

Indoor Localization with Radio Tomographic Imaging using Channel State Information

M. M. N. D Rodrigo



Indoor Localization with Radio Tomographic Imaging using Channel State Information

Nethmi Rodrigo
Index No : 19001462

Supervisor: Dr Chamath Keppetiyagama
Co - Supervisor: Dr Asanka Sayakkara

May 2024

Submitted in partial fulfilment of the requirements of the
B.Sc. (Honours) in Computer Science Final Year Project



Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: Nethmi Rodrigo

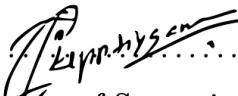
.....


Signature of Candidate

Date: April 22, 2024

This is to certify that this dissertation is based on the work of Ms. M. M. N. D Rodrigo under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Principle Supervisor's Name: Dr. Chamath Keppetiyagama

.....


Signature of Supervisor

Date: April 22, 2024

This is to certify that this dissertation is based on the work of Ms. M. M. N. D Rodrigo under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Co-Supervisor's Name: Dr. Asanka Sayakkara

.....


Signature of Co-Supervisor

Date: April 22, 2024

Abstract

This research study aimed to develop a non-learning-based approach for indoor localization using Channel State Information (CSI). The study successfully created a linear model that could perform the task of indoor localization with limited data. The research dataset contained CSI data pertaining to an occupant who was engaged in performing five distinct activities, and each data point in the dataset came with annotations that detailed specific locations and activities. The study also provided valuable insights into how different activities impact CSI in different ways, how amplitude and phase differences produce various outcomes, and how the size of the comparison vector is an important factor in determining the accuracy of CSI data. While the algorithm proposed to locate an individual within a particular area showed promising results, it had some limitations. Overall, the findings of this research study provide valuable contributions and insights that could assist not only in indoor localization but also in activity recognition.

Preface

Indoor localization has recently emerged as a crucial area of research, driven by the growing need for precise location tracking within buildings using wireless signals. While traditional methods rely heavily on learning-based algorithms, limitations in available datasets have hindered their accuracy and applicability.

In response to this challenge, this dissertation presents a novel non-learning-based approach to indoor localization using CSI. The study explores the impact of various activities on CSI, providing insights into wireless signal propagation and reception within indoor environments.

Throughout this work, I transparently acknowledge the contributions of others, particularly in the development of data collection setups in collaboration with my supervisor. However, the data collected for evaluation and the subsequent analysis are entirely my own, demonstrating the originality and autonomy of this research. By adhering to principles of academic integrity, this dissertation aims to contribute authentically to the field of indoor localization, advancing our understanding of wireless signal behaviour in indoor settings.

Acknowledgement

I would like to express my sincere gratitude to Dr. Chamath Keppetiyagama and Dr. Asanka Sayakkara for their invaluable guidance, unwavering support, and insightful feedback throughout the entire journey of this thesis. Their expertise, patience, and encouragement have been instrumental in shaping this work.

I am deeply thankful to the University of Colombo School of Computing for providing me with the resources and environment conducive to research and learning. The academic rigor and opportunities for intellectual growth offered by the institution have significantly contributed to the development of this thesis.

My heartfelt appreciation goes to my parents for their unconditional love, unwavering belief in my abilities, and constant encouragement. Their sacrifices and encouragement have been my pillars of strength, enabling me to pursue my academic aspirations.

I extend my sincere thanks to the members of my evaluation panel for their invaluable guidance, constructive criticism, and insightful suggestions, which have helped refine this thesis and enhance its quality.

Lastly, I am grateful to all those who have supported me in various ways, directly or indirectly, throughout this academic endeavor. Your support and encouragement have been indispensable in the completion of this thesis.

Table of Contents

Acronyms	1
1 Introduction	2
1.1 Motivation	2
1.2 Scope	4
1.3 Research Gap	4
1.4 Research Questions	5
1.5 Aims and Objectives	6
2 Background	7
2.1 Applications of Indoor Localization	8
2.1.1 Military and Rescue	8
2.1.2 Security	9
2.1.3 Domestic and Commercial Use	9
2.2 Methods of Indoor Localization	10
2.2.1 Visual-Based Approaches	10
2.2.2 Environment-Based Approaches	10
2.2.3 Radio Signal Based Approaches	10
2.3 Introduction to Radio Tomographic Imaging	12
2.4 Radio Tomographic Imaging (RTI) in Indoor Localization	15
2.4.1 RSSI and CSI	16
2.4.2 CSI-based Implementations	19
2.4.3 RTI Setup	19
3 Design and Implementation	23
3.1 Datasets Used	23
3.1.1 Wireless Indoor Localization Dataset (WILD)	24
3.1.2 A Wireless Fidelity (Wi-Fi) Channel State Information (CSI) and Received Signal Strength (RSSI) dataset for human presence and movement detection	28
3.2 Data Collection	33

3.2.1	Collection Setup	33
3.2.2	Collection Methodology	36
3.3	Development of the model	37
3.3.1	Overview of the algorithm	37
3.3.2	Challenges with the proposed algorithm	38
3.3.3	Data Processing	39
3.3.4	Vectorization	39
3.3.5	Heatmap Generation	40
4	Results and Evaluation	42
4.1	Results	42
4.2	Evaluation	47
4.2.1	Evaluation Methodology	47
4.2.2	Evaluation Results	47
4.2.3	Improving Accuracy	49
5	Conclusion	51
5.1	Conclusion on the Research Questions	51
5.1.1	Is it possible to develop a linear model for indoor localization using CSI-based RTI?	51
5.1.2	How to accommodate a sparse device setup for a linear model of CSI-based RTI?	51
5.2	Contributions	52
5.2.1	Main Contributions	52
5.2.2	Auxiliary Findings	52
5.3	Limitations	53

List of Figures

1.1	An example of an RTI resultant image (adapted from RtiWithWirelessNetworks)	2
1.2	An example RTI setup	4
2.1	Categorization of localization methods (adapted from [8615537]) . .	11
2.2	Illustration of an RTI network and links that pass through a localized area (adapted from [jwilson])	13
2.3	An overall view of a RTI setup (adapted from [8615537])	14
2.4	Illustration of a localized area with people walking inside to depict the LOS blockage and multipath effect (adapted from [8444589]) .	14
2.5	Layout of subcarrier types in the WiFi frequency domain (adapted from [subcarriers])	18
2.6	The non-learning based models (adapted from [wifiSensingWithCSI])	20
2.7	Pattern-based vs Model-based solutions (adapted from [patternToModel])	20
2.8	An illustration of an RTI network (adapted from [RtiWithWirelessNetworks])	21
2.9	Experimental setup with a dense network used by Wilson and Patwari in 2011 (adapted from [seeThroughWallTrackingUsingVRTI])	22
2.10	Experimental setup used by Kaltiokallio, Bocca and Patwari in 2013 (adapted from [enhancingAccuracy])	22
3.1	MapFind (right) is an autonomous robotics platform that creates a map (left) for indoor navigation and collects ground truth labelled CSI data for neural network training (adapted from [wild])	25
3.2	Complex high multipath None-line-of-sight (NLOS) environment (1500 sq. ft.) (adapted from [wild])	25
3.3	Simple LOS based environment (500 sq. ft.) (adapted from [wild])	26
3.4	Schematic of the four rooms, drawn to scale, where the experiments were conducted (adapted from [scheurer'improving'nodate]) . .	29
3.5	Example of a CSI sample (adapted from [scheurer'wi-fi'2020]) . .	29
3.6	Schematic of the room "1.28A" drawn to scale (adapted from [scheurer'improving'nodate]) . .	29

3.7	Voxel breakdown drawn with scale	35
3.8	Schematic of the room that the data was collected in	36
3.9	Architecture diagram of the data collection setup	37
4.1	A grid with the marked voxels	43
4.2	Expected output for when the occupant is in the first voxel	43
4.3	Output generated when all generated plots of all activities and sub-carriers were layered on top of each other for when the occupant's actual location was in voxel 1	44
4.4	Output generated with phase vector for each activity separately for when the occupant was in voxel 1	45
4.5	Output generated with amplitude vector for each activity separately for when the occupant was in voxel 1	46
4.6	Accuracy for when vector difference was calculated using amplitude_vector	48
4.7	Accuracy for when vector difference was calculated using phase_vector	49

List of Tables

3.1	Data dictionary of the CSI data	27
3.2	Data dictionary for the CSI samples	30
3.3	Data dictionary for the relation for each link	31
3.4	Data dictionary for the ground-truth annotations	32
3.5	Data dictionary for the ground truth relation	33
3.6	Data dictionary for the relation 'threshold_per_occurrence'	34
3.7	Data dictionary for the relation "distance_vector"	40

Acronyms

- CSI** Channel State Information
RTI Radio Tomographic Imaging
RT Radio Tomography
RSSI Received Signal Strength Indicator
AP Access Point
CTI Computed Tomographic Imaging
NLOS None-line-of-sight
LOS Line Of Sight
RF Radio Frequency
RFID Radio-frequency identification
DFL Device Free Localization
COTS Commercial-Off-The-Shelf
LIDAR Light Detection and Ranging
WILD Wireless Indoor Localization Dataset
NIC Network Interface Card
Wi-Fi Wireless Fidelity
OFDM Orthogonal Frequency-Division Multiplexing
MIMO Multiple Input Multiple Output
LTFS Long Training Symbols

Chapter 1 - Introduction

Radio Tomographic Imaging (RTI) is a device-free localization method that uses the effect of objects on a signal to localize/track the said object that is causing the changes in the signal [[aSurveyOnDetection](#), [arti](#)]. This follows the fundamental principle that radio signals would display a dip in their characteristics when the path between the receiver node and the transmitter node (known as a link) has an obstruction [[blindrti](#)]. RTI works by placing RF nodes around an area of interest, which are capable of communicating with each other and thus form links to communicate that cover the said area. When any object crosses/obstructs a link, this will create a shadowing effect on the power of the signal. The receivers calculate the loss in power of the signal of each link and construct an image that depicts where the attenuation occurs [[deviceFreeLocalizationReview](#)]. RTI utilizes the change in attenuation of signal strength caused when radio signals collide with objects in an area to form an image of the said area, such as Figure 1.1 [[RtiWithWirelessNetworks](#)].

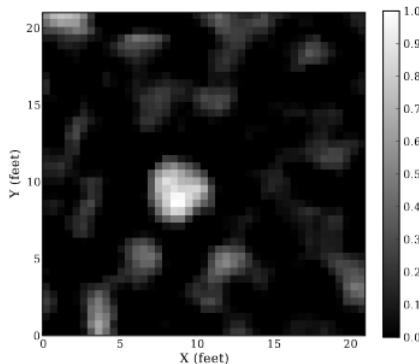


Figure 1.1: An example of an RTI resultant image (adapted from [RtiWithWirelessNetworks](#))

1.1 Motivation

Attempting to locate or localize individuals within an enclosed space is a problem that comes up very often in many rescue operations. To address the problem, various technological solutions have been presented over time, ranging from attempting

to sneak a device into the area of interest to "passive" solutions that do not require direct access to the said area. Imagine such an emergency rescue situation where a building has collapsed, and it is necessary to locate the survivors within the rubble or a partly collapsed area. In such situations, it would be difficult to access the area directly. RTI aims to provide a solution to gain valuable information about the location of individuals in such situations by utilizing the effect that obstacles have on radio frequency signals.

As of now, there has been considerable work done in the area of RTI, going as far as being able to detect changes in the breathing of individuals by monitoring the change in radio signals in the area. Despite there being various applications for monitoring the effect that human activity has on radio frequency signals, this research, in particular, aims to tackle the problem of how to localize individuals within an enclosed space using RTI by developing a linear model as opposed to the usage of learning models. The reasoning behind developing a linear model is due to the necessity of requiring sufficient datasets for learning models. Such datasets are rarely available freely and are not portable to any environment or scenario. Further investigation into linear models can help develop systems that can be developed with a lesser amount of data than learning models and that do not require calibration in the deployment environment.

In 2009, J. Wilson and N. Patwari introduced a linear model that maps human activity to changes in a WiFi signal. While this model provided satisfactory results, given that it was not a learning algorithm, it required an ample number of devices. In order to monitor an area of interest, the said area needed to be covered by devices placed uniformly as depicted in Figure 1.2, with the exact distances between the nodes placed and the dimensions of the area of interest needing to be known measurements. This requirement makes the solution less than ideal in practical environments, and such a high number of devices (or nodes) may not always be available.

These issues proposed the possibility of a solution that would accommodate more robustness in the device deployment strategy (or topology) and how one could develop the existing linear model to achieve the same goal of locating people within an area of interest.

