significant distance in a short time frame, resulting in an exaggerated velocity value. This explains the presence of outliers in Figure (5.11), where the velocity values spike due to the rapid change from no detection to detection.

Subsequently, the points where the target was not detected were removed from the dataset to obtain more accurate data values. As shown in Figure 5.12, this resulted in a significant reduction in the time evaluation, passing approximately from 72 to 28 seconds, aligning more closely with the actual movement patterns of the person.

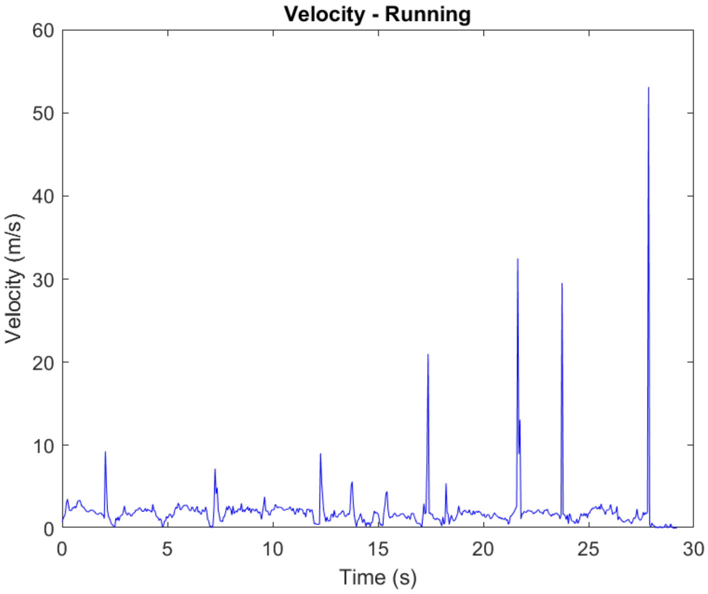


Figure 5.12: Velocity data with just detection values.

The next crucial step is to address the outliers in the data. This is achieved using the *rmoutliers()*[1] function in MATLAB, resulting in a more accurate representation of the velocity data. For this function, an outlier is identified as a data point whose value deviates by more than three times the scaled MAD from the data median [61]. The process of *rmoutliers()* and identifying the outliers can be described in the next 4 steps:

1. **Calculate the Median:** The first step involves finding the median value of the dataset;
2. **Calculate the Median Difference:** Calculate the absolute difference between each data point in the dataset and the median value. This will give a set of difference values. Then, is calculated the median of the set of difference values obtained;
3. **Set a Limit (MAD):** In this step, a limit is established. It is calculated 3 times of the median difference calculated in step 2;
4. **Identify Outliers:** All values greater than limit calculated in 3rd, are assumed to be outliers.

Figure 5.13 demonstrates the data after the removal of outliers.

[1] The *rmoutliers()* MATLAB function is a useful tool for detecting and removing outliers from the data.

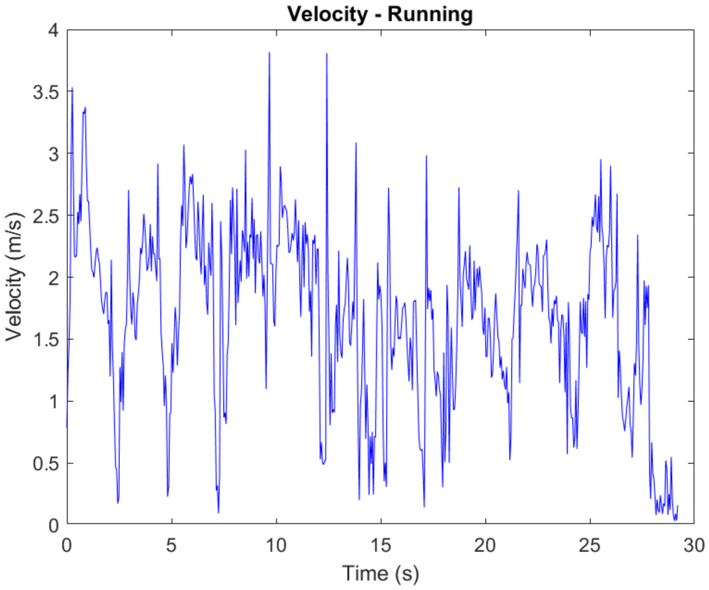


Figure 5.13: Velocity data without outliers.

The same procedure is repeated for the acceleration values, following the equation (5.4).

$$acceleration = \frac{\sqrt{(x_2, y_2)^2 - (x_1, y_1)^2}}{t_f^2} \tag{5.4}$$

After performing these calculations, the mean values of velocity and acceleration at all times are presented in Table (5.4).

Table 5.4: Velocity and acceleration values for the Outliers Remotion with MAD method.

Calculus	**Stopped**	**Walking**	**Running**
Velocity	0.5380	0.7866	1.6617
Acceleration	10.7606	15.7329	33.2344

The expectation is that the velocity when a person is stopped should be approximately zero. However, even when a person is stopped, other subtle motions, such as body balance or diaphragm motion during breathing, can influence the calculations. This means that the velocity when a person is stopped is indeed close to 0 m/s, due to the micro-Doppler variations. Nevertheless, the ability to distinguish between these three movements is still achievable.

In the case of speed, for example, when a person is running, the value is more likely to be in a small range, such as 5.9821 km/h (1.6617 m/s). This illustrates that the system can effectively differentiate between different movements.

Set of frames Method

This method initially resembles the second method, where values of velocity equal to zero are eliminated. However, in this approach, additional steps are taken to remove outliers from the data. Velocity values are estimated over multiple frames, and through experimentation, it was determined that the best interval for calculating the velocity value is within a span of 20 frames. The choice of using 20 frames was based on achieving the best and most realistic results, as shown in Table (5.5). This decision involved conducting multiple iterations of the test with different frame spans to obtain realistic velocity values while maintaining a strong correlation in the dataset with fewer outliers and greater correct data consistency. Opting for a higher number of frames would have resulted in significantly longer testing periods, leading to a higher probability of encountering outliers and potentially inflating the velocity values. Therefore, the main reason for selecting a specific set of frames was to reduce the impact of outliers by effectively "comparing" them to the other "correct" values in the dataset segment. This approach was aimed at minimizing the influence of outliers by ensuring that their deviations from the norm were less significant. So, the selection of a set comprising 20 frames was made through a process of trial and error. This specific quantity was chosen as it emerged as the most optimal after a series of systematic experiments and evaluations.

This method can be visually represented in Figure 5.14, and it is further described by Equation (5.5).

$$Velocity = \frac{x_{20} - x_0}{20 \times t_{frame}} \tag{5.5}$$

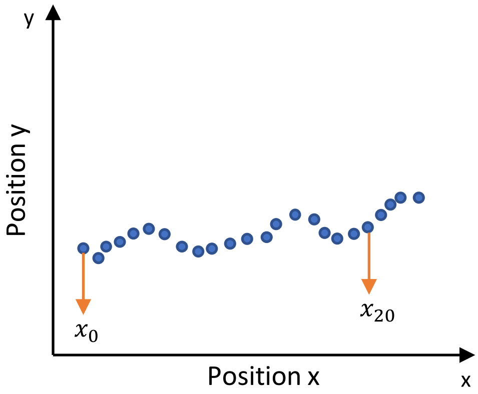


Figure 5.14: Velocity estimation between several frames.

Table (5.5) displays the velocity values calculated using this method. These values are more representative of real-world scenarios, showing relative differences in velocity between the three different movements. This approach appears to provide more accurate and meaningful results in terms of velocity estimation.

Table 5.5: Velocity values for the set of frames Method.

Movement	Final Velocity (m/s)
Stopped	0.1158
Walking	0.9530
Running	1.2549

Kalman Filter Method

The fourth method involves the implementation of a filter to eliminate outliers in the velocity data.

The Kalman Filter is particularly useful in scenarios where velocity values may contain outliers, which is quite relevant in the context of radar data processing. Non-detections by the radar can introduce errors into the data, making it essential to have a mechanism for handling such situations.

The Kalman Filter operates by making predictions about the expected values at each data point based on the assumption that a person acceleration is constant. It then compares these predictions with the actual data and makes corrections accordingly. This process enables the filter to generate a more accurate trajectory for the person being tracked.

The use of the Kalman Filter is illustrated in Figure 5.15, demonstrating how it helps in removing outliers and improving the accuracy of velocity and trajectory estimation.

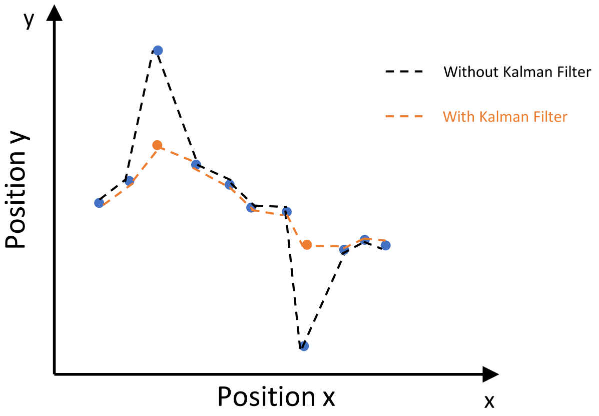


Figure 5.15: Kalman Filter.

So the velocity estimation can follow the next steps (with visualisation):

- Step 1:

The values of velocity are taken by using the values of positions on x and y axes and the 50 ms of time frame (Figure 5.16). This figure is accomplished with resource to MATLAB.

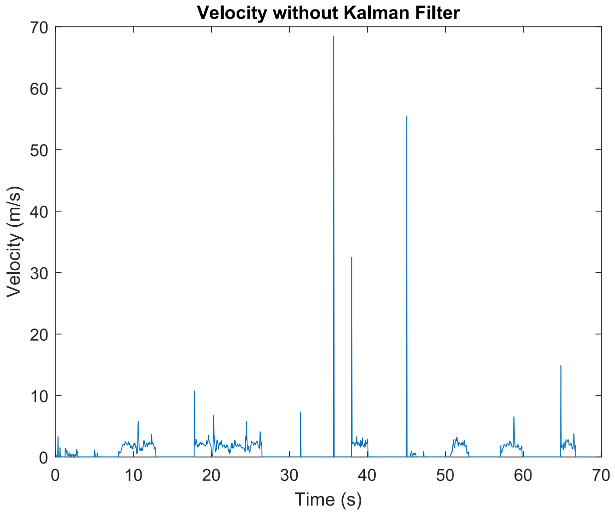


Figure 5.16: Velocity without Kalman Filter (frame by frame).

