Politecnico di Torino

Master Thesis

Indoor positioning using BLE for monitoring eldest patients

Author:
Ahmadreza Ghadirzadeh

Supervisor:
Prof. Marco Piras

A thesis submitted in fulfilment of the requirements for the Master Degree ICT for smart societies

in the

Collegio di Ingegneria elettronica, delle telecomunicazioni e fisica
Dipartimento di Elettronica e Telecomunicazioni

July 14, 2021

POLITECNICO DI TORINO

Abstract

Faculty Name
Dipartimento di Elettronica e Telecomunicazioni

Master Degree ICT for smart societies

Indoor positioning using BLE for monitoring eldest patients

by Ahmadreza Ghadirzadeh

Thanks to advances in IoT (Internet of Things) related technology, requests for highly accurate indoor positioning systems and services remarkably increased. Indoor positioning defined as a system that determines the exact location of objects in a closed structure such as supermarkets, hospitals, universities, airports.

Access to the location of objects, people, and services is one of the basic requirements in intelligent environments.

The proliferation of smartphones and other wireless devices over the past few years have led to the use of a wide range of services, including internal location system services. Indoor positioning is the process of locating a device or a user indoors. Internal locators have been extensively studied in recent decades and are mainly used in industrial environments, wireless sensor networks and robotic systems.

Every year, much media attention is given to Indoor Positioning Systems (IPS). Similarly, Bluetooth low energy (BLE) is extensively used for IPS because of its low cost and ease of use. In addition, techniques such as trilateration, triangulation, and fingerprinting are widely studied in IPS.

Here, we proposed to use the state-of-the-art machine learning approach for Indoor positioning in a hospital with taking advantage of utilising Artificial Intelligence (AI). This paper proposes a BLE based machine learning location and tracking system for indoor positioning.

Acknowledgements

First and foremost, I would like to thank God for giving me the strength, knowledge, opportunity and health to undertake this project and persevere and complete it satisfactorily.

I want to thank the Polytechnic of Turin, who opened the doors to study and complete the ICT for smart societies Master degree. To all the professors who were part of my professional development throughout the university stage.

I will thank my advisor Marco Piras for his presence, collaboration and dedication that helped me to complete this project correctly.

To the CWS Team, who hugged and helped me with their knowledge and presence to achieve the goal. I will thank all the members Luca Passini, Andrea Romanello, Stefano Galati, and Marco Mancini.

To my family and colleagues, who day by day gave me their support and encouragement to move forward and be the person I am.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	**A**rtificial **I**ntelligence
AP	**A**ccess **P**oint
BLE	**B**luetooth **L**ow **E**nergy
CAGR	**C**ompound **A**nnual **G**rowth **R**ate
GPS	**G**lobal **P**ositioning **S**ystem
IMU	**I**nertial **M**easurement **U**nit
IPIN	**I**ndoor **P**ositioning and **I**ndoor **N**avigation
IOT	**I**nternet **O**f **T**hings
LOS	**L**ine **O**f **S**ight
NN	**N**eural **N**etworks
PDF	**P**robability **D**ensity **F**unction
RNN	**R**ecurrent **N**eural **N**etwork
RP	**R**eference **P**oint
RSSI	**R**eceived **S**ignal **S**trength **I**ndication
STA	**S**tation
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

In recent years, the Smart Environments market has had remarkable growth nationally and globally, in the case of Italy; 2018 has witnessed a growth of 52 % concerning 2017. Moreover, due to the development of technology in positioning and the pervasiveness of smartphones in today's society, we see that people, instead of using paper maps and atlases of Global Positioning System (GPS) signals and digital maps to determine their position, use outdoors. Nonetheless, when we enter indoor environments, GPS signals do not work. Instead, maps and signs mounted on the walls as the primary reference for internal navigation in hospitals, universities, museums, shopping malls, and other gigantic structures.

With the increasing advancement of emerging technologies in pervasive computing, positioning systems that can be prepared indoors have attracted more attention and interest. Indoor positioning systems, along with the positioning alert of the person in a specific place and in real-time, accordingly are among the primary applications of comprehensive calculations. For example, positioning elderly patients in a hospital that can be predicted in the quickest duration of time (any movement of the patient) is exemplary, and as a result, the nurses will be on-site immediately in the occurrence of a problem.

As mentioned above, this project emerges from the initiative of CWS Srl company, with expertise in ICT technologies based in Turin, in collaboration with the Department of Environment, Land and Infrastructure Engineering (DIATI) of the Polytechnic of Turin (POLITO). This project aims to monitor the elderly of a hospital located in Bolzano, Italy, called MELITTAKLINIK; we will describe the structure of the hospital (Test site) in the chapter 3.

The main focus of this thesis is related to the Back-end and the core of the system with various experiments performed on various sensors and Antennas, including simulation of Antennas on Raspberry pis and their comparison to having a better result for the system plus more accurate positioning in the hospital.

The core of prediction and positioning in this thesis is using machine learning algorithms and the Neural Networks (NN) methods that we will discuss in Chapter 5. In addition to a Machine Learning (ML) model, an Artificial Intelligence (AI) system has implemented to increase the positioning accuracy to prevent errors and noise in the system.

Design and implement an intelligent indoor positioning system for a five floors hospital that can position 180 patients in real-time to inform nurses of any problems or emergencies. Additionally, to evaluate the movement and behaviour of patients for doctors and nurses who can use this information to have better control over patients.

Obtaining position information in indoor positioning is concretely challenging for several reasons: indoor environments are more intricate as there are sundry objects (such as walls, equipment, and people) that reflect signals and lead to multipath and delay quandary.(Saleem, 2019)

In the hospital, we need more precision for the system because there is no possibility of high error in emergencies, and a person may be in a dangerous condition.

As the distance from the source increases, the position accuracy decreases. Moreover, The indoor obstacles may cause different attenuation coefficients for RSSI (Received signal strength indication) signals. Therefore the establishment of a precise indoor propagation model is very arduous.

This thesis subdivided into eight separate chapters. The First Chapter deals with the "State of Arts" and illustrates the results of previous research and reported in the international technical-scientific literature. In the second chapter, the aim of the study presented along with study cases, technologies used, and the methodology to be followed (2).

In the third chapter (3), we talk about the site of the project and the technologies and equipment used in the experiments.

In the fourth chapter, the actual study is the Training and Experiments (4). The first part of this section refers to the Training study case. This section discusses how to store data for the Training section and complete the training for each hospital region.

In the second section, the Experimental Results study case exhibited. This section defines several experiments, including the effects of the Transmit Power (TX) parameter on the RSSI signal. Additionally, it compares the performance of the Antenna and the Raspberry pi (s). Plus, RSSI signal thresholds in each Zone and each hospital floor further examine variations over the RSSI signal over a long time.

The fifth chapter is related to Data analysis and Modeling (5), which at the beginning of this chapter deals with how to clean data and the importance of data cleaning in machine learning algorithms. Then the results obtained on neural network models are discussed.

The sixth chapter deals with Testing (6), which trades with how to store data in a database and synchronize the time plus steps that the positioning system goes through to obtain the final prediction. In addition, it deals with errors that may occur in the system, and the system may have problems.

Chapter 7 first discusses the Artificial Intelligence (AI) algorithm in this project to increase positioning accuracy. Then, the other section discusses the multi-threaded system in the project.

Finally, in the eighth chapter, the results obtained are discussed, conclusions from the present work exhibited, and recommendations for future works based on the present experience presented.

1.1 State of Arts

Different indoor positioning systems are depending on the type of signals used. All of them have their limitations, and none of the options is perfect. Below, we explain some of the most common technologies.

The global Indoor Positioning and Navigation (IPIN) Market by Component, Technology, Application and End-Use Industry: Global Opportunity Analysis and Industry Forecast, 2018 - 2025," the global indoor positioning and indoor navigation (IPIN) market was valued at $ 2,642 million in 2017 and projected to reach $ 43,511 million by 2025, growing at a Compound Annual Growth Rate (CAGR) of 42.0 % from 2018 to 2025.(Lanjudkar, 2018)

Especially in Europe, Indoor Positioning would witness market growth of 40.8 % CAGR during the forecast period (2017 – 2023). Technological advancements have primarily benefited the market participants within the business landscape.

Indoor Positioning and Navigation (IPIN) Market Growth Drivers are among the main drivers in the global Indoor Positioning, and Navigation Market is the growing adaption of these technologies in many industries like healthcare and logistics. There is a high return on investment using these technologies as using these indoor positioning and navigation technologies will result in low operational costs, thereby maximizing efficiency and profit.(Lanjudkar, 2020)

Chapter 2

Introduction of solution for Indoor positioning

This chapter discusses the relevant technologies for providing an indoor positioning system for movement patients in the hospital and discusses the differences between various technologies in this domain.

2.1 Types of IPS in terms of technology

Currently, the technologies used in the world for the indoor positioning system include Radio Frequency Technologies, Ultrasound System, Infrared Light, Inertial Measurement Unit (IMU) and Zigbee.

Different indoor positioning systems are depending on the type of signals used. All of them have their limitations, and none of the options is perfect. Below, we explain some of the most common technologies.

2.1.1 Radio Frequency Technologies

RFT is the most popular IPS based on radio frequency signals. There are two reasons for this. First, some systems reuse deployed technology, including WIFI networks, Bluetooth, and mobile phones; this drastically reduces the cost of deployment and makes the technology easily accessible to a more significant number of people (e.g. by installing an app).

The second one is that, as these signals can traverse obstacles, they can work in real-world settings where obstacles are unavoidable, including commercial areas. In some cases, these systems can also provide more extensive coverage. Below, The most used IPS systems using radio frequency are Radio Frequency Identification (RFID), WIFI & Bluetooth and Ultra-Wideband (UWB).

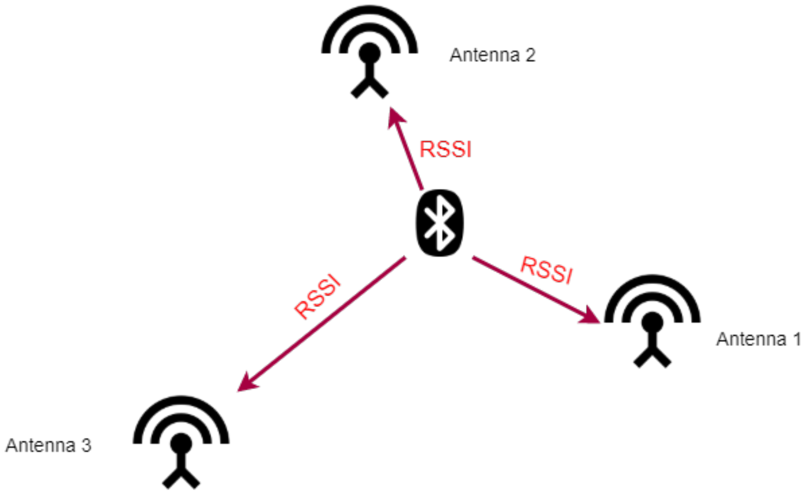


Figure 2.1: Bluetooth-Based Indoor positioning

2.1.2 Ultrasound System

Ultrasound systems use sound instead of light. "Ultrasound wave is a mechanical wave that is an oscillation of pressure transmitted through a medium." (Al-Ammar, 2014). It does not conflict with electromagnetic waves and does not require a line of sight. The system needs a set of anchors and a tag. It uses Time of Flight, that is, the time required by sound to travel from an anchor to a tag or, conversely, to predict the distance between them. As illustrated in figure 2.2 once to have at least three distances available, the position can be computed using trilateration.

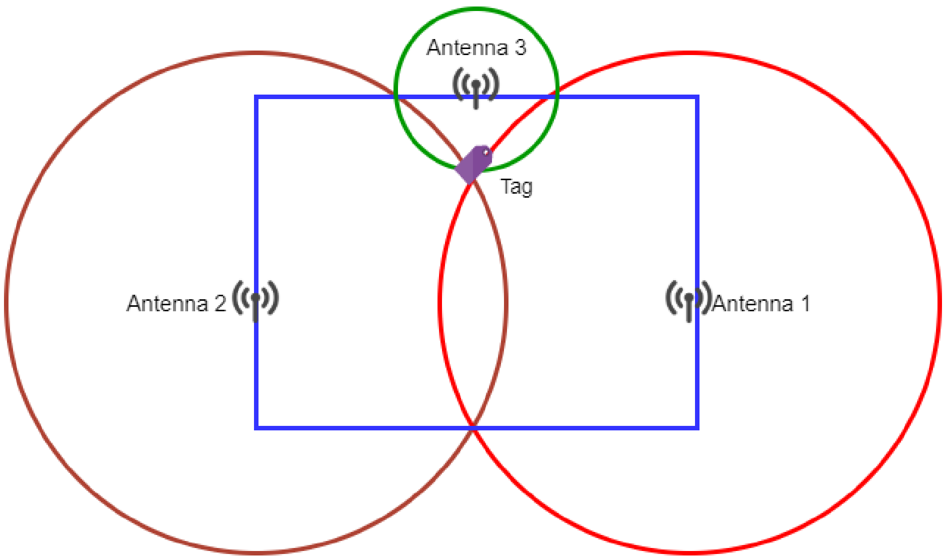


Figure 2.2: The basic concept is to find the intersection of the three circles defined by the three distances between the Antennas and the Tag.

2.1.3 Infrared Light

These systems used infrared light pulses like a TV remote control. They require an unobstructed Line of Sight (LOS) between the anchor and the tag. This type of system can be used as a very reliable room detector. Since light cannot traverse walls, a tag cannot detect light from an anchor without being in the same room. For precise localization, they require installing many anchors and can struggle due to the low quality of the signal strength measurements required to compute the position from multiple anchors. A similar approach is currently used in VR headsets to locate the person in a room, using multiple light sources and reflecting objects. (Bitbrain, 2017)

2.1.4 Inertial Measurement Unit (IMU)

Inertial Systems inform about the relative positioning of the tag with the combination of several sensors, such as accelerometer, magnetometer, and gyroscope, in a tiny module. These sensors help determine the direction and orientation of the movement. Combined, they can provide a prediction of the corresponding action concerning the previous position. This information usually obtained by combining all available signals using algorithms such as Dead Reckoning. Dead reckoning is "the process of estimating known current position based on the last position and incrementing that position based on known or estimated speeds over elapsed time." (Farid, 2013)

One of the benefits of this technology is that it does not need anchors in the environment. Regrettably, the accuracy of this type of system is usually poor, as the error accumulates over time and can be on the order of meters after just a few seconds.

For that reason, it is common to use it in combination with other technologies to smooth the results and reject outliers. Additionally, identifying movement can be helpful to recognize whether the tag and, therefore the participant, has stopped.

2.1.5 Zigbee

Zigbee indoor positioning method on the distance measurement technology is based on the RSSI (Received Signal Strength Indication) and has widely used in positioning and navigation. An analysis has been conducted using Zigbee wireless network system, and the results confirmed that the positioning accuracy of the Zigbee system had been improved, and the average error is 1.57 m.(Guo, 2016)

To conclude which technology is fitting to develop our IPS, some features have to be taken into account and balanced for the required specifications of the project.

- **Accuracy**

  Defined as "the average Euclidean distance between the estimated position and the true position." (Alhadhrami, 2014) Accuracy is acknowledged as the main feature in most indoor mapping systems and the most challenging to develop. The best solutions usually require ad-hoc deployments, which increase cost and complexity. For that reason, when accurate positioning is not significant, cheaper and, simpler technologies are used.

Table 1 presents the best accuracies that can typically be obtained with the systems described previously.

	WIFI	**Infrared**	**Bluetooth**	**Zigbee**	**UWB**	**RFID**	**Ultrasound**
Accuracy (m)	1-10	5-10	2-15	1-5	0.1-1	0.5-2	0.01-0.1

Table 2.1: Technologies Accuracy in meters.

- **Coverage**

  Coverage is the essential characteristic of approaching accuracy. Coverage is the area where the location information is available. The coverage of IPS usually ranges from a room to scalable systems that can cover multi-room environments or large areas such as warehouses or commercial centres. Often there is a trade-off between coverage and accuracy, where technologies with more extensive coverage typically imply smaller precision.

- **Cost**

  The cost includes deployment costs, operational costs, and maintenance during the lifetime of the system. Some technologies require fixed installations, while others are mobile or can use the existing infrastructure. Those technologies using trilateration usually require calibration, which is costly in time, especially if the system's installation is not permanent.

Thanks to the invention of the Bluetooth positioning technology, it provides us accuracy down to the meter or centimetre level, depending on configuration and enhancements. For instance, it is more suited for applications that do not require precise locations. Further, the BLE based positioning technology can deliver adequate results, and its cost and ease of deployment make it an ideal choice for most applications.

The indoor positioning depends on Bluetooth Low Energy (BLE) beacons installed on objects, walls, ceilings, and other places where they transmit radio signals at planned intervals. Devices within the emission area can then detect the signals, which helps establish if the two (emitter and receiver) are within the range of each other.

Although one Beacon is enough to establish an object's presence, it cannot pinpoint the specific location. Regularly, the location accuracy increases with the number of beacons. After installing that two objects are near each other, the Bluetooth location services can use the Received Signal Strength Indicator (RSSI) to estimate the distance between them.

2.2 RSSI

RSS is the power of the received signal estimated by a receiver's RSSI circuit. Frequently, RSS is reported as estimated power, i.e., the squared magnitude of the voltage. RSSI is a reception parameter easily accessible from the manufacturer's driver. This quantity measures signal power in dBm (decibels concerning one mW) and goes typically from -40 dBm (near the transmitter) to -95 dBm (on the limits of the coverage area). The noise floor in a standard indoor environment is around -100 dBm. However, to estimate the distance from an analytical propagation model, we need to compute the path loss due to the separation between the transmitter (STA) and receiver (AP).(Aguilera, 2017)

It is straightforward to transform the amount of power to distance in inaccessible environments (free space) such as environments where we do not have corridors, walls and other materials that can increase or decrease the signal strength. Therefore, as we have walls and passages in the hospital, we can take advantage of machine learning algorithms and artificial intelligence to solve this problem. We will go into more detail in later chapters.

Chapter 3

Test Site

In this chapter, we will discuss the project site and the technology used in this project.

This project aims to monitor the elderly of a hospital located in Bolzano, Italy, called MELITTAKLINIK, a private hospital in Bolzano. The structure consists of five floors with 180 beds in total, of which 90 dedicated to post-acute rehabilitation and intensive neurorehabilitation and ten dedicated to patients in acute treatment after surgery (orthopaedic, neurosurgery, plastic), a clinic that boasts a salt department for surgery.

Inside the structure, there are two therapeutic areas (a pool and a gym) which the latter equipped with "fitness" equipment for sports rehabilitation and 25 clinics for rehabilitation, Ergotherapy, and speech therapy accessed by patients, doctors, and assistants.

In addition to this, there are 80 dedicated beds in the MELITTA CARE ward, a residence facility for the elderly that can accommodate up to 40 Alzheimer's patients.(Cumino, 2021)

3.1 Antennas

3.1.1 Cambium antenna

Cambium antennas (cnPilot™ e600) installed at specific distances in corridors and projected locations.(figure 3.1).The figure 3.2 is a screenshot of some part of the map of the first floor in the hospital, and the point with the green marks are the antennas installation location.

Figure 3.1: Cambium antenna (cnPilot™ e600). Figure reproduced from (cambiumnetworks, 2021)

As depicted in the figure 3.2, the location of the Cambium antennas, which is shown by the green signs, is in the passways of the hospital. The aim of placing them in the passway is to take advantage of the Radio Frequency Technologies, in other words, to recognize the Beacons with the antennas placed in the passways.

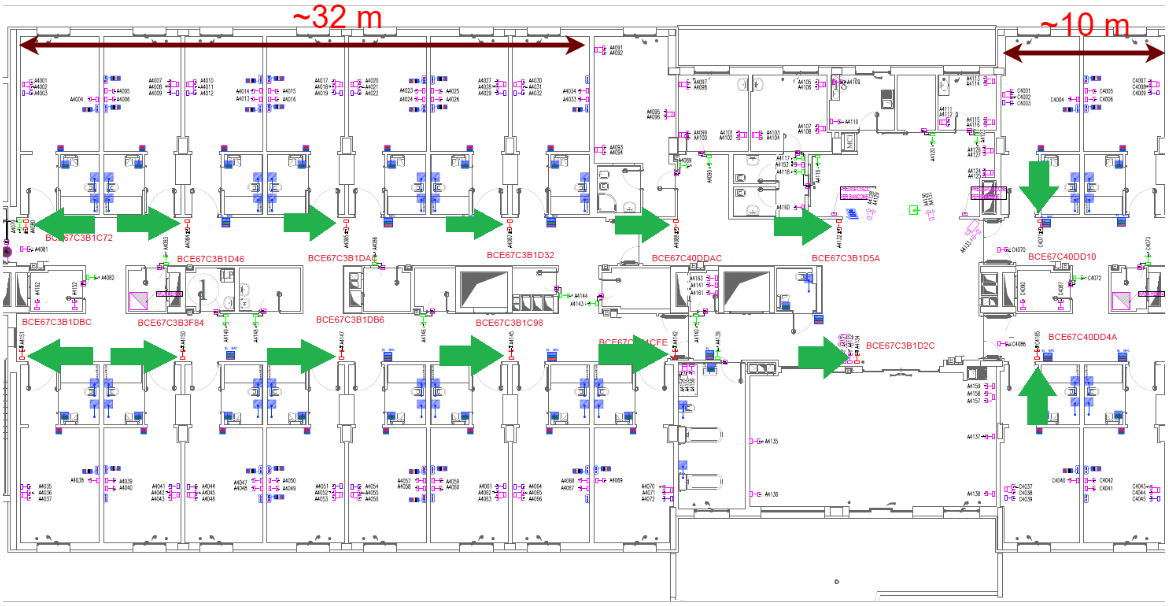


Figure 3.2: Part of the first floor Map of Hospital with Cambium antennas (Green signs).