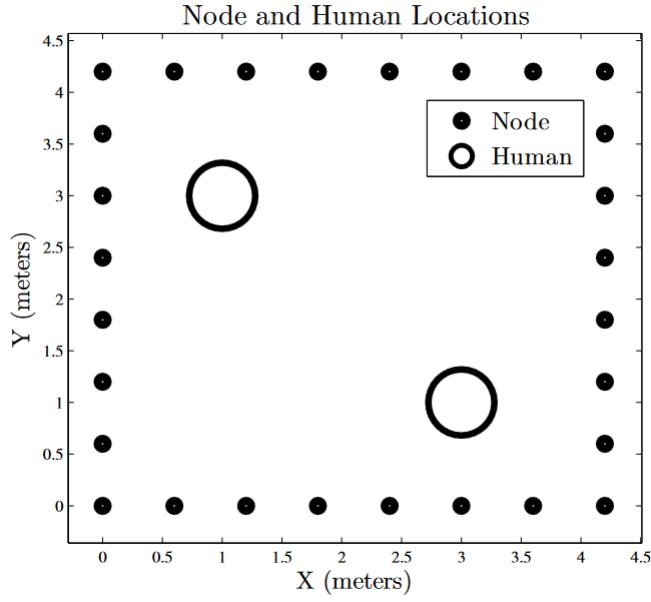


Figure 1.2: An example RTI setup

1.2 Scope

In Scope

- Investigating the capability of linear RTI systems with a lesser number of nodes.
- Developing the existing linear model to enhance the capability to detect occupancy using CSI.

Out of Scope

- The project's feasibility will not be tested in large or outdoor environments.
- The project does not aim to accommodate situations where the node locations are unknown.

1.3 Research Gap

A clear need for a passive, non-intrusive localization and tracking system arose due to the situations of needing to locate people without having direct access to an area, such as in disaster and rescue operations, military operations and shopper analysis. It was these requirements that brought forth RTI, an inexpensive system

capable of localizing objects within an enclosed area even in through-wall situations. Although RTI has much to offer in terms of its practical use, it is not yet in a stage where it is a commercially and predominantly available technology. It was identified during the preliminary literature review that not much focus was given to performing RTI when the setup has an irregular topology. In roughly all the papers reviewed, the nodes were set up uniformly around the area. Whilst such setups can yield optimal results, they reduce the system's practicality in nature. In addition, it was also identified that the reviewed literature on CSI-based implementations developed a machine-learning model to solve the problem of human detection or gesture recognition.

1.4 Research Questions

1. **Is it possible to develop a linear model for indoor localization using CSI-based RTI?**

Hypothesis: CSI has an advantage over Received Signal Strength Indicator (RSSI) in providing more detailed information that can result in a linear model with higher accuracy. As there is a lack of datasets available when developing a learning-based approach, a linear model is a preferred approach.

2. **How to accommodate a sparse device setup for linear model of CSI-based RTI?**

Hypothesis: The statistical linear model presented in the paper by **RtiWithWirelessNetw** assumes that enough devices cover the entire area of interest. However, when the network setup consists of only a few nodes placed randomly throughout the area, this assumption may not hold. It could significantly impact the effectiveness of the system. It is possible to modify the standard model to account for such irregular device placement with minimal impact on the successful implementation of RTI.

1.5 Aims and Objectives

Project Aim

This project focuses on exploring the ability to image an area of interest with RTI using CSI when the setup is deployed in a nonuniform and irregular manner and a sparse network of nodes is used.

Objectives

- Exploring the ability to form an image/detect the occupancy in an enclosed area when the setup used is sparse and deployed in a non-uniform manner.
- Developing a linear RTI model that utilizes CSI as the primary parameter.

Chapter 2 - Background

Indoor localization is novel research that is gaining popularity due to its wide applicability in various fields, especially in NLOS environments where certain human activity in an enclosed space needs to be monitored in situations where it may be difficult to get direct visual input [s20010043] [liu2020human]. Indoor localization brings forth a multitude of potential applications, such as recognizing patterns in smart home management, disaster management, emergencies, rescue situations, traffic management, assisted living, being a few [s20010043] [Device-Free-Crowd-Counting] [electronic-frog-eye] [liu2020human] [zou2018device].

Despite being applicable in a myriad of situations, indoor localization is a complex research problem. In terms of visual-based approaches, where indoor localization is performed by processing the feed obtained by a camera, it brings in problems such as scale invariance, where the scale of the object depends on its distance to the camera, and other problems such as environmental changes such as fog that can affect the output [Modolo'2021'WACV]. Moreover, in terms of non-visual based approaches, overlapping objects and clutter in the environment, the activities that the individuals are engaged in (some individuals may be stationary, some may be moving), unpredictable human behaviour, indoor localization proves to be a challenging task[7603465]. In addition, [electronic-frog-eye] notes that solutions should be "scale-able and minimally intrusive" and need to function as intended in various scenarios regardless of the crowd density [Modolo'2021'WACV].

Researchers have looked into various methods to tackle this problem, ranging from visual-based approaches such as using cameras combined with image processing, environmental science and radio signals [9043195]. Radio signal-based approaches use Radio Frequency (RF) sensing devices such as Radio-frequency identification (RFID) tags, Bluetooth, or processes the effect on Wireless Networks [8255034]. Although computer vision has long since been used to achieve this, it has many limitations, is not applicable in NLOS environments, and poses privacy concerns [Device-Free-Crowd-Counting]. Thus, the need arose for a technology that is capable of passively monitoring human activity within an enclosed space without

requiring direct access to that specific area.

RTI is a radio-based novel approach. Formed from the Greek word "tomos", which translates to section or sectioning, tomography is concerned with the computation of an image of a sectioned area using the interaction of signals with the objects in the area of interest. When this technology is performed using radio signals, it is called Radio Tomography (RT) [7829922].

RTI provides an effective method for through-wall tracking compared to the other methods [shukri2017device]. It uses the attenuation caused by a signal when an object obstructs the links in an area covered by a wireless network to compute information about the object [6214374]. RT using wireless networks is being greatly researched into as the infrastructure required is readily available, easier to deploy, low cost and less intrusive than camera-based approaches [liu2020human]. Wireless tomography uses the properties in a wireless network, such as the RSSI or CSI, and how they interact with the objects in an environment to compute an image or calculate the number of people in a certain localized area.

2.1 Applications of Indoor Localization

This section aims to give a general overview of how indoor localization is applied in different use cases. The applications can be generalized to military and rescue, security, domestic and commercial use, and assisted living.

2.1.1 Military and Rescue

Indoor localization can be used in various situations, albeit the most popular usage is in military situations. [6214374] notes that it has wide applications in the areas of "security, surveillance, search-and-rescue, and military" and can be used for tracking systems that can help to locate survivors or individuals trapped during disasters. Usage of passive localization systems could result in fewer injuries for law enforcement officers as such imaging systems would assist them in assessing an environment and locating offenders in situations where they would otherwise have no visual input [jwilson] [6214374] [7829922]. The military can utilize such imaging systems to detect fighter jets and drones or the number of soldiers

and their locations, which would assist in setting up and planning counterattacks and defensive mechanisms [s20010043]. In addition, in situations such as concerts with massive crowds, such technologies can be used to manage the crowd to avoid disastrous situations [s20010043]. Another possible application is detecting if the amount of people in a gathering exceeds a safe threshold, which would be required in crises such as the COVID-19 crisis where large gatherings were prohibited [counting-a-staionary-crowd].

2.1.2 Security

Traditional methods to provide security are camera-based approaches or trip-wire-based approaches. Tripwire systems have a major limitation of being able to identify only if a boundary has been crossed and cannot provide information about the location or number of people [jwilson]. While security cameras do have the advantage of providing a visual output that can be used to identify people, it has limited coverage, cannot see through objects such as walls and may not always work as intended in low light situations [7829922]. On the contrary, passive localization systems, such as RTI based systems, could be used to be alerted when an intruder crosses a boundary as well as to track the intruders [jwilson].

2.1.3 Domestic and Commercial Use

Localization systems can be integrated with smart buildings or homes to optimize energy consumption according to the individuals present in specific regions and control the air conditioning, lighting and heating accordingly [7829922] [jwilson] [8444589]. In terms of commercial use, it can be used in buildings such as supermarket stores to assess the areas that attract more customers [8444589]. It can also assist in urban planning to better distribute the resources based on the occupancy [8444589]. In addition to urban planning, indoor localization also helps in event management to identify and estimate the number of people sitting in a certain region [counting-a-staionary-crowd].

2.2 Methods of Indoor Localization

The problem of indoor localization has been approached in different ways. A noted classification of the approaches is as image-based approaches and non-image-based approaches [electronic-frog-eye] [8254522].

2.2.1 Visual-Based Approaches

Traditionally, camera-based or special hardware-based approaches have been used [liu2017wicount]. Camera-based approaches add an image-processing algorithm to a visual output to extract information such as the identity or the number of people [electronic-frog-eye] [8444589] [electronics10030315]. As such, the inclusion of image processing algorithms leads to an additional computational overhead [6128688] [8255034]. To add to its limited applicability, video-based systems would require a network of cameras to be set up, which would add on a deployment cost in addition to its inability to function as expected in low light situations [8444589]. In addition to bringing in the added complication of privacy concerns, being applicable only in direct line-of-sight situations, most camera-based systems require favourable environmental and lighting conditions to function as intended [zou2018device] [electronic-frog-eye] [8255034].

2.2.2 Environment-Based Approaches

Research has also been done in utilizing environmental science such as carbon dioxide concentration, dew point and humidity to identify the number of people [8615537]. Similar to the downside of visual-based approaches, this approach requires direct access to the area of interest.

2.2.3 Radio Signal Based Approaches

Radio signal-based approaches can be further classified into two main categories [8444589] [8615537].

1. Device-based active methods which rely on people to carry a communication device

- Device-free passive methods which use the interaction of signals with the people in the area of interest. These methods are based on the principle that the presence and motion of humans affect radio signals in a significant manner [8615537].

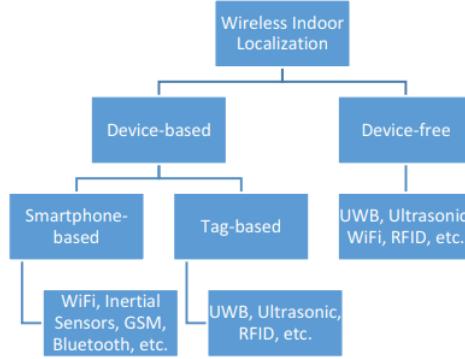


Figure 2.1: Categorization of localization methods (adapted from [8615537])

Since device-based active methods are limited in applicability due to the requirement of people in the area of interest needing to carry communication devices, device-free passive methods are being largely researched. Indoor localization inherently poses the requirement to utilize device-free passive methods to be practically usable in many situations as they need to be rapidly deployed, especially in emergencies, without having direct access to the area of interest [6214374]. Modern requirements have brought forward the need for passive methods, which is also known as Device Free Localization (DFL) or “sensorless-sensing” [shukri2017device].

Device-Based Active Approaches

RFID is a contact-less communication technology whose system consists of a tag and a reader [liu2020human]. RFID systems required the individuals to carry tags, which would also enable the individuals to be uniquely recognizable [shukri2017device]. RFID tags are used since the RFID signals reflect off of the human body. This reflection is then used for human activity sensing [liu2020human]. However, the readers required by RFID systems are expensive hardware [shukri2017device]. Device-based approaches carry the disadvantage of requiring the people in the crowd to be carrying mobile devices [liu2017wicount]. Since RF based approaches use specialized sensors and have a device-based active nature, they cannot support large-scale deployment [8255034].

Device-Free Passive Approaches

RTI is a radio-based, device-free passive approach. To achieve NLOS tracking, using radio signals in imaging systems has an advantage over signals such as infrared since they are long-range and can travel through obstructions such as walls [jwilson].

Using Wireless Signals with RTI

With the recent boom of wireless networking and mobile devices, wireless infrastructure has been deployed widely in both commercial and residential areas, leading to most areas constantly being covered by wireless networks [liu2020human] [8255034] [counting-a-staionary-crowd] [shukri2017device] [abeywardane2017case].

Due to this abundantly available network coverage and most Commercial-Off-The-Shelf (COTS) devices being WiFi enabled, using WiFi has become a popular approach to indoor localization [counting-a-staionary-crowd]. Although ZigBee transceivers have outperformed WiFi in indoor environments, in situations where there is less scattering and in outdoor environments, WiFi has had better outcomes in localization. [dilushi].

Despite WiFi infrastructure being readily available and abundant coverage by wireless networks, usage of WiFi poses its drawbacks. Its sensing range is limited, and the signals can be affected by environmental factors. Moreover, WiFi cannot support both motion sensing and communication simultaneously [liu2020human].

Despite the existence of other techniques such as camera, capacitance, infra-red, ultrasonic, and environmental science, most of them have major drawbacks such as raising privacy concerns, requiring training in a deployed environment, high deployment cost, occlusion, with the most critical being their inability to support NLOS tracking. Due to its ability to support NLOS tracking, regardless of its drawbacks, RTI imaging remains a popular solution for indoor localization.

2.3 Introduction to Radio Tomographic Imaging

Radio tomography stems from the concepts of two widely used imaging systems. One is radar systems, which transmit a signal and, based on the echo, identify the

existence and distance of objects in the environment. The other is imaging systems in the medical industry, such as Computed Tomographic Imaging (CTI), which sends out a signal through a medium and uses the signal measurements to compute an image of the spatial field of the paths that the signals went through. RTI, similar to radio systems, uses radio frequencies and measures the signal strengths across different paths that the signal travels through in a certain area [jwilson]. RTI is a technology that enables the imaging of a localized area based on the attenuation of wireless signals caused by the physical objects within the area of interest [jwilson]. Wireless tomography combines the concepts of RTI using wireless communication technologies [smallbon2015implementation].