- Step 2:

The Kalman Filter is applied (Figure 5.17).

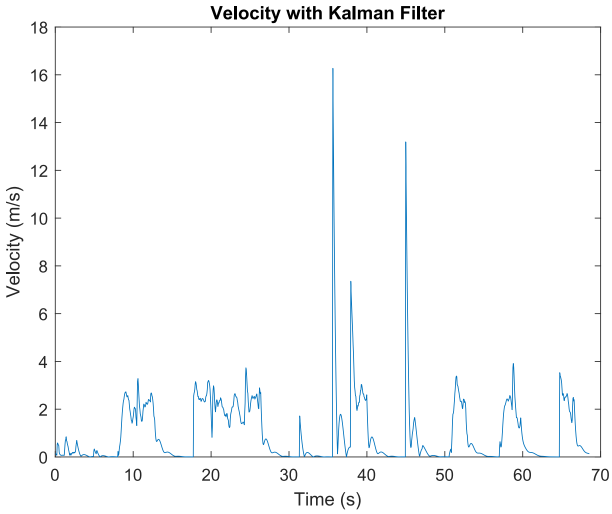


Figure 5.17: Velocity with Kalman Filter (frame by frame).

In Figure 5.17, there are still some instances where the velocity values are equal to zero. These zero values correspond to the moments when the radar is not detecting the person, and the velocity data is unavailable or not measured.

- Step 3

During this step, only the velocity values above 0 m/s are utilized, as shown in Figure 5.18. This filtering process excludes velocity values of 0 m/s, focusing on instances when the radar detects movement.

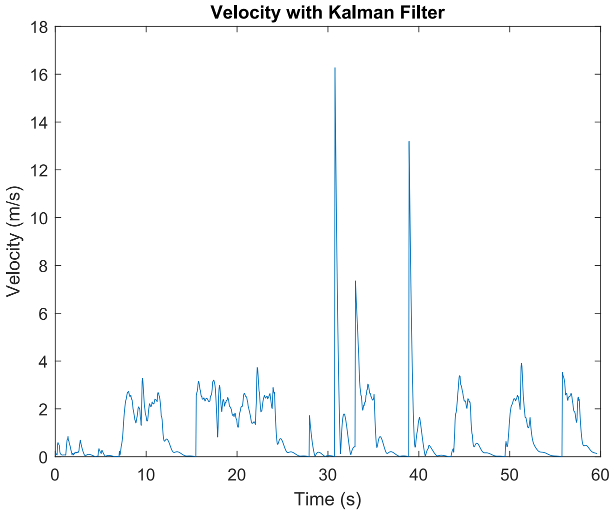


Figure 5.18: True values of velocity with Kalman Filter (frame by frame).

However, it is important to acknowledge that the dataset contains some outliers that do not accurately represent real-life scenarios. For instance, there are velocity values reaching approximately 43.2 km/h (12 m/s), which is highly improbable for an individual to achieve in the given conditions.

To address this issue, the previous steps are repeated, but with a modification similar to the third method. Velocity is calculated at intervals of 20 frames. The resulting velocity values are presented in Figure 5.19 and summarized in Table (5.6). This adjustment aims to provide more realistic velocity values by smoothing out the outliers.

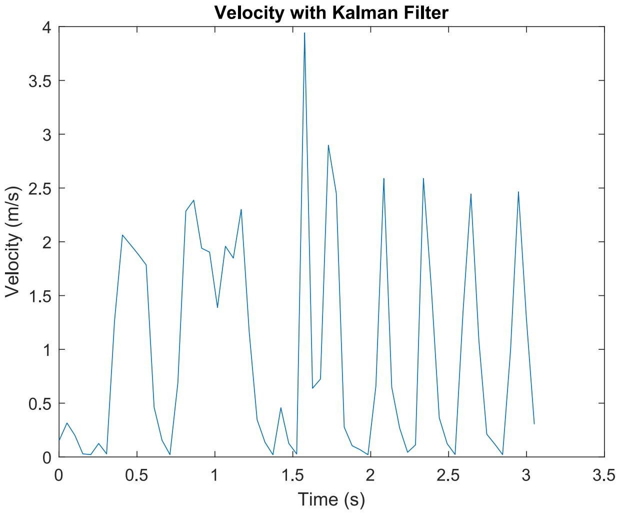


Figure 5.19: True values of velocity with Kalman Filter (20 by 20 frames).

Table 5.6: Velocity values for the Kalman Filter method.

Movement	Final Velocity (m/s)
Stopped	0.1198
Walking	0.8642
Running	0.8816

In these values, the velocities for walking and running are very close, indicating insufficient differentiation between these two types of movements.

In conclusion, the outliers remotion method, using *rmoutliers()*) MATLAB function, appears to be the most effective for estimating target velocity because it provides the best differentiation of velocity values between the three movements and can closely represent real-time conditions.

The set of frames method also yields good results, but the choice of the frame window for calculating velocity can introduce some variability. Selecting 20 frames may be appropriate in some situations, while in others, 25 or 15 frames might be more suitable. This uncertainty in the optimal frame window makes the set of frames method less preferable.

Additional Features

Doppler values are essential for distinguishing movements. If a person is stationary, the Doppler value is nearly zero (m/s), but it increases significantly when the person is walking.

Doppler information is crucial because it allows the radar to differentiate between a stationary person and one running on a treadmill, even though their positions may appear the same.

Similarly, the range, or the distance of the target from the radar over time, is another crucial feature. It is expected that the range will remain relatively constant when a person is stationary but will vary when the person is moving in the room.

Expanding on the concept of range, the displacement that the target undergoes over time is another valuable feature. The total displacement made by a person walking in the room should be more significant when compared to when they are stationary. This can be calculated using the position values along the x and y axes. Additionally, the displacement made solely along the x and y axes can be used as separate features to distinguish different labels.

Using the values of displacement and employing the chosen method (5.4.1) for estimating target velocity, features such as the mean and median velocity of the target, as well as the mean acceleration calculated using Equation (5.4), can be generated.

Another potential feature to extract is the Jerk value, which can provide information about how quickly the velocity or acceleration is changing. Equation (5.6) can be used to calculate Jerk.

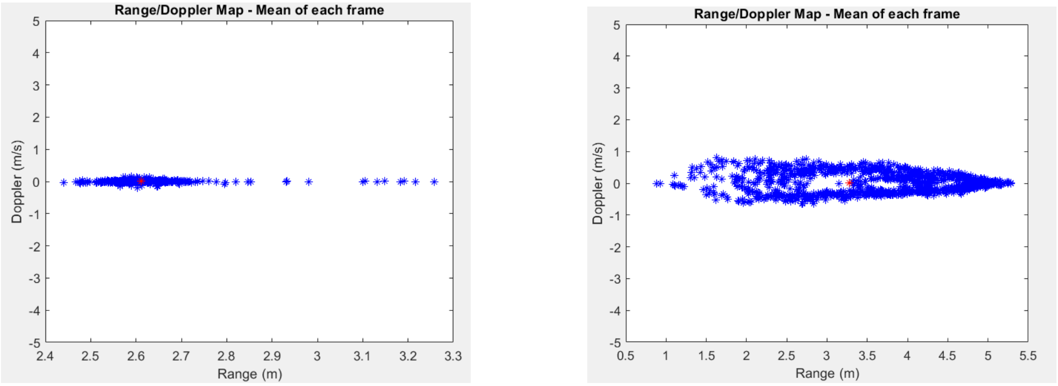
$$Jerk = \frac{Acceleration_2 - Acceleration_1}{t} \tag{5.6}$$

The procedure involves calculating $Acceleration_2$ as the acceleration in the next iteration and $Acceleration_1$ as the acceleration of the present point for each movement tested, in time frame t_f seconds.

Utilizing the Range and Doppler values to distinguish different movements, an additional feature has been created, which measures the mean distance of all points to the centroid of a Range-Doppler graph.

In Figure 5.20, the Range-Doppler cloud is depicted, with the red point representing the centroid. The analysis suggests that the distance to the centroid is expected to be shorter when a person is stationary.

While using the measurement of the furthest point as a reference can be a practical approach, it does have limitations, particularly when tracking a person movements across varying distances within a room. To mitigate potential errors, a more robust strategy involves calculating the mean distance of all points to the centroid. This method takes into account the distribution of points across the room, providing a more accurate representation of the overall tracking scenario.



(a) Range Doppler Graph - Target Stopped.

(b) Range Doppler Graph - Target Walking in the room.

Figure 5.20: Range-Doppler Graph.

By utilizing the SNR values in dB from Equation (5.7), it becomes possible to assess the energy dispersion, which serves as a potential feature for distinguishing between labels. To estimate this energy dispersion, Equation (5.8) is employed.

$$SNR_{dB} = 10 \times \log_{10}(SNR) \tag{5.7}$$

$$s = \frac{1}{n}\sum_{i=1}^{n}(xi - \overline{x})^2 \tag{5.8}$$

Where x is the value of SNR and $\overline{x}$ is the mean value.

Additional features can be derived from formula (5.8), such as the variance of positions on the x and y axes, as well as the variance of the norm of every x and y position. This implies that one feasible feature to determine is the variance of positions along the x-axis, which can be calculated using Equation 5.8. In this equation, x will represent the value of the position along the x-axis, and $\overline{x}$ denotes the mean value of the positions along the x-axis. The same procedure can be repeated for positions along the y-axis, as well as for the Euclidean norm of both the x and y positions.

Kurtosis, Skewness, and Standard Deviation are essential statistical measurements used in data analysis.

Kurtosis quantifies the extent to which the tails of a probability distribution deviate from a normal distribution, indicating the "fatness" or presence of outliers in the distribution. Higher kurtosis values suggest that the values are more concentrated around the mean, while lower values indicate more dispersion and a greater distance from the mean.