3.1.2 Raspberry Pi

According to the experiments that performed in more detail in the Experimental Results (4.2) chapter, in the farthest location of some rooms, simultaneously we stayed outside the threshold of corridor antennas, and we had two solutions to solve the problem. The first solution is to add a Cambium antenna in the rooms, and the second solution is to install a Raspberry Pi in the rooms. Due to the difference in the price and the place where it can be installed.

The Raspberry Pi(s) are installed on the back of the TVs as shown in the figure 3.3 (Raspberry pi symbols) and send all the devices in a visible range to the server using a node-beacon-scanner module.

The node-beacon-scanner is a Node.js module that allows us to scan BLE beacon packets and parse the packet data. This module supports iBeacon, Eddystone, and Estimote. In addition, we designed the Raspberry Pi (s) to automatically restart the system to restart if turned off for any cause.

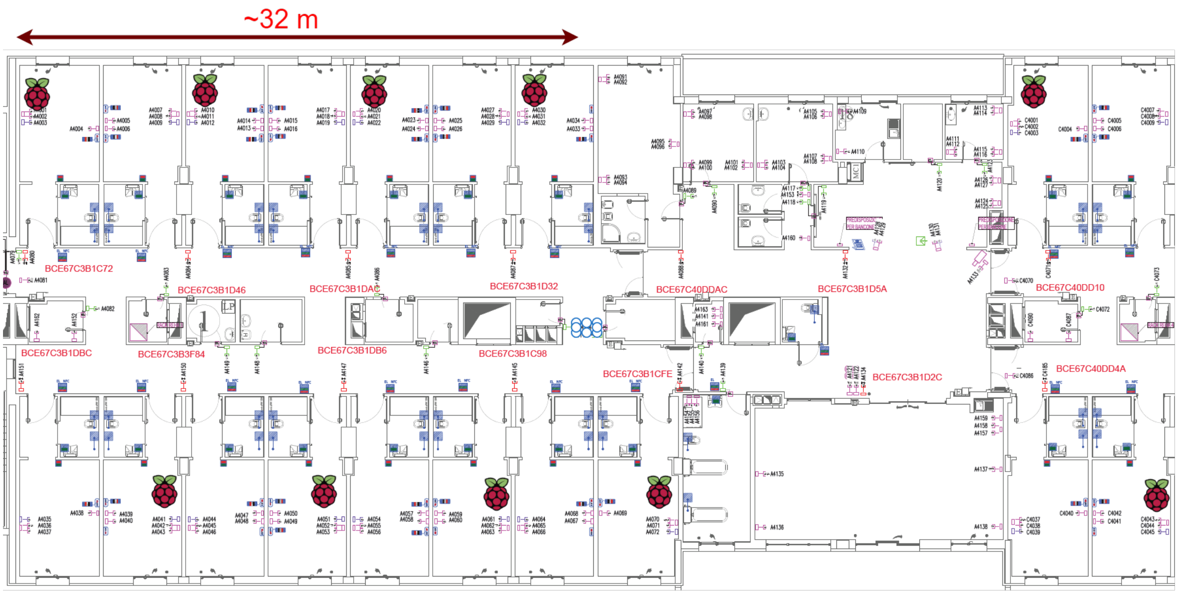


Figure 3.3: Part of the first floor Map of Hospital with Raspberry Pies

3.2 Disk Beacony

Disk Beacony is a smart BLE (Bluetooth Low Energy/version 4.2) Beacon, optionally available with an accelerometer and temperature and humidity sensor. The new plastic housing, designed with press-fit closure, facilitates battery replacement. It guarantees high performance (up to 200 meters reading distance) in addition to a very affordable cost; it has a replaceable CR2032 battery.(figure 3.5).

We attached them to bracelets for a more convenient usage for patients (Alzheimer disease). (refer to the figure 3.4)

Figure 3.4: Wristband Beacon. Figure reproduced from (Global-Tag, 2021)

The Beacon's parameters (such as broadcast interval, Tx power, friendly name, Major, Minor, UUID and other available services) if its parameters changing the system will experience a change, that in the experimental results (4.2) chapter, we examine some of these parameters.

Figure 3.5: Smart BLE Beacon Device. Figure reproduced from (Global-Tag, 2021)

Chapter 4

Training and observation of IPS

The development of our indoor positioning system is divided into two phases: a training phase (offline phase) and a testing phase (online phase). In the training phase, RSSI readings at each predefined RP are collected into a database.

In the Testing section, the data is stored in the same way as Training, except that it has a different format. Furthermore, by using artificial intelligence algorithms, we increase the accuracy of the positioning system. Chapter 7 presents more details.

4.1 Training

As mentioned in the previous chapter, converting an RSSI signal to distance is not achievable in environments with different materials, so we have to implement the system using machine learning algorithms in the next step.

As illustrated in figure 4.1, we see the workflow of storing data in the database for the training part. First, all antennas and Raspberry Pi(s) at an asynchronous time send RSSI signal that is visible from all BLE beacons such as mobile phones and other devices to the server, then by using a filter, the existing one sends all the demanded BLE Beacons to Logstash and, after that Logstash is transformed to a format which we need with time (Unix time) and location index.

We set the BLE Beacons in a different room for the training section, receive various signals, and store them in Elasticsearch. We do the same for other rooms and places in the same way to have the data for all the hospital rooms.

Subsequently, the data will be stored in Elasticsearch. Still, since the packet received from antennas and Raspberry Pi(s) could be so much from different antennas and different Beacon to avoid losing any record, we need a Message broker for translating a message to the Elasticsearch in this regard, the software of the Message broker that we use is RabbitMQ for generating a queue and not to miss messages.

- **Logstash**: Logstash is a lightweight, open-source, server-side data processing pipeline that collects data from various sources, transforms it on the fly, and sends it to the desired destination. It is most frequently used as a data pipeline for Elasticsearch, an open-source analytics and search engine. Because of its tight combination with Elasticsearch, powerful log processing capabilities, and over 200 pre-built open-source plugins that can help index data easily, Logstash is a popular choice for loading data into Elasticsearch.(aws.amazon, 2019)

- **Unix time**: The UNIX timestamp is a way to track time as a running total of seconds. This count starts at the Unix Epoch on January 1, 1970, at UTC. Therefore, the UNIX timestamp is merely the number of seconds between a particular date and the Unix Epoch.(unixtimestamp, 2015)

- **Elasticsearch**: Elasticsearch stores data as JSON documents. Each document correlates a set of keys (names of fields or properties) with their corresponding values (strings, numbers, Booleans, dates, arrays of values, geolocations, or other data types).

  In addition, Elasticsearch uses a data structure called an inverted index, which is designed to allow quick full-text searches. An inverted index lists every unique word in any document and identifies all of the documents each word occurs.

  During the indexing process, Elasticsearch stores documents and builds an inverted index to make the document data searchable in near real-time. Indexing is initiated with the index API, through which one can add or update a JSON document in a specific index.(elastic, 2019)

- **Message broker**: A message broker is an intermediary computer program module that translates a message from the sender's formal messaging protocol to the receiver's formal messaging protocol. First, Message brokers are elements in telecommunication or computer networks where software applications communicate by exchanging formally defined messages. Second, Message brokers are building blocks of message-oriented middleware (MOM) but typically are not replacing traditional middleware like MOM and remote procedure call (RPC).(wikipedia, 2020a)

- **RabbitMQ**: RabbitMQ (Function A.3) is open-source message-broker software that initially implemented the Advanced Message Queuing Protocol (AMQP) and has since been extended with a plugin architecture to support Streaming Text Oriented Messaging Protocol (STOMP), MQ Telemetry Transport (MQTT) and other protocols.(wikipedia, 2020b)

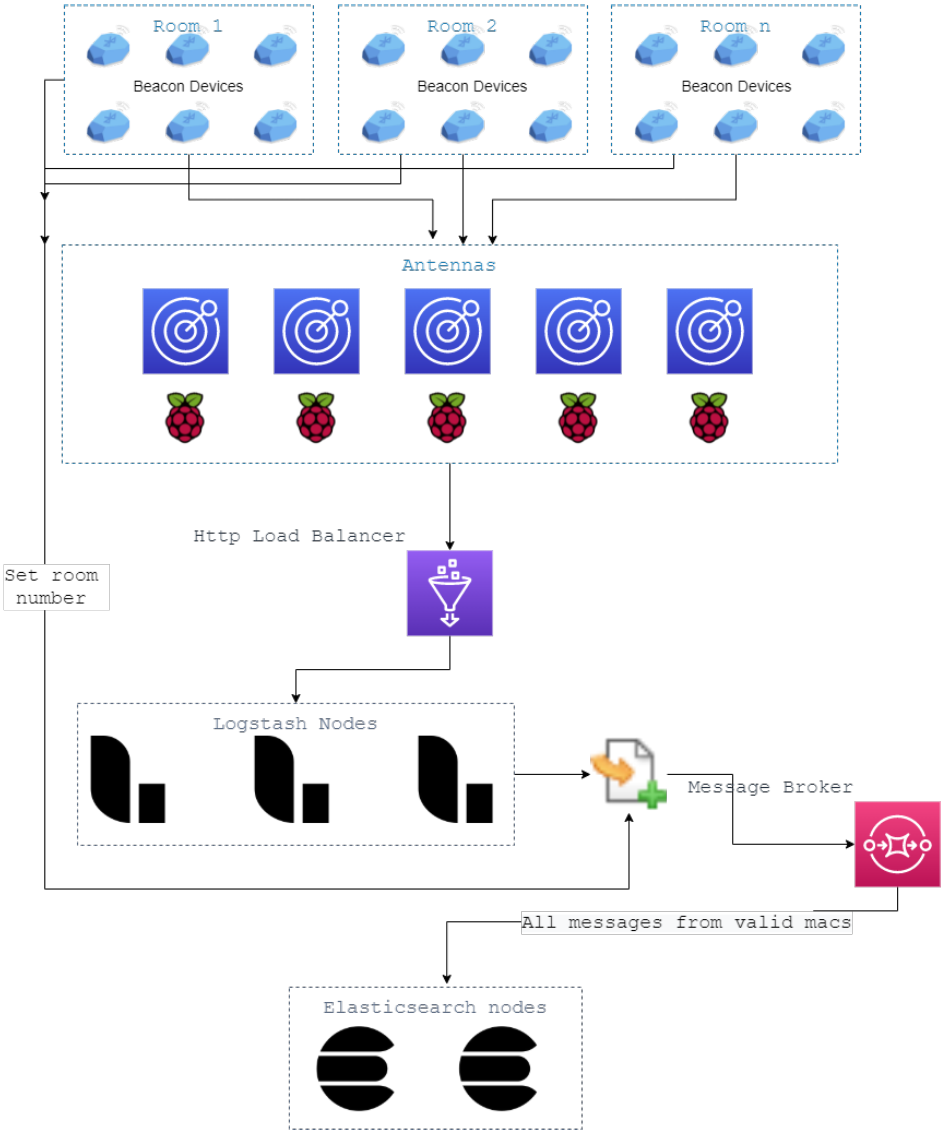


Figure 4.1: Workflow of storing data in Training part

Before starting the Training section for each floor, using the program that we implemented, we check that all the antennas and Raspberry Pi(s) are fully functional and that their data is appropriately collected in the database. The reason for this is that in each floor is examined, solely the data of the antennas and Raspberry Pi(s) of the same floor are explained in more detail in the experimental results (4.2) chapter.

However, if antennas do not operate correctly during Training, we will face problems in the Testing part; therefore, we must restrain them first. Besides, We must be concerned that this operation must be done and controlled frequently in the Testing section in real-time.

4.2 Experimental Results

This section will review the Experimental Results to improve the system in the testing and Artificial Intelligence (AI) parts.

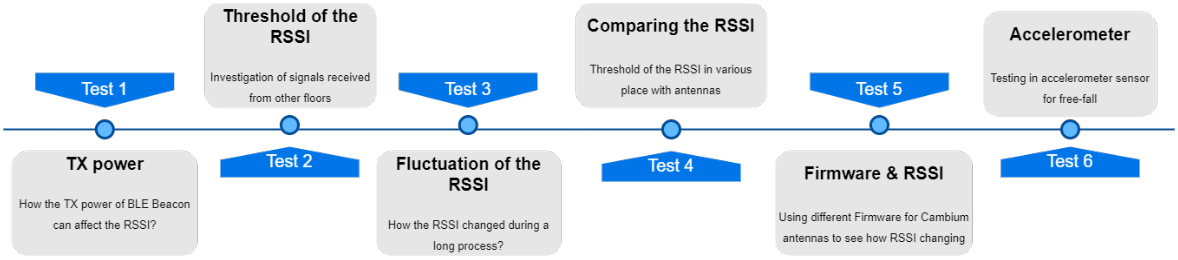


Figure 4.2: Workflow of the experiments

As it is evident in workflow 4.2, we have done six tests which each one of which is essential for the final result:

- By changing the TX power of the Beacon, we want to scrutinize how RSSI will change.
- Having placed the Beacon on different floors, we want to check if we can receive a signal or no.
- In a specific time slot (two or three months), we will examine the RSSI variation. To put it another way, we have to observe whether the RSSI experiences fluctuations or no?
- Placed on a specific floor in various zones, we will examine whether the Antenna can recognize the Beacon or no?
- By changing the Firmware of the Antenna, we examine how the RSSI will change.
- As the final test, we will check if the accelerometer can inform us about the falling of the patient or no?

4.2.1 Correlation between power and precision in RSSI

As indicated in Chapter 3.2, the BLE Beacon Devices has parameters that can be changed.

We can change the amount of signal transmission power in the Beacon. Refer to the table 4.1 it has four transmission modes for setting the Beacon, that the "0x02" is the default mode. In this experiment, we receive information and modify it by changing the power. We test -23 dBm and four dBm and analyze them to determine which mode is suitable for our project.

We fixed two beacons, the first of which had a value of 4 dBm and the second of which had a value of 0 dBm transmitting power, together with placing them in one of the Antenna of the hospital and saving the received data. Undoubtedly, Antennas and beacons fixed during the experiment.

Service	Data format	Function	Range	Remark
0xFF86	UINT8 1 byte	Transmitting power	0∼ 3	$0 \Rightarrow -23dBm0x00(shortreadingdistance)$ $1 \Rightarrow -6dBm''0x01''$ $2 \Rightarrow 0dBm''0x02''(longreadingdistance)$ $3 \Rightarrow 4dBm''0x03''$
Default: 0x02 (TX Power set to 2->0 dBm)				

Table 4.1: TX power

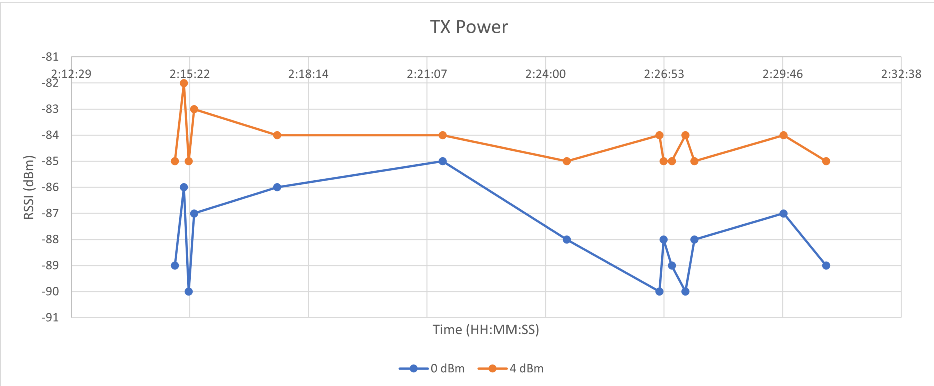


Figure 4.3: Transmitting power in a different range

As depicted in figure 4.3, The orange chart represent the Beacon with a TX power of 4 dBm, and the blue chart represents the Beacon with a TX power of 0 dBm. It is crystal clear that the average of the RSSI in the orange chart is lower than the blue chart.(Invoke in table 4.2) otherwise saying, when the average is lower, it indicates that the signal's power is high.

Time (HH:MM:SS)	0 dBm	4 dBm
2:15:00	-89	-85
2:15:13	-86	-82
2:15:20	-90	-85
2:15:28	-87	-83
2:17:29	-86	-84
2:21:30	-85	-84
2:24:31	-88	-85
2:26:46	-90	-84
2:26:52	-88	-85
2:27:04	-89	-85
2:27:24	-90	-84
2:27:37	-88	-85
2:29:47	-87	-84
2:30:49	-89	-85
Min	-90	-85
Max	-85	-82
Average	-88	-84.285

Table 4.2: Broadcasting Power

Additionally, broadcasting Power (Transmit Power) directly impacts the signal range. To put it another way, the more power, the longer the range. Additionally, increasing the power can make the signal more stable; consequently, it can harm battery life.

The Beacon's range is technically up to 70 m (+ 4dBm) for Proximity Beacons and up to 200 meters for Location Beacons. However, in real-world conditions, we should expect up to 40-50 m and 120-150 m, respectively.(Estimote, 2019)

The achievement of this analysis is that since there is countless external noise in the hospital, and it may raise the positioning failure, regardless of the battery life, we use more power for Beacons(4 dBm) to have less error. Henceforward we will analyze the battery life and failure of the system in other Experiments. (4.2.3)

4.2.2 Threshold of the RSSI in various floors

As mentioned previously, the hospital has six floors. The purpose of this examination is to control the RSSI signal from other floors. For instance, if the Beacon is on the second floor, which antennas will recognize it? Will it recognize the first or third floors antennas of the device if it recognizes any? How much is its power?

To perform this experiment, we placed a considerable number of beacons on all floors and stored all the data of the desired Beacon. Its crystal clear that we have to examine all the antennas on all floors to send the data correctly so as not to miss data from the Antenna.

As reflected in the figure 4.4, To deepen the analysis of the result, we only examine the stored data of a part of the hospital altogether, take into account that this data is related to beacons located in the same region.

Figure 4.4: The part of the hospital where we examine the Beacons data (The region marked in red)

According to Tables 4.3 and figure 4.5, it is evident that the data we collected from each Beacon on each floor recognize only the antennas of the same floor; furthermore, the antennas of the other beacons did not recognize the other floor.

Consequently, we only use data from the same floor in the Training and AI (Artificial Intelligence) section. It is crystal clear that while patients are in the elevator, we may have an error, which we will explain in more detail in the testing chapter.

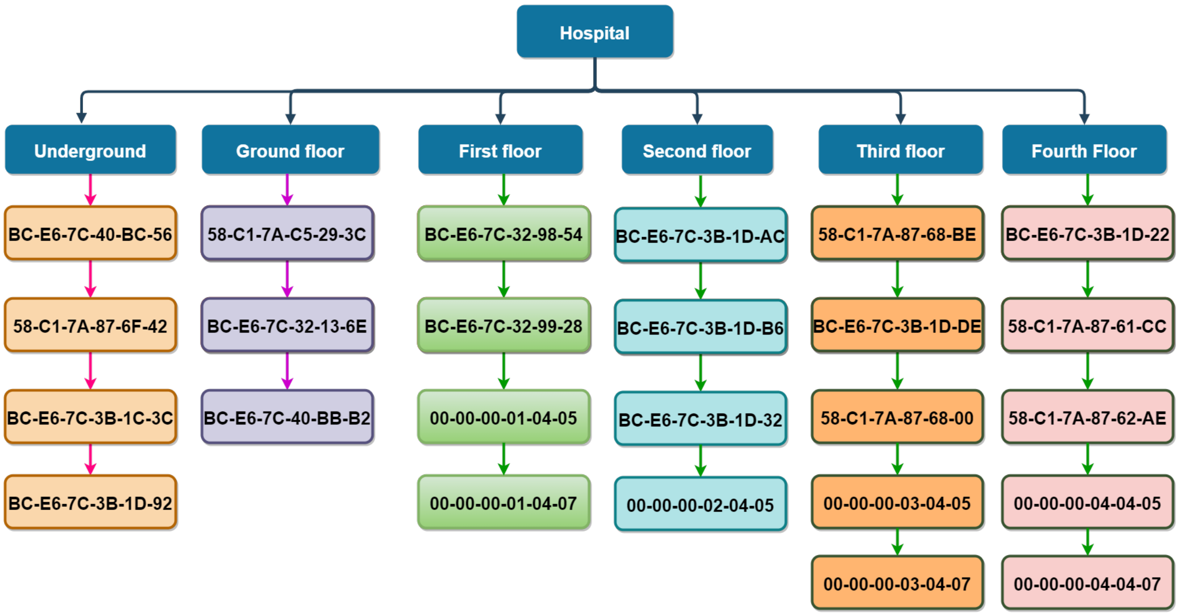


Figure 4.5: The chart of antenna installation locations based on floor numbers in a particular region.

Floor Number	Mac id of beacons	Antennas	Average RSSI (dBm)
Underground	F9-A9-F0-C5-7F-D9	BC-E6-7C-40-BC-56	-90
		58-C1-7A-87-6F-42	-68
		BC-E6-7C-3B-1C-3C	-89
		BC-E6-7C-3B-1D-92	-82
Ground floor	CF-C8-30-2E-C9-6C	58-C1-7A-C5-29-3C	-90
		BC-E6-7C-32-13-6E	-68
		BC-E6-7C-40-BB-B2	-89
First floor	FE-DF-30-B6-3A-E6	BC-E6-7C-32-98-54	-92
		BC-E6-7C-32-99-28	-72
		00-00-00-01-04-05	-82
		00-00-00-01-04-07	-88
Second floor	D1-79-85-79-A6-38	BC-E6-7C-3B-1D-AC	-90
		BC-E6-7C-3B-1D-B6	-68
		BC-E6-7C-3B-1D-32	-72
		00-00-00-02-04-05	-93
Third floor	D6-50-52-6A-84-66	58-C1-7A-87-68-BE	-90
		BC-E6-7C-3B-1D-DE	-68
		58-C1-7A-87-68-00	-70
		00-00-00-03-04-05	-88
		00-00-00-03-04-07	-84
Fourth floor	EA-B4-18-DD-3E-45	BC-E6-7C-3B-1D-22	-68
		58-C1-7A-87-61-CC	-83
		58-C1-7A-87-62-AE	-80
		00-00-00-04-04-05	-84
		00-00-00-04-04-07	-86

Table 4.3: Data received from Beacons in a particular region

4.2.3 The transformation of RSSI in the long term

This examination aims to monitor and determine the power of the received signal (RSSI) in a long-term process. To put it simply, if the bracelet works for a long time, will the power of the signal we received remain unchanged in the constant circumstance or no?