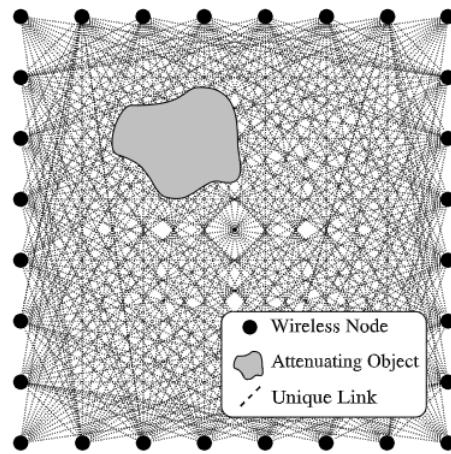


Figure 2.2: Illustration of an RTI network and links that pass through a localized area (adapted from [jwilson])

[jwilson]

In a localized area of interest that is covered by a wireless network, the obstruction of the links by the objects in the said area would cause an attenuation in the signals, and the link would experience shadowing losses [shukri2017device] [jwilson]. In RTI, multiple radio transceivers, or RF nodes, are set up around an area such that they create a wireless network grid covering the area [shukri2017device]. A link is used to describe the straight line between two RF nodes [8615537]. A voxel is a three-dimensional element in the coverage area, similar to a pixel in a two-dimensional space [8615537].

The impact on a link by the objects in an environment can be categorized into two types: [8444589] [jwilson].

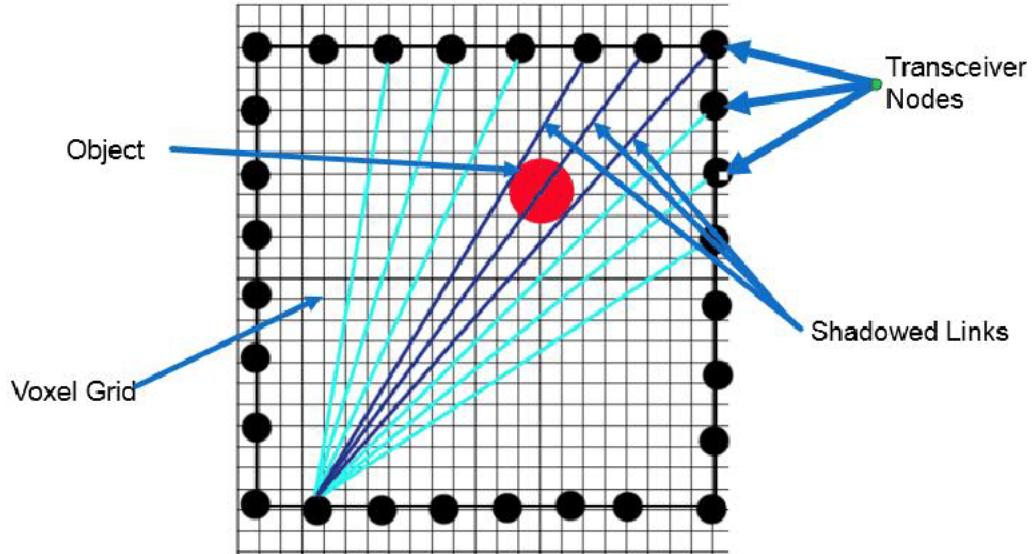


Figure 2.3: An overall view of a RTI setup (adapted from [8615537])

1. Line Of Sight (LOS) blockage - When an object is obstructing a direct path between a transmitter and receiver. This causes a significant attenuation in the RSSI [dilushi].
2. Multipath effect - The signals can reflect and bounce off objects in the area. This would cause the RSSI to fluctuate. The interference with the signal could either be a constructive interference, where the reflected signals combine to form a stronger signal than the transmitted, or a destructive signal, where the reflected signals have cancelled each other out.

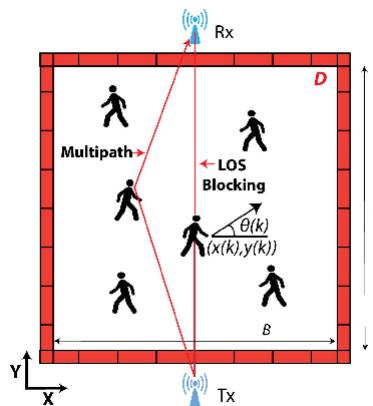


Figure 2.4: Illustration of a localized area with people walking inside to depict the LOS blockage and multipath effect (adapted from [8444589])

RTI would typically have three main components [8615537].

1. The weighting models - To identify the voxels which contribute to a link and

their contribution to said link.

2. Noise - Noise is identified to be the variation in RSSI measurements when there is no object obstructing a link that could cause an attenuation.
3. Image reconstruction - Concerns with constructing a map/image of the area under inspection such that the effect of noise is as minimal as possible. The reconstruction of the image is an inverse problem.

Inverse Problem

A forward problem is any typical problem where a solution to a system is obtained by applying data to known parameters. An inverse problem attempts to discover the parameters using the observations or result of a system [[tarantola2006popper](#)]. In the context of RTI, the inverse problem would be, knowing the emitted and received RSSI measurements, what would be the signal loss at each voxel.

2.4 RTI in Indoor Localization

Radio signals are inherently affected by the phenomenon of "radio irregularity", which refers to signals being absorbed, reflected or scattered by objects in their path [[lin2013exploiting](#)]. RTI uses this phenomenon to its advantage to track the objects that caused the attenuation [[7346782](#)]. It exploits changes in the radio signals to locate objects of the body without them needing to carry a device or be actively participating in the localization process [[shukri2017device](#)].

RTI can be mainly categorized into two.

1. Shadowing-based RTI

It uses the shadowing effect of the human body to correlate to the location of an individual. The shadowing effect is the phenomenon of the signal values displaying a considerable loss due to an obstacle being in the direct path between a receiving and sending device. [[8615537](#)] [[shukri2017device](#)].

2. Variance-based RTI

A vector of the RSSI variance on each link is used to compute an image that can detect motion within the voxels. It is based on the fluctuations of RSSI measurements over time due to the motion of individuals. These systems require an initial calibration without the presence of objects in the room

[shukri2017device] [7346782].

2.4.1 RSSI and CSI

RTI works by analyzing the effect that objects in an environment have on a signal. This analysis is performed utilizing either of the one-channel properties RSSI or CSI.

RSSI

RSS measurements can be calculated by most commercial off-the-shelf (COTS) electronic devices. It is the received power of a signal, which is the squared magnitude of the complex baseband voltage, in decibels [wang2015devicefree]. It is a coarse indicator of the received power obtained from the MAC layer [compressiveSensingBased]. The received signal strength can be mathematically represented as,

$$y_i(t) = P_i - L_i - S_i(t) - F_i(t) - v_i(t) \quad (2.1)$$

where,

- P_i : Transmitted power in dB.
- $S_i(t)$: Shadowing loss in dB due to objects that attenuate the signal.
- $F_i(t)$: Fading loss in dB that occurs from constructive and destructive interference of narrow-band signals in multipath environments.
- L_i : Static losses in dB due to distance, antenna patterns, device inconsistencies, etc.
- $v_i(t)$: Measurement noise [RtiWithWirelessNetworks].

CSI

CSI serves as the metric in Orthogonal Frequency-Division Multiplexing (OFDM) systems to characterize amplitude and phase fluctuations among subcarrier frequencies during signal transmission from a transmitter to a receiver. Channel estimation is the method employed to discern these fluctuations across subcarriers in OFDM systems by transmitting a designated set of known shared pilot symbols arranged in a comb-type pilot pattern. This pattern consistently utilizes the same subset of subcarriers as pilot subcarriers over time.

CSI measurements are fine-grained channel information obtained from the physical layer, including the different time delays, amplitude attenuation and the phase shift of multiple paths in each subcarrier [**deviceFReeOccupancy**]. While CSI amplitude variations can be observed to depict different patterns for different individuals, activities and gestures, phase shifts in the frequency domain display, which are related to transmission delay and direction, can be utilized to obtain information regarding the location of humans [**wifiSensingWithCSI**]. CSI represents how wireless signals can propagate along multiple different paths from a receiver to a transmitter at certain carrier frequencies [**wifiSensingWithCSI**].

A WiFi channel with Multiple Input Multiple Output (MIMO) has multiple subcarriers divided by OFDM. In order to measure CSI, a transmitter sends Long Training Symbols (LTF), which contains pre-defined symbols for each sub-carrier in the packet preamble. When the receivers receive the LTFs, the CSI is calculated using the received signal and the original LTF. For each sub-carrier, the channel is modelled by the following equation,

$$y = Hx + n \quad (2.2)$$

where,

- y: Received Signal
- H: The estimated CSI matrix (calculated by the receiver)
- x: The transmitted signal
- n: Noise vector

OFDM systems utilize multiple subcarriers, each with slightly different frequencies, to prevent interference issues and ensure reliable data transmission. Some subcarriers serve as pilot subcarriers, carrying known symbols that help the receiver understand variations across the entire OFDM symbol. This enables the correction of signal fluctuations across different subcarriers through subcarrier equalization. For example, in a standard WiFi channel with a 20MHz bandwidth, there are 64 subcarriers, with some designated as pilot subcarriers as illustrated in Figure 2.5. These pilot subcarriers, along with others carrying encoded data, are organized around a central frequency, allowing for efficient data transmission and reception.

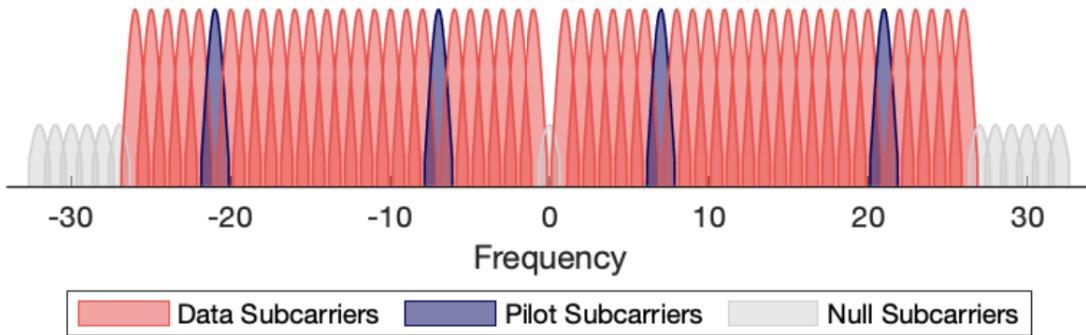


Figure 2.5: Layout of subcarrier types in the WiFi frequency domain (adapted from [subcarriers])

RSSI vs CSI

RSSI has been a popular approach because it does not need expensive hardware to form a dense network. Since the number of nodes used increases the accuracy of the system, a low-cost dense network would “ result in an RTI system with significant capability [dualRti, wang2015devicefree]. When an object obstructs the line-of-sight (LOS) path in a wireless network, the RSS in this path will undergo a significant drop. Although a majority of the RSS-based approaches rest on this principle, due to the multipath effect, the RSS signal may not always undergo a drop. Based on whether the interference is constructive or destructive, due to the signals getting reflected or diffracted, the signals could either increase, decrease or even remain unchanged [compressiveSensingBased, wang2015devicefree, dualRti]. The dips in the signals which are not caused by objects are called fake-shadowing-decreased measurements [dualRti]. If the fake/incorrect measurements are taken into account in the image construction phase, it can cause a significant drop in the accuracy. Researchers have since then come up with concepts, such as the fade level and using regularization, to identify the fake measurements from the informative ones and to reduce the effect of noise [dualRti, rtiWithLowRank]. CSI can provide fine-grained information, but it is susceptible to inherent noise that conventional low pass filters cannot remove [deviceFreeOccupancy]. CSI values are very sensitive to even the most minor disruptions in the environment and more resilient to noise [electronicFrogEye]. Due to its sensitivity, it has been used in related work for human gesture recognition, human activity recognition, and even

to detect breathing analysis [**deviceFReeOccupancy**]. Unlike in the case of RSS, most commodity WiFi chipsets do not support CSI extraction [**democratizedRT**]. CSI has the upper hand over RSS since it is more robust to the multipath effect, supports channel and spatial diversity and provides more information than RSS that can [**compressiveSensingBased**].

2.4.2 CSI-based Implementations

In a survey done in 2020 by Ma et al., different theoretical models, statistical models and algorithms are mentioned as depicted in Figure 2.6 [**wifiSensingWithCSI**]. Despite identifying and describing several models, the paper presents only two related literature that follow the mentioned mathematical models. Unfortunately, both papers track mobile devices specifically and not objects or humans in the environment [**spotfi**] [**DecimeterLevel**]. Another survey conducted in 2017 by Yousefi et al. only reviews learning-based models for CSI-based implementations for behaviour recognition [**behaviorRecogSurvey**]. A survey conducted by Zhang et al. in 2017 does a more comprehensive review of the model-based approaches, which they identify as depicted in Figure 2.7. During their survey, the authors identified the Fresnel zone model as a powerful model as they conducted experiments in which they observed that it is capable of interpreting where and at what orientation a person's respiration can be sensed. When considering the related work mentioned that focuses on localizes and tracking, the passive tracking (CSI-speed model-based) method utilized can only work when the sensor devices are deployed inside the tracking area along the walls such that objects and humans do not obstruct the LOS [**widar**]. The Angle-of-Arrival (AOA) based method (MaTrack), comparatively, can only work in situations when there is only one target to track [**matrack**]. WiDir, the Fresnel zone model-based approach, only estimates the indoor movement direction of a target and does not localize or track the said target, and similar to MaTrack is only for singular subjects [**widir**].