Skewness measures the degree of asymmetry in a probability distribution, indicating whether the distribution is symmetric or shifted towards a particular direction. Higher skewness values imply that the tail is extended to the right, lower values indicate extension to the left, and if the values are equally distributed, the tails are balanced, and the distribution is centered.

Standard Deviation shows how far the data points are from the mean value. The greater the dispersion of values from the mean, the larger the standard deviation, and vice versa.

These statistical metrics can be applied to describe various aspects of movements. By computing these measurements for velocity, acceleration, Doppler, and positions on the x and y axes, a wide range of features can be created to capture different aspects of the movements.

Table (5.7) provides an overview of the mean feature values obtained through the different labels.

Table 5.7: Feature Values where STD is the standard deviation.

Results	**Stopped**	**Workout**	**Moving**
Mean Doppler	0.0009	0.0319	0.0027
Mean Velocity	0.4027	0.9578	0.7940
Median Velocity	0.3781	0.9716	0.8160
Max Velocity	1.0511	2.0622	1.5231
Mean Acceleration	8.0531	19.1553	15.8800
Max Acceleration	21.0225	41.2446	30.4628
Kurtosis Velocity	2.9334	2.9450	3.2890
Skewness Velocity	0.4779	0.1193	-0.3106
STD Velocity	0.2063	0.4222	0.2663
Kurtosis Doppler	55.3599	8.7398	26.3639
Skewness Doppler	0.7879	0.6741	0.2348
STD Doppler	0.0228	0.2836	0.2356
Mean distance from all points to centroid (Range – Doppler)	0.1575	0.5547	0.8655
Jerk	161.0613	383.1051	317.6009
Energy dispersion	10.8847	19.0248	21.0778
Kurtosis Position x	5.6034	4.9048	3.2581
Skewness Position x	0.4408	-0.0862	0.3718
STD Position x	0.0974	0.5591	0.5944
Kurtosis Position y	5.2993	4.3767	2.6602
Skewness Position y	-0.2736	-0.0430	-0.1254
STD Position y	0.1057	0.6133	1.0734
Total Displacement	19.3384	38.5425	65.9623
Displacement Positions x	16.3514	25.5888	33.7321
Displacement Positions y	8.6019	23.6458	49.4104
Variance position x	0.0194	0.5726	0.4798
Variance position y	0.0517	0.6693	1.4329
Variance position norm(x,y)	0.0619	0.9640	1.5458

5.5 Feature Statistical Analysis

At the moment, 27 are the number of features used to apply in ML algorithms, however, too many features can cause redundancy passing the same information twice or more.

Redundancy should be avoided in classification data inputs, since it may bias the model. To reduce the number of features, feature importance in the classification problem may be one of the options of select the most representative ones. For this purpose, a statistical analysis will be performed. Furthermore, the correlation between features should be validated. In the presence of features correlated, only one of them should be selected.

To decide which statistical test should be applied to the data, the first step is to know if features present a normal distribution and for that, Kolmogorov-Smirnov (kstest), Anderson-Darling (adtest), and Lilliefors (Lillietest) tests are performed. When all the three tests do not reject the null hypothesis, it can be assumed that features have a normal distribution and the one-way ANOVA can be applied.

When the null hypothesis is rejected, the normal distribution of the data is not assumed, and the Kruskal-Wallis test should be applied. In case this happens, the next step is to perform the Kruskall-Wallis test for each feature.

The process of feature selection is summarized in Figure 5.21.

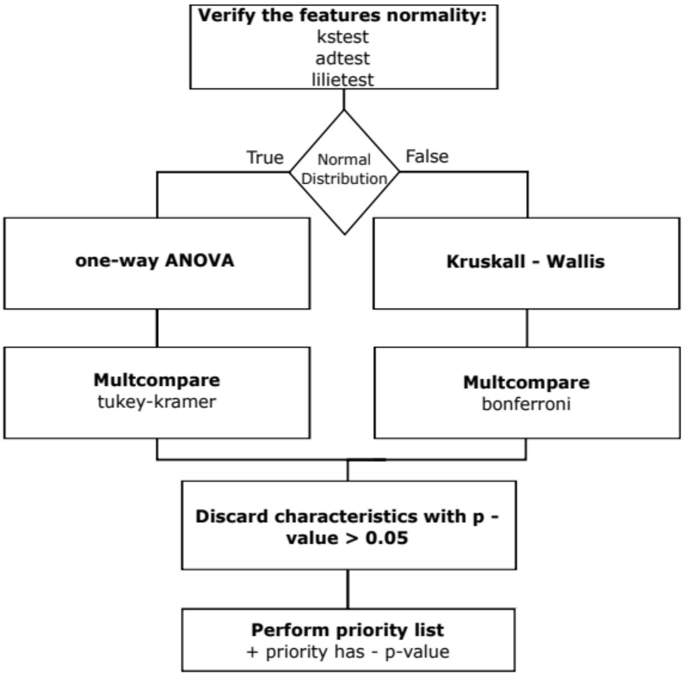


Figure 5.21: Feature Statistical Analysis (Adapted from: [62]).

In all the 27 tested features, the null hypothesis was rejected, therefore the Kruskal-Wallis test was applied to all of them. That said, the first approach allow to understand if there is at least one pair of conditions that may be distinguished by the evaluated feature. At a second level the multcompare test, with the Bonferroni correction as posthoc test was used to infer which pair of conditions are being differentiated.

When the p-value of the test is higher than the significance level of 0.05, then that feature is discarded. With the Kruskall-Wallis and Multcompare test performed, the feature "Skewness Position y" is the only one with a p-value superior to 0.05, presenting a value of 0.25402. For that reason, that feature is discarded.

For instance, the feature "STD Doppler" exhibits a p-value of 4.94×10^{-32}, and as shown in Figure 5.22, this feature can effectively differentiate between the three labels.

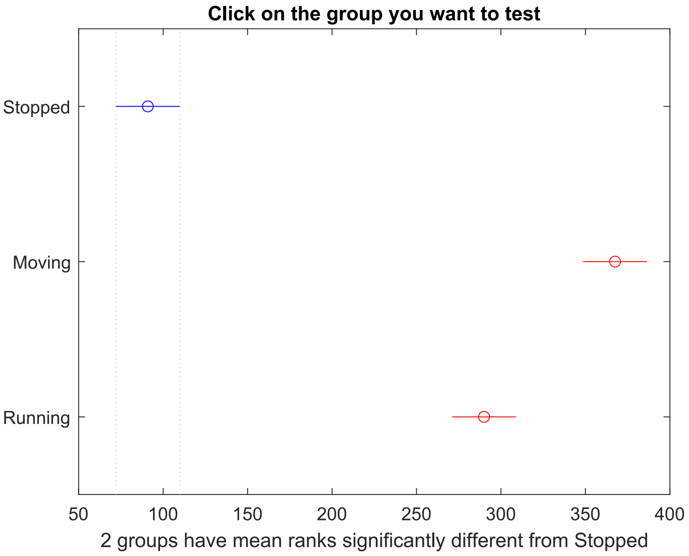


Figure 5.22: Multcompare Feature - STD Doppler.

Conversely, the feature "Displacement x" demonstrates the ability to differentiate only between label 3 (Moving) and labels 1 (Stopped) and 2 (Workout) because this feature carries the same meaning for labels 1 and 2, as depicted in Figure (5.23). In this instance, the p-value is also 4.94×10^{-32}.

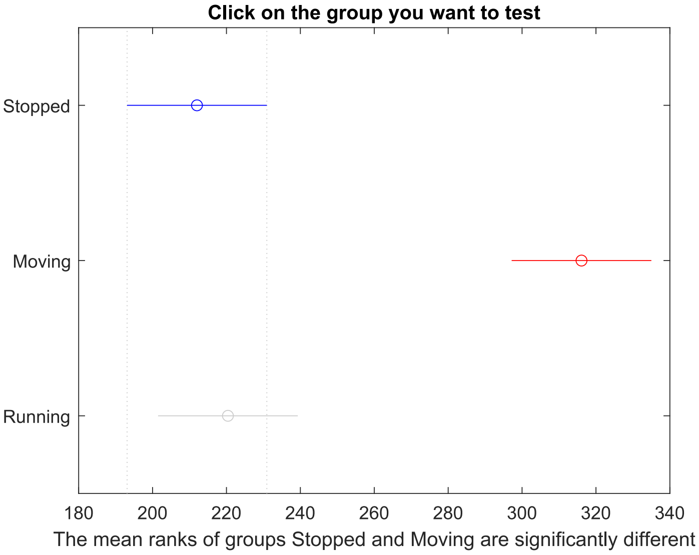


Figure 5.23: Multcompare Feature - Displacement x.

This feature remains valuable as it can differentiate at least one label from the others. However, the challenge arises with the feature "Skewness Position y", and the rationale for this is evident in Figure 5.24. This particular feature is unable to distinguish any label, as it holds the same significance for all labels.

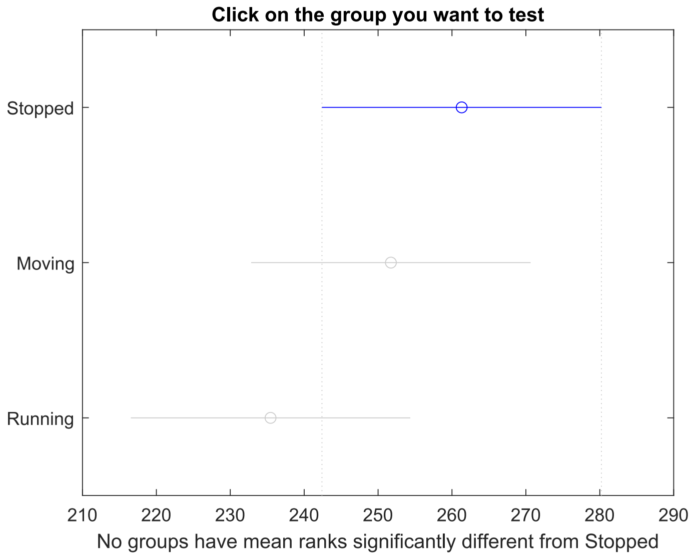


Figure 5.24: Multcompare Feature - Skewness Position y.

Using the p-values for each feature, it is feasible to establish a priority list with the most valuable features having the lowest p-values. This priority list is presented in Table (5.8).

Table 5.8: First Priority Features List.