To start, we place the bracelet fixed in a room and check the data we received from only one Antenna for a long time. For example, data received from October 21, 2020, 9:53:21 to Sunday, February 28, 2021, 9:16:33 PM. Unfortunately, the data available through this period is only 20000 records for one Antenna, which can not be tabulated due to the high volume of data.

Note to mention that the data as depicted in figure 4.6 have not been cleaned data; moreover, the total test duration is 130 days, and all of the Beacons fixed in a room. The bracelets may be moved by people in the hospital, which is standard for cleaning rooms. These movements have been identified and have no impact on the test outcomes.

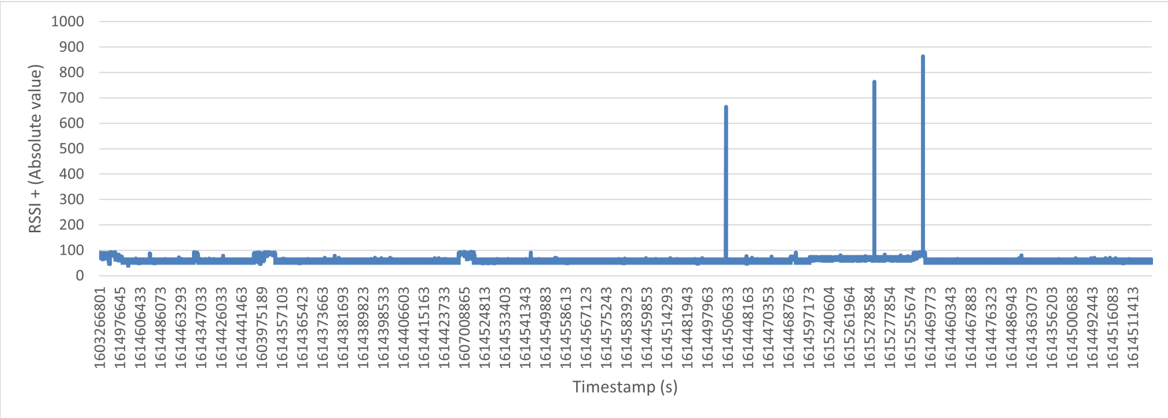


Figure 4.6: RSSI during a long process

According to the chart 4.6, first, the numbers indicated in RSSI turned to absolute value, put the matter another way, there is a reverse relationship which means that when there is an increase in the chart, the power of the received signal decreased.

The system experienced a sharp increase (Three sharp) in the received data, which we have to ignore because it is an error that happened while restoring the data. As mentioned before, the data is not cleared, and we know that the absolute value of RSSI is between zero and one hundred, so it is not reasonable to have values of 700 or 800, and this is an error in sending data or in the section of storing it, which is common in all practical projects.

When we enlarge the chart 4.6, we can see the illustration figure 4.7 here are some fluctuations (Marked with red circles.) in the chart.

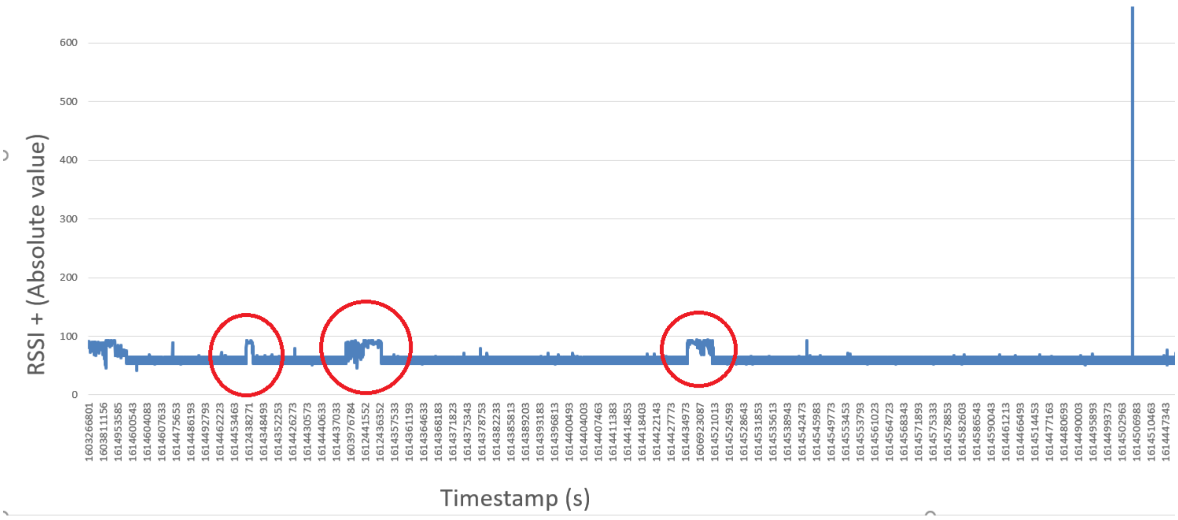


Figure 4.7: Zoomed in on chart 4.6 the part of moving the bracelets

All the data we see in this thesis is fundamental and has been extracted from the hospital database. Moreover, throughout all 130 days of the experiment, we were not present at the site to prevent the movement of the bracelets, afterwards as depicted in figure 4.7. Therefore, we have three increases and decreases in the chart for 2 and 3 days. After the investigations, we realized that at this time, a group was present to clean the hospital and move the bracelets for this period; therefore, it was the reason for these fluctuations that the system experienced.

On the other side of Figure 4.6, it can be seen that the volume of the signal received increases over time. For a better view, we recognize it in the zoomed figure 4.8.

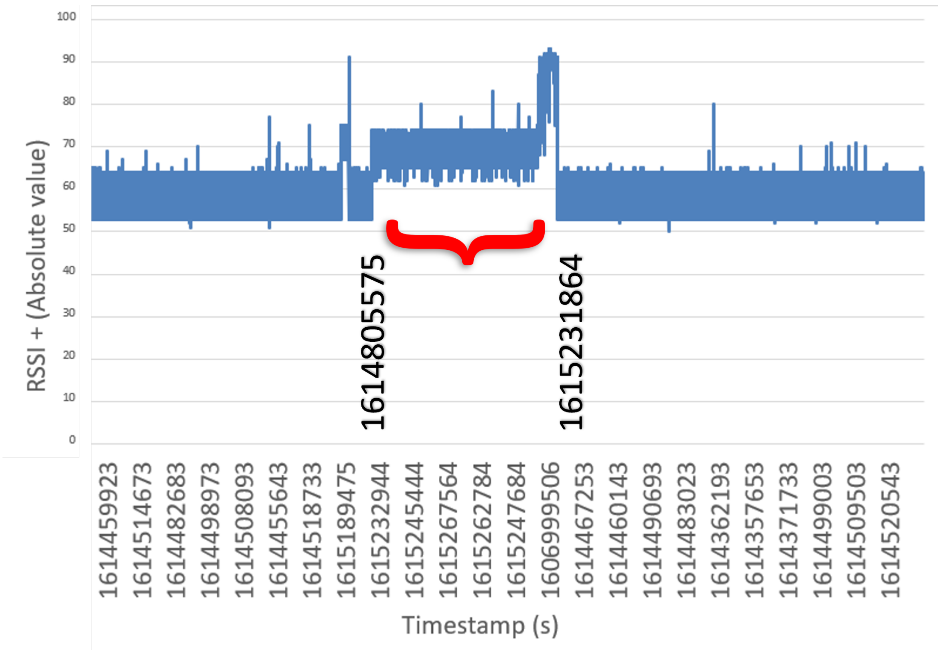


Figure 4.8: Zoomed in on chart 4.6 the part of the changing battery.

As illustrated in figure 4.8, we have an increase in the RSSI value between 1614805575 and 1615231864; in other words, when the RSSI value increases, i.e. the power decreases because the position of the bracelet and Antenna fixed.

The duration of this sudden increase is about seven days. For observing this division explicitly, (from 1615231864), I replaced the bracelet's battery and put a new battery; it was evident that the data we received decreased (the power of signal increased); note to mention that the position of the bracelet and antennas fixed.

As reflected in figure 4.8, after replacing the battery and placing the new battery in the bracelet, Immediately, the received signal changed to the previous state. As a result, the received signal decreases; contrastingly, the signal strength has increased.

Consequently, using the battery for a long time without replacing it with a new one will experience more errors in the positioning system.

4.2.4 Threshold of the RSSI in various place with antennas

The objective of this experiment was to investigate the threshold for receiving RSSI signals from antennas at various locations of the hospital.

First, we place the bracelet in different places and collect the data received from all the antennas. Then, we place this test in other hospital locations for the same time as before and compare the data.

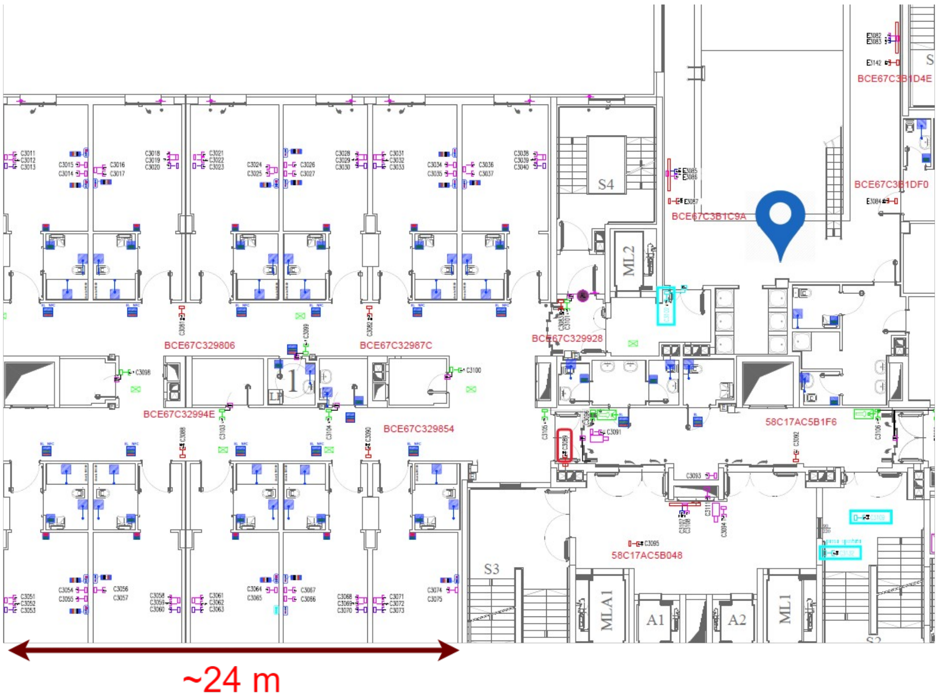


Figure 4.9: Part of the ground floor map of the first location.

As reflected in figure 4.9, the bracelet located in the specified area (blue location icon), and all antennas controlled before beginning the test to send their information correctly. The Mac ID of the antennas is written in red in Figure 1, and the Antenna's position is fixed.

The bracelet is located in a prominent place, and many antennas should recognize it; besides, there are not many hindrances between the Antenna and the tag. We can see all the available data for a fixed time in Table 4.4.

Antennas	RSSI (Average) (dBm)	Number of packets	Min	Max
BC-E6-7C-3B-1D-F0	-80	96	-85	-74
BC-E6-7C-3B-1D-4E	-89	90	-91	-83
BC-E6-7C-3B-1C-9A	-71	97	-80	-61
BC-E6-7C-32-99-28	-95	60	-99	-93
Min	-95	60		
Max	-71	97		

Table 4.4: Data received when the bracelet is in location one.

As it is evident in Table 4.4, four antennas can be observed from the bracelet with the position defined in Figure 4.9. Furthermore, the lowest value of RSSI is related to the antenna "BC-E6-7C-32-99-28", and the highest value of RSSI is related to the antenna

"BC-E6-7C-3B-1C-9A". In other words, "BC-E6-7C-32-99-28" antennas have higher signal strength than other antennas; therefore, it is closer. Additionally, the Antenna has less signal strength than the "BC-E6-7C-3B-1C-9A". Consequently, it is farther than other antennas. Nevertheless, it has been mentioned before that this theory is when we are outdoors(free space), but we have many obstacles in this area. For instance, it can be seen that the antenna "BC-E6- 7C-3B-1D-4E" is further away from the bracelet than the "BC-E6-7C-32-99-28" antenna; however, the signal they received does not have this relativity, and the "BC-E6-7C-32-99-28" antenna has power. Nevertheless, it has less signal than the other, and this is due to the obstacles that exist between the bracelet and the "BC-E6-7C-32-99-28" antenna, including the presence of two large metal doors that were closed during the examination.

On the other side of the coin, there is a correlation between signal strength and the number of received packets. It is clear that the higher the received signal strength, the more packets received, and this is relevant because the farther the Antenna is, the less probability to receive packets.

We placed the bracelet in a different place in the subsequent step and saved the received data, but a better comparison can be made simultaneously, as discovered in the previous experiment.

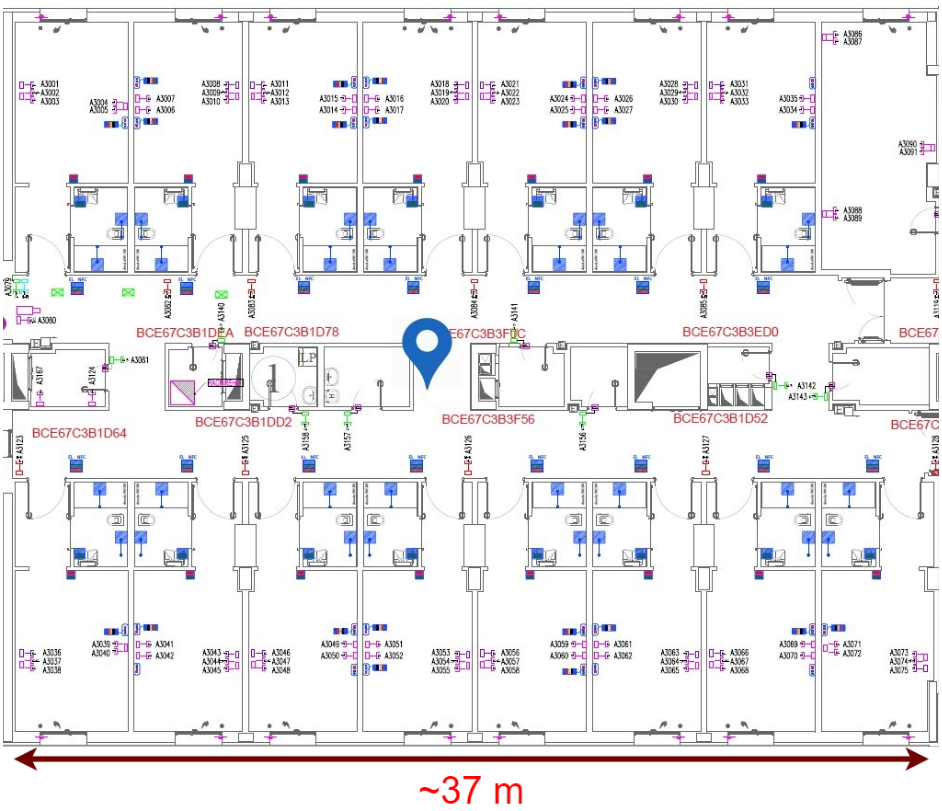


Figure 4.10: Part of the ground floor map of the second location.

As shown in Figure 4.10, we located the bracelet in another location and collected all the received data for the same period. Following Table 4.5 presents the maximum and minimum values of the received signal.

Antennas	RSSI (Average) (dBm)	Number of packets	Min	Max
BC-E6-7C-3B-3F-0C	-71	95	-75	-68
BC-E6-7C-3B-1D-78	-83	89	-86	-79
BC-E6-7C-3B-3F-56	-71	94	-73	-65
BC-E6-7C-3B-1D-D2	-83	87	-85	-80
BC-E6-7C-3B-1D-64	-96	54	-98	-95
BC-E6-7C-3B-3E-D0	-95	57	-98	-94
BC-E6-7C-3B-1D-52	-94	52	-96	-92
BC-E6-7C-3B-1D-EA	-90	63	-95	-90
Min	-96	54		
Max	-71	95		

Table 4.5: Data received when the bracelet is in location two (corridor).

As indicated in table 4.5, more antennas are detected on the bracelet than in the first position. Because the bracelet is in the corridor, there are fewer obstacles between the Antenna and the bracelet; therefore, antennas recognize more tags.

Moreover, the nearest Antenna has more packets and also more signal strength. Similarly, in the case of the farthest Antenna, we have fewer packets and less signal strength.

According to this experiment, as mentioned, it can be concluded that the number of packets depends on the distance or vicinity of the antennas and also the presence of obstacles. We now have fewer antennas on the sides, and this is a challenge in positioning in the system because the fewer antennas, the less information is received, and the system cannot position correctly.

4.2.5 Analysis of the performance changing the Firmware for Cambium antennas

This analysis wants to switch the antenna firmware to comprehend a variation in the positioning system. As we are aware, Firmware is a software program or set of instructions programmed on a hardware device. It provides instructions on how to connect the device to other computer hardware.

In this experiment, we use two existing Firmware related to Cambium Networks company. For this purpose, we use two antennas with firmware "4.1-a4" and "4.2-r15 "and two bracelets with the same distance from the antennas to collect the received Data for the same duration.

As reflected in the table 4.6, we can observe the average, minimum and maximum RSSI values. For better comparison, all the data have their absolute value written, and also because we had a high number of records, it is not possible to display all the data in one table, and we can see all the data behaviour in Figure 4.11.

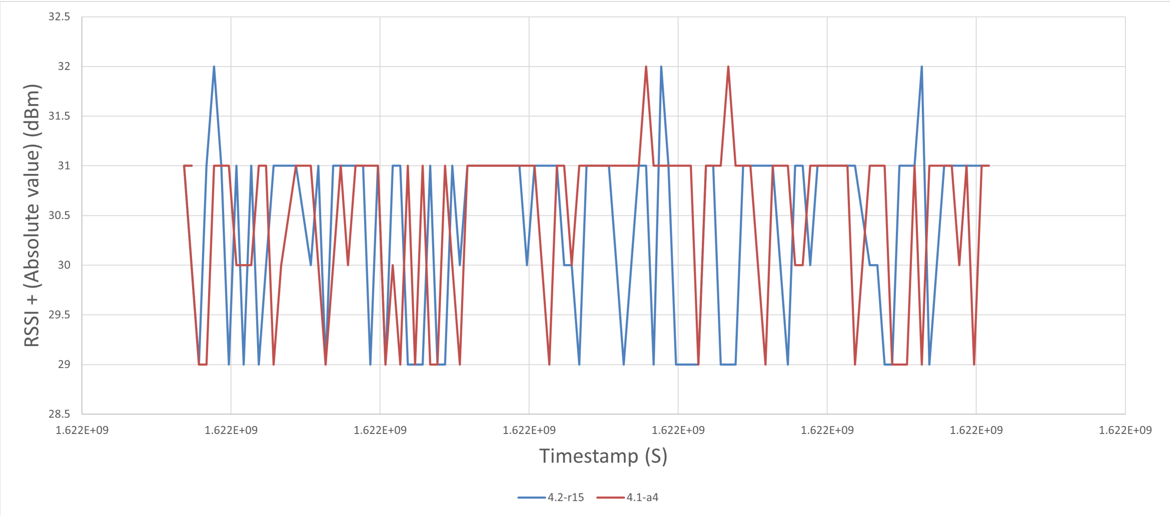


Figure 4.11: Absolute value data received by two antennas with different firmware.

Firmwares	**4.2-r15**	**4.1-a4**
Average	30.4158 dBm	30.51 dBm
Max	32 dBm	32 dBm
Min	29 dBm	29 dBm

Table 4.6: Analyze data received from two antennas with different firmware.

As indicated in Table 4.6 and Figure 4.11, there is no difference in the signal because the average and range of data fluctuation are equivalent. For more details, we will introduce a part of the data received from both Firmware.

```
% Part of the data received from the Antenna
% with the firmware "4.1-a4".
{
   "ap_mac":"58-C1-7A-87-69-86",
   "ap_name":"E600-876986",
   "version":"2.2",
   "ble_discased_clients":[
      {
         "bt_timestamp":"0] 2021-05-24 09:07:49.083218\n",
         "bt_mac":"4F-7E-71-3F-A6-AA",
         "bt_rssi":" -30 dBm ",
         "bt_uuids":"Unknown (0xff80)\n"
      }
```

```
    ]
}
```

```
% Part of the data received from the antenna
% with the firmware "4.2-r15".
{
   "ap_mac":"58-C1-7A-87-69-86",
   "ap_name":"E600-876986",
   "version":"2.2",
   "ble_discaseed_clients":[
      {
         "bt_timestamp":1621847269,
         "bt_mac":"4F-7E-71-3F-A6-AA",
         "bt_rssi":-31,
         "bt_uuids":"Unknown (0x180f)\n"
      }
   ]
}
```

As can be seen in the preceding formats, there is a slight difference between them, including:

- "ble_discaseed_clients" & "ble_discased_clients".
- The difference in "bt_timestamp" field.
- The difference is in the "bt_rssi" field in firmware 4.1-a4 with the RSSI unit (dBm) but in firmware "4.2-r15" without the unit.

The conclusion that can be drawn from this analysis is that distinct Firmware does not change the RSSI signal, although we must be attentive about filtering and inserting data in Elasticsearch. This firmware change causes an error.