2.4.3 RTI Setup

For most typical RTI setups that attempt to perform through-wall tracking or occupancy detection, wireless nodes would be placed in a uniform manner sur-

Modeling-based:

- (1) modeling X by theoretical models based on physical theories or statistical models based on empirical measurements;
- (2) inferring $f(\cdot)$ by the model of X ;
- (3) predicting Y by the modeled function $f(\cdot)$ and measurements of X , sometimes assisted by optimization algorithms.

Theoretical Models: Fresnel Zone Model, Angle of Arrival/Departure, Time of Flight, Amplitude Attenuation, Phase Shift, Doppler Spread, Power Delay Profile, Multi-Path Fading, Radio Propagation: Reflection, Refraction, Diffraction, Absorption, Polarization, Scattering; **Statistical Models:** Rician Fading, Power Spectral Density, Coherence Time/Frequency, Self/Cross Correlation; **Algorithms:** MUSIC, Thresholding, Peak/Valley Detection, Minimization/Maximization

Figure 2.6: The non-learning based models (adapted from [wifiSensingWithCSI])

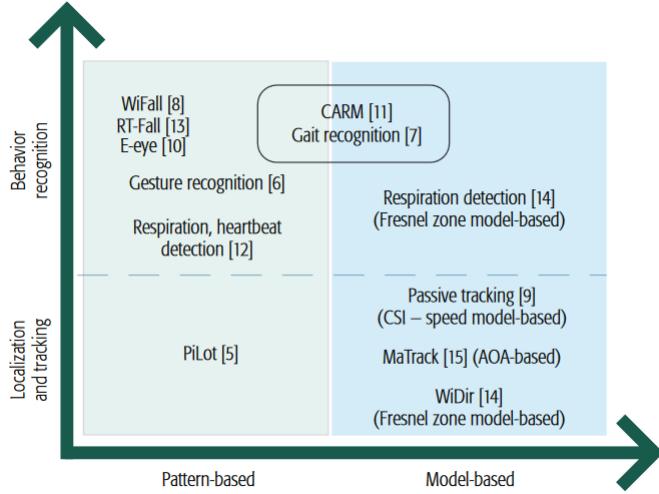


Figure 2.7: Pattern-based vs Model-based solutions (adapted from [patternToModel])

rounding the area of interest in order to form a dense network of nodes. Since many RTI models attempt to image the attenuation per voxel, as per Figure 2.8, a sparse network of nodes would not form a regular grid, thus making it difficult to image as the voxel sizes would differ. While dense networks utilize a large number of nodes, such as 20-30 nodes, a sparse network would contain a much lesser number of nodes, such as three nodes. This becomes an issue in situations such as in disaster areas, where the placement of nodes cannot be done uniformly and would be a random deployment [deploymentStrategiesReview]. Random deployment could bring forth many issues, such as poor network connectivity and insufficient coverage. Self-deployment of nodes has been a proposed technique where the nodes themselves would be capable of organizing themselves to form an optimal topology [deploymentStrategiesReview]. This would be a diffi-

cult task to perform as finding the optimal topology is an NP-Hard problem and would require trade-offs when considering the different factors such as network connectivity, coverage, energy consumption, and the battery lifetime of the nodes [deploymentStrategiesReview, strategiesForNodePlacementSurvey].

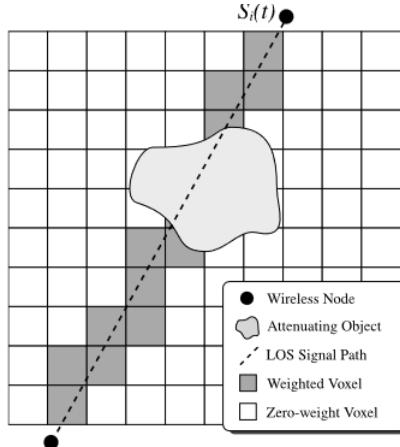


Figure 2.8: An illustration of an RTI network (adapted from [RtiWithWirelessNetworks])

Regular Topology

Figure 2.9 and Figure 2.10 are a few examples of setups that follow a uniform/regular topology. When a regular topology is set up, an imaginary grid with voxels of a specific size can be used. When the topology becomes irregular, maintaining a uniform grid with equally sized voxels can be difficult and would complicate the mathematical models thus developed.

In addition, **seeThroughWallTrackingUsingVRTI** mentions how the number of nodes utilized increases the accuracy of the system rather than using nodes with high computational power, which would create a dip in accuracy in setups with a sparse network. This research aims to explore the methods to increase/maintain the accuracy achieved whilst using a lesser number of nodes and when the grid created by the links in a network cannot form a uniform grid.

The work done thus far has focused on aspects such as increasing the accuracy of the system, attempting to track multiple targets since most papers have a limit of being able to track at most three/four people at a time [**deviceFreeIndoorMulti**] and not much focus has been given to improve the practical usability of RTI. Although

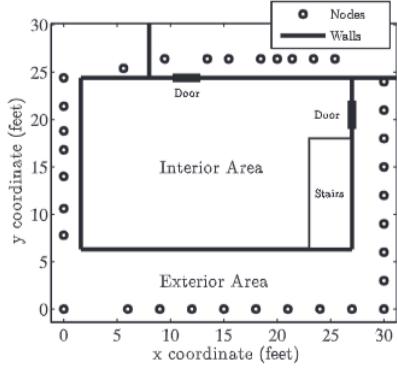


Figure 2.9: Experimental setup with a dense network used by Wilson and Patwari in 2011 (adapted from [[seeThroughWallTrackingUsingVRTI](#)])



Figure 2.10: Experimental setup used by Kaltiokallio, Bocca and Patwari in 2013 (adapted from [[enhancingAccuracy](#)])

it has been proposed that future advancements lie in increasing the deployment strategies of an RTI setup, not much work has been done on the topic. Therefore, there is a gap in terms of obtaining a satisfactory result when using a mathematical model in irregular setups with a sparse node network.

Chapter 3 - Design and Implementation

To understand RTI without the availability of a non-learning model’s public implementation, the only apparent solution was to replicate a typical configuration illustrated in Figure 1.2, which would necessitate using at least four or more devices. However, while attempting to replicate the setup, several obstructions were encountered.

1. A significant number of devices were required
2. A protocol needed to be implemented for the devices to communicate and transmit the CSI data to a computer.
3. The node’s firmware must be familiarized to implement such a protocol.

It is imperative to note that successful implementation of such a protocol would require an in-depth understanding of the firmware, communication protocols, and data transmission mechanisms. Given the constraints of the project timeline and the desired output, it was determined that replicating existing setups would be excessively time-consuming. As such, allocating resources to this task would be detrimental to the project’s overall success.

3.1 Datasets Used

Publicly available datasets that meet the project requirements were sought to develop a non-linear model. A suitable dataset would have the following properties:

1. Accurate information regarding the environment in which the data was collected, such as the room’s dimensions. This information is critical to segment the area of interest into voxels.
2. Clearly stated information on the exact location of every occupant and the devices used to collect the data. This attribute is crucial for establishing a correlation between the CSI data and occupant presence and the relationship between an occupant’s location and the device’s location.
3. A comprehensive and detailed description of the dataset, including informa-

tion about the exact number of subcarriers and their corresponding data format.

3.1.1 WILD

In June 2020, WILD was released. The dataset was collected to develop a Deep Learning-based Wireless Localization for Indoor Navigation called "DLoc". It is claimed to be the first of its kind and the largest location labelled Wi-Fi CSI dataset. The authors utilized the dataset to create a platform that facilitates indoor navigation. DLoc platform enables the utilization of any Commercial-Off-The-Shelf (COTS) devices (such as smartphones) to access a map of the environment and estimate their location concerning the map. Once trained using collected data, DLoc enables new users to estimate their location with their smartphone [wild]. The authors mounted an off-the-shell Wi-Fi transmitter onto the Turtlebot2 platform, a low-cost, open-source robot development kit to collect the CSI data and generate the maps. The robot was equipped with Light Detection and Ranging (LIDAR), a camera, and odometry to collect accurate location estimates for wireless channels. To gather accurate data, the authors placed the Wi-Fi transmitter on the robot's top at a height typically used by an average smartphone user. The robot was operated through the Robot Operating System (ROSKinetic) with the help of a laptop that was equipped with an 8th Gen Intel Core i5-8250U mobile processor and 8GB of RAM.

The data was collected in two different environments: a complex NLOS environment with high multipath, as shown in Figure 3.2, and a simple LOS environment, as shown in Figure 3.3 [wild]. For Figure 3.2, data was collected for five different setups; for Figure 3.3, data was collected for three different setups.

The CSI dataset was provided in MATLAB files stored using HDF5 file structure, with one file for each setup in each environment, and each file was said to contain the variables given in Table 3.1.

Challenges with the dataset

Upon loading the dataset, it was discovered that the actual attributes differed from those listed in the description. Specifically, the attributes `RSSI`, `channel`, `ap_aoa`,



Figure 3.1: MapFind (right) is an autonomous robotics platform that creates a map (left) for indoor navigation and collects ground truth labelled CSI data for neural network training (adapted from [wild])

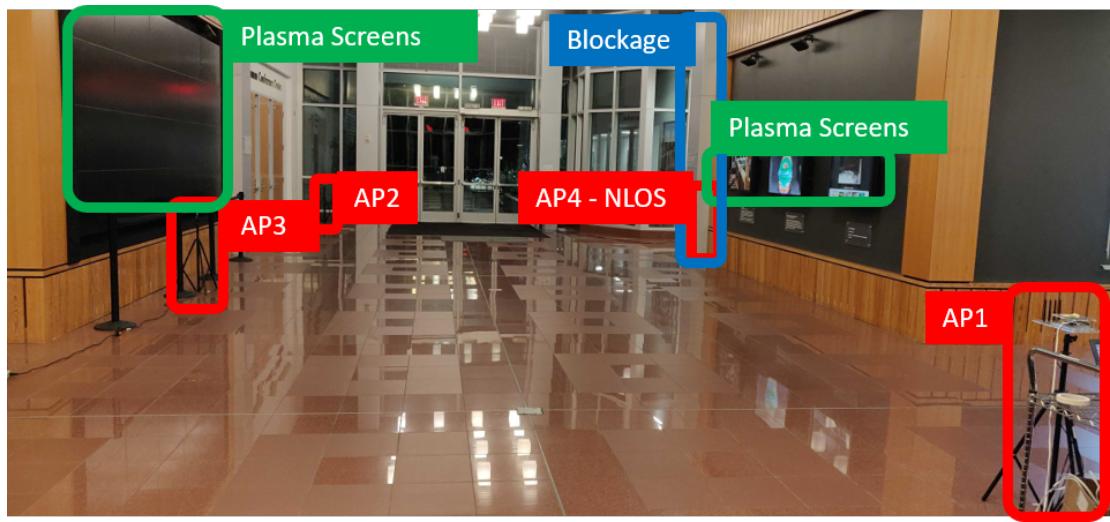


Figure 3.2: Complex high multipath NLOS environment (1500 sq. ft.) (adapted from [wild])

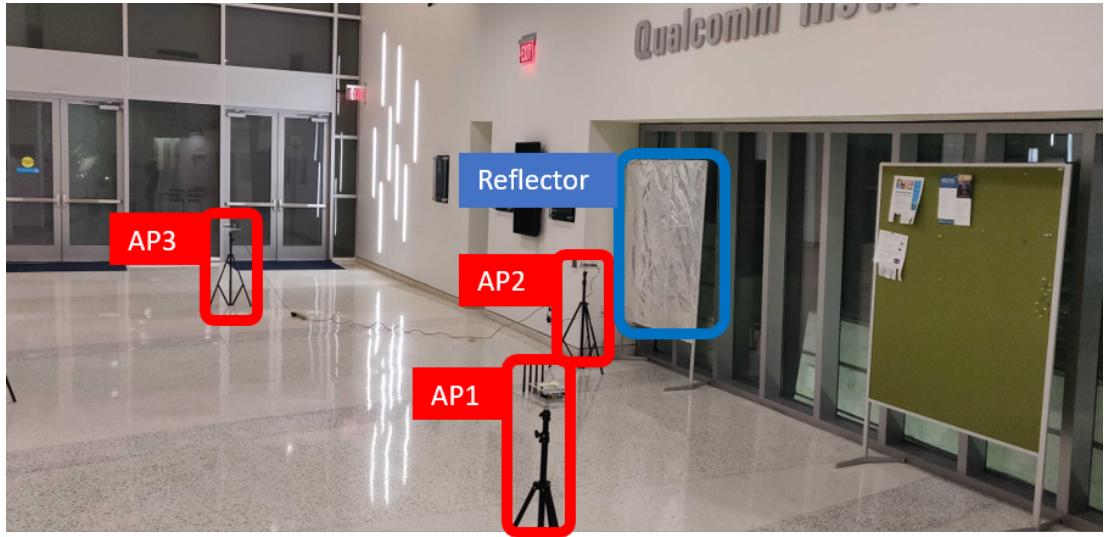


Figure 3.3: Simple LOS based environment (500 sq. ft.) (adapted from [wild])

`d1` and `d2` were not present, while the attributes `#refs#`, `channels_w_offset`, `channels_without_offset` and `tof_gnd` were included despite not being listed in the data dictionary.