Priority	Feature	P-value
1^{st}	0	Mean Velocity
2^{nd}	0	Mean Acceleration
3^{rd}	0	Skewness Velocity
4^{th}	0	Mean distance from all points to centroid (Range – Doppler)
5^{th}	0	Jerk
6^{th}	0	Median Velocity
7^{th}	0	Kurtosis Doppler
8^{th}	0	STD Doppler
9^{th}	0	Kurtosis Posy
10^{th}	0	STD PosY
11^{th}	0	Displacement Posy
12^{th}	0	Variance Posy
13^{th}	0	Norm Posxy
14^{th}	1.4013×10^{-45}	STD Posx
15^{th}	1.4013×10^{-45}	Variance Posx
16^{th}	5.6052×10^{-45}	Total Displacement
17^{th}	6.5861×10^{-44}	Energy Dispersion
18^{th}	4.9407×10^{-32}	Kurtosis Posx
19^{th}	2.1408×10^{-25}	Mean Doppler
20^{th}	9.8925×10^{-20}	Kurtosis Velocity
21^{st}	3.2783×10^{-15}	Max Velocity
22^{nd}	3.2783×10^{-15}	Max Acceleration
23^{rd}	2.3068×10^{-12}	Displacement Posx
24^{th}	8.7656×10^{-07}	STD Velocity
25^{th}	2.4669×10^{-03}	Skewness Posx
26^{th}	8.8288×10^{-03}	Skewness Doppler

In Table (5.8), some p-values are displayed as zero, but in reality, the values are extremely small to the point where MATLAB cannot accurately represent them.

The next step is to identify which features are correlated. By using the MATLAB command corrcoef(), it is possible to evaluate the correlation values between features. Features with a correlation value exceeding 0.7 and holding a lower position in the priority list (5.8) are removed.

The correlation matrix is presented in Figure 5.25, leading to the creation of a second and final priority list represented in Table (5.9). This process helps in selecting the most relevant and uncorrelated features for further analysis or modeling.

Table 5.9: Second Priority Features List.

Priority	**P-value**	**Feature**
1^{st}	0	Mean Velocity
2^{nd}	0	Skewness Velocity
3^{rd}	0	Mean Distance Centroid
4^{th}	0	Kurtosis Doppler
5^{th}	0	Kurtosis Posy
6^{th}	0	Displacement Posy
7^{th}	4.9407×10^{-32}	Kurtosis Posx
8^{th}	2.1408×10^{-25}	Mean Doppler
9^{th}	9.8925×10^{-20}	Kurtosis Velocity
10^{th}	2.4669×10^{-03}	Skewness Posx
11^{th}	8.8288×10^{-03}	Skewness Doppler

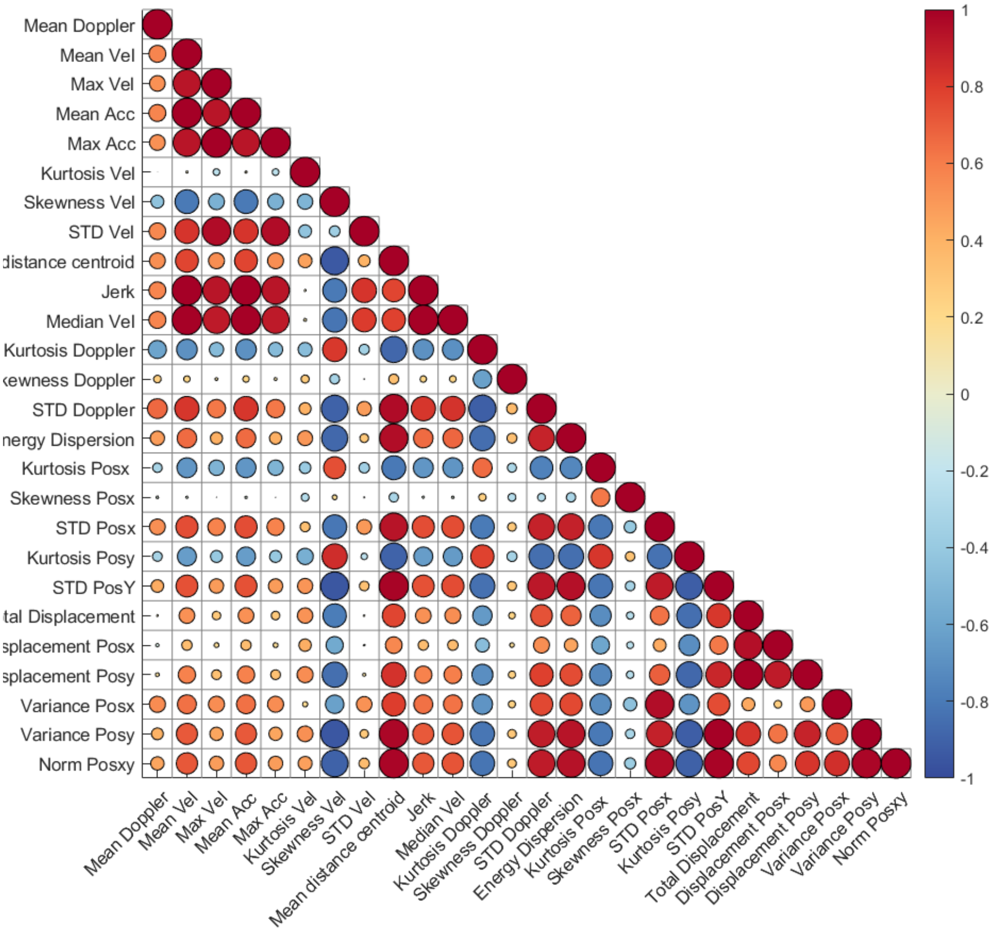


Figure 5.25: Correlation Matrix - First Priority List.

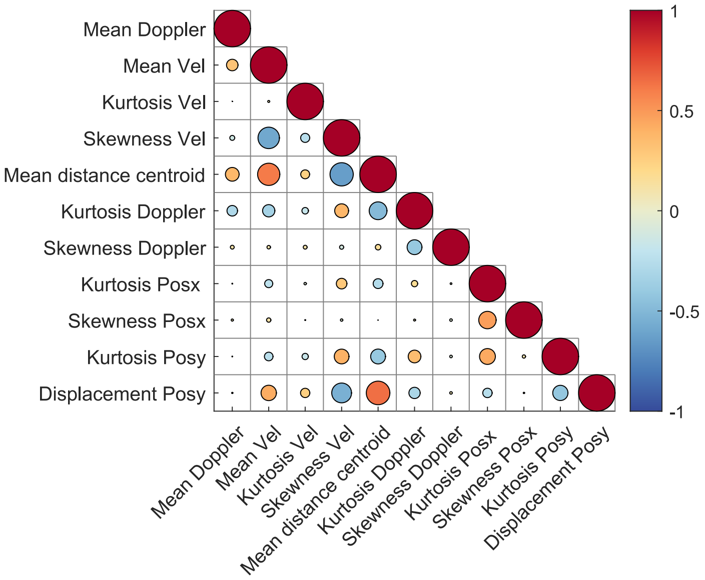


Figure 5.26: Correlation Matrix - Second Priority List.

5.6.1 Accuracy of Machine Learning Algorithms without Feature Selection

The dataset will be partitioned into distinct training and testing subsamples. The training subsamples will be employed to train the ML model, while the test samples will be used to validate the model.

To ensure the model is well-trained, it is essential to have a diverse set of training data. Simultaneously, to assess whether the model correctly classifies movements, some data should be reserved for testing to evaluate the accuracy of the algorithms.

The dataset for the division process is summarized in Table (5.10)

Table 5.10: Data Division for training and testing.

Dataset	**Division Percentage**	**Number of observations**
Training	70%	210
Testing	30%	90

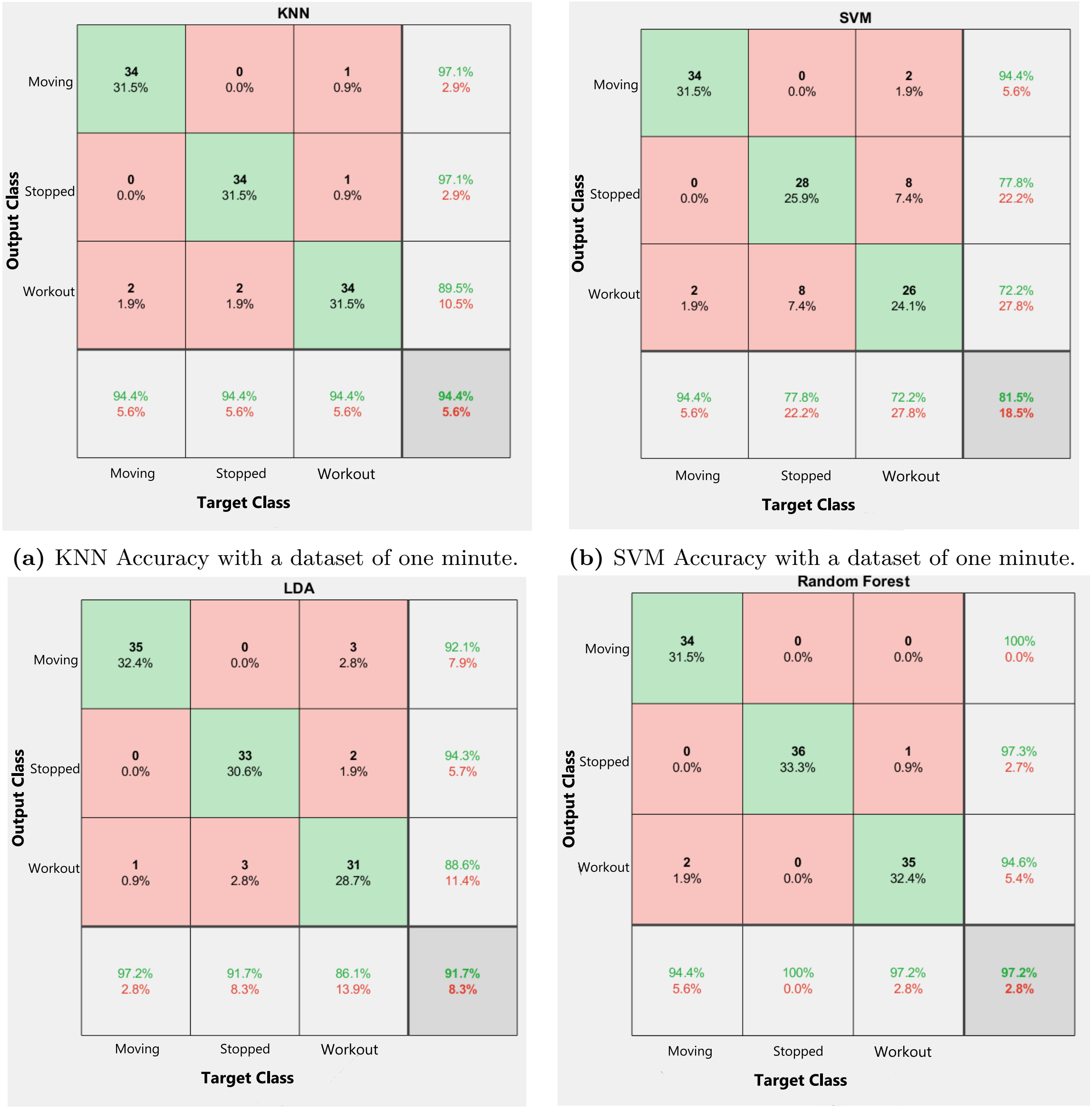
In the KNN model, a parameter known as *k* determines the number of neighbors considered when making a prediction. In this case, *k* is set to 5, which means that the model will take into account the five closest neighbors when making predictions.