4.2.6 Testing in accelerometer sensor for free-fall

As mentioned in the beginning, the diagnostic feature of the project is to alert the system if the patient falls so that nurses can come to the patient's location by taking advantage of the indoor positioning system. This warning system is sent to the nurses' system in the back end, though how does it detect a fall in the core of this system?

The Beacon has integrated an accelerometer sensor that displays the coordinates of the three-axis (X, Y, Z). Its crystal clear that when the battery is removed from the Beacon, the sensor will turn OFF.

The accelerometer sampling is the frequency of the G-sensor to wake itself to read the coordinates; the MCU sampling is the frequency of the MCU to read the coordinates of the G-sensor. The higher the value of the MCU sampling, the higher the frequency of the Beacon to read the coordinates and the battery consumption.

There are four operating modes of accelerometer:

- **Connection Mode** : The accelerometer values can be read ONLY via service (FF:94).
- **Advertising Mode** : The accelerometer values can be read in Scan Response.
- **Trigger Mode** :This mode allows the user to choose a sensitivity for the accelerometer:
  - Slight Mode
  - Moving Mode
  - Critical Mode

  It also allows the user to choose an interval (in seconds) in which the Beacon will send Advertising Packets as soon as it senses an acceleration based on the sensitivity chosen.

  The Watchdog function can also be used (see below for details of service FF: E3), allowing to receive advertising packets if the Beacon is not moving (stationary).
- **Real Time Mode** :This mode allows the user to choose a sensitivity for the accelerometer:
  - Slight Mode
  - Moving Mode
  - Critical Mode

  The Beacon will send Advertising Packets when it senses an acceleration based on the sensitivity chosen, without a custom interval, but only when it senses it. When the Beacon stops, no packets will be sent.
- **CONNECTION MODE**

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Read/Write	1Byte	Set moving mode	1 Byte	0x00 $\Rightarrow ConnectionMode(DEFAULT)$

Table 4.7: 0xFFE1: Moving mode service

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Notify	6 Byte (Big Endian)	accelerometer	6 BYTE BIG ENDIAN	XYZ each direction 2 Byte

Table 4.8: Accelerometer control in connected mode

For instance, in the Accelerometer control in connected mode section (refer in table 4.8), if we receive the following data:

0x000100010100 ={x=1,y=1,z=256}

- **ADVERTISING MODE**

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Read/Write	1Byte	Set moving mode	1 Byte	0x01 ⇒ *AdvertisingMode*

Table 4.9: Moving mode service

- **TRIGGER MODE**

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Read/Write	1Byte	Set moving mode	1Byte	0x03 ⇒ *SlightMode* 0x04 ⇒ *MovingMode* 0xff ⇒ *CriticalMode*

Table 4.10: 0xFFE1: Moving mode service

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Read/Write	2Byte	Set adv. time	From 0s to 100s	2s. ⇒ 5*dec.* ⇒ 0*x*5*HEX* 10s. ⇒ 25*dec.* ⇒ 0*x*19*HEX* 60s. ⇒ 150*dec.* ⇒ 0*x*96*HEX* 100s. ⇒ 255*dec.* ⇒ 0*xFFHEX*

Table 4.11: 0xFFE2: Moving advertising time service

As illustrated in table 4.11, the range values are only even.

- **REAL TIME MODE**

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Read/Write	1Byte	Set moving mode	1Byte	0X03 ⇒ *SlightMode* 0x04 ⇒ *MovingMode* 0xff ⇒ *CriticalMode*

Table 4.12: 0xFFE1: Moving mode service

Service	Executable operation	Data format	Function	Range	Remark
0xFF80	Read/Write	28Byte	Set adv. time	0s. and 2s. to 100s.	Default: 0x00

Table 4.13: 0xFFE2: Moving advertising time service

As indicated in table 4.13, in this mode, the advertising sent while moving Beacon unexpectedly stopped working.

According to these methods, if the acceleration horizontally is higher than the configuration value, a packet will be sent to the server, which is a binary message and has a bracelet ID ("bt_mac") and time("bt_timestamp"). For this design, a notification is created in the back end and the indoor positioning system.

4.3 Experiments on Raspberries

As mentioned and examined in the previous sections and chapters (3.1.2), we concluded that in some rooms and areas of the hospital, we have fewer antennas available. We call those places dark regions, and it is crystal clear that for indoor positioning with a few antennas, the system's failure is relatively high. Like the Global Positioning Systems (GPS) that requires at least two satellites for 2D positioning, indeed, as mentioned earlier, the internal positioning system is entirely different from the GPS; however, in this project, we used the ideas of this system, which we will review in more detail in the section on artificial intelligence(AI).

Because the patient is more likely to be in the rooms and it is one of the black areas, we decided to install antennas. There are two solutions to this problem; first, buy an antenna and install it in Rooms and others Purchase and install Raspberry Pi(s) and have done the installation in rooms.

Accordingly, in these experiments, we examine which solution is more suitable.

4.3.1 Comparing the threshold of the RSSI on Raspberries in various places

In this section, we want to examine the threshold of the receiving signal, monitor and compare the division of both the signal received from the Cambium antennas and the signal received from the Raspberry pi (s).

First, the Cambium antennas and Raspberry pi (s) have been installed and fixed in the ("BC-E6-7C-3B-1C-C4") along with bracelets which have been put in different locations after that; we can compare the data received from different locations. Then, as clearly depicted in Figure 4.12 the location of the antenna and Raspberry pi(s) are determined, the bracelets are depicted with the numbers, and we received various data in every zone. Before starting the test, we controlled all the antennas that send the data correctly; furthermore, storing data in the database is the same for all three tests.

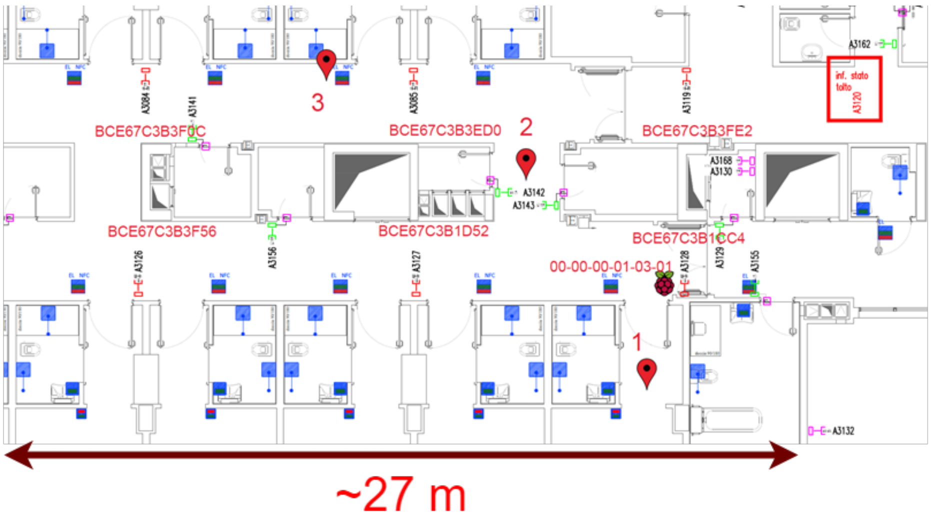


Figure 4.12: First floor map, three bracelet locations to compare antenna and Raspberry pi(s).

- **Location 1**
  In this test, the bracelet is planted in place number 1, which is inside the corridor of the room, also; to try not to put an additional obstacle between the Antenna and the bracelet in all three tests, the door of the room will be kept open during the test. Note: The room's door is metal, and if we keep the door closed, the received signal will be different.

  In location N.1, we received the data mentioned in table 4.14. As stated in the table, the average power of the signal received from the Raspberry pi(s) is higher; however, the location of the Cambium antenna (BC-E6-7C-3B-1C-C4) and the Raspberry pi(s) were the same. Note that the packets received are more than the packets received from the Raspberry pi(s).

Antennas	RSSI (Average)	Number of packets	Min	Max
00-00-00-01-03-01	-66 (dBm)	53	-68	-61
BC-E6-7C-3B-1C-C4	-70 (dBm)	35	-79	-65
BC-E6-7C-3B-1D-52	-89 (dBm)	20	-98	-80
Min	-89	20		
Max	-66	53		

Table 4.14: Comparison between antenna and Raspberry pi(s) in position number 1.

Figure 4.13: Comparison between antenna and Raspberry pi(s) in position number 1.

As illustrated in Figure 4.13, straightforward that the farther the antennas are, the lower the signal strength received, furthermore, since the antennas and the Raspberry are in the same place, the signal strength received from Raspberry pi(s) has more authoritative than antennas furthermore the number of received packages is higher than antennas.

Another point that can be mentioned is the difference between antennas and Raspberry pi(s). In fact, there is a variation between the minimum amount of signal and the Maximum amount of signal received from antennas and Raspberry pi(s). In other words, the standard deviation of the signal in antennas is far less than the Raspberry Pi.

- **Location 2**

  In location N.2, we received the data mentioned in table 4.15; we have received more data because we were in a place with a higher threshold of the Antenna. It is also clearly visible that the received data from the Cambium antenna ("BC-E6-7C-3B-1C-C4") and the Raspberry pi(s) ("00-00-00-01-03-01") decreased because we stayed far from them, but as already mentioned, there is a remarkable difference between Antenna and the Raspberry pi(s).

Antennas	RSSI (Average)	Number of packets	Min	Max
BC-E6-7C-3B-1C-C4	-82 (dBm)	28	-92	-75
BC-E6-7C-3B-1D-52	-71 (dBm)	30	-83	-67
00-00-00-01-03-01	-77 (dBm)	52	-82	-70
BC-E6-7C-3B-3F-E2	-82 (dBm)	27	-92	-73
BC-E6-7C-3B-3F-0C	-89 (dBm)	20	-98	-80
BC-E6-7C-3B-3F-56	-90 (dBm)	17	-98	-79
Min	-90	17		
Max	-71	52		

Table 4.15: Comparison between antenna and Raspberry pi(s) in position number 2.

Figure 4.14: Comparison between antenna and Raspberry pi(s) in position number 2.

As we can see in Figure 4.14, the number of packets received from closer antennas is still higher; however, another significant point is the fact that the Raspberry besides due to the maximum and minimum amount of received signal, the standard deviation of the Raspberry is more than the antennas.

- **Location 3**

  In location N.3, we received the data mentioned in table 4.16, however, as you see, we did not receive any data on the Cambium antenna ("BC-E6-7C-3B-1C-C4") as we were not in the threshold of it; however, we still received data from the Raspberry pi, but the amount of the data was noticeable (the power of the signal was low) put it another way, we are far from the Raspberry pi ("00-00-00-01-03-01").

Antennas	RSSI (Average)	Number of packets	Min	Max
BC-E6-7C-3B-1C-C4	– (dBm)	–	–	–
BC-E6-7C-3B-1D-52	-80 (dBm)	24	-94	-73
00-00-00-01-03-01	-87 (dBm)	31	-94	-86
BC-E6-7C-3B-3F-E2	-81 (dBm)	15	-92	-76
BC-E6-7C-3B-3F-0C	-76 (dBm)	27	-85	-71
BC-E6-7C-3B-3F-56	-82 (dBm)	26	-90	-76
BC-E6-7C-3B-3E-D0	-70 (dBm)	35	-78	-66
Min	-87	15		
Max	-70	35		

Table 4.16: Comparison between antenna and Raspberry pi(s) in position number 3.

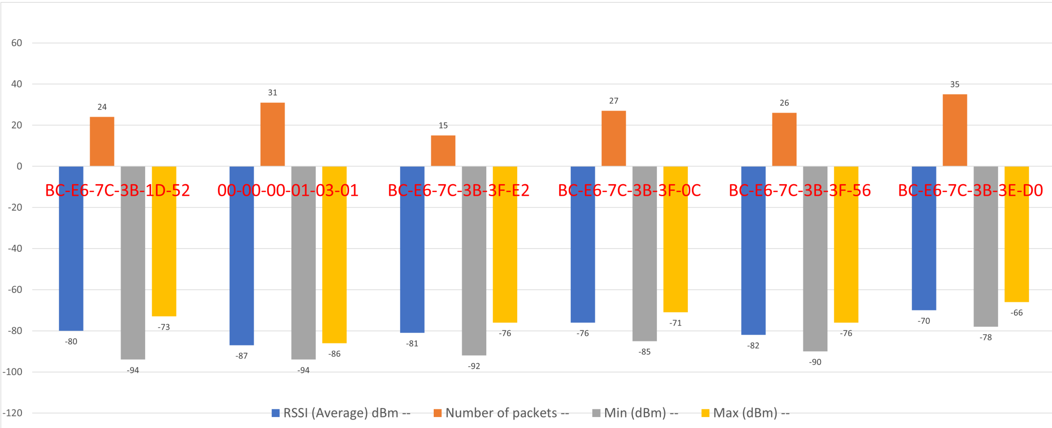


Figure 4.15: Comparison between antenna and Raspberry pi(s) in position number 3.

As illustrated in Figure 4.15, the farther the Antenna is, in addition to reducing the number of received packets, the standard deviation also decreases concerning the closer antennas, and it has a more significant impact on the Raspberry.

One of the many differences that may be seen in this experiment is the difference in data between the "BC-E6-7C-3B-3F-E2" and "BC-E6-7C-3B-3F-0C" antennas. Consequently, we will notice that the antenna number "BC-E6-7C-3B-3F-E2" has less power of the received signal, including the number of received packets, than the antenna "BC-E6-7C-3B-3F-0C", even though the distance between these two antennas is approximately equal to the bracelet.

As detailed in Figure 4.15, approaching the antenna "BC-E6-7C-3B-3F-E2", there is an emergency door that is closed during the test, and this door opens only in case of emergency, and also this is a metal door. Therefore, it is thick; that is the reason why we have a difference in receiving data.

The conclusion that can be extracted from these three experiments to solve the problem of black areas is that in addition to their low price compared to antennas in the indoor positioning system and detection of BLEs, Raspberries are better not only than the antennas but also than the observation threshold. Note to mention that there are more Raspberries than antennas, so we use them in this project.

We programmed the raspberry pies so that the sent data to the database is precisely the same amount of data the Antennas will send to the database; therefore, there will be no difference between the raspberry pie and Antenna in the Training and Testing section. In the following tests, we will go into more detail about Raspberries, which will accommodate us better in our project.

4.3.2 RSSI Performance using different Raspberry Pi

As discovered in the previous experiment, the best solution to resolve the black areas is installing and setting up Raspberries in those areas to collect more data from the antennas. Because implementing it is the most optimal choice for our project.

We can use Raspberries version 3 and 4 for this project, but what is the difference between these two versions of Raspberries in our project? Does it affect our project and the amount of signal?

To answer the above questions, we start here by placing two Raspberries with different versions in the same place. We try to make both Raspberries with Plastic cases that have the same test conditions. Additionally, place a Disk Beacony at an equal distance from both Raspberries. (Figure 4.16)

Figure 4.16: Two Raspberry Pi(s) with different versions (white Raspberry Pi 3 and black Raspberry Pi 4)

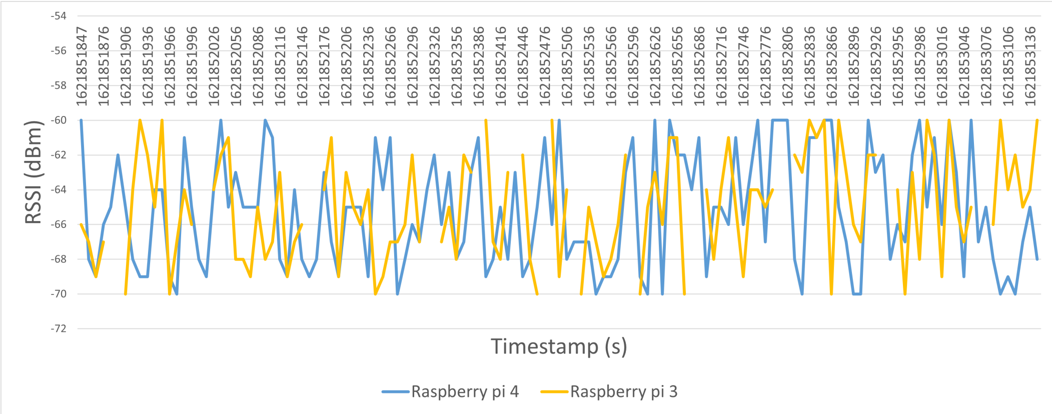


Figure 4.17: Data received from BLE beacon by two Raspberry Pi(s) 3 and 4 with Plastic case.

As indicated in Figure 4.17, data received from the BLE beacon by two Raspberry Pi(s) 3 and 4 with Plastic cases with the same test time. The duration of the examination is approximately 21 minutes, and due to the high volume of data received from the BLE beacon by two Raspberry Pi(s), we see only the data analysis table in 4.17.

Raspberry Pi(s)	**RSSI (Average)**	**Number of packets**	**Test duration**
Raspberry Pi 4	-64.916 (dBm)	132	1309 (s)
Raspberry Pi 3	-65.09 (dBm)	111	1309 (s)

Table 4.17: Analysis of data received from BLE beacon by two Raspberry Pi(s) 3 and 4 with Plastic case.

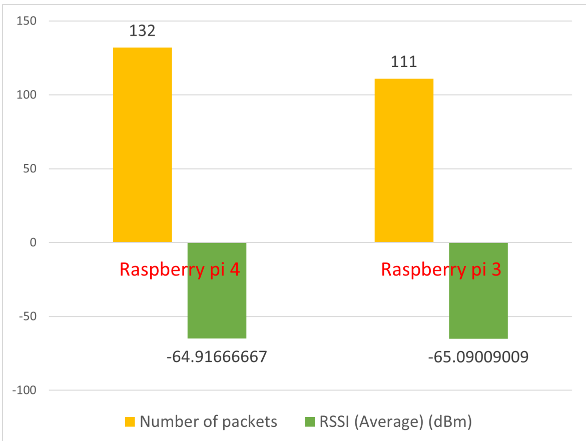


Figure 4.18: Analysis of data received from BLE beacon by two Raspberry Pi(s) 3 and 4 with Plastic case.

As indicated in Figure 4.18 and Table 4.17, there is not much difference between the Raspberry Pi 3 and 4 in the RSSI signal. Although there is a difference in the number of packages received by Raspberry Pi(s)and Raspberry Pi 4, the number of packages received is more than Raspberry Pi 3. Before studying the difference in the number of received packages of Raspberry Pi 3 and 4., we better duplicate the early experiment without having a Plastic case as shown in Figure 4.19 and save the received data.

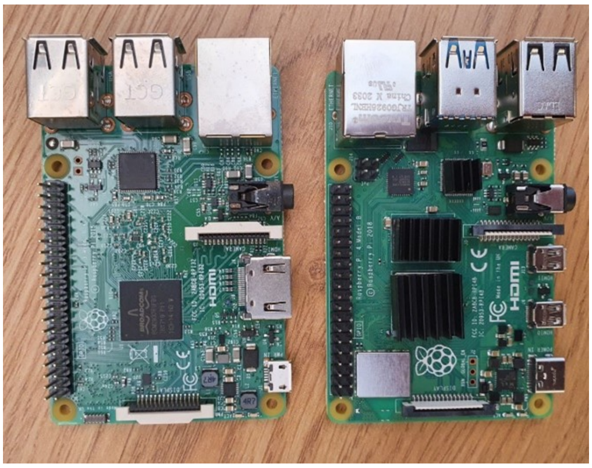


Figure 4.19: Raspberry Pi 3 model B and 4 model B without Plastic case.

As reflected in Figure 4.20, all data received from the BLE beacon by two Raspberry Pi(s) 3 model b and 4 model b is without the case, which is done in the same time for approximately 25 minutes.

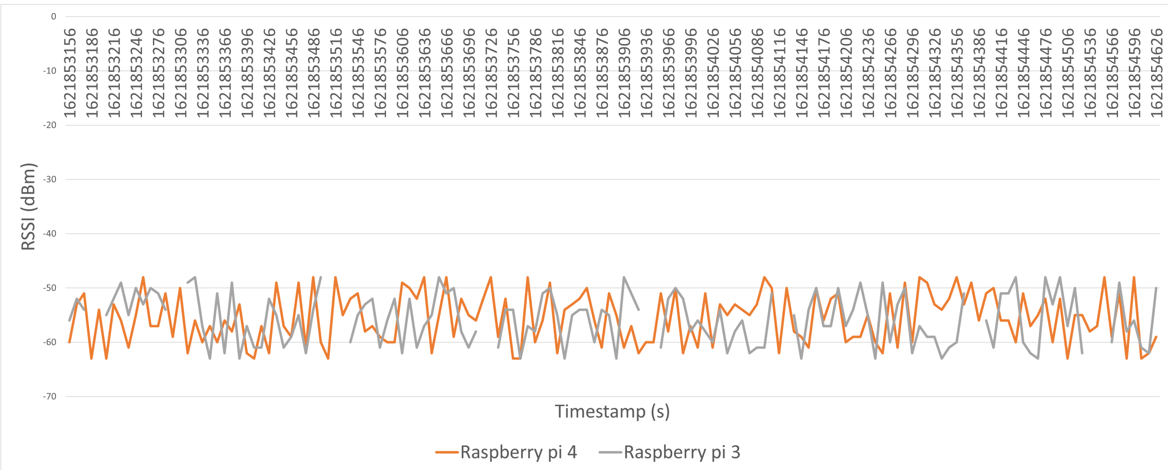


Figure 4.20: Data received from BLE beacon by two Raspberry Pi(s) 3 and 4 without case.

Raspberry Pi(s)	**RSSI (Average)**	**Number of packets**	**Test duration**
Raspberry Pi 4	-55.7583 (dBm)	149	1480 (s)
Raspberry Pi 3	-55.3664 (dBm)	131	1480 (s)

Table 4.18: Analysis of data received from BLE beacon by two Raspberry Pi(s) 3 and 4 without case.