The dataset contained several discrepancies that posed obstacles to the analysis process. The following issues were identified:

1. The `labels` variable did not provide the relationship between the `x` and `y` locations and the environment's actual dimensions, making it challenging to determine the spatial locations of the data points.
2. The data format for the Access Point (AP) locations lacked information about the entire axis's scale, as it was supposed to be in the missing variables `d1` and `d2`, which made it challenging to comprehend the location concerning the entire axis.
3. The `channels_w_offset` and `channels_wo_offset` variables contained a 4D array that can be assumed to be the `channels` variable given in the data dictionary. However, the dataset lacked any information about whether the single value contained in the variable was the amplitude or phase of the CSI data. CSI data is usually given as a tuple of imaginary and complex numbers to calculate both amplitude and phase.
4. The task of dividing the area into voxels and calculating the dimensions of each voxel was hindered by the unavailability of the scale for the `x` and `y` axes.

Variable	Data format	Description
channels	[No of datapoints x No of subcarriers x No of antennas X No of access points]	4D complex channel matrix
RSSI	[No of subcarriers x No of access points]	2D RSSI matrix
labels	[No of subcarriers x 2]	2D XY labels
opt.freq	[No of subcarriers x 1]	1D vector that describes the frequency of the subcarriers
opt.lambda	[No of subcarriers x 1]	1D vector that describes the wavelength of the subcarriers
opt.ant_sep	NA	antenna separation used on all of our APs
ap	[No of access points]	Each element corresponding to [No of antennas x 2] XY locations of the number of antennas on each AP
ap_aoa	[No of access points x 1]	Vectors that contain the rotation that needs to be added to the AoA measured at each AP (assumes that the AoA is measured about the normal to the AP's antenna array)
d1	NA	The sampled x-axis of the space under consideration
d2	NA	The sampled y-axis of the space under consideration

Table 3.1: Data dictionary of the CSI data

However, the most significant obstacle was the lack of information about the robot and AP locations, as it was impossible to identify which voxel an AP and occupant would be in. As a result of the challenges encountered, it was required to pursue a different dataset.

3.1.2 A Wi-Fi Channel State Information (CSI and Received Signal Strength (RSSI) dataset for human presence and movement detection

”A Wi-Fi Channel State Information (CSI) and Received Signal Strength (RSSI) dataset for human presence and movement detection”, released in February 2020, claimed to be the first at the time to release a dataset that combines antenna-wise RSSI with human presence annotations [**scheurer-wi-fi-2020**]. The authors collected the dataset to develop a machine-learning-based model that detects human presence using RSSI.

The data was collected from four testbeds, depicted in Figure 3.4. The Wi-Fi APs, displayed as blue disks in Figure 3.4, were regular COTS computers running Ubuntu with an Intel 5300 Network Interface Card (NIC) and three dipole antennas. Each AP had a whitelist of client device addresses. Three clients were in each room, and each client had three antennas. The Raspberry Pi 3 model was used as the client device with an integrated Cypress CYW43455 Wi-Fi chip connected to a PCB Proant Dual Band Niche antenna in the larger rooms. On the other hand, Raspberry Pi 2 was used in the smaller rooms, with an assortment of single-antenna Wi-Fi USB dongles. The dotted lines found in Figure 3.4 represent connections between the AP and a client and are referred to as ”Wi-Fi links”.

The ”Intel CSI” tool [**Halperin-csitoold**] was utilized to capture and save the CSI information from every antenna of each client. The collected CSI data was stored in separate JSON files for each AP-client link. The JSON file’s data dictionary for the CSI data is presented in Table 3.2. Figure 3.5 provides a sample of the CSI data, where each CSI object is listed on a separate line in the JSON file.

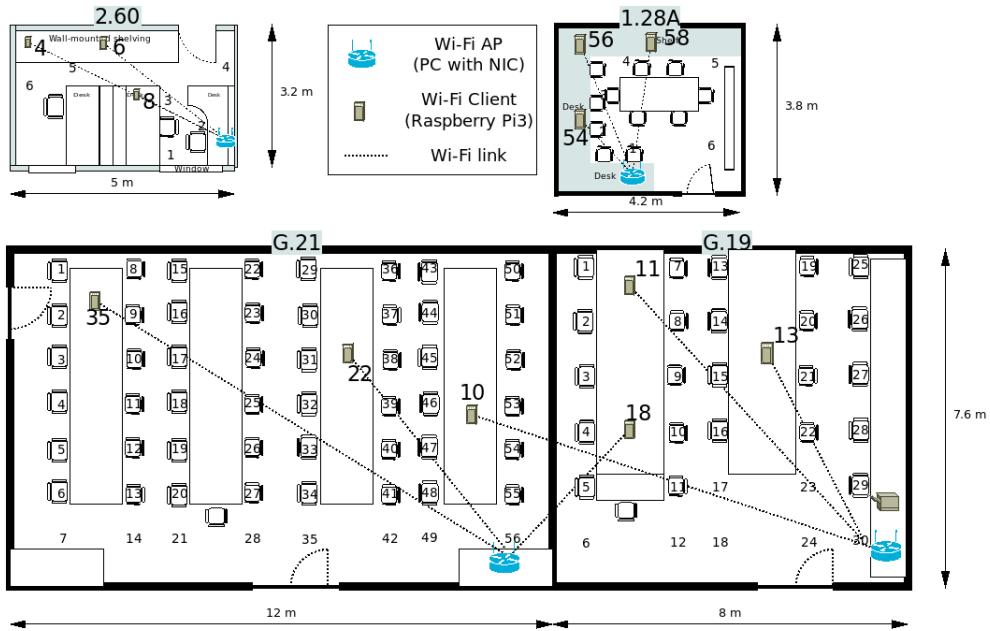


Figure 3.4: Schematic of the four rooms, drawn to scale, where the experiments were conducted (adapted from [scheurer'improving'nodate])

[scheurer'improving'nodate]

```
{
  "t": 1532944712.235510,
  "csi": [
    [{"r": 26, "i": -11}, {"r": 5, "i": -2}, {"r": -13, "i": -4} ],
    [{"r": 8, "i": -32}, {"r": 2, "i": -2}, {"r": -11, "i": 10}],
    [{"r": -24, "i": -25}, {"r": -3, "i": -3}, {"r": 4, "i": 16}],
    // and so on, for a total of 30 arrays, each of which
    // corresponds to one subcarrier
    [{"r": -23, "i": 23}, {"r": 24, "i": -4}, {"r": 15, "i": -13}]
  ]
}
```

Figure 3.5: Example of a CSI sample (adapted from [scheurer'wi-fi'2020])

Attribute	Data-type	Description
t	double	*NIX epoch timestamp, rounded to 6 decimal digits.
CSI	array[30][]	This is a two-dimensional array that consists of one array for each of the 30 subcarriers. Each subcarrier array consists of several complex number pairs antenna.
r	integer	The real part of a complex number
i	integer	The imaginary part of a complex number

Table 3.2: Data dictionary for the CSI samples

Data Processing

The data from one of the rooms, specifically "1.28A" depicted in Figure 3.6 with the voxels annotated, was analyzed. The necessary CSI data was stored in JSON files, and due to the formatting, loading the dataset was a time-consuming and storage-consuming task. To optimize the loading and assist in processing, the file was processed in chunks, and the data was processed and saved into a SQLite database in a format that would allow faster and more refined access.

The room had three links, one for each client, and the data for each link was given in a separate file, which was saved in individual relations per link in the database. The data, initially in the format shown in Table 3.2, was formatted to save in its respective link's relation, which had the format shown in Table 3.3.

The ground truth values were stored in a CSV file with the format shown in Table 3.4 and were gathered through a mobile app which the occupants utilized to record their upcoming activities. The information was stored in a separate relation in the database as shown in Table 3.5.

A relationship between the level of CSI data and the occupant's presence needed

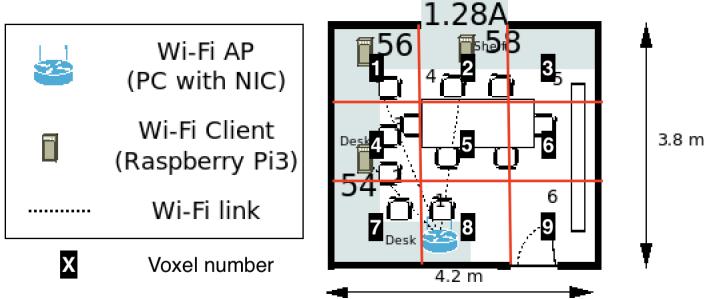


Figure 3.6: Schematic of the room "1.28A" drawn to scale (adapted from [scheurer'improving'nodate])

Column Name	Data format	Description
timestamp	Unix timestamp saved as a string	The timestamp of the data instance
antenna_no	int	Each client had three antennas. This attribute was to save the CSI from each antenna separately for each data instance.
csi	int[]	The array containing the complex number tuple for the relevant antenna for all subcarriers of the particular data instance.

Table 3.3: Data dictionary for the relation for each link

to be established to determine if an occupant was present in a particular area. To do so, an average CSI value or level at which an activity occurs could be identified. This level will be referred to as the threshold. Whenever the CSI data crosses this threshold, it would indicate that some activity had occurred in the area.

To establish a threshold for each activity, the timestamps of every instance of that activity were obtained from the ground truth relation. Since an activity can occur multiple times, there were multiple timestamp ranges for each activity. The data within each timestamp of every occurrence of an activity was then retrieved from each link's relation. For each occurrence of an activity, the average, minimum, and maximum values were calculated for each subcarrier and saved in a separate

Column	Data-type	Description
begin_time	double	Timestamp (*NIX epoch), rounded to 6 decimal digits that mark the beginning of the period the annotation corresponds to.
end_time	double	Timestamp (*NIX epoch), rounded to 6 decimal digits, marks the end of the period the annotation corresponds to.
room	string	The room that the annotation corresponds to, which would be either "G21", "G19", "260", "128a", or "NA" (for "Not Applicable").
oid	string	The occupant ID ("01" or "02").
label	string	This is the ground truth, i.e., the occupancy state of the room during the indicated period.

Table 3.4: Data dictionary for the ground-truth annotations

relation called 'threshold_per_occurrence'. The data dictionary for this relation is provided in Table 3.6.

Challenges with the dataset

The structure of the relation given in Table 3.6 and the calculation of averages, minimum and maximum values were under the assumption that an activity could be associated with a voxel. This would enable the identification of the threshold at which an activity occurs in a specific voxel. Consequently, each voxel would have its threshold for every activity, allowing us to detect occupancy at each voxel.

begin_timestamp	double	The starting timestamp of a labelled activity.
end_timestamp	double	The ending timestamp of a labelled activity.
activity	string	There were five activities: "Gone", "Approach", "Enter", "Mobile", "Stationary", "Exit", and "Departure". Each activity would indicate the room's occupancy.

Table 3.5: Data dictionary for the ground truth relation

It was discovered that a dataset, which was being analyzed to establish a threshold per voxel, was missing location information for occupants. This meant that identifying a threshold per voxel would not be possible. The dataset included five different activities, but only two activities, "Enter" and "Departure," could be associated with a voxel number (voxel nine, as seen in Figure 3.6). Therefore, thresholds could only be identified for these two activities for only a singular voxel. Unfortunately, the other two activities, "Stationary" and "Mobile," were said to occur randomly in the room, and thus, their impact on the CSI data could not be analyzed. The absence of this critical information made it difficult to gain valuable insights into how the CSI data would react to both stationary and active occupants. Upon examining two datasets, the latter of which contained data from only three links, it was discovered that it might be feasible to generate a dataset from scratch utilizing just two devices. This would eliminate the requirement for a communication protocol, which was the primary issue at hand.

3.2 Data Collection

3.2.1 Collection Setup

An empty room of 4.2m by 4.2m was divided into nine equally sized voxels, each voxel being 1.4m by 1.4m. The room was divided into nine voxels instead of a

begin_timestamp	double	The starting timestamp of the occurrence of the activity
end_timestamp	double	The ending timestamp of the occurrence of the activity
activity	string	The name of the activity
link_no	int	The threshold of all links were saved in the same relation. Each link was given an identifying number.
antenna_no	int	The number determining which of the three antennas.
subcarrier_no	int	The subcarrier index, ranging from 0-29.
min	double	The lowest CSI value encountered during the specific time period.
max	double	The highest CSI value encountered during the specific time period.
avg	double	The average of the CSI values during the time period.
voxel_no	int	The voxel at which the activity occurred.