Conversely, the SVM model, uses binary learners and follows a one-versus-one coding design. For each binary learner, one class is positive and the rest are negative. When classifying a new sample in a multiclass scenario, the one-versus-one method involves each binary SVM classifier providing a vote, and the class is assigned based on the majority of votes. This approach simplifies the classification process and allows SVM to handle multiclass problems efficiently, with more information detailed in [63].

In the LDA model, a linear discrimination is applied. This means that LDA seeks to identify an approximate linear decision boundary that effectively separates different classes within the data, as detailed in [64]. Although RF model, training is performed for classification tasks using a total of 100 trees to train the model.

The accuracy of each ML algorithm is assessed using the complete dataset and the chosen data division, with default hyperparameters selected from MATLAB. Confusion matrices are used to evaluate the performance of each algorithm by assessing correct and incorrect identifications. Various performance metrics can be calculated using the confusion matrix, and for the purpose of this work, accuracy has been chosen as a key metric for comparing the results.

The confusion matrices for each algorithm can be found in Figures 5.27, which provide a visual representation of the model performance in terms of true positives, true negatives, false positives, and false negatives.



(a) KNN Accuracy with a dataset of one minute.

(b) SVM Accuracy with a dataset of one minute.

(c) LDA Accuracy with a dataset of one minute.

(d) RF Accuracy with a dataset of one minute.

Figure 5.27: Confusion Matrices for each ML Algorithm with a dataset of one minute.

Table (5.11) provides a summary of the results, with a dataset comprising 90 testing data points and 210 training data points, utilizing 27 features for classification.

Table 5.11: Accuracy values for each ML Algorithm with a dataset of one minute.

ML Algorithm	Accuracy
KNN	94.4%
SVM	81.5%
LDA	91.7%
Random Forest	97.2%

Based on the evaluation of the values obtained for each respective ML algorithm, it appears that the RF algorithm has achieved the highest accuracy among the tested algorithms. This suggests that, in the context of this work and with the specific dataset and problem at hand, the Random Forest algorithm may be the best choice for classifying movements and achieving the highest accuracy results. However, it is important to consider other factors.

5.6.2 Time Window - Accuracy

The values presented in Table (5.11) correspond to a dataset with movements lasting approximately one minute each. However, given the potential influence of movement duration on the identification process, it was crucial to investigate the impact of shorter durations, such as thirty-second windows, on the results.

Table (5.12) displays the values for each Machine Learning algorithm with the dataset containing shorter movements.

Table 5.12: Accuracy values for each ML Algorithm with a dataset of thirty seconds.

ML Algorithm	Accuracy	Number of Features	Testing dataset	Training dataset
KNN	93.5%	27	180	420
SVM	94.4%			
LDA	89.7%			
Random Forest	98.1%			

Once more, the RF algorithm stands out as the one with the highest accuracy.

By further dividing the dataset in terms of time, where movements with a duration of one minute are split into fifteen-second segments, it significantly increases the volume of data available for analyzing movements. The accuracy results obtained through this approach are presented in Table (5.13).

Table 5.13: Accuracy values for each ML Algorithm with a dataset of fifteen seconds.

ML Algorithm	Accuracy	Number of Features	Testing dataset	Training dataset
KNN	83.6%	27	360	840
SVM	79.4%			
LDA	86.0%			
Random Forest	88.8%			

Upon examining the tables, specifically Table (5.11), Table (5.12), and Table (5.13), the collective information is summarized in Table (5.14). The summary reveals that when movement durations are reduced to fifteen seconds, the accuracy of the classification tends to decrease compared to cases where the movement duration is either thirty seconds or one minute, using 27 features

Table 5.14: ML Algorithms Accuracy with Time Division.

Time Window	KNN Accuracy	SVM Accuracy	LDA Accuracy	RF Accuracy	Testing dataset	Training dataset
15 Seconds	83.6%	79.4%	86.0%	88.8%	360	840
30 Seconds	93.5%	94.4%	89.7%	98.1%	180	420
One Minute	94.4%	94.4%	91.7%	97.2%	90	210

The accuracy values achieved for movement durations of either thirty seconds or one minute are found to be similar. In this context, the critical factor for determining the optimal time window for movement duration is the quantity of data available for both training and testing ML models. Given this consideration, a thirty-second duration emerges as a reasonable and practical choice for movements for both training and testing purposes.

However, the accuracy values can change when the data selected for training and testing is altered. To account for this, variations of the chososen training and testing data are made fifteen times, and the mean accuracy is calculated from the obtained values, as represented in Table (5.15). This process helps to assess the robustness of the ML models in handling different datasets.

Table 5.15: ML algorithms Robustness using 27 features.

ML Algorithm	Mean Accuracy ± Standard Deviation	Number of Features	Number of Tests	Number of Trainers
KNN	93.17% ± 1.72	27	180	420
SVM	84.99% ± 7.29			
LDA	91.86% ± 3.26			
Random Forest	99.00% ± 0.79			

Table (5.15) offers valuable insights, highlighting that the RF algorithm achieves the highest accuracy with a minimal standard deviation. Additionally, when examining the variations in the data used for training and testing, the RF algorithm exhibits robustness, making it a reliable choice for this context.

5.6.3 Accuracy of Machine Learning Algorithms with Feature Selection

Feature Selection Analysis was conducted to evaluate the accuracy of ML algorithms. The results are presented in Table (5.16) consisting of 180 training data points, 420 testing data points and 11 features. When comparing these accuracy values to those in Table (5.15), the accuracy achieved with feature selection is nearly equivalent to that achieved with all 27 features.

Table 5.16: ML Algorithms Robustness using 11 features.

ML Algorithm	Mean Accuracy ± Standard Deviation	Number of Data Variations
KNN	90.24% ± 2.64	15
SVM	79.00% ± 11.40	
LDA	91.67% ± 2.25	
Random Forest	98.89% ± 0.82	

This indicates that utilizing feature selection can be advantageous, primarily due to the reduced number of features, 11 in this case. This approach not only provide comparable accuracy, but it also results in lower computation time, as demonstrated in Table (5.17).

The reduction in time for model creation and label prediction underscores the efficiency and practicality of feature selection in the context of developing and using ML models. This approach can significantly improve the computational efficiency of the ML process.

Table 5.17: Necessary time (Seconds) of the ML algorithm.

ML Algorithm	With Feature Selection	Without Feature Selection
KNN	0.064742	0.081173
SVM	2.771071	5.843736
LDA	0.045244	0.049770
Random Forest	9.785501	13.496827

In addition to computational efficiency, memory usage is another crucial factor to consider when working with a dataset with 27 features, as illustrated in Table (5.18). The use of feature selection not only reduces computation time but also helps conserve memory resources, which can be a valuable advantage in scenarios with limited memory capacity. This demonstrates the multifaceted benefits of employing feature selection in ML applications.

Table 5.18: Necessary memory (Bytes) to create the ML algorithm and respective prediction.

ML Algorithm	With Feature Selection	Without Feature Selection
KNN	35621	69461
SVM	54590	96366
LDA	41480	82424
Random Forest	1342871	1425727

The SVM algorithm exhibits the lowest accuracy compared to the other algorithms and demands more time and memory resources than KNN and LDA. Conversely, LDA is the fastest algorithm, while KNN consumes the least amount of memory.

5.6.4 ML RF algorithm optimization

For instance, the RF algorithm requires more time and memory than the others, especially when considering the number of Decision Trees, which is set at 100. A study was conducted to determine the minimum number of decision trees that could be used while maintaining the same accuracy to reduce both time and memory usage.

Figure 5.28 visually represents the relationship between accuracy and the number of decision trees. As expected, accuracy tends to increase as the number of decision trees grows. However, it is worth noting that accuracy stabilizes at 15 trees, where it remains relatively constant, even as the number of decision trees continues to increase. This 15-tree configuration represents the optimal number of decision trees to achieve a reasonable level of accuracy for the given dataset and model.

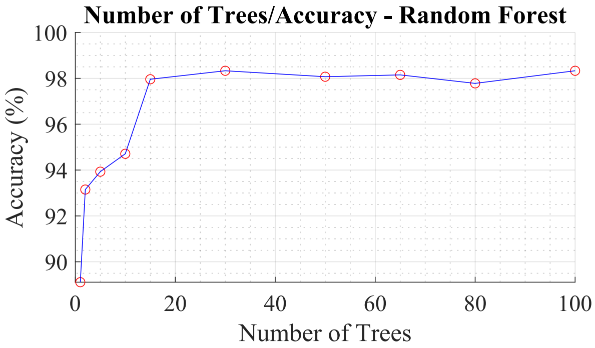


Figure 5.28: Number of Trees/Accuracy - RF.