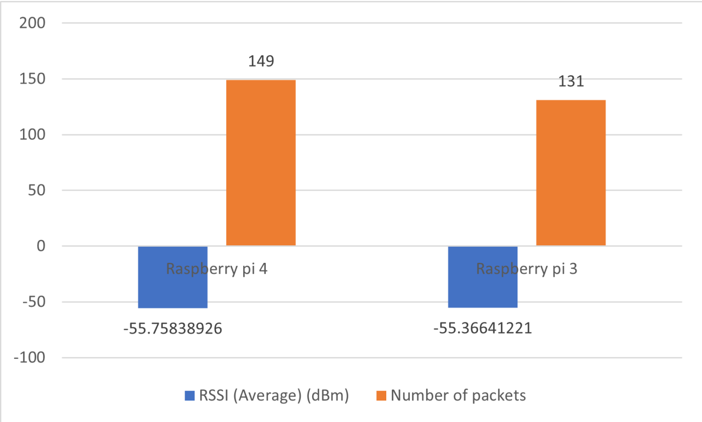


Figure 4.21: Analysis of data received from BLE beacon by two Raspberry Pi(s) 3 and 4 without case.

As reflected in table 4.18, there is no difference in the received signal between the two Raspberry Pi(s) 3 and 4; however, there is still a difference in the number of packets received from the Raspberry Pi(s) 3 and 4. Furthermore, there is a difference between Tables 4.17 and 4.18 in the received signal (RSSI), which is due to the Plastic case of the Raspberry Pi(s), which despite the lack of case, the power of the received signal is higher because there will be the minor obstacle between the Bluetooth sensor and BLE beacon.

The difference in the number of packets received from Raspberry Pi 3 and 4 is due to the sensors' difference. As illustrated in table 4.19, In Raspberry Pi 3 is installed "Broadcom BCM43438" Bluetooth Chipset and in Raspberry Pi 4 is installed "Cypress CYW43455" Bluetooth Chipset.

Board	**Bluetooth Chipset**	**Bluetooth Supported**
Raspberry Pi 3 Model B	Broadcom BCM43438	Bluetooth 4.1
Raspberry Pi 4 Model B	Cypress CYW43455	Bluetooth 5.0

Table 4.19: Bluetooth Chipset on different Raspberry Pi with Bluetooth Supported.

The updated version Bluetooth 5 is faster than 4. Bluetooth 4 supported 2 Mbps, whereas five gives us up to 5 Mbps. Furthermore, Bluetooth 4 supports 50 meters in the outdoor range and 10 meters in an indoor location. In comparison, Bluetooth 5 supports 5 meters of the outdoor environment, 200 meters (Around 800 feet) in the indoor environment, and 40 meters in the internal environment. In Message Capability, Bluetooth 4 can give us highly 31 bytes, whereas Bluetooth 5 can give us 255 bytes.(Estimote, 2019)

Based on the experiment, we conclude that the Raspberry Pi 4 works better than the Raspberry Pi 3, but because the material used in the hospital must be standard, we necessarily need to use a plastic case so that it does not pose a danger for patients. Its crystal clear that other cases are available, which we will discuss in the subsequent experiments.

4.3.3 Using different cases for Raspberry pi to see how RSSI changing

As already mentioned, having a case for Raspberry pi has a different result in receiving the signal. Therefore, in this experiment, we want to monitor which case is suitable for our project by placing various cases on Raspberry pi.

To start, we use two Raspberry pi with the same model, one with an aluminium case (Figure 4.22b) and the other with a Plastic case (Figure 4.22a), and we use two BLE beacons to collect all the values of the RSSI signal.

.

(a) By Plastic Case.

(b) By Aluminum Case.

Figure 4.22: Raspberry Pi 4 Model B with various case.

The test takes about fifteen minutes at the same time for both Raspberry Pi(s), including we see all the packages received from both Raspberry Pi with different cases in Figure 4.23. As it is clear, Raspberry Pi does not work well with an aluminium case and only receives some packets, which are also high-noise signals.

Raspberry Pi(s) 4	RSSI (Average)	Number of packets	Test duration
By Plastic Case	-64.7777 (dBm)	90	890 (s)
By Aluminum Case	-77.223 (dBm)	9	890 (s)

Table 4.20: Analysis of data received from two Raspberry Pi (s) with different Plastic and Aluminum case.

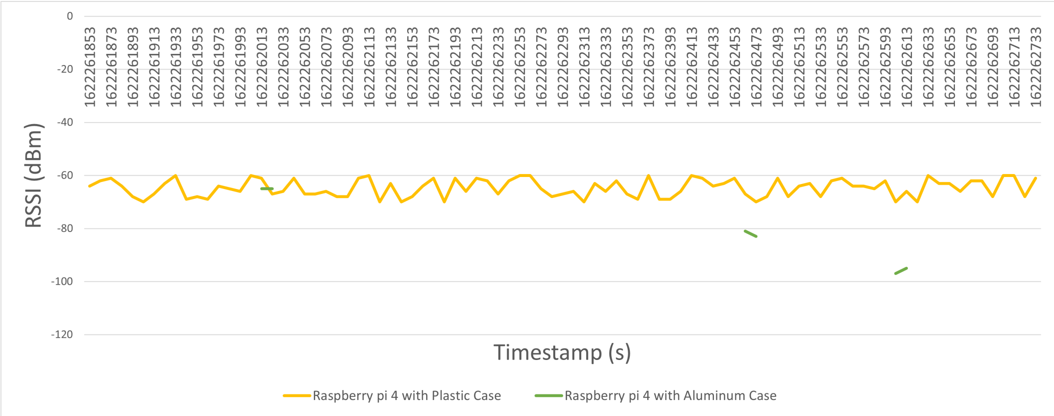


Figure 4.23: Data received from two Raspberry Pi (s), version 4 with different Plastic and Aluminum cases.

Aluminium cases are conductors of electricity. Practically any conductor blocks radio waves because radio waves are a form of electromagnetic radiation. Otherwise saying conducting materials convert waves of electromagnetic radiation into thermal energy. Electromagnetic waves that hit the conductive material are partially reflected and partially refracted. So part of the wave may be heading back to us.

The result of this experiment is that we have to use Plastic cases for the project because the Bluetooth sensor has a problem with the aluminium and iron cases.

4.3.4 Check the difference of RSSI through rotating Raspberry pi

In this experiment, we want to investigate the effect of an RSSI signal by rotating a Raspberry Pi in a static location. First, we have a Raspberry Pi 4 with a plastic case and a BLE beacon located at the same distance in Raspberry Pi that should be rotated for this test.

To start the next test, we had a Raspberry pi version 4 with the plastic case and a bracelet fixed in a place as it is indicated in Figure 4.25, we rotate the raspberry pie, and in each rotation, we received the data mentioned below, it is apparent that we will have various data because the sensor of the receiving signal of the Raspberry will come close or go away with each rotation of the Raspberry pi.

As reflected in Figure 4.24, the position of the Bluetooth sensor in the Raspberry Pi 4 is depicted.

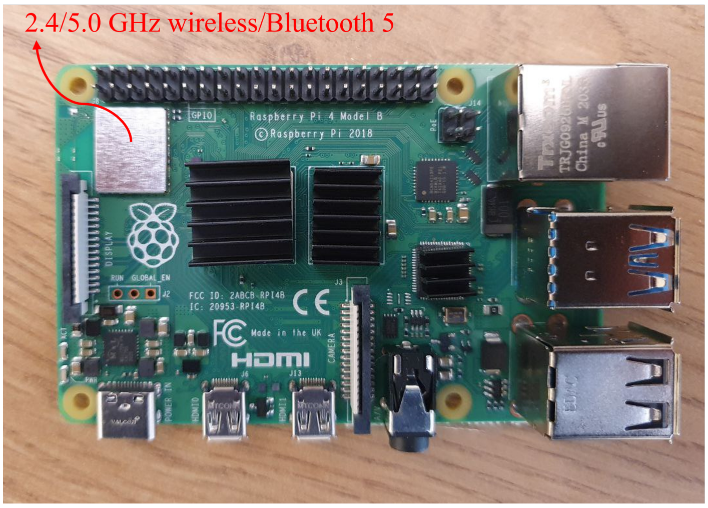
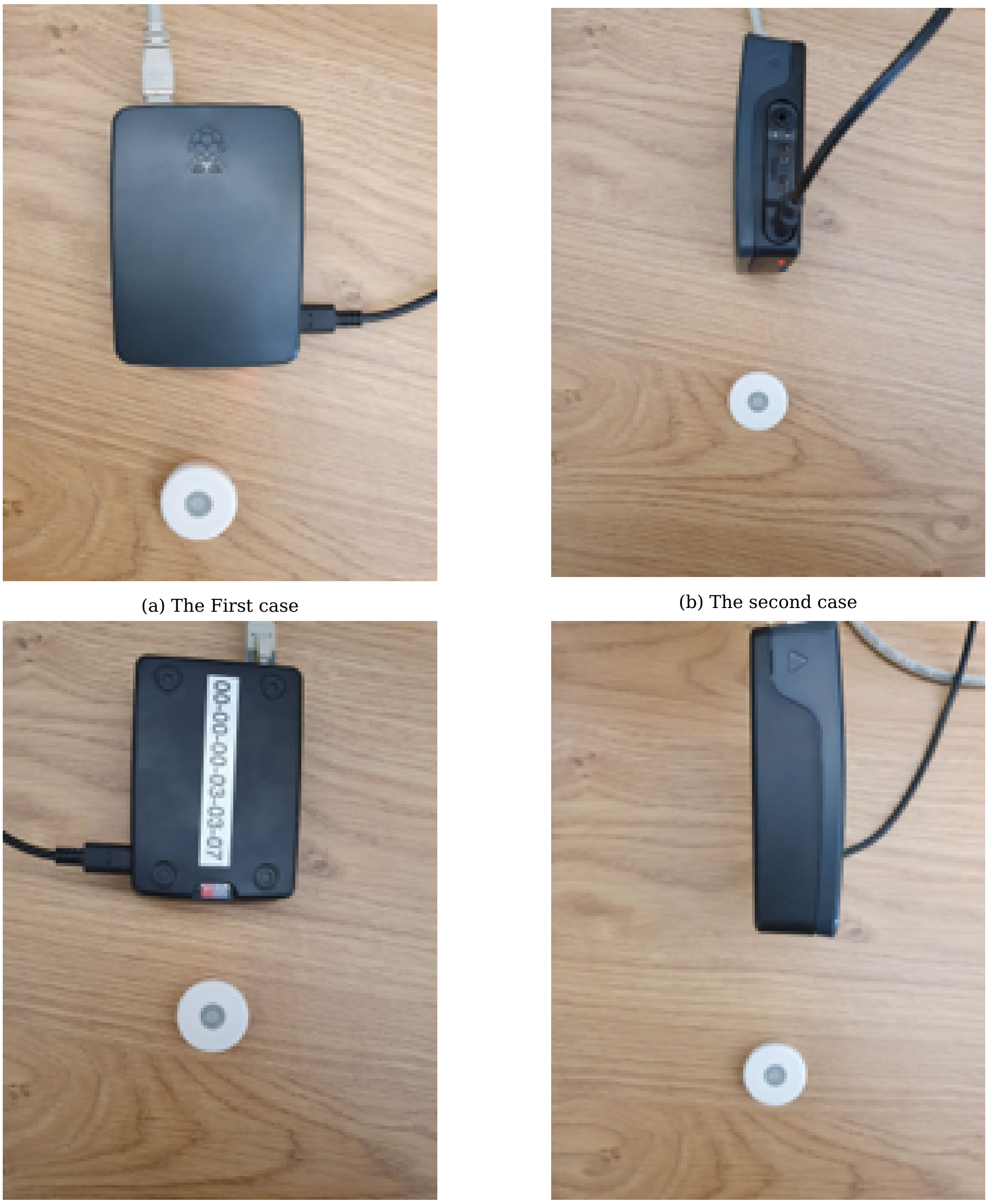


Figure 4.24: Raspberry Pi 4 Model B



(a) The First case

(b) The second case

(c) The third case

(d) The fourth case

Figure 4.25: Raspberry Pi 4 with plastic cover in RSSI comparison analysis with Raspberry Pi rotation.

Due to the position of the Bluetooth sensor in the Raspberry pi 4 (Figure 4.24) moreover the various rotations of the Raspberry pi in different cases, we collected all the data received from each case, including to be the same test conditions, all 4 cases were performed in the same time. As a result, all received data as presented in the chart 4.27.

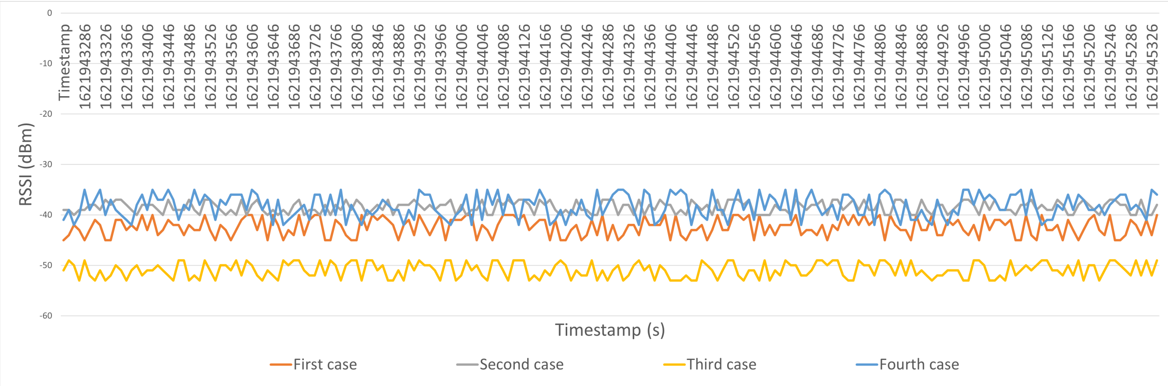


Figure 4.26: Difference of RSSI through rotating Raspberry pi.

Raspberry Pi(s) 4	**RSSI (Average)**	**Number of packets**	**Test duration**
First Case	-42.6 (dBm)	210	2090 (s)
Second Case	-38.4857 (dBm)	208	2090 (s)
Third Case	-51 (dBm)	211	2090 (s)
Fourth Case	-38.509 (dBm)	209	2090 (s)

Table 4.21: Analysis of data received through rotating Raspberry pi 4.

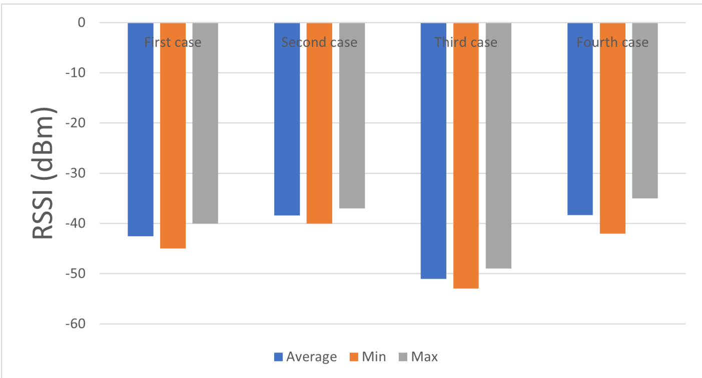


Figure 4.27: Review of data received through rotating Raspberry pi 4.

As illustrated in Table 4.21 and Figure 4.27, the number of received packets in various rotations is almost the same, except the received signal is different.

It is evident that the average amount of the received signal in the case of numbers two and four is practically the same, and this is because if we see the position of the Bluetooth sensor in Figure 4.24, the BLE beacon is closer to the sensor and therefore the received signal strength is more significant than in the case of number three, the average signal is different due to the presence of obstacles.

Based on the experiment, we conclude that the installation location of the Raspberry Pi in the hospital is critical, and by rotating the Raspberry Pi, we receive different signals, so in Training and Testing mode, the position of Raspberry Pi does not change because we will have noise and failure in the system.

Chapter 5

Data analysis and Modeling

5.1 Data Cleaning

One of the main steps in the overall process of data processing is data cleaning. Data cleaning, additionally called data cleansing or scrubbing, deals with detecting and removing errors and inconsistencies from data to improve data quality.

Data quality problems are present in single data collections, likewise files and databases, e.g., due to misspellings during data entry, missing data, or other invalid data. For instance, in test (4.2.3), we have invalid data in the database.

Because every practical project has noise and error, we must clean the data before creating a model and using Training data. Similarly, in our project, we have to do the same because the data is in the signal format, and the external noises affect them, therefore the steps in which the data will be stored in the Elastichsearch, and in each of these steps, there could be noises that will disturb the system; therefore, to solve this problem after extracting the data from the database, we have to clean the data urgently.

Likewise, in the Testing section, these clearing operations are performed because storing data in Training and Testing is not much different, which we will discuss in more detail in the next chapter.

In the data cleaning section (Function A.2) of our project, there are two steps. First, some of the data was incorrect when collected, which has no meaning at all, and for instance, in the RSSI value, we received the value -600 (dBm), and this Certainly, there is an error in storage because the RSSI value is between 0 and -100, and this is typical in storage projects because much data is sent to Logstash at any given time, and there may be an error in some parts.

The second step of cleaning the data according to the data storage format (A) is related to data that have less information from each room number. (Data B)

For example, As illustrated in Figure 5.1, we extract the stored data of the Training section (Function A.1) from room number "2-4-1" (second floor, zoom number four and first room); we will see a summary of the data in Table 5.1.

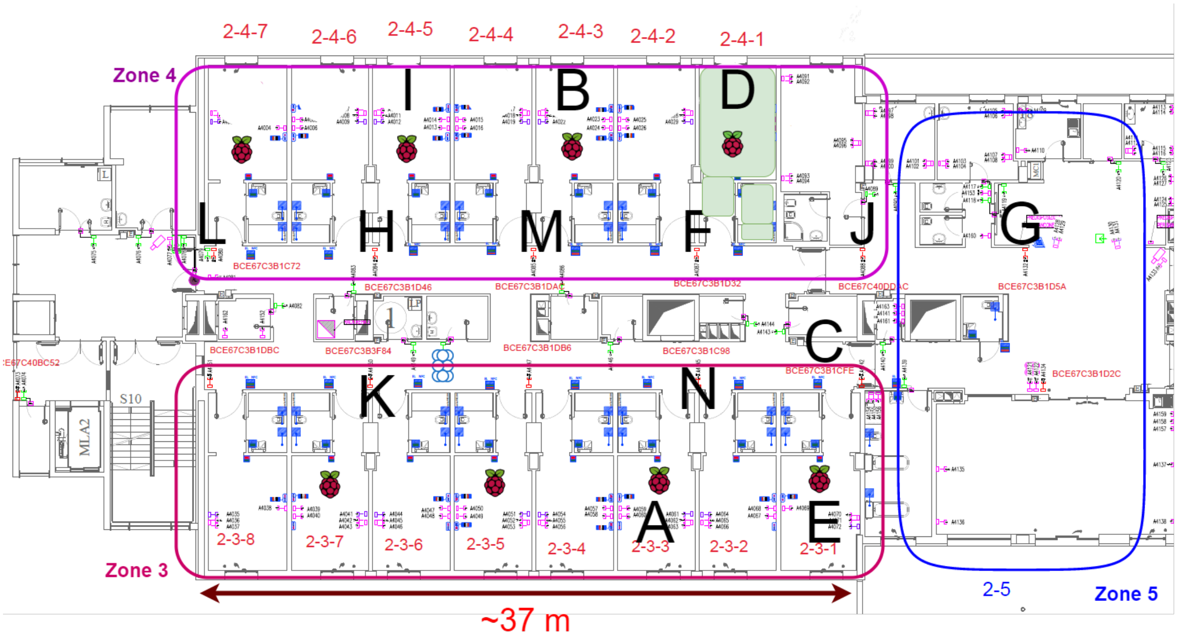


Figure 5.1: Map of the second floor.

Label on map	Antennas	Number of records
A	00-00-00-02-03-03	146
B	00-00-00-02-04-03	6265
C	BC-E6-7C-3B-1C-FE	251
D	00-00-00-02-04-01	6265
E	00-00-00-02-03-01	152
F	BC-E6-7C-3B-1D-32	1107
G	BC-E6-7C-3B-1D-5A	35
H	BC-E6-7C-3B-1D-46	16
I	00-00-00-02-04-05	314
J	BC-E6-7C-40-DD-AC	955
K	BC-E6-7C-3B-3F-84	3
L	BC-E6-7C-3B-1C-72	10
M	BC-E6-7C-3B-1D-AC	360
N	BC-E6-7C-3B-1C-98	313

Table 5.1: Data received in the Training section of Room No. "2-4-1". The number of records is over 282 are shown with the green highlight, and the ones lower than 282 are shown with the red highlight. (282 is the Median number in the list)

As reflected in table 5.1, we have numerous records for room "2-4-1" without cleaning the data. Regarding the test chapter, the antennas are further away; we received few packages; for example, the antenna with the ID "00-00-00-02-04-01" has the highest record in the database, which is related to the Raspberry pi that is installed in room number "2-4-1" behind the TV.

As it is clear, the more antennas in Training, the more features to build the model, so we have a better model; however, to create a great model, the compelling features must be used. Accordingly, it is better to ignore these antennas (features) to create a good classification model; we use antennas that receive data from the identical antennas in Testing mode.

Antennas	**Number of records**
00-00-00-02-04-03	4643
00-00-00-02-04-01	6265
BC-E6-7C-3B-1D-32	1107
00-00-00-02-04-05	314
BC-E6-7C-40-DD-AC	955
BC-E6-7C-3B-1D-AC	360
BC-E6-7C-3B-1C-98	313

Table 5.2: All data for room number "2-4-1" which is used after cleaning the data.

The antennas we use are for room number "2-4-1" in table 5.1. We do not consider antennas with less than 300 records. Also, As reflected, we did not consider antennas that placed at great distances.