Table 3.6: Data dictionary for the relation 'threshold_per_occurrence'

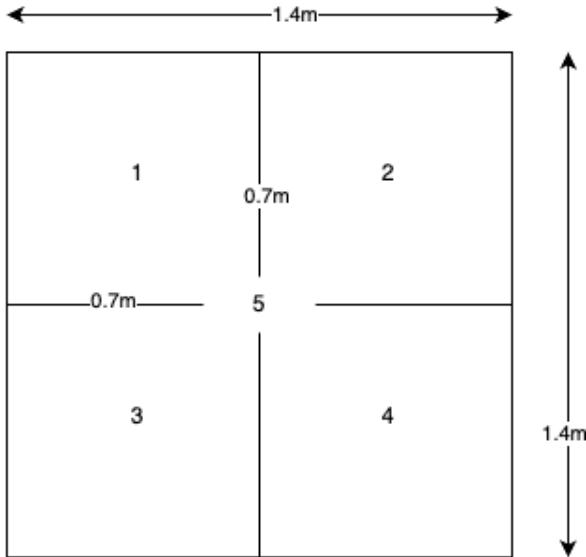


Figure 3.7: Voxel breakdown drawn with scale

larger count to make identifying thresholds easier. If the voxels were required to be smaller for a finer output, each was further divided into four smaller voxels, called "inter-voxels." Each inter-voxel size was 0.7m by 0.7m. The inter-voxels were numbered from 1 to 4 with the midpoint of the voxel being numbered as 5 as illustrated in Figure 3.7.

Two ESP32s were utilized for the experiment, where one of them was used as the transmitter, and the other one was used as the receiver, functioning as an AP. Neither of the devices had any external antennae attached to them. The firmware utilized in both ESP32 devices was obtained from an open source CSI collection tool, "ESP32 CSI Toolkit" [Hern2006:Lightweight]. All the configurations were maintained as per the tool's documentation, except for the tick rate, which was adjusted to 100 Hz.

The laptop running Ubuntu 22.04 LTS was connected to the receiver, which received the CSI data from the receiver through the serial port. The transmitter was placed on the voxel numbered "2", while the receiver was placed on the voxel numbered "8", as shown in Figure 3.8.

A web application was created to gather annotated CSI data that allowed the user to indicate their activity and position at any given time. The web page was hosted on the same laptop computer connected to the receiver and made accessible over the network. The user accessed the web page using a tablet device (iPad). The

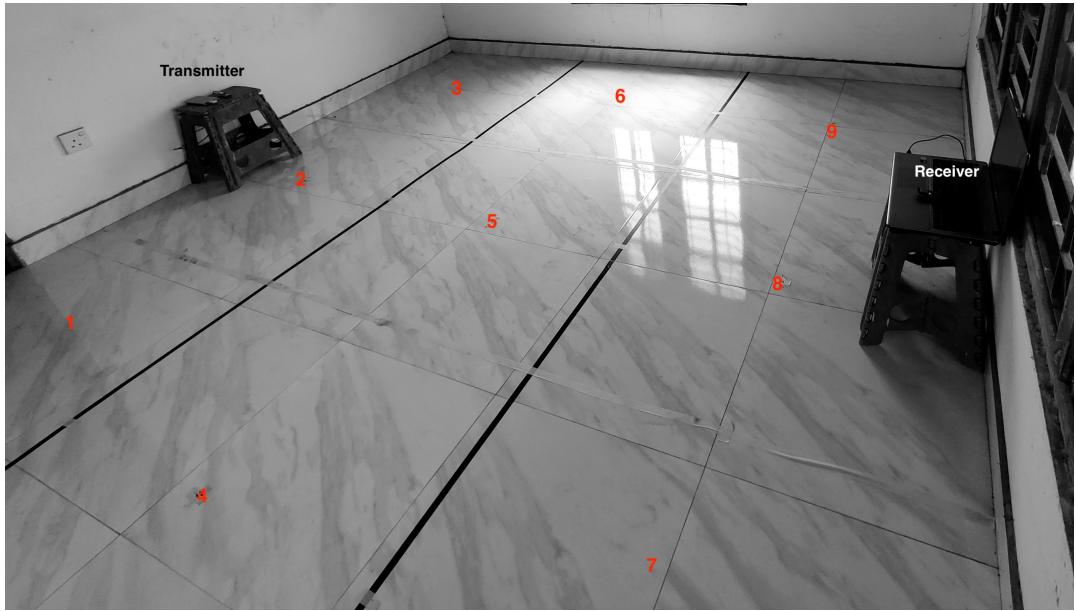


Figure 3.8: Schematic of the room that the data was collected in

web page was connected via a web socket to a Python server. Whenever the user updated their location or activity, the Python server added annotations to the CSI data read through the serial port. The annotations included the user's activity and location and were then saved in a file on the laptop computer. The setup's architecture is illustrated in Figure 3.9.

The Python libraries Pandas and Numpy were used for data manipulation and processing. SQLite3 was utilized for database operations, the math module for mathematical calculations, and Matplotlib's pyplot module for data visualization.

3.2.2 Collection Methodology

To establish thresholds for each voxel, it was necessary to collect configuration data, the CSI data obtained from occupants. To collect this data, a single occupant was asked to remain at the centre of each voxel for thirty seconds and perform one of the five activities:

1. "Standing-Stationary": standing motionless.
2. "Standing-Movement": standing still in the same location but moving their arms.
3. "SittingDown-Stationary": sitting down cross-legged with no movement.
4. "SittingDown-Movement": sitting down cross-legged while moving their arms.

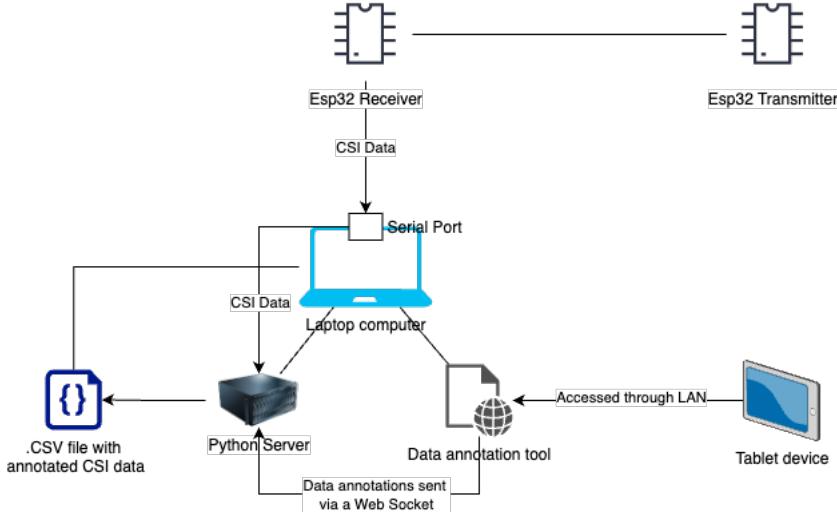


Figure 3.9: Architecture diagram of the data collection setup

5. "Walking": walking within a voxel.

In addition, a "None" activity was also recorded to represent the CSI data when no one was in the room. Each activity was performed for 30 seconds, and this resulted in 335 data points per activity per voxel.

To collect the testing data, the occupant walked around the room for a random amount of time while annotating their current location. The annotation was done to obtain ground-truth values. It is worth noting that for both configuration data and testing data, only one occupant was present in the room at any given time.

3.3 Development of the model

3.3.1 Overview of the algorithm

To estimate the occupied voxel for one row of input data, the algorithm would perform the following steps:

1. Set a threshold for each voxel based on activity and subcarrier. There would be 52×5 thresholds for each voxel, one for each subcarrier per activity.
2. For each of the 52 subcarriers in the input data, the algorithm would check which voxel's respective subcarrier threshold is exceeded, and it would do this for all five activities.
3. If the threshold is exceeded, the corresponding voxel on the plot will be coloured. This process would be repeated for all 52 subcarriers and each of

the five activities, resulting in 52 x 5 plots.

4. Finally, all outputs would be layered on top of each other to obtain the final output.

3.3.2 Challenges with the proposed algorithm

1. The algorithm proposed needed to address a crucial issue: determining the threshold of activity. Although calculating an average seemed the most straightforward solution, taking a single average for 335 data points would result in data loss.

Solution - Instead of calculating a single average value for all data points, represent the 335 data points in the configuration data as a vector.

2. Using a vector as a threshold helps eliminate the risk of having low accuracy when using a singular value as a threshold. However, it is important to select a vector size that is large enough to provide sufficient accuracy while being small enough to avoid the need for large configuration data.

Solution - To handle the configuration data, which contains 335 data points for each activity per voxel, there are two options: divide it into five blocks, each with 67 rows or divide it into 67 blocks, each with five rows. A vector size of 5 x 1 per activity per voxel per subcarrier is a good starting point. Therefore, the configuration data was divided into five blocks of 67 rows each. Each block was then averaged into a single value, resulting in 5 averages from the five blocks used to form the final vector.

3. The goal is to develop an algorithm that can use CSI data to estimate the occupant's location within a specific voxel. To achieve this, the input CSI data must be compared with a threshold vector to estimate the voxel. The main challenge is to find a reliable method for comparing the configuration vector and input data.

Solution - Calculate the distance between the vectors. By determining the configuration vector of the voxel with the shortest distance from the input vector, the algorithm can identify and colour the estimated voxel. Calculating the similarity between vectors is a common problem in computing.

3.3.3 Data Processing

A row from the raw data obtained from the serial port would contain the columns `type`, `id`, `mac`, `rssi`, `rate`, `sig_mode`, `mcs`, `bandwidth`, `smoothing`, `not_sounding`, `aggregation`, `stbc`, `fec_coding`, `sgi`, `noise_floor`, `ampdu_cnt`, `channel`, `secondary_channel`, `local_timestamp`, `ant`, `sig_len`, `rx_state`, `len`, `first_word`, `data`. Of these columns, only the `data` column would be of value as it contained the CSI data. The CSI data was in an array with 128 elements, with each pair of elements belonging to a subcarrier, thus representing 64 subcarriers. The pair of elements represented the real and imaginary parts values for each subcarrier, with every second element being the imaginary value.

3.3.4 Vectorization

The raw CSI data had to undergo several processing steps after being loaded into a Pandas data frame to obtain insightful data such as amplitude and phase per subcarrier. These steps were as follows:

1. Identifying the null subcarriers: According to the ESP32 CSI Toolkit's documentation, out of the 64 subcarriers, only 52 are data subcarriers. The indexes of the null subcarriers were provided in the toolkit's documentation and were removed from the dataset accordingly.
2. Separating the real and imaginary parts of each subcarrier: Each subcarrier had a real and imaginary part but was in the same array. It was necessary to extract each subcarrier's real and imaginary parts separately from the CSI array.
3. Calculating the amplitude and phase: The amplitude was calculated by taking the square root of the sum of the squares of the imaginary and real values. The phase was calculated by finding the arc tangent from the real and imaginary values using the Python `atan2` function. For each row of data, the amplitude and phase were calculated for each of the 52 data subcarriers and saved in separate columns in the data frame.

$$amplitude = \sqrt{i^2 + r^2}$$

$$phase = \text{atan2}(i, r)$$

Column	Data type	Description
voxel	int	The voxel number.
subcarrier	int	The index of the subcarrier
activity	string	The activity label
amplitude_vector	double[]	The 5×1 amplitude vector.
phase_vector	double[]	The 5×1 phase vector.

Table 3.7: Data dictionary for the relation ”distance_vector”

4. The amplitude and phase vectors were calculated for each configuration data file and saved in a relation named “distance_vector” in an SQLite database. The data dictionary of the relation is given in Table 3.7.

3.3.5 Heatmap Generation

A heatmap of a 3×3 grid would be generated to visualize the estimated voxel where the occupant is present. To estimate the voxel and visualize the output, the proposed solution is as follows:

1. 5 rows of input were read from the test data as the configuration vector size is 5.
2. The null subcarriers were removed, and the amplitude and phase of each subcarrier were calculated.
3. Separate input vectors for amplitude and phase were made for each subcarrier using their respective amplitudes and phases that were previously calculated. The size of each input vector was the same as the configuration vectors.
4. The Euclidean distance was used to calculate the vector distance between input vectors and corresponding configuration vectors. This calculation was performed for every subcarrier in each voxel, and the resulting vector differences were then saved in separate 3×3 Numpy arrays for both amplitude and phase. Since there were a total of 52 subcarriers, 52 arrays were generated for both amplitude and phase. Therefore, for five activities, there were a total of $52 \times 9 \times 2 \times 5$ Numpy arrays.
5. Heatmaps were created using Matplotlib’s pyplot module. They were layered

on top of each other using alpha blending with decreasing opacity, using the Numpy arrays as input.

Chapter 4 - Results and Evaluation

4.1 Results

The results discussed in the following are obtained from a subset of test data in which the occupant was “walking”, and the location was voxel number 1. As described in subsection 3.3.5, the vector size is five. Thus, a subset of five rows was extracted from the test data that corresponds to the occupant’s walking activity in voxel number 1. The voxel locations are provided in Figure 4.1. An overall evaluation is described in section 4.2.