The RF performance values in (5.17) and (5.18) have been updated to Table (5.19), reflecting the improvements achieved by reducing both time and memory RF algorithm. These enhancements were achieved by using only 15 decision trees in the RF algorithm and implementing feature selection.

Table 5.19: Performance impact by using 15 Trees in RF with feature selection.

Performance	100 Trees	15 Trees
Time (Seconds)	9.785501	0.819489
Memory (Bytes)	1342871	271187
Mean Accuracy ± Standard Deviation	98.89% ± 0.82	97.96% ± 1.21

5.6.5 Performance ML summary

Table (5.20) provides a comprehensive overview of all the values presented in the spider chart in Figure (5.29). This table consolidates the performance metrics for various machine learning algorithms, offering a detailed comparison of their accuracy, execution time, and memory consumption.

Table 5.20: Performance Values of ML Algorithms.

Algorithm	Creation Time (s)	Prediction Time (s)	Memory (Kbytes)	Accuracy (±STD)
KNN	0.064742	0.010284	35.621	89.06% (±2.58)
SVM	2.771071	0.004020	54.590	83.28% (±9.15)
LDA	0.045244	0.001457	41.480	89.08% (±3.63)
Random Forest	0.945343	0.002114	272.891	97.96% (±1.21)

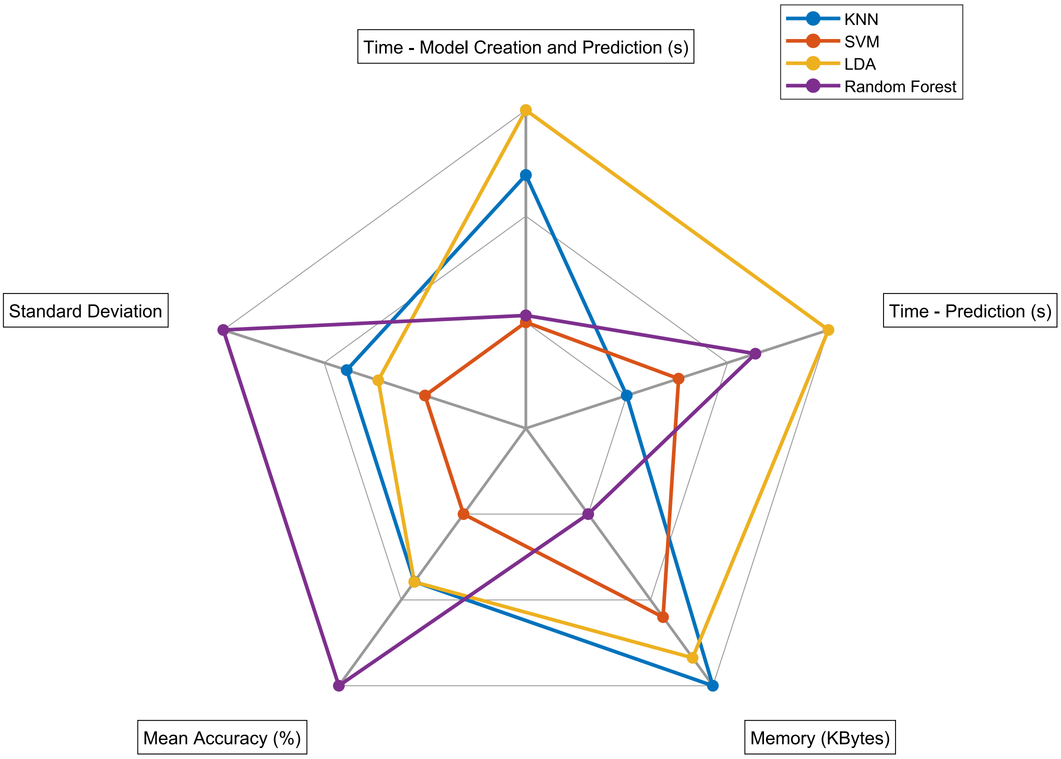


Figure 5.29: Algorithm performance - Spider Chart.

By analyzing the values in Table (5.20) and referring to the spider chart presented in Figure (5.29), one can make an informed choice regarding the most suitable ML algorithm for a specific purpose.

In essence, if an individual is facing a decision between LDA and RF, the choice can be based on their specific requirements. If the primary concern is achieving a highly accurate prediction value, and the time required for that prediction is not a critical factor, then RF is the preferred option. However, if time is a crucial parameter, and it is acceptable to have a slightly lower accuracy value, then LDA is the algorithm of choice. This is because LDA is the algorithm that requires less time for model creation and prediction.

5.7 Real-Time Detection and Movement Classification

A Smart-Home application of this work can be summarized in Figure 5.30. This flowchart illustrates the system workflow based on the classified movement of the individual that can be applied as a product or solution that Bosch can implement within their products. Depending on the detected movement, the system adjusts the room temperature, either increasing or decreasing it, to enhance the person comfort.

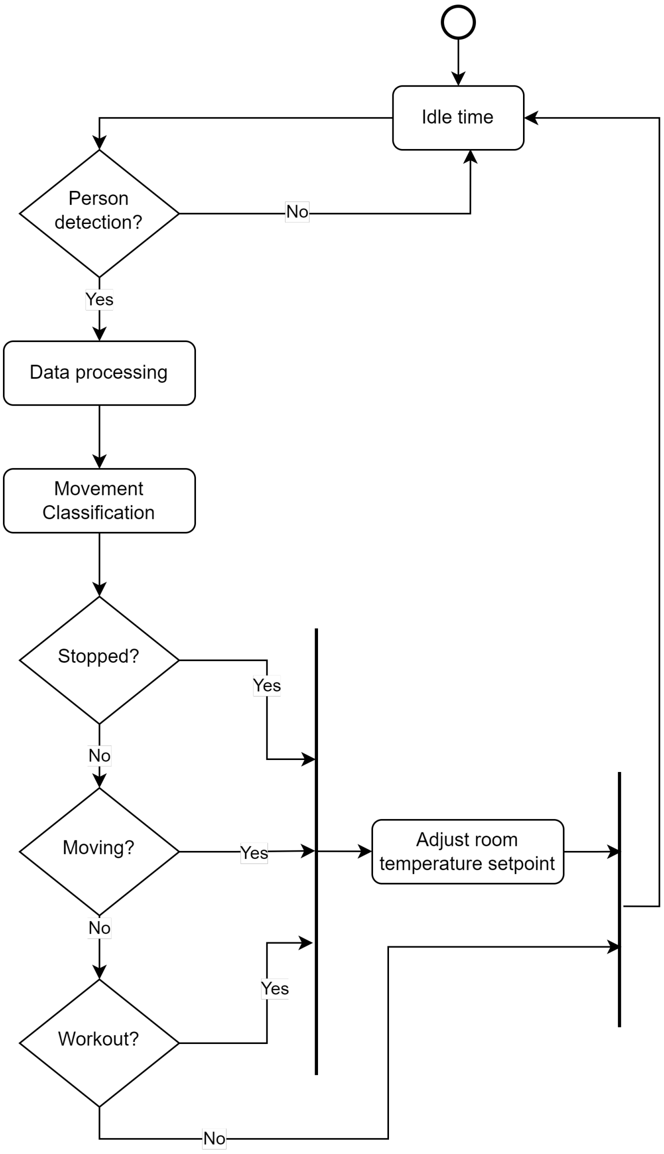


Figure 5.30: Example of a flow chart of the present work for Smart Homes applications.

This work addresses this need by proposing the utilization of Radar technology, relying on signal processing and data analysis. Such an approach facilitates the development of a system proficient at accurately assessing movement without compromising privacy.

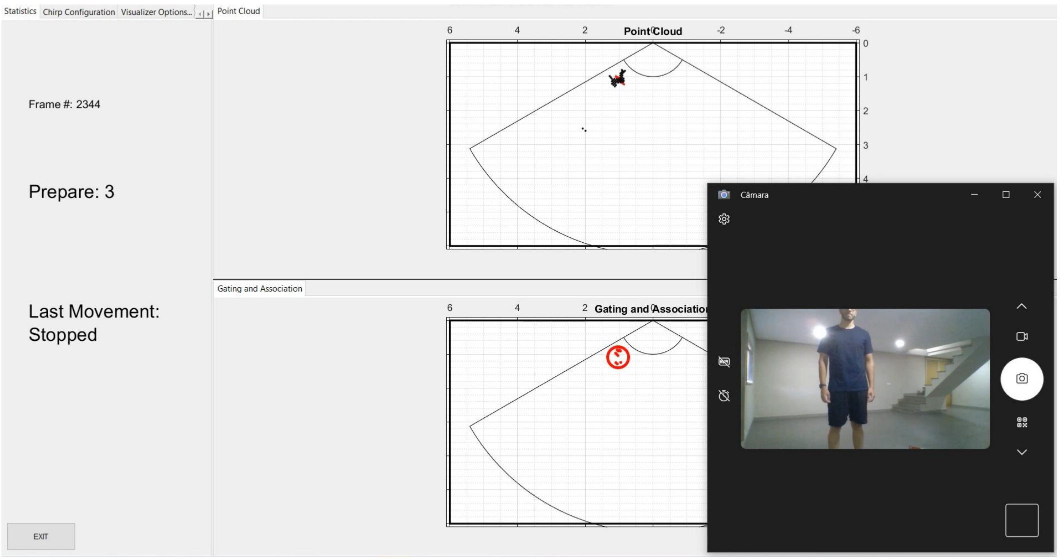
A real-time tool for visualization and movement classification was developed using the GUI of the People Counting Software as a foundation.

RF were employed to classify a person movement within a specified time window. After testing the software, it was determined that using a very short time window (e.g., one or two seconds) was insufficient for accurate classification. Conversely, a time window that was too long (e.g., thirty seconds) increased the likelihood of multiple movements occurring within the same window. Therefore, a fifteen-second time window was selected as the optimal compromise.

The GUI of the People Counting Software was modified to aid movement preparation and communication with the individual. In this setup, the person had three seconds to prepare for a movement within the fifteen-second time window, and the movement was classified at the end of this window. The GUI displayed the time windows for preparation and movement, making it easier for the subject.

The real-time tests with the GUI demonstrated high accuracy in detection and movement classification, providing proof of concept.

Figure 5.31 illustrates the GUI real-time test example.