We normalized the RSSI values using (Equation 5.1) to simplify and to speed up the training process. After this normalization process, every RSSI value is in the 0–1 range.

$$RSSIn_i = \frac{(RSSI_i + 100)}{60} \tag{5.1}$$

5.2 Design the model

In this section, we will describe the Recurrent Neural Networks model for classification and analyze the accuracy.

The RSSI probability density function (PDF) is considered to have empirical parametric distributions (e.g., Gaussian, double-peak Gaussian, lognormal), which may not be undoubtedly accurate in possible situations. In order to establish better performance, non-parametric methods did not make any assumption on the RSSI PDF; however, they require a large amount of data at each reference point (RP) to form the smooth and accurate PDF.(He, 2016)

Because the number of rooms in the hospital are so much, Does reducing classes in a classification method improve accuracy?

It will enhance our performance if the classes we combine are similar and have many misclassified samples because it will decrease the errors. However, if the classes are not similar in the group, it will most likely not improve our accuracy since we will not decrease errors. Because the data we receive from each room is very similar, and in some cases, we receive data from the antennas of another hall, it is sufficient to do zoning to reduce the number of classes. As illustrated in Figure 5.1, we categorized each floor into different zones.

Applying the corridors, we separated the zones, and in each zone, we have several classes; for instance, The first floor is divided as chart 5.1.

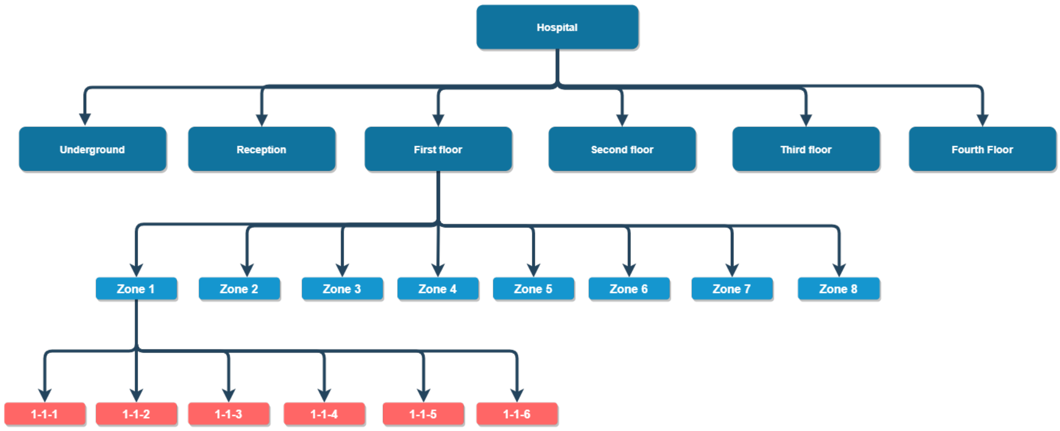


Figure 5.2: Classification of the first zone of the first floor of the hospital.

As reflected in Figure 5.2, the classification of the first zone of the first floor of the hospital shows that the other floors are classified in the same way. However, some zones have two rooms on each floor.

5.2.1 Recurrent Neural Networks (RNN)

Neural networks are an excellent selection for classification problems. It has been used for many modern classification problems, such as handwriting recognition, self-driving car, speech recognition, and sentiment analysis.

RNN is related in many ways to a feed-forward Neural network. The main difference is that RNN does not always have a forward connection. Instead, a recurrent connection connects a neuron from the current layer into a neuron in the previous layer or into the neuron itself. The recurrent connection allows the networks to make use of past contexts. The example of a recurrent connection is defined in Figurer 5.3.

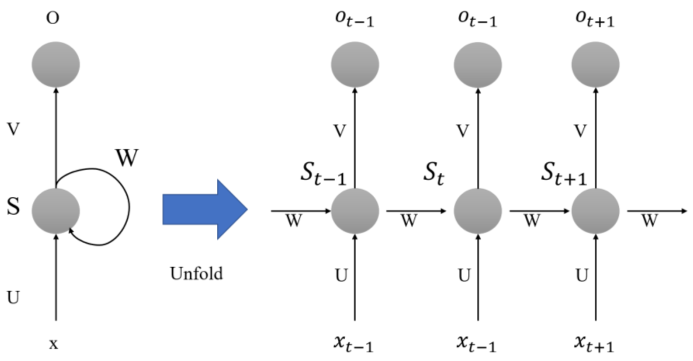


Figure 5.3: Recurrent Connection and the Unfolding in Time

As reflected in table 5.3 are the parameters used for RNN model training. We used k-fold validation for our RNN model evaluation. Our model is constructed with two fully connected recurrent layers with a "SoftMax" activation function. We also used categorical cross-entropy between predictions and targets as the loss function for Training and validation.

Parameter	**Value**
Class count	100 classes
Recurrent layer count	2 layers
Recurrent layer type	Elman Simple RNN
Neurons in the recurrent layer	128 neurons
Neurons in the input layer	150 neurons
Optimization method	Adam Optimizer
Learning rate	0.0025
Number of epochs	1000 epochs
K in k-fold validation	10 folds

Table 5.3: Parameters used for Training in RNN model

According to table 5.4 and Figure 5.4, for every Fold, the training process took about four hours to be completed because we only used the CPU. To complete the Training and evaluation of all the folds we need about 40 hours. The average accuracy of our RNN model with 100 classes is 82.51%. This result is high.

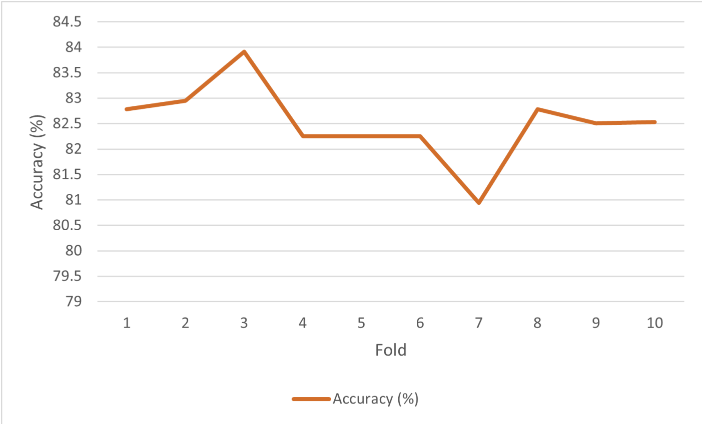


Figure 5.4: Accuracy of RNN model based on the number of folds.

Fold	**Accuracy (%)**
1	82.78
2	82.95
3	83.91
4	82.25
5	82.25
6	82.25
7	80.94
8	82.78
9	82.51
10	82.53
Average	**82.515**

Table 5.4: The precision of the RNN model based on the number of folds.

We have successfully developed the RNN model, trained and evaluated the model. Based on the experiment results, we can conclude that RNN produces adequate accuracy. Our model still needs some tweaking on training parameters to be able to surpass K-Nearest Neighbors performance. Note to mention that our model's performance is affected by the fluctuation of the loss value during Training. We suggest using an artificial intelligence (AI) system alongside it to reduce the noise and increase accuracy. We will explain this system's artificial intelligence system and diagram separately in the feature research.(Yogyakarta, 2017)

Chapter 6

Testing

6.1 Introduction

The testing section is very critical in this project because all cases and details must be addressed. As mentioned already, the hospital has six floors; on each floor, several Raspberry pies and antennas have been installed. Properly have those antennas in the database; otherwise, according to the examinations performed in the previous chapter, the positioning will behave inaccurately. Furthermore, the time of sending packets from the antennas to the database is not Synchronize.

6.2 Data Storage and Time Synchronization

In the Training section, we did not have a problem with the data submission time because time does not affect our data. Although in the testing section, receiving the packages is essential for us too because the patients move at any time, and we have to predict the patient's position in the shortest time.

For this reason, we must synchronize the data we receive so that we can extract the best data from the database and clean it, and then use the model we created to predict the patient's position.

As illustrated in Figure 6.1, the diagram shows the data received from each of the antennas to the database.

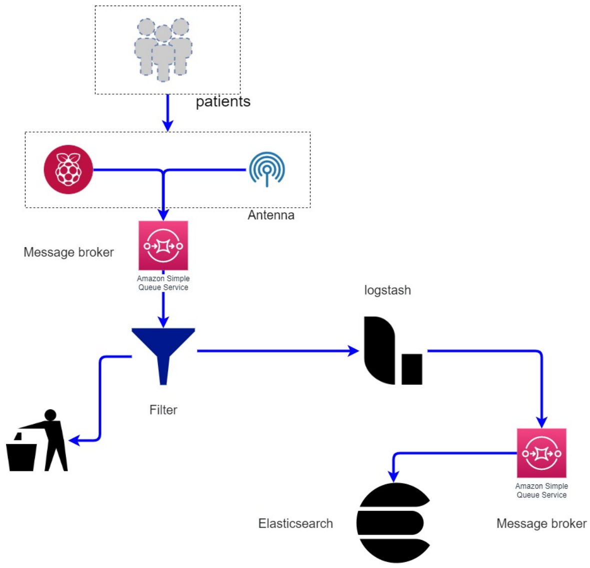


Figure 6.1: Data storage diagram of antennas in Elasticsearch.

According to Figure 6.1, the data received from the antennas and Raspberry pi (s) are sent to the address specified in the antenna and Raspberry pi (s) configuration (Post).

As a consequence, much data is sent. Further, each record sent from the antenna to the server has time to use the server time for convenience. Because a considerable amount of data is sent to the server at any given time and the database alone can not receive, analyze, and store all these packages in the database, designing a system to solves this problem.

Therefore, using Message Broker, we queue all the packages we receive. Moreover, by using the filter we have, we send only the BLE beacons we want to the next step and discard the rest. Finally, in the Logstash section, we convert the data to the desired format and store it in the database. For example, one of the sample data in the Testing section is as follows.(Data B)

```
% Sample format of testing section
{
    "bt_mac": "User1",
    "timestamp": 1502984373838719200,
    "Data_antenna":
        [
          {
            "ap_mac": "antenna1",
            "bt_rssi": -48
          },
          {
            "ap_mac": "antenna2",
            "bt_rssi": -94
          },
```

```
            {
              "ap_mac": "antenna3",
              "bt_rssi": -90
            },
            {
              "ap_mac": "antenna4",
              "bt_rssi": -86
            }
          ]
  }
```

As mentioned earlier and shown in Figure 6.2, each antenna does not send its data in synchronized form. For instance, when the second packet is sent from the third antenna, we may send the third packet from the second antenna at the same time. Therefore, to solve this problem, we consider all the before and after data to not miss any data using a virtual window. All the data in this window is normalized using normalization methods and is prepared to predict the patient's position.

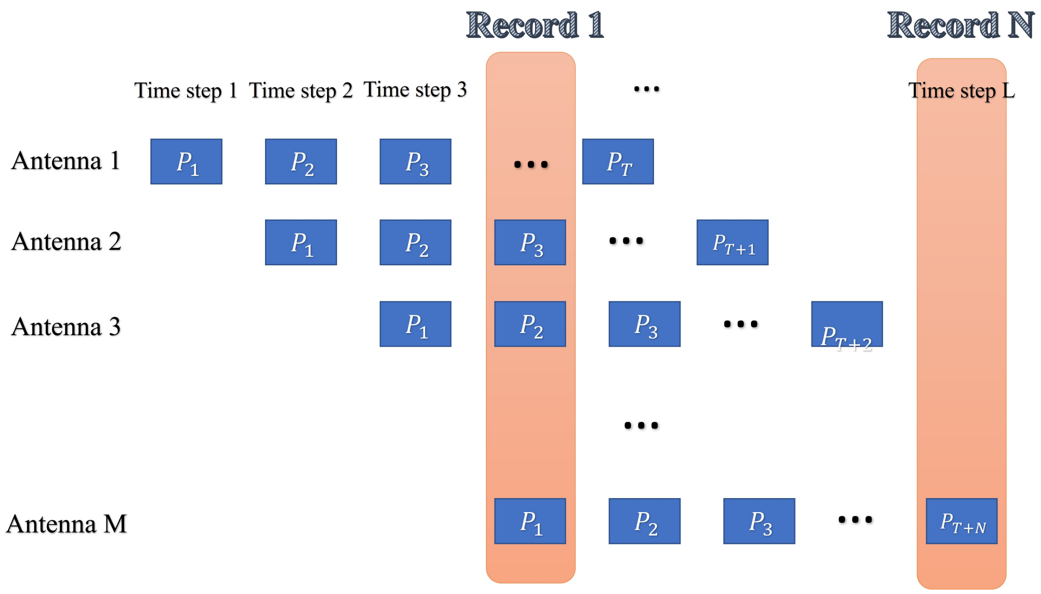


Figure 6.2: Antenna time diagram.

6.3 Control the antennas in real-time

In total, we installed around two hundred antennas and Raspberry pi (s) throughout the hospital, which in the Training section at each time to store and collect data for each room independently checked to ensure that the antennas are working correctly. However, in the Testing section, this control must be done in real-time because if there is a problem with any of the antennas, the positioning will be confused.

For instance, in Room No. 2 in the Training section, we received data from four antennas A, B, C, and D; furthermore, we collected all that data, and based on that, we created our model. However, in the Testing section, despite the problem, it is stated that one of the antennas does not work correctly at the moment, and we do not have information about that antenna in the database, so the model is not able to position correctly because that feature no longer exists.

To solve this problem, we can control in real-time whether the antenna is on or off by using the cloud management system for antennas.

6.4 Workflow of Testing

When the data is collected in the database in real-time, the correct operation of the antennas is guaranteed, and we can use our model to predict the patient's position.

In general, the indoor positioning system operates according to Figure 6.3, each element in a figure will be examined separately in the next section.

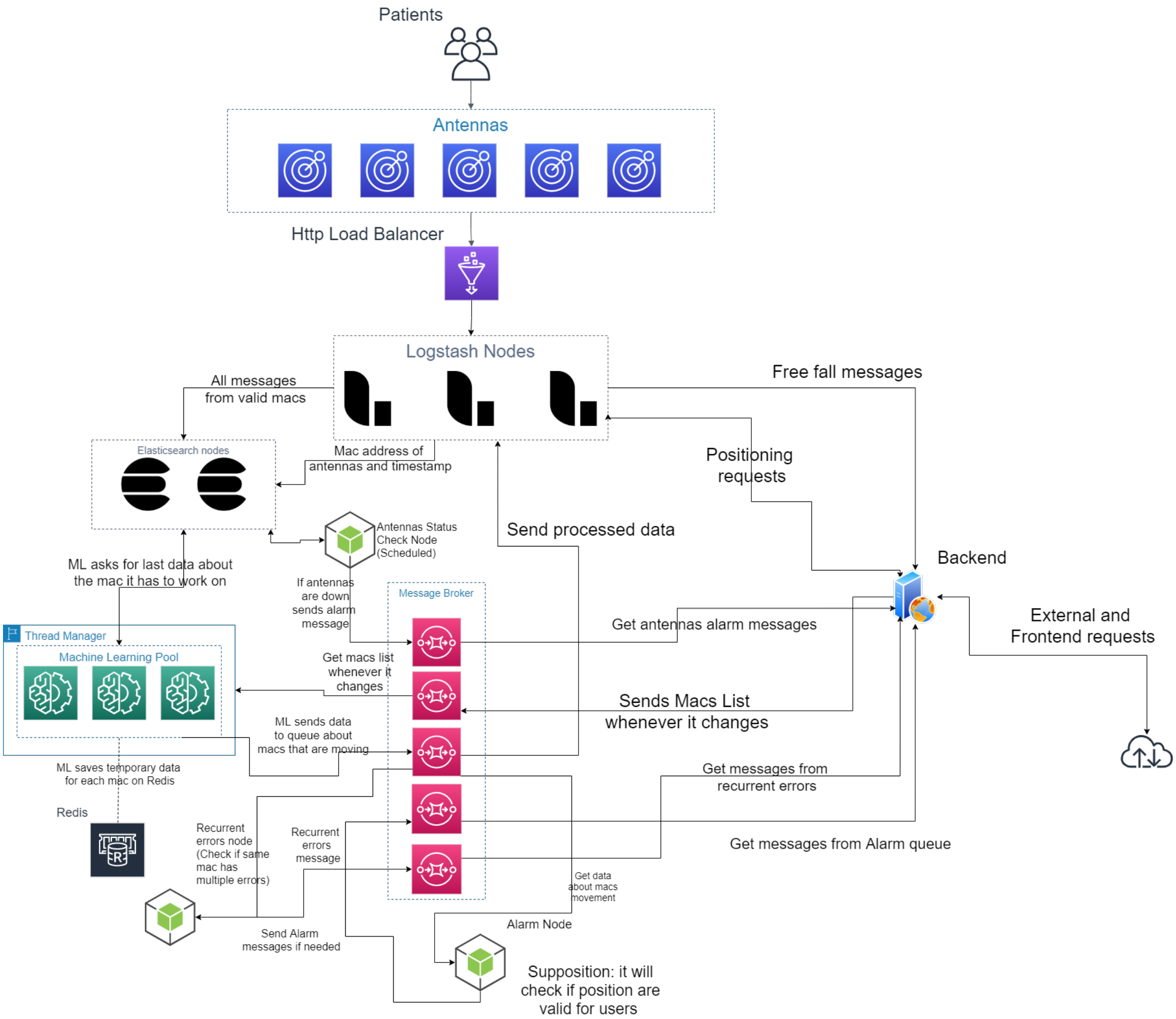


Figure 6.3: Workflow

According to Figure 6.4, there are several steps involved in predicting a patient's position. Each of the sections has more details that we will cover.

- Initial control of all important parts before starting the system.
- Receiving data and storing it in the database. (6.2)
- Extract the latest data received from Elasticsearch for the specified patient.
- Clean and normalize data.
- Predict the floor number.
- Predict the zone number.
- Predict the Room number.

- Checking the positioning accuracy using artificial intelligence algorithms (explained in more detail in the next chapter).
- Controlling the displacement of the patient's position using the previous position.
- Announcing the patient's position.

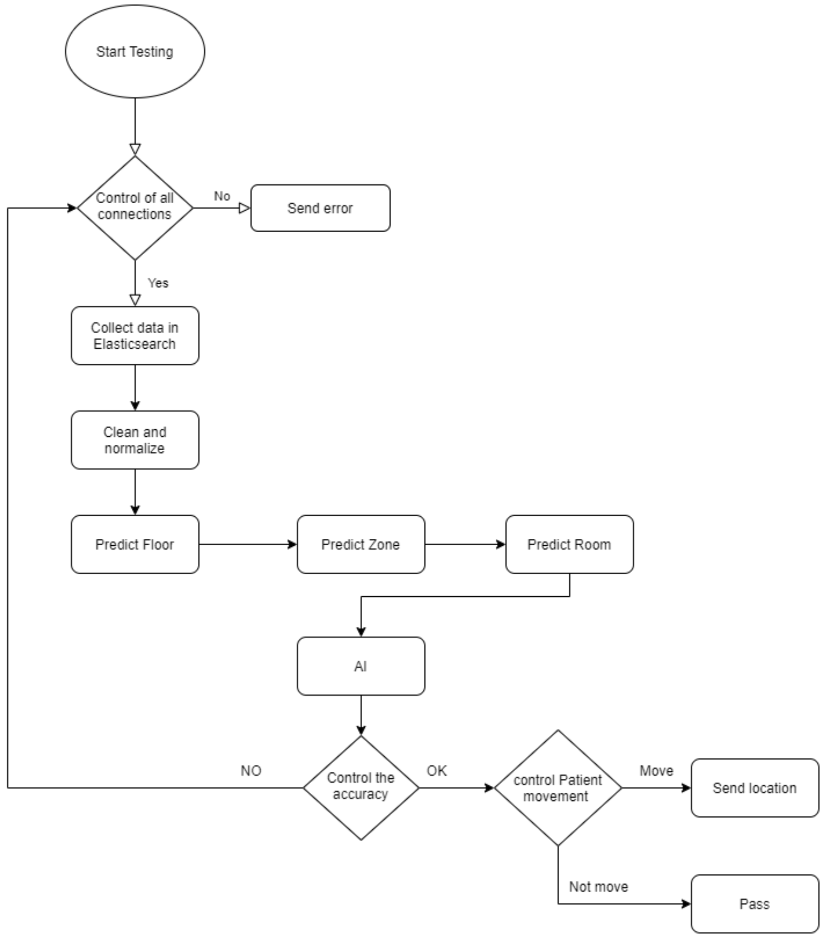


Figure 6.4: Flow chart of the testing section

6.4.1 Analysing the connectivity

According to Figure 6.5, it is a summary of the connections of the project to other components. Therefore, before starting the positioning system, all the connections should be verified, and in case of any problem in each system, it should automatically make the right decision.

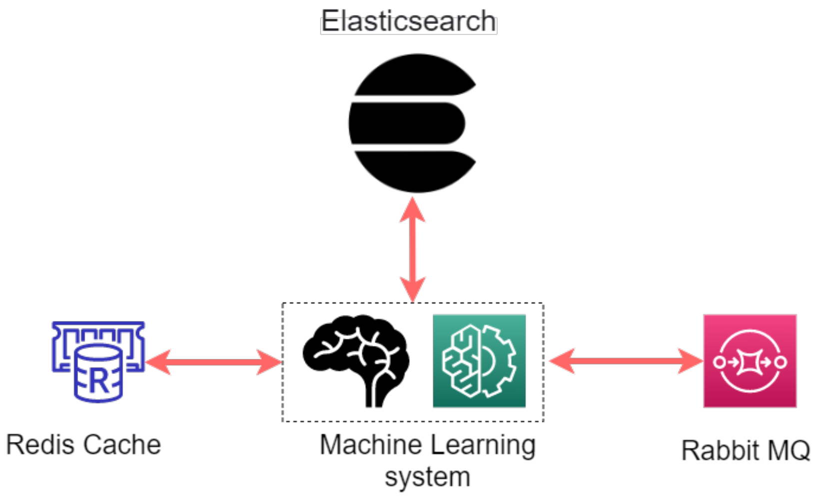


Figure 6.5: General diagram of connections of other components to the positioning system.