As per the implementation, after plotting all possible activities of each voxel per subcarrier and layering them on top of each other, a scattered heatmap was generated, which estimated the occupant to be present in almost every voxel. The output generated with amplitude vectors is depicted in Figure 4.3a, while the output generated with phase vectors is described in Figure 4.3b. An example expected output of 100% accuracy is depicted in Figure 4.2.

The output accuracy was suboptimal due to the overlapping of all activities in the analysis. However, this was expected given that the test data examined was drawn from a subset where the occupant was solely walking. It is worth noting that all activities were accounted for in the analysis to ensure a comprehensive evaluation. The approach was necessitated by the fact that the algorithm would not perform activity recognition. As such, to account for any potential activity by the occupant, all activities were considered in the analysis. After plotting the activities individually and by considering the phase and amplitude vectors separately, the output for the phase vectors is shown in Figure 4.4. In contrast, the output for the amplitude vector is depicted in Figure 4.5.

Amplitude vs. Phase

The results obtained using the amplitude vectors indicate low accuracy, as none of the activities were correctly estimated to have only voxel 1. However, the outputs generated with the phase vector showed relatively better results. Specifically,

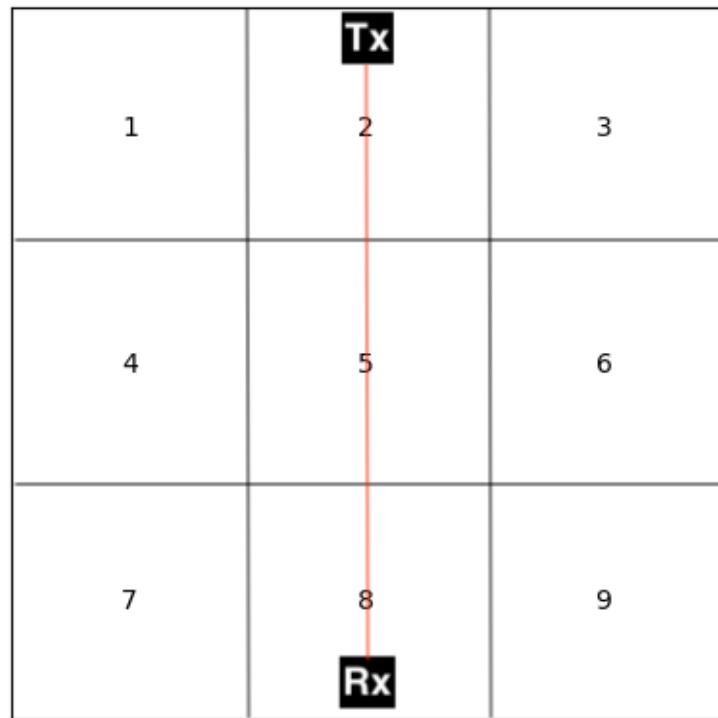


Figure 4.1: A grid with the marked voxels

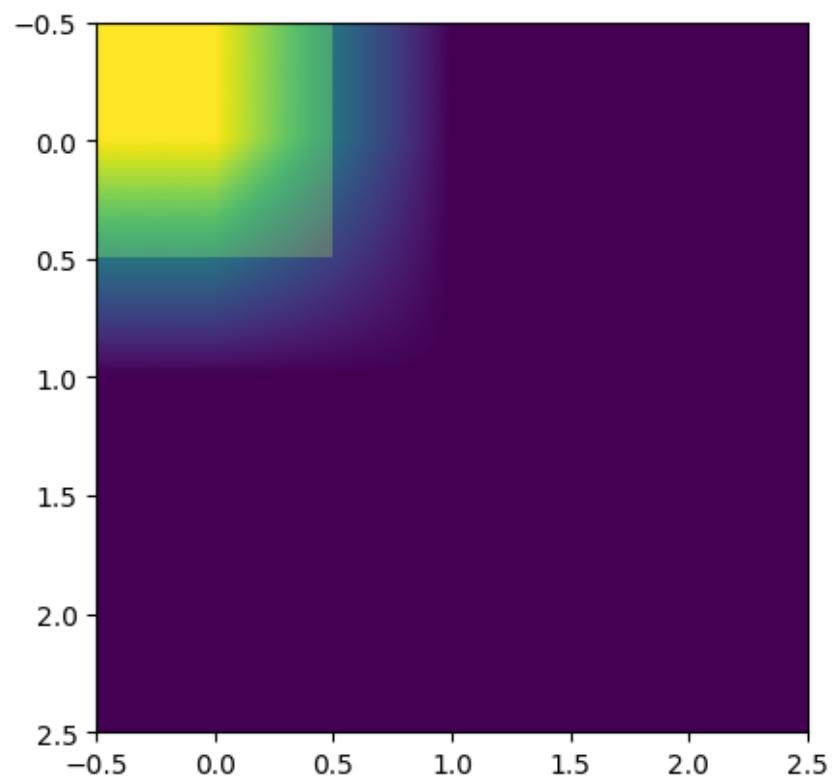
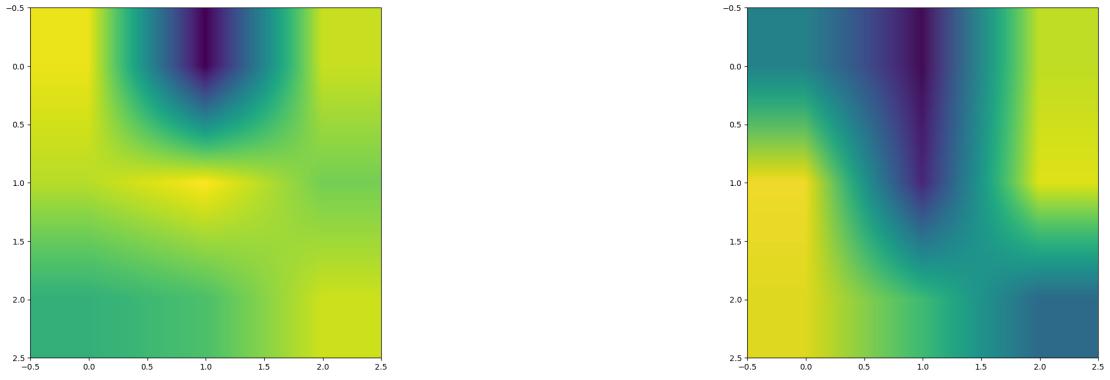


Figure 4.2: Expected output for when the occupant is in the first voxel



(a) Output generated with the amplitude vector

(b) Output generated with the phase vector

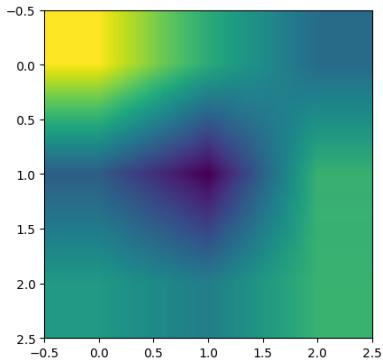
Figure 4.3: Output generated when all generated plots of all activities and subcarriers were layered on top of each other for when the occupant's actual location was in voxel 1

for two of the activities, **Standing-Stationary** (as shown in Figure 4.4a) and **Standing-Movement** (as shown in Figure 4.4b), voxel 1 was accurately estimated as the most probable voxel that the occupant would be in.

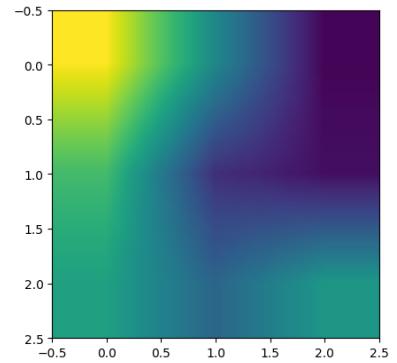
Based on research conducted by [phase is better], it has been found that the phase difference caused by movements remains stable in individual subcarriers. Furthermore, it is also observed that the phase differences are not affected by fluctuations caused by noise, interference in nearby correlated electromagnetic fields, or the multi-path effect [freeSense]. Since the phase is less susceptible to noise, it is expected to provide higher accuracy from the phase vector.

Activity Analysis

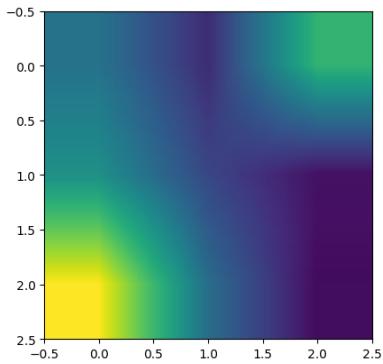
The amplitude of a signal is known to be highly sensitive to noise and movements, which can result in the generation of inaccurate configuration vectors. In contrast, the phase vector has been shown to be less susceptible to noise and is thus considered to be the more accurate parameter. After analyzing the phase vector outputs for different activities, it is evident that the “Standing stationary” and “Standing movement” activities have provided more accurate estimations than the others. While examining the voxels with the least possibility, the **Standing-Movement** phase vector outperformed the **Standing-Stationary** one. This outcome was ex-



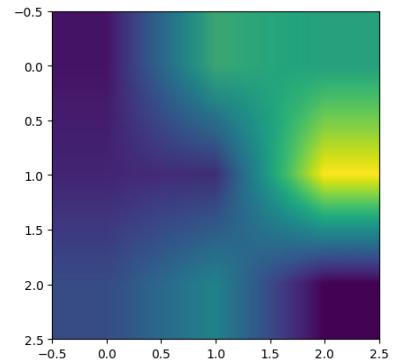
(a) Output generated with the phase vector for the activity
Standing-Stationary



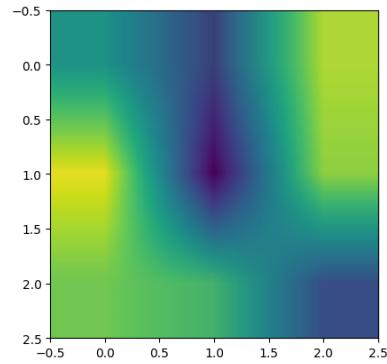
(b) Output generated with the phase vector for the activity
Standing-Movement



(c) Output generated with the phase vector for the activity
Walking

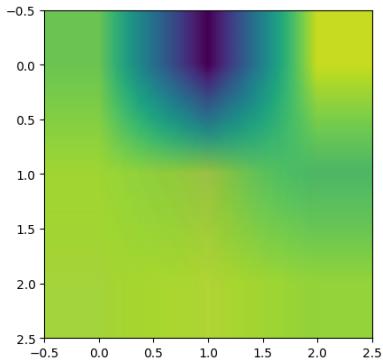


(d) Output generated with the phase vector for the activity
SittingDown-Stationary

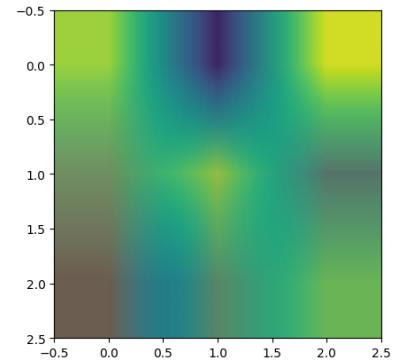


(e) Output generated with the phase vector for the activity
SittingDown-Movement

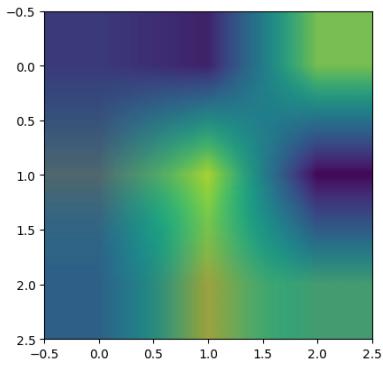
Figure 4.4: Output generated with phase vector for each activity separately for when the occupant was in voxel 1



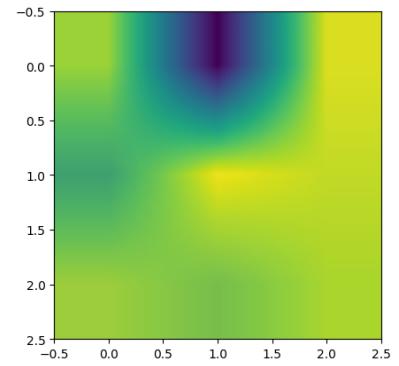
(a) Output generated with the amplitude vector for the activity **Standing-Stationary**



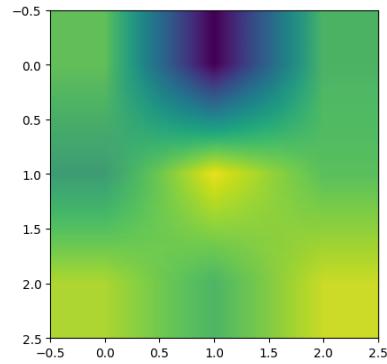
(b) Output generated with the amplitude vector for the activity **Standing-Movement**



(c) Output generated with the amplitude vector for the activity **Walking**



(d) Output generated with the amplitude vector for the activity **SittingDown-Stationary**



(e) Output generated with the amplitude vector for the activity **SittingDown-Movement**

Figure 4.5: Output generated with amplitude vector for each activity separately for when the occupant was in voxel 1

pected as the subset of the testing data used for generating the output was obtained while the occupant was walking. Since the data considered was from a small time frame, the overall walking activity was not completely captured. Hence, it is more likely to be similar to “Standing-Movement.” The time frame considered was small enough to capture only a part of the occupant’s walking activity and not large enough to encompass all the movements involved in walking. In order to enhance the accuracy, it is imperative to have prior knowledge of the activity to select a comparison vector as the parameter.