Frame #: 2344

Prepare: 3

Last Movement:
Stopped

Figure 5.31: GUI for the Real-Time proof of concept.

CHAPTER 6

Conclusion and Future Work

This chapter outlines the primary conclusions drawn from the research conducted in this study. Additionally, it highlights potential areas for future exploration that can lead to improved outcomes in the current work.

6.1 Conclusions

The primary objectives of this dissertation, stated in section (1.2), were achieved.

Placing the radar on the ceiling or floor can limit its ability to detect entirely the movements made by individuals in the room. However, when positioned at middle height, certain challenges may arise, particularly in the center of the room, due to the line of sight of the radar, the closest corners to the radar may not be fully trackable if a person passes through those areas. This limitation in tracking is a trade-off that should be considered when determining the ideal radar placement within a room. The study determined that the most suitable placement for the radar in an indoor environment is at a middle height position in one of the room corners. This placement offers a practical compromise, changing the signal parameters to increase the maximum range value, for covering the entire room area.

The study demonstrated that using just one FMCW Radar with an extensive dataset containing data from 10 different individuals performing 30 distinct movements, it is possible to achieve a high level of accuracy, 97.96% using RF ML algorithm, in movement classification.

Four different ML algorithms, namely KNN, SVM, LDA, and RF, were thoroughly tested, and their performance was comprehensively analyzed and evaluated.

The findings indicated that the KNN algorithm is the most memory-efficient, LDA is the fastest, and RF presented the best results in terms of standard deviation and accuracy, achieving an accuracy of approximately 97.96%.

However, it is worth noting that among the four machine learning algorithms examined, SVM exhibited the least favorable performance. This was primarily attributed to its comparatively higher computational demands in terms of both time and memory usage. Additionally, SVM displayed less desirable results in terms of standard deviation and accuracy when compared to the other algorithms.

With this comprehensive performance evaluation of each algorithm, it becomes possible to choose the most appropriate one for implementation in real-time systems. Depending on specific requirements, RF may be the preferred choice for achieving maximum accuracy, LDA for the fastest performance, and KNN for memory-efficient applications.

6.2 Future Work

During the implementation , several challenges and issues emerged that had an impact on the overall success. During the data collection phase, an observation was made that the radar has a blind spot in the frontal direction. In other words, when targets pass directly in front of the radar at 0 degrees azimuth, the radar is unable to collect any data, and the detection of individuals in this specific situation is not possible. At the same time, the radar antenna configuration does not permit elevation angle measurements. The elevation angle presented by the AWR1642BOOST is very small, resulting in data being collected in a two-dimensional plane without elevation information. For future work, addressing these limitations is a priority. The first step would involve using a radar system that allows for Three Dimensional (3D) visualization, for example *IIWR6843ISK-ODS*, and that is not blind at 0 degrees of azimuth. This would enable a more comprehensive understanding of the environment. Additional specifications and details about the *IIWR6843ISK-ODS* radar can be found in the reference [65].

Once a 3D radar system is in place, the data collection process can be enhanced. Instead of the three labels currently studied (as represented in Figure 5.8), the system could be expanded to accommodate more labels that are relevant for various Smart Home applications.

For instance, with 3D data collection, it would be possible to distinguish between a person standing up and a person lying down. This distinction can be crucial, especially for healthcare applications, where it is essential to detect if someone has fallen.

In settings like a gym, it would be important to differentiate between different types of workouts. For example, if multiple people are running on treadmills, the room ventilation requirements might be different compared to an area where people are lifting weights. Each sub-label represented in Figure 5.8 could potentially become a global label.

Additionally, future work could involve the detection and classification of movements with multiple people in the same cluttered space. The current study focused on movement detection and classification with only one person in the room and did not address scenarios involving multiple individuals. Implementing and refining this concept for crowded environments would be a significant advancement.

Besides, implementing a fusion of multiple radars in future work is a promising approach to enhance the results and address blind spots in the monitoring process. Different radars can offer diverse perspectives and views of a person movements, leading to more accurate classification and tracking. By combining the data from these multiple radars, it becomes possible to obtain a more comprehensive and detailed understanding of the subject actions and behaviors.

Finally, a potential avenue for future work involves the development of a radar system that is not only cost-effective but also compact in size. This system could be designed for integration into a product or solution that Bosch can implement within the company.

Bibliography

[1] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, and C. Loge, “The smart home concept : Our immediate future,” in *2006 1ST IEEE International Conference on E-Learning in Industrial Electronics*, 2006, pp. 23–28. DOI: `10.1109/ICELIE.2006.347206`.

[2] L. C. De Silva, C. Morikawa, and I. M. Petra, “State of the art of smart homes,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 7, pp. 1313–1321, 2012, Advanced issues in Artificial Intelligence and Pattern Recognition for Intelligent Surveillance System in Smart Home Environment, ISSN: 0952-1976. DOI: `https://doi.org/10.1016/j.engappai.2012.05.002`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S095219761200098X`.

[3] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, “A review of smart homes—past, present, and future,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012. DOI: `10.1109/TSMCC.2012.2189204`.

[4] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, “Monitoring activities of daily living in smart homes: Understanding human behavior,” *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 81–94, 2016. DOI: `10.1109/MSP.2015.2503881`.

[5] L. Chen, C. D. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, 2012. DOI: `10.1109/TKDE.2011.51`.

[6] A. Fleury, M. Vacher, and N. Noury, “Svm-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 274–283, 2010. DOI: `10.1109/TITB.2009.2037317`.

[7] D. Ding, R. A. Cooper, P. F. Pasquina, and L. Fici-Pasquina, “Sensor technology for smart homes,” *Maturitas*, vol. 69, pp. 131–136, 2 Jun. 2011, ISSN: 03785122. DOI: `10.1016/j.maturitas.2011.03.016`.

[8] J. Toomay and P. J. Hannen, *Radar Principles for the Non-specialist*, 3rd. SciTech Pub, 2001, ISBN: 1891121286.

[9] MathWorks, *Understanding radar principles: Pulse-doppler radar - matlab simulink*, [Accessed in July-2023]. [Online]. Available: `https://www.mathworks.com/videos/radar-basics-part-4-pulse-doppler-radar-1655124097873.html`.

[10] A. Figueroa, B. Al-Qudsi, N. Joram, and F. Ellinger, “Comparison of two-way ranging with fmcw and uwb radar systems,” in *2016 13th Workshop on Positioning, Navigation and Communications (WPNC)*, 2016, pp. 1–6. DOI: `10.1109/WPNC.2016.7822856`.

[11] N. Levanon, “Radar,” in *Encyclopedia of Physical Science and Technology (Third Edition)*, R. A. Meyers, Ed., Third Edition, [Accessed in June-2023], New York: Academic Press, 2003, pp. 497–510, ISBN: 978-0-12-227410-7. DOI: `https://doi.org/10.1016/B0-12-227410-5/00973-X`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B012227410500973X`.

[12] J. Geisheimer, W. Marshall, and E. Greneker, “A continuous-wave (cw) radar for gait analysis,” in *Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat.No.01CH37256)*, vol. 1, 2001, 834–838 vol.1. DOI: `10.1109/ACSSC.2001.987041`.

[13] X. Li, Y. He, and X. Jing, "A survey of deep learning-based human activity recognition in radar," *Remote Sensing*, vol. 11, no. 9, 2019, ISSN: 2072-4292. DOI: 10.3390/rs11091068. [Online]. Available: https://www.mdpi.com/2072-4292/11/9/1068.

[14] I. Komarov and S. Smolskiy. 2003.

[15] M. Yu, N. Kim, Y. Jung, and S. Lee, "A frame detection method for real-time hand gesture recognition systems using cw-radar," *Sensors*, vol. 20, no. 8, 2020, ISSN: 1424-8220. DOI: 10.3390/s20082321. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2321.

[16] M. J. Al-Dujaili and A. M. Al-Awadi, "Chirplet signal design by fpga," *International Journal of Electrical and Computer Engineering*, vol. 11, pp. 2120–2127, 3 Jun. 2021, ISSN: 20888708. DOI: 10.11591/ijece.v11i3.pp2120-2127.

[17] S. Rao and T. Instruments, "Introduction to mmwave sensing: Fmcw radars," [Accessed in July-2023]. [Online]. Available: https://www.ti.com/content/dam/videos/external-videos/2/3816841626001/5415528961001.mp4/subassets/mmwaveSensing-FMCW-offlineviewing_0.pdf.

[18] S. Li and R. Hishiyama, "A field people counting test using millimeter wave radar in the restaurant," 2021.

[19] A. Santra, R. V. Ulaganathan, and T. Finke, "Short-range millimetric-wave radar system for occupancy sensing application," *IEEE Sensors Letters*, vol. 2, no. 3, pp. 1–4, Jul. 2018. DOI: 10.1109/LSENS.2018.2852263.

[20] J. Weiß, R. Pérez, and E. Biebl, "Improved people counting algorithm for indoor environments using 60 ghz fmcw radar," pp. 1–6, Sep. 2020. DOI: 10.1109/RadarConf2043947.2020.9266607.

[21] X. Yang, W. Yin, and L. Zhang, "People counting based on cnn using ir-uwb radar," in *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, IEEE, Oct. 2017, pp. 1–5. DOI: 10.1109/ICCChina.2017.8330453.

[22] Z. Yang, G. Qi, and R. Bao, "Indoor regional people counting method based on bi-motion-model-framework using uwb radar," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Mar. 2022. DOI: 10.1109/LGRS.2022.3159590.

[23] J.-H. Choi, J.-E. Kim, and K.-T. Kim, "People counting using ir-uwb radar sensor in a wide area," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5806–5821, Apr. 2021. DOI: 10.1109/JIOT.2020.3032710.

[24] Y. Zou, C. Ding, H. Hong, C. Li, and X. Zhu, "Indoor non-rhythmic human motion classification using a frequency-modulated continuous-wave radar," *The Journal of Engineering*, vol. 2019, no. 20, pp. 6928–6930, 2019. DOI: https://doi.org/10.1049/joe.2019.0560. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/joe.2019.0560.

[25] C. Ding, Y. Zou, L. Sun, H. Hong, X. Zhu, and C. Li, "Fall detection with multi-domain features by a portable fmcw radar," in *2019 IEEE MTT-S International Wireless Symposium (IWS)*, IEEE, May 2019, pp. 1–3. DOI: 10.1109/IEEE-IWS.2019.8804036.