Given that this project operates automatically and must make the best decision automatically in all problems at different times, we need to design a system that makes the right decision according to the circumstances.

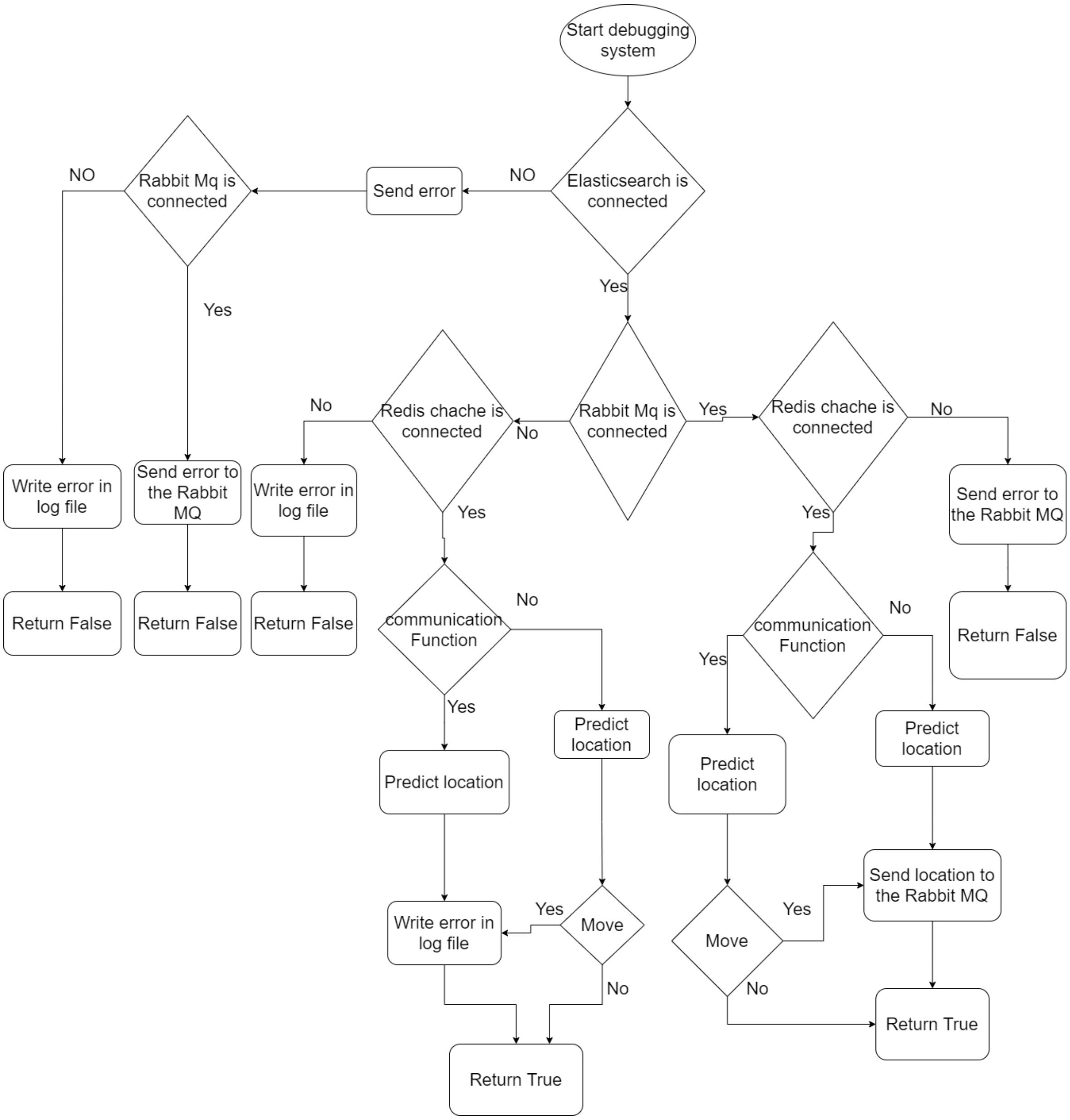


Figure 6.6: Flow chart of the error function.

As illustrated in flow chart 6.6, several components need to be checked before starting the system, each of which we will address here.

Elasticsearch control

Elasticsearch is constantly storing new data, and if we want to do new positioning, we need to extract the latest BLE beacons-related data from Elasticsearch. Therefore, if the Elasticsearch connection is disconnected and it is impossible to extract new data, it is impossible to locate, and the system will receive an error, and the Backend must be informed of the error.

The first error is that Elasticsearch is not connected, and the second is that the user we are looking for does not exist. In other words, the user with the desired ID is not registered in the database.

The latest control we need to look at in Elasticsearch Section is the expiration of the information in the database. For instance, if we want the position of patient A, we have to extract the latest information stored in bracelet A. Following extracting and checking the data storage time, we found that the latest data is related to the previous day, and if the same expired data is used, positioning is completed, but it is not real. Firstly, one of the most common causes of this issue is that the bracelet runs out of battery, and secondly, the bracelet may be broken. However, due to the control of time expiration in this section, it is possible to prevent this error and if this error occurs, send the error to the Backend.

RabbitMQ control

Rabbit MQ has to handle all the output of the positioning system with Backed. In other words, if the patient's position is predicted or if there is an error in the system, we must send the output to the queue, so if the Rabbit MQ connection is not established, it will be impossible to send information.

However, positioning is possible, and it is possible to do positioning despite the antenna data and being connected to Redis, but the output must be written to the log file. For this reason, after completing the positioning and announcing the lack of communication with Backen in the log file should be written. Moreover, the final output of the system is declared positive (True) because there is no problem in positioning.

Redis cache control

Redis (Function A.4) has two critical duties in the positioning system. The first task of Redis is that if it does not connect to RabbitMQ, then Redis must be known; according to the communication function, it will be declared that communication was not possible, and we wrote the output in the log file. Additionally, another purpose of Redis is to be an auxiliary memory for the AI section, which we will discuss in more detail in the AI chapter.

In conclusion, if we do not connect to Redis, Indoor positioning is not possible, and the system can not position, so the system output is negative (False).

Communication Function

As described in the previous sections, in this function, if RabitMQ is not connected and Redis is connected, we set the flag of communication as false means we do not communicate with RabbitMQ; then, In the next step, if all connections with RabbitMQ and Redis are valid. Then, we can predict the patient's position; we have to send the last position to the system at the first opportunity.

Monitor patient movement

One of the components shown in flow chart 1 is the patient's movement monitoring function. In this section, having the patient's previous position and the patient's predicted position, we can also check whether the patient has moved or not.

This function is designed to notify the Backend system only if the patient is moved. For this purpose, fewer messages are sent between the positioning system and the queue so that we do not have a problem losing messages in the future. This previous patient information is stored in Redis, and if the movement is confirmed (high probability), which is controlled by the artificial intelligence function, the current position will be updated in Redis.

6.4.2 Predict position of the Floor number

As mentioned in the previous chapter, due to the large area of the hospital, the positioning system was divided into various sections.

We required first to predict the patient's position in the floor number. In this section, we have a list of all the antennas and their IP Id and the places where they are installed. Then, using the Python dictionary, we categorized the location of the antennas based on the number of floors.

After extracting the latest available data from Elasticsearch and mapping the data with the existing dictionary, the floor number can be predicted. However, in some cases, such as when the patient is in an elevator or on the stairs, we may additionally receive two or more antennas from different floors. To solve this problem, we were able to sort the signals received from each antenna and setting the coefficient based on the power of each signal.

For instance, if we received the data in Table 6.1 from Elsatichsearch and the category classification dictionary is in Table 6.2, the calculations are performed as follows.

Antennas	**RSSI (dBm)**
BC-E6-7C-3B-1D-B2	-95
58-C1-7A-C5-B0-48	-90
BC-E6-7C-40-DD-4A	-76
BC-E6-7C-40-DD-9C	-81
BC-E6-7C-40-BA-D8	-79

Table 6.1: The latest record of extracted data for a random BLE beacon.

Antennas	Floor number
58-C1-7A-C5-B1-F6	1
58-C1-7A-C5-B0-48	1
BC-E6-7C-3B-1D-B2	1
BC-E6-7C-3B-1D-3C	1
BC-E6-7C-3B-1D-5A	2
BC-E6-7C-3B-1D-2C	2
BC-E6-7C-40-DD-10	2
BC-E6-7C-40-DD-4A	2
BC-E6-7C-40-DD-9C	2
BC-E6-7C-40-BA-D8	2
BC-E6-7C-40-DE-04	2

Table 6.2: Part of the dictionary of antenna IP classified by floor number.

$$List_{\alpha} = Map(Min(RSSI)) \times \alpha \tag{6.1}$$

$$List_{R} = Map(All - List_{\alpha}) \tag{6.2}$$

Max frequent element

$$Max_{f} = Max([List_{\alpha} + List_{R}]) \tag{6.3}$$

For Tables 6.1 and 6.2:

$$Min(RSSI) = "BC - E6 - 7C - 40 - DD - 4A" \tag{6.4}$$

$$\alpha = 2 \tag{6.5}$$

$$List_{\alpha} = [floor2, floor2] \tag{6.6}$$

$$List_{R} = [floor1, floor1, floor2, floor2] \tag{6.7}$$

Max frequent element:

$$Max_{f} = [floor2] \tag{6.8}$$

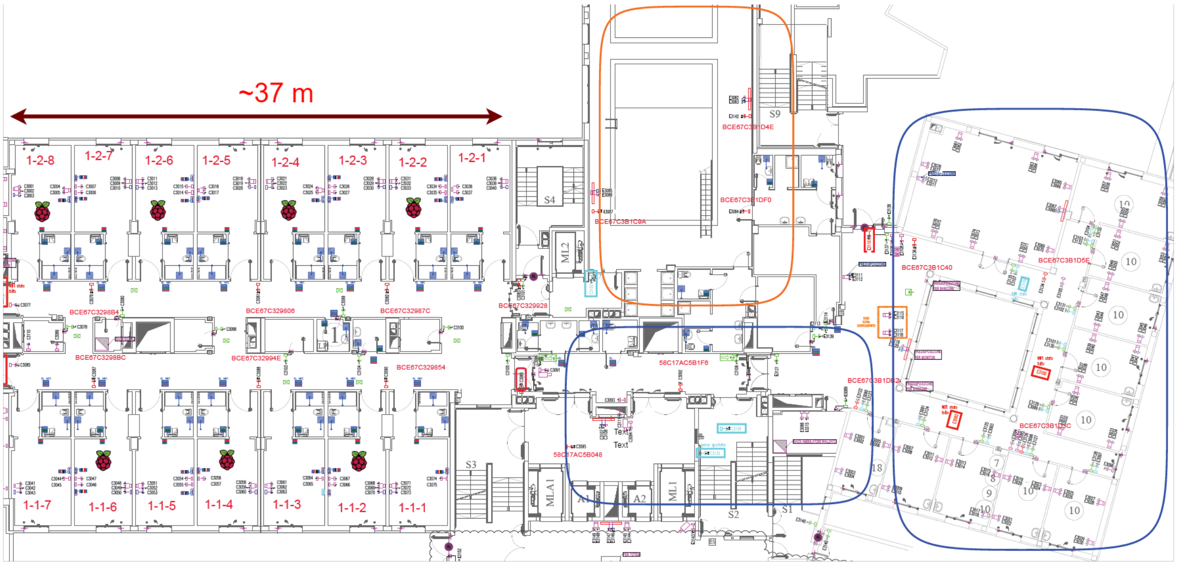
Probability:

$$P = (\frac{4}{6}) * 100 = 66.66\% \tag{6.9}$$

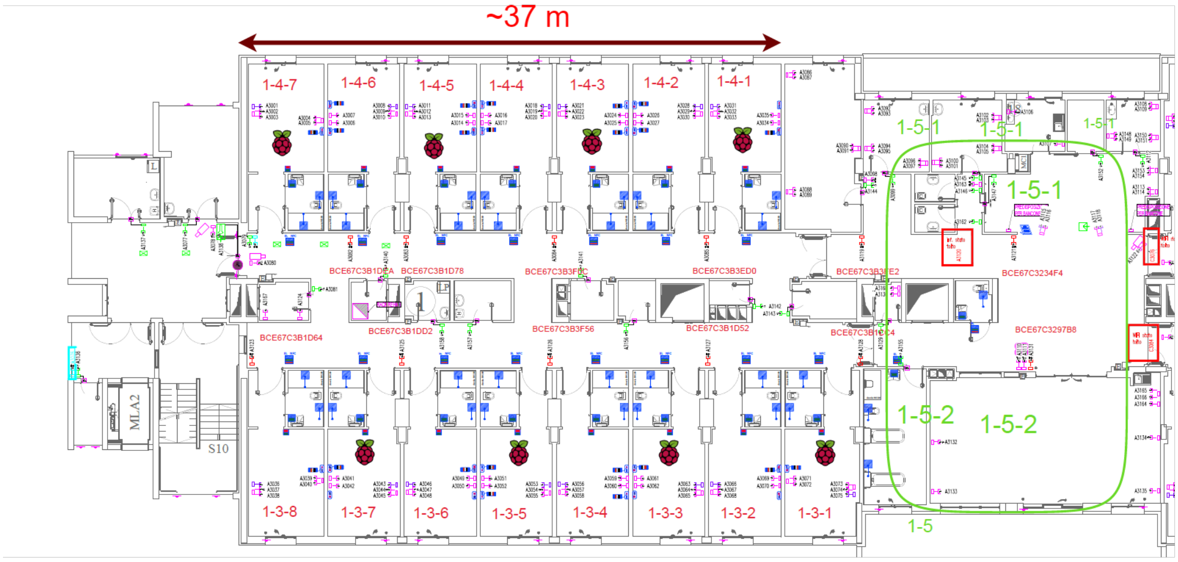
As illustrated in 6.9, the probability that the patient is on the second floor is 66.66 %. For the positioning system to work more precisely, we can not position the position when the probability is less than the set value(configuration file), and it sends to the system the error of not predicting the position of the floor. Furthermore, when the probability of predicting the floor is excellent, we move on to the next stage of prediction, which is to predict the zones.

6.4.3 Predict position of the Zone number

As reviewed in the previous sections, each floor is divided into different zones. This organization is the result achieved in the experiments section (4.2). We examined the threshold of the signal received in the antenna, and as reflected in map 6.7, the zones are organized according to the separation of rooms by corridors. To put it another way, we took advantage of the structure of the corridors and specified each side of the floor as an independent zone.



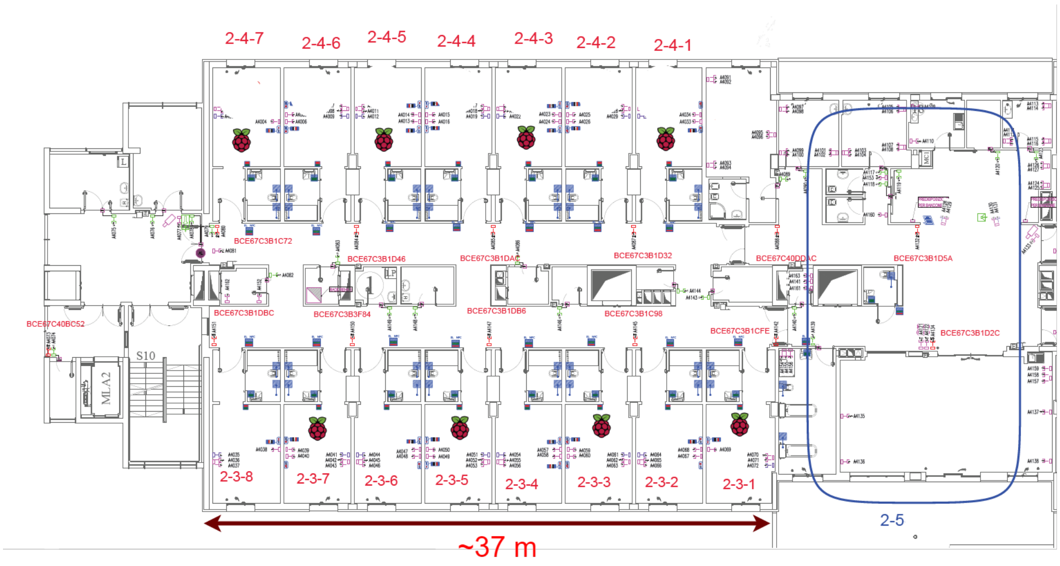
(a) First floor (zones 1,2,6,7)



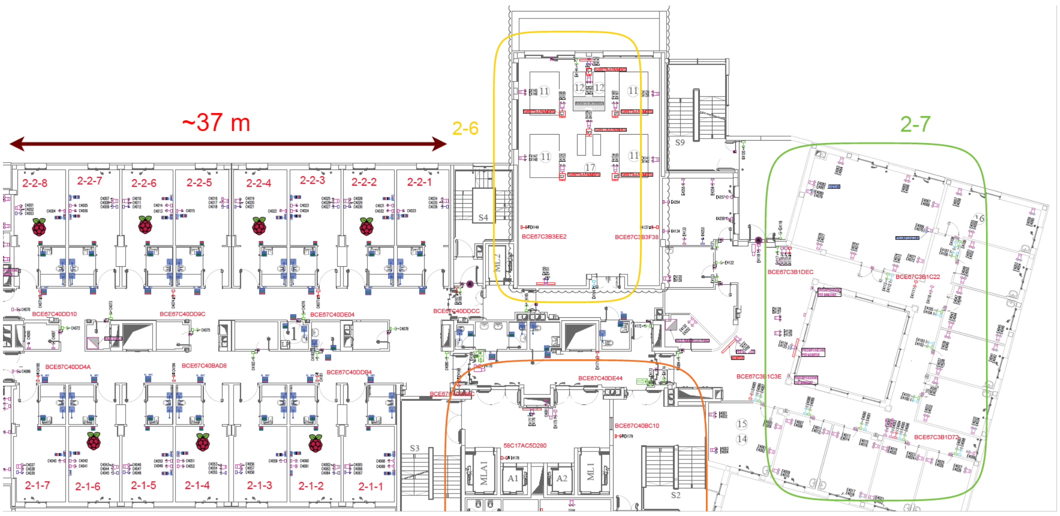
(b) First floor(zones 3,4,5)

Figure 6.7: Map of the first floor of the hospital and organization based on zoning.

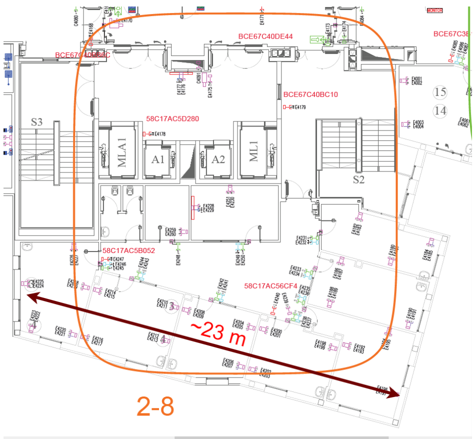
As can be seen in Map 6.7, it corresponds to the maps of the first floor of the hospital, which is generally divided into seven zones. Zones one, two, three, and four correlate to the zone where the patient room is located where the accuracy of positioning in these areas should be high. Moreover, zone five (1-5) is related to the nursing and dining departments, and zone six (1-6) and seven (1-7) are related to the pool and the doctors' room department, respectively.



(a) Second floor (zones 3,4,5)



(b) Second floor(zones 1,2,6,7)

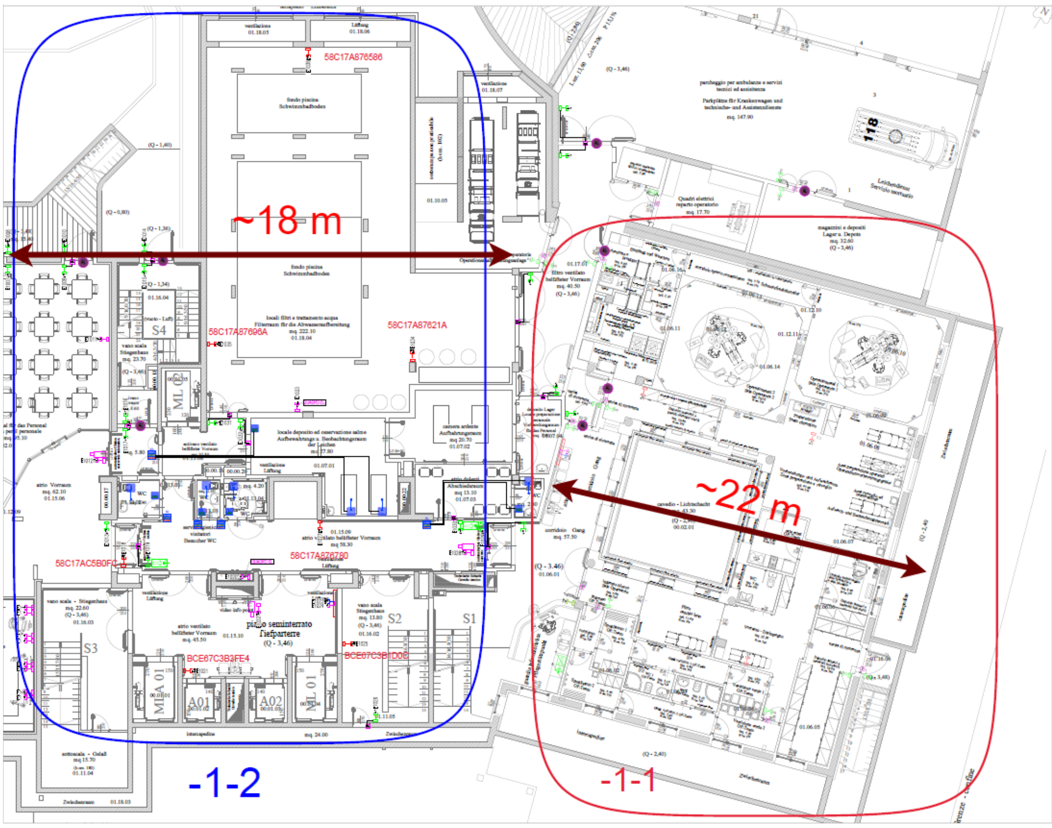


(c) Second floor(zone 8)

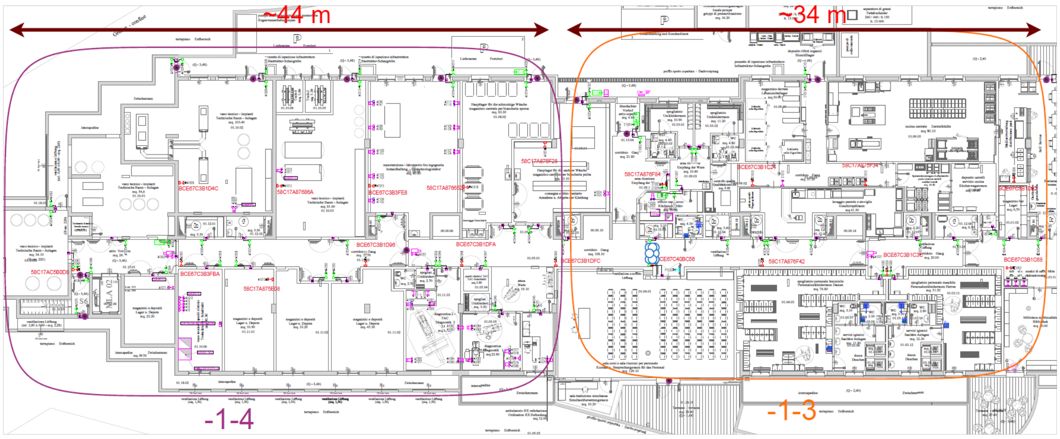
Figure 6.8: Map of the Second Floor of the hospital and organization based on zoning.