4.2 Evaluation

4.2.1 Evaluation Methodology

The algorithm produces heatmaps based on the vector differences between the vectors generated from the test data and the configuration data. However, evaluating visualizations can be difficult. Therefore, the evaluation process follows the following steps to determine the accuracy level per subcarrier.

1. Iterates over a dataset in batches of five rows
2. Calculates amplitude and phase for each subcarrier using vectorized operations
3. Computes Euclidean distances between the dataset and reference vectors for each subcarrier and voxel
4. Estimates the voxel for each subcarrier based on the minimum phase and amplitude difference separately
5. Counts the number of correct estimates (hits) and incorrect estimates (misses)
6. Determines the most frequently predicted voxel and its count
7. Evaluates the accuracy based on the hits and total estimates

4.2.2 Evaluation Results

The evaluation data was collected when the occupant was randomly walking around the room for a variable duration and standing still at certain locations for varied durations. However, to evaluate test data, only the data collected when the occupant was walking was considered. Therefore, the configuration vectors for the activities

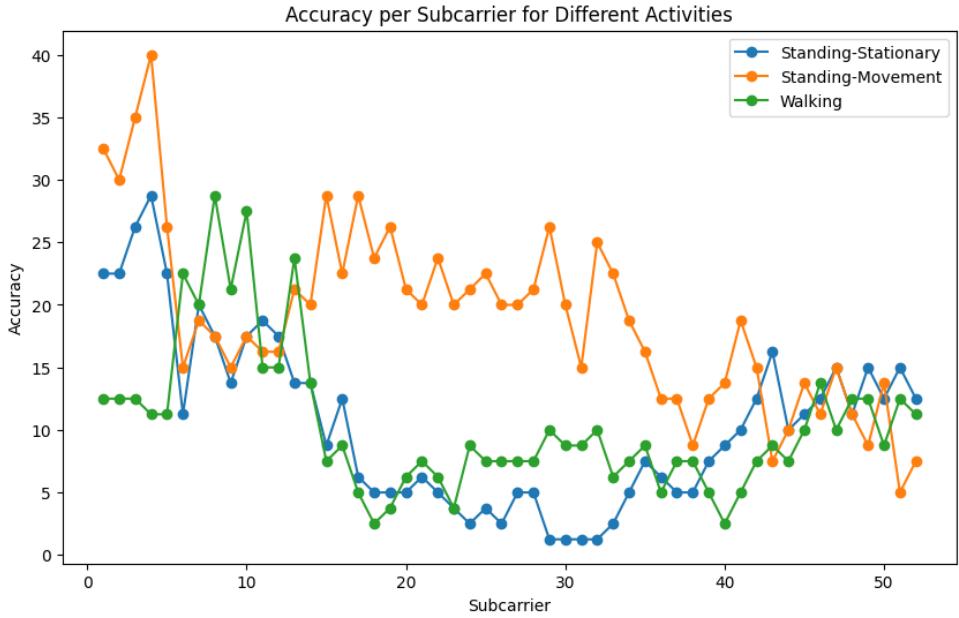


Figure 4.6: Accuracy for when vector difference was calculated using amplitude_vector

SittingDown-Stationary and SittingDown-Movement will not be considered for evaluation.

The accuracy of the amplitude and phase vectors is presented in Figure 4.6 and Figure 4.7, respectively. The accuracy percentage represents the number of occasions where each of the estimated voxels was accurate.

It is important to note that the accuracy levels presented in Figure 4.6 and Figure 4.7 represent the precision per subcarrier. Specifically, this refers to the number of times that the vector differences between the configuration vector and test vector accurately estimated the correct voxel. It is worth emphasizing that the accuracy of the visualization is contingent upon the number of subcarriers that accurately estimate the voxel for each instance of the test data. The heatmap generated is based on the number of subcarriers that estimate the same voxel. The plots from each subcarrier will be layered on top of each other to create the final output. Therefore, the more subcarriers that estimate the same voxel, the more highlighted it will be in the final output. It should be noted that when calculating the vector difference with the "phase_vector", the accuracies of the subcarriers are less erratic and neighbouring subcarriers display similar accuracies.

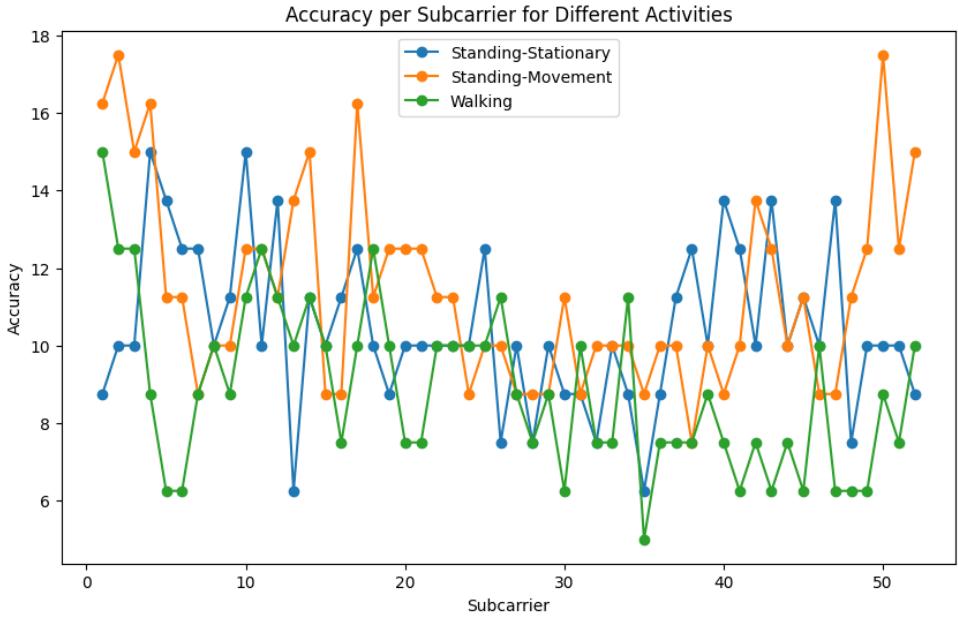


Figure 4.7: Accuracy for when vector difference was calculated using phase_vector

4.2.3 Improving Accuracy

The framework developed to provide a non-learning-based solution to indoor localization faced weaknesses when it came to accuracy. During the evaluation phase, it was noted that the following points were identified, which would assist in improving the accuracy.

1. Expanding the size of the configuration vector will result in a larger amount of data to be compared, which may improve the similarity between the test vector and the configuration vector. However, it is important to carefully select a suitable configuration vector size because if it is too large, too many changes will be recorded in a single vector, causing the accuracy to decrease. Therefore, additional analysis is recommended to determine the appropriate size of the configuration vector.
2. During the analysis of subcarriers, it was observed that certain voxels were frequently misidentified as specific other voxels. For instance, voxel 1 was frequently mistaken for voxel 8. This inconsistency might be attributed to the fact that when occupants are in specific voxels with similar spatial locations, their impact on the CSI data is indistinguishable. For instance, standing on two sides of a node may have an identical effect. By considering this factor,

the algorithm can be improved to increase accuracy.

3. It is possible to select the most prominent estimated voxel before generating the final heatmap. In case there are voxels that have not been estimated by many subcarriers, they can be considered as noise and removed. This can improve the accuracy of the output, as removing the noise subcarrier predictions can reduce interference.

Chapter 5 - Conclusion

This section presents the results of a research study that aimed to develop a non-learning-based approach for indoor localization using CSI. The discussion covers the limitations identified during the project, potential future advancements, and overall findings.

5.1 Conclusion on the Research Questions

5.1.1 Is it possible to develop a linear model for indoor localization using CSI-based RTI?

The primary reason for exploring the development of a linear model with CSI was the unavailability of publicly accessible datasets of adequate size to create an accurate learning-based model. Two datasets were evaluated, but neither of them had enough labelling information, making them unsuitable for the task of indoor localization. Therefore, to develop a model, a dataset had to be created with appropriate labelling. The dataset was formulated by linking each voxel with a CSI-based vector and calculating the vector difference. Consequently, the study was able to create a linear model that could perform the task of indoor localization successfully.

Additionally, the study provided insight into how amplitude and phase differences produce various outcomes and how different activities impact CSI in different ways. This could assist not only in indoor localization but also in activity recognition. In conclusion, the research question of whether such an approach was possible was successfully resolved by developing a linear model.

5.1.2 How to accommodate a sparse device setup for a linear model of CSI-based RTI?

The research dataset was formed with only two nodes deployed in the area of interest, where one node acted as the receiver and the other as the transmitter. This resulted in sparse coverage since only one Line-of-Sight (LOS) link was available in

the entire area. However, using CSI to develop a model allowed for the utilization of fine-grained information. This paved the way for the support of a limited number of devices, in this case, only two devices.

5.2 Contributions

The research conducted in this study has provided significant contributions and valuable insights, which are summarized below:

5.2.1 Main Contributions

1. The dataset in question contains CSI (Channel State Information) data pertaining to an occupant who is engaged in performing five distinct activities. Each data point in the dataset comes with annotations that detail specific locations and activities. This dataset would be a valuable resource for researchers and developers who are working on projects related to activity recognition and indoor localization.
2. A non-learning-based algorithm was developed to determine the location of the occupant within a specified area of interest. This algorithm is designed to leverage the CSI data available in order to determine the occupant's location accurately and provide insightful analysis and a framework that can be used for indoor localization with limited data.

5.2.2 Auxiliary Findings

In addition to the main contributions, the study also yielded some valuable insights, which are summarized below:

1. Voxel analysis involves examining the changes in the CSI data when an obstacle is present in different locations around a link. By analyzing the data in this way, we can gain insights into how the presence and location of obstacles affect wireless signal propagation and reception.
2. Activity analysis is a technique for studying how different activities, such as moving or standing still, affect the CSI data. By analyzing the data in this way, we can gain insights into how activities affect wireless signal prop-

agation and reception, which can be useful for optimizing wireless network performance in various scenarios.

3. Amplitude and phase are two important parameters of wireless signals that can affect the accuracy of CSI data. By comparing and contrasting the differences between amplitude and phase, we can gain a better understanding of how they affect the accuracy of CSI data, and how to optimize wireless network performance based on these parameters.
4. The size of the comparison vector is an important factor in determining the accuracy of CSI data. By analyzing the effects of different comparison vector sizes on the accuracy of CSI data, we can gain insights into how to optimize wireless network performance based on this parameter.

5.3 Limitations

The algorithm proposed to locate an individual within a particular area showed promising results, but it had some limitations.

1. The experiments were conducted in a relatively small room, and only one link was sufficient to detect changes in CSI data when the occupant was in any voxel. However, if the room were larger, there could be blind spots where the occupant could not be detected if there weren't enough links covering the area of interest. This study aimed to develop a linear model with limited data and a sparse network of devices, and considering the performance of the model in locations with large dimensions was considered out of scope.
2. To accurately detect an occupant's location, the activity they are engaged in must be known since each activity has its comparison vector. Without knowing the activity, comparing to every possible vector would result in heavy inaccuracies.
3. The devices' placement in the experiments created a line across the room, dividing it into two. Different placements of devices could have provided different results, but the study considered only one particular device placement.
4. Splitting a room into nine voxels may not be sufficient to obtain an acceptable level of accuracy in a larger room. In a smaller room, nine voxels may be too large to differentiate a change from each voxel. This study did not take this

factor into account.

5. The room could be divided into elliptical voxels instead of squares, but this study did not consider voxels of a different shape when developing a solution.
6. Noise was not taken into account, which can affect the accuracy of the amplitude vector. However, the effect of noise can be minimized using signal processing and regularization techniques.
7. Increasing the tick rate of the devices can result in collecting more CSI data points for the configuration data. This can help in capturing more accurate movement data, but the effect of different tick rates was not considered during this study.

This study aimed to explore the possibility of developing a linear-based model that could perform indoor localization even when there is a low number of devices in the area. The objective was motivated by the realization of a gap between the potential of RTI in practical applications and its current progress. The proposed solution can be further improved by addressing the constraints discussed above. By supporting a random placement of a low number of devices, RTI would benefit various fields, including military, smart home applications, healthcare, and rescue operations since the major hindrance when applying such a solution in real-world applications is its lack of tolerance for robustness. Unlike learning-based models, which require a large training dataset and configuration phase in the deployment area, non-learning-based models could be developed with far fewer data and still tolerate robustness in the device topology.

In conclusion, the development of a linear-based model for indoor localization presents promising prospects for enhancing real-world applications across diverse fields. With further refinement to address existing constraints, such advancements hold the potential to revolutionize industries and improve efficiency, safety, and performance in various scenarios.