[26] M. Hämäläinen, L. Mucchi, S. Caputo, *et al.*, "Ultra-wideband radar-based indoor activity monitoring for elderly care," *Sensors*, vol. 21, no. 9, 2021, ISSN: 1424-8220. DOI: 10.3390/s21093158. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3158.

[27] Y. Yang, C. Hou, Y. Lang, G. Yue, Y. He, and W. Xiang, "Person identification using micro-doppler signatures of human motions and uwb radar," *IEEE Microwave and Wireless Components Letters*, vol. 29, no. 5, pp. 366–368, Apr. 2019. DOI: 10.1109/LMWC.2019.2907547.

[28] L. Senigagliesi, G. Ciattaglia, A. De Santis, and E. Gambi, "People walking classification using automotive radar," *Electronics*, vol. 9, no. 4, 2020, ISSN: 2079-9292. DOI: 10.3390/electronics9040588. [Online]. Available: https://www.mdpi.com/2079-9292/9/4/588.

[29] S.-w. Kang, M.-h. Jang, and S. Lee, "Identification of human motion using radar sensor in an indoor environment," *Sensors*, vol. 21, no. 7, 2021, ISSN: 1424-8220. DOI: 10.3390/s21072305. [Online]. Available: https://www.mdpi.com/1424-8220/21/7/2305.

[30] C. Y. Aydogdu, S. Hazra, A. Santra, and R. Weigel, “Multi-modal cross learning for improved people counting using short-range fmcw radar,” in *2020 IEEE International Radar Conference (RADAR)*, 2020, pp. 250–255. DOI: 10.1109/RADAR42522.2020.9114871.

[31] D. Wang, S. Yoo, and S. H. Cho, “Experimental comparison of ir-uwb radar and fmcw radar for vital signs,” *Sensors*, vol. 20, no. 22, 2020, ISSN: 1424-8220. DOI: 10.3390/s20226695. [Online]. Available: https://www.mdpi.com/1424-8220/20/22/6695.

[32] O. Boric-Lubecke, V. M. Lubecke, A. D. Droitcour, B.-K. Park, and A. Singh, *DOPPLER RADAR PHYSIOLOGICAL SENSING*, 1st. Wiley-IEEE Press, Dec. 2015, ISBN: 1118024028.

[33] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*, 2nd. Chapman and Hall/CRC, 2005, ISBN: 9780429125416. DOI: 10.1201/9781420057072.

[34] MathWorks, *Understanding radar principles: Understanding the radar equation - matlab simulink*, [Accessed in July-2023]. [Online]. Available: https://www.mathworks.com/videos/radar-basics-part-3-understanding-the-radar-equation-1655123512443.html.

[35] H. J. Li and Y. W. Kiang, “Radar and inverse scattering,” *The Electrical Engineering Handbook*, pp. 671–690, Jan. 2005. DOI: 10.1016/B978-012170960-0/50047-5.

[36] B. R. Mahafza, *Introduction to Radar Analysis*, 2nd. Chapman and Hall/CRC, 2017.

[37] C. I. Radar, S. Rao, and T. Instruments, “The fundamentals of millimeter wave radar sensors,” 2020, [Accessed in July-2023]. [Online]. Available: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1698256734757&ref_url=https%5C%253A%5C%252F%5C%252Fwww.google.com%5C%252F.

[38] T. Instruments, *Mmwave radar sensors | ti.com*, [Accessed in July-2023]. [Online]. Available: https://www.ti.com/video/series/mmwave-training-series.html#tab-1.

[39] I. Kononenko, “Machine learning for medical diagnosis: History, state of the art and perspective,” *Artificial Intelligence in Medicine*, vol. 23, no. 1, pp. 89–109, 2001, ISSN: 0933-3657. DOI: https://doi.org/10.1016/S0933-3657(01)00077-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S093336570100077X.

[40] V. Nasteski, “An overview of the supervised machine learning methods,” *HORIZONS.B*, vol. 4, pp. 51–62, Dec. 2017.

[41] J. Alzubi, A. Nayyar, and A. Kumar, “Machine learning from theory to algorithms: An overview,” *Journal of Physics: Conference Series*, vol. 1142, no. 1, p. 012 012, Nov. 2018. DOI: 10.1088/1742-6596/1142/1/012012. [Online]. Available: https://dx.doi.org/10.1088/1742-6596/1142/1/012012.

[42] J. P. B. Castanheira, “Machine learning methods for radar-based people detection,” Master´s thesis, Universidade de Aveiro, 2019. [Online]. Available: https://ria.ua.pt/handle/10773/30403.

[43] M. Bansal, A. Goyal, and A. Choudhary, “A comparative analysis of k-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning,” *Decision Analytics Journal*, vol. 3, p. 100 071, Jun. 2022, ISSN: 2772-6622. DOI: https://doi.org/10.1016/j.dajour.2022.100071. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772662222000261.

[44] S. Raschka, “Stat 479: Machine learning lecture notes,” 2018, [Accessed in June-2023]. [Online]. Available: https://sebastianraschka.com/pdf/lecture-notes/stat479fs18/02_knn_notes.pdf.

[45] S. Ray, *Svm | how to use support vector machines (svm) in data science*, [Accessed in July-2023]. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

[46] M.-P. Hosseini, A. Hosseini, and K. Ahi, “A review on machine learning for eeg signal processing in bioengineering,” *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 204–218, Jan. 2021. DOI: 10.1109/RBME.2020.2969915.

[47] A. Arora, *12 unique ml interview questions on linear discriminant analysis*, [Accessed in July-2023]. [Online]. Available: https://analyticsarora.com/12-unique-ml-interview-questions-on-linear-discriminant-analysis/#What-is-Linear-Discriminant-Analysis.

[48] N. M. Abdulkareem and A. M. Abdulazeez, “Machine learning classification based on radom forest algorithm: A review,” 2021. DOI: 10.5281/zenodo.4471118.

[49] A. Sharma, “Decision tree vs. random forest-which algorithm should you use?” [Accessed in July-2023]. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/.

[50] T. Instruments, “Ti mmwave radar device regulatory compliance guide,” 2021, [Accessed in June-2023]. [Online]. Available: https://www.ti.com/lit/an/spracp3c/spracp3c.pdf?ts=1698320396400&ref_url=https%5C%253A%5C%252F%5C%252Fwww.google.com%5C%252F.

[51] T. Instruments, “Awr1642 evaluation module (awr1642boost) single-chip mmwave sensing solution,” 2017, [Accessed in June-2023]. [Online]. Available: https://www.ti.com/lit/ug/swru508c/swru508c.pdf?ts=1698259947698&ref_url=https%5C%253A%5C%252F%5C%252Fwww.google.com%5C%252F.

[52] T. Instruments, “Mmwave demo visualizer user’s guide mmwave demo visualizer,” 2017, [Accessed in June-2023]. [Online]. Available: https://www.ti.com/lit/ug/swru529c/swru529c.pdf?ts=1698263088529.

[53] T. Instruments, *Uniflash software programming tool | ti.com*, [Accessed in June-2023]. [Online]. Available: https://www.ti.com/tool/UNIFLASH.

[54] T. Instruments, “Mmwave sdk user guide,” 2022, [Accessed in June-2023]. [Online]. Available: https://dr-download.ti.com/software-development/software-development-kit-sdk/MD-PIrUeCYr3X/03.06.00.00-LTS/mmwave_sdk_user_guide.pdf.

[55] T. Instruments, *Mmwave sensing estimator software*, [Accessed in July-2023]. [Online]. Available: https://dev.ti.com/gallery/view/mmwave/mmWaveSensingEstimator/ver/2.3.0/.

[56] T. Instruments, *Mmwave demo visualizer*, [Accessed in july-2023]. [Online]. Available: https://dev.ti.com/gallery/view/mmwave/mmWave_Demo_Visualizer/ver/3.6.0/.

[57] T. Instruments, “Main page modules data structures files millimeter wave (mmw) demo for xwr16xx,” [Accessed in June-2023]. [Online]. Available: http://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/user_5F00_guide.pdf.

[58] A. Trange, “Fmcw mmwave radar for detection of pulse, breathing and fall within home care,” Master’s thesis, School of Electrical Engineering and Computer Science (EECS), 2021. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?dswid=-6310&pid=diva2%5C%3A1541369&c=1&searchType=SIMPLE&language=en&query=AXEL+TRANGE&af=%5C%5B%5C%5D&aq=%5C%5B%5C%5B%5C%5D%5C%5D&aq2=%5C%5B%5C%5B%5C%5D%5C%5D&aqe=%5C%5B%5C%5D&noOfRows=50&sortOrder=author_sort_asc&sortOrder2=title_sort_asc&onlyFullText=false&sf=undergraduate.

[59] T. Instruments, “Understanding the out of box demo data output,” [Accessed in June-2023]. [Online]. Available: https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/6153.understand_5F00_OOB_5F00_output.pdf.

[60] T. Instruments, “People count user guide,” [Accessed in July-2023]. [Online]. Available: https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/Overhead_5F00_People_5F00_Tracking_5F00_and_5F00_Stance_5F00_Detection_5F00_users_5F00_guide.pdf.

[61] MathWorks, *Detect and remove outliers in data - matlab rmoutliers*, [Accessed in June-2023]. [Online]. Available: https://www.mathworks.com/help/matlab/ref/rmoutliers.html.

[62] B. L. Soares, “Study of physiological and structural variability in the acquisition of vital signs with bio-radar,” Master´s thesis, Instituto Superior de Engenharia de Lisboa, 2022.

[63] MathWorks, *Fit multiclass models for support vector machines or other classifiers - matlab fitcecoc*, [Accessed in June-2023]. [Online]. Available: https://www.mathworks.com/help/stats/fitcecoc.html#bufm0zb-1.

[64] MathWorks, *Fit discriminant analysis classifier - matlab fitcdiscr*, [Accessed in June-2023]. [Online]. Available: https://www.mathworks.com/help/stats/fitcdiscr.html.

[65] T. Instruments, *Iwr6843isk-ods evaluation board | ti.com*, [Accessed in July-2023]. [Online]. Available: https://www.ti.com/tool/IWR6843ISK-ODS.