As it is evident in Map 6.8, it corresponds to the maps of the second floor of the hospital, which is generally divided into eight zones. Zones one, two, three, and four correlate to the zone where the patient room is located where the accuracy of positioning in these areas should be high (Similar to the first floor). Moreover, zone five (1-5) is related to the nursing and dining departments, and zone six (1-6) and seven (1-7) are related to the gym and the Physiotherapy department, respectively. On this floor, we have another zone, which is the doctor's room section, which is marked with zone eight (6.8c).

The third and fourth floors of the hospital are the same as the first floor (6.7) of the hospital and are divided into seven zones.



(a) underground (zones 1 and 2)



(b) underground (zones 3 and 4)

Figure 6.9: Map of the underground of the hospital and organization based on zoning.

Though the underground has a different map and different zoning, due to the low probability of the presence of patients in the underground, the divisions have been made more widely.

As reflected in Map 6.9, the underground is divided into four zones, the first of which deals with surgery rooms and the second of which deals with stair, pool facilities and

Nurses and doctors Restaurants. The central kitchen and cinema are located in the third zone, and finally, the fourth zone is related to the general facilities of the hospital.

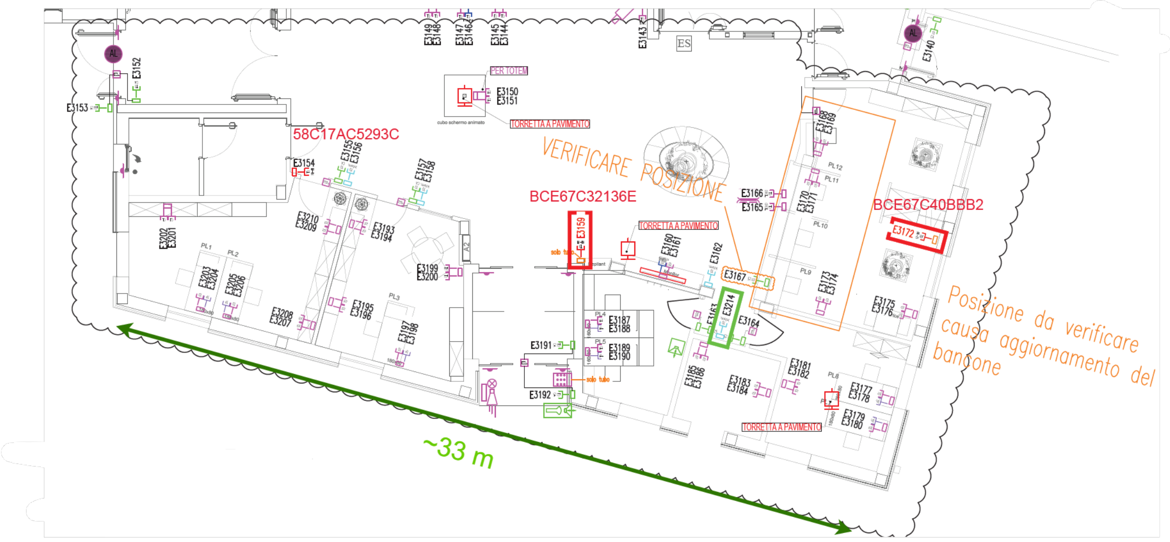


Figure 6.10: Hospital reception map.

As depicted in Map 6.10, it belongs to the ground floor of the hospital, which is generally divided into awards due to its small zone.

As reported in the hospital maps, the zones where the patients' rooms are and the patients are more likely to be present have been made more accurate and smaller divisions, and in other parts, the divisions have become more general.

Predicting the patient's position in the zones is the same as positioning the floor number, except that several coefficients have been added. In other words, the lowest value (RSSI) is given the maximum coefficient, the second-lowest value is given the lowest coefficient compared to the first coefficient, and the third coefficient is given less than the second, and the probability is given less than the Second calculated identically.

Furthermore, if the probability is low, regional positioning will not be possible, and it will be sent to the error system.

There are two purposes for the low regional probability.

The first is due to defects in the antennas on the floors that the antenna may not work correctly. Furthermore, the second reason is when the patient is placed between two zones, so the system cannot predict the zone. To solve the second problem, we will deal with this problem using the artificial intelligence algorithms described in the AI chapter.

6.4.4 Predict position of the Room number

Hitherto we have been able to predict the patient's position from the floor and the zone. The last step is to predict the patient's position in the room where he is located.

Abundantly clear that the deeper we get into the system of positioning, the process will be more laborious. Further, the probability of error is higher. For illustration, the probability of a floor number prediction error if the antennas work well and we do not have an error in the system's background is about 8%, which is when the patient is in the elevator we are between floors.

Although, the probability of system error without using an artificial intelligence system to predict room number is more than 20%.As depicted in workflow 6.11, In the positioning system, each divided zone has a model which, after predicting the area, is given to the model using the latest data extracted from Elasticsearch and normalized and cleaned to predict the patient's position.

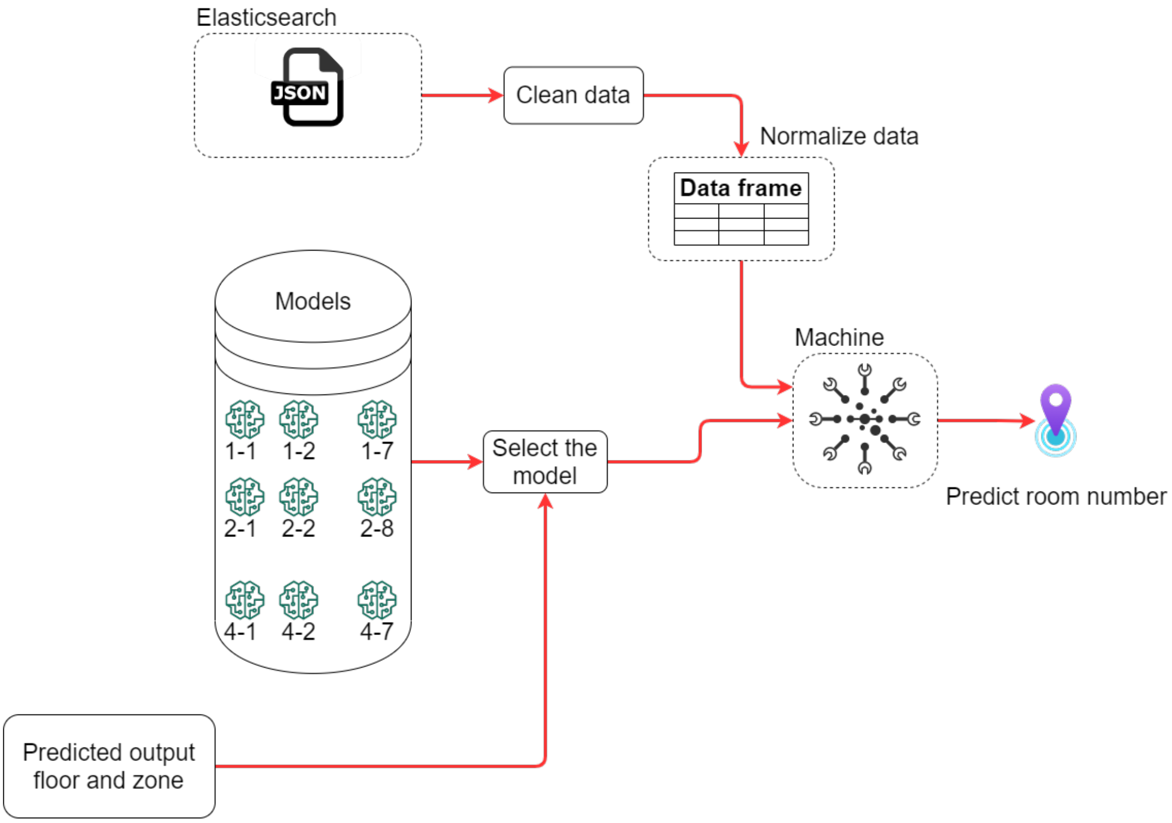


Figure 6.11: Flowchart of prediction room number.

The predicted position will not be sent to Backend because two more steps are needed to check the correctness of the position. These two steps are because we will have a minor error in the system. After all, if they have a false error, nurses will not trust the system after a while.

For this reason, after the prediction is made, the obtained position is sent to the artificial intelligence section, and then it is checked whether the patient has been relocating it or not. We will consider these issues in more detail in the next chapter (7).

Chapter 7

Artificial Intelligence & Multithreading

7.1 Artificial Intelligence

An intelligent positioning approach based on artificial intelligence algorithms has been presented in an indoor positioning for accurate positioning and decision-making through adaptation and learning. Firstly, Implement a system such as Global Navigation Satellite System (GNSS) that stores a person's last position and makes accurate predictions based on it. Then, control the correct positioning in the Testing step.

As demonstrated in the machine learning chapter, the AI section requires more accurate positioning and reducing error. Moreover, according to flowchart 7.1, the AI function is connected to the machine learning department and needs to use a limited memory to perform calculations and store its activities. For this purpose, we solved this problem by using Redis cache.

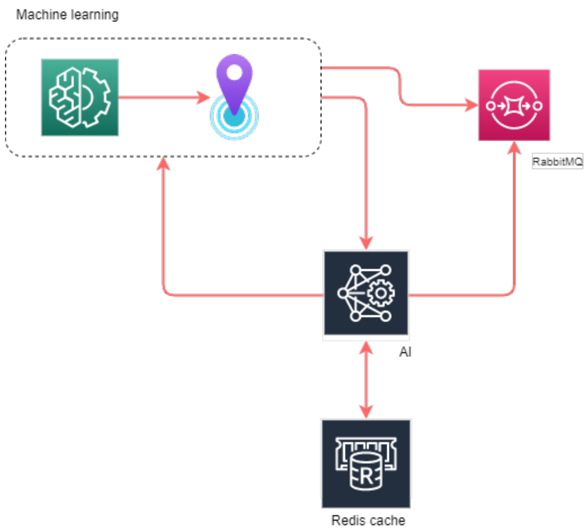


Figure 7.1: Diagram of Artificial intelligence section.

Redis Cache, which stands for Remote Dictionary Server, is a fast, open-source, in-memory key-value data store. Redis maintains data with a key-value system, and thanks to this feature, access to and retrieval of this information will be much easier since there is no complex relationship between the data.

Key-Value is a server or storage location where information is stored in pairs of keys and values. For instance:

- Name: Marco Mancini
- Tag_id: "F9-A9-F0-C5-7F-D9"
- Location: Floor 2, Zone 4 and Room 6
- Time=1623534698

Data stored in Redis Cache.

- key="F9-A9-F0-C5-7F-D9"
- Value="2-4-6/1623534698/0"

As provided in the example earlier, all information about each patient, including the patient's last position and time, is stored in Redis.

After receiving each new record from antennas in the artificial intelligence function, we can predict the patient's position. Because the data we received is a signal and the noise in the signals depends on many factors that this article does not have the opportunity to address, the machine learning section predicts the patient's position after receiving each record.

By receiving at least four records of patient data and positions from ML, we can make a more precise prediction. In addition, the probability of the patient's presence in the place, in other words, if we receive four records of data and in the machine learning section, all four records of the data predict the patient's presence in room number 8, then the probability of the patient's presence in room 8 is very high. We can also make better judgments by using the patient's previous position.

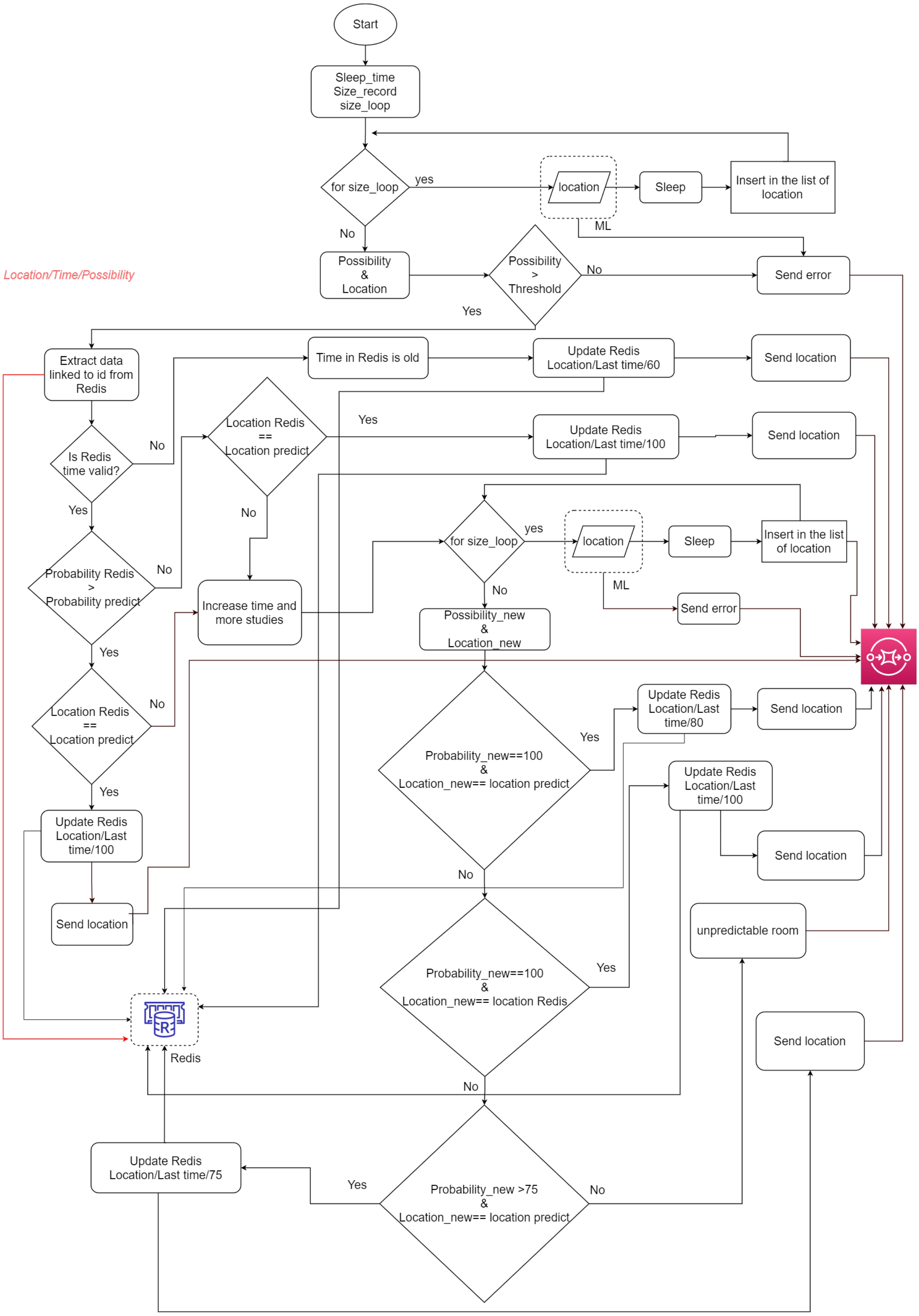


Figure 7.2: Flowchart of Artificial intelligence section.

As depicted in Complex Flow Chart 7.2, patient information is reviewed at all stages. First, by extracting the data from Elasticsearch and using the obtained model, we also predict the patient's position. Then we wait a few seconds for the new data to be stored in the database and do the positioning again. We do this process with the number "Size_loop".Then by "Size_loop", we have the predicted position of the patient. For example:

"Size_loop = 4" results in:

list_location = [2,2,2,3]

Final_location = 2

Probability = 75 %

In the next step, if the probability is less than the threshold, the queue will be notified that the positioning system will not position this patient to predict the room. However, as mentioned before, we have been able to locate the zone, and the floor in the previous steps in this level to use the calculations we have done, and this information is helpful to us.

If the patient has changed according to previous information recorded in Redis, the floor or zone will be reported to the queue that we could not locate the exact room number, but the patient has been moved and is present in this zone and floor. For this purpose, we generally do not lose the patient in the event of an accident to the patient.

As described in the flow chart, to check the patient's latest position and the use of this data in the artificial intelligence system, the time of this data must not have expired. In other words, the time of the patient's last position must be valid. Therefore, before using the data, it is checked that the time of the last data is accepted.

If the time has expired, the Redis information is not valid, and we have to do the positioning without using this information. However, after positioning, we register in Redis that the probability of positioning is low to be corrected in the next step if the position or probability is correctly predicted.

Then, if the prediction made is less probability than the probability stored in Redis and the predicted position is equal to the position stored in Redis, two important conclusions can be drawn from this information. First, the predicted position is most likely correct, and the error is low at this stage; moreover, we update in Redis that the prediction has been made with high probability.

On the other hand, if the predicted position is not equal to the position recorded in Redis. This means that we selected a higher probability of correct prediction, but now the patient's position has changed. To ensure that whether this is correct and not send the wrong alarm to the system, we reposition the system. However, this time, we have increased the number of repetition loops to do more research.

After receiving information from the new situation, if the probability was 100% and the situation we had further investigation was equal to the predicted situation, we conclude that the patient has been moved and the nurses should be informed. However, if the probability was low or the predicted position was not equal, then there is a possibility of error, and it is better to avoid sending with low probability. This is because if the system misinforms nurses about many handling errors, after some time, trust in the system decreases, and eventually, the system will become useless.

As mentioned before, if RabbitMQ is connected and communication is possible, in other words, information is exchanged between machine learning and Backend, it is reported in Redis that the connection is established. As indicated, in artificial intelligence, efforts have been made to increase the system's accuracy. However, Redis is also critical in this section, which is why in section 6.4.1 we considered that if Redis is not established, it is not possible to position correctly.

7.2 Multithreading

Thread is a lightweight process that executes code sequence and all the data supporting structures. If we want to run the code in parallel, making programming is simple. It takes benefit from the architectures of multi-CPU. It can also run multiple processes or multiple threads within one process.

A thread is a way to break itself within more than two parts concurrently while running the jobs for a program. There is a difference between the processes of each operating system with another. However, usually, a thread consists of a process, and they are distinct in the same process share similar resources; in the meantime, diverse processes in the same multitasking operating systems cannot be utilised.

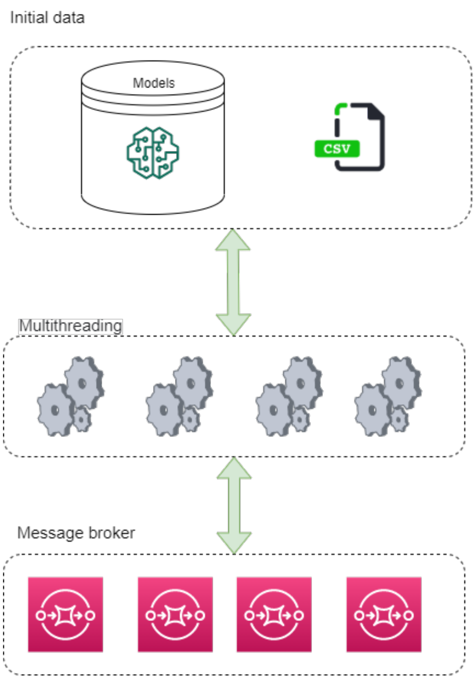


Figure 7.3: Diagram of Multithreading.

Concerning this project, we need Multi threading because, for each bracelet, we need to do the computations independently.

As illustrated in figure 7.3, The algorithm runs continuously in a circular way; the program starts reading all available mac addresses from a file and putting each of them in an internal queue. Then it generates the threading pool with a configured number of threads and listens to the same queue. For each message, it calls the tag positioning prediction and waits for the result. Independently from the success or the failure of the prediction, the mac id is reinserted to the queue to be reprocessed. The program also listens to a Rabbit queue for any changes in the mac list (macs can be dynamically added or removed) and update the corresponding file (it is necessary in case of restart). For this reason, a set is used; only elements in the set are enqueued again (eliminated tags are not added to the set).

7.3 Final Result

To enhance the quality of life for Alzheimer's patients or regular patients in the hospital, in this work, we focused on developing a hybrid technology to help these persons and doctors in the hospital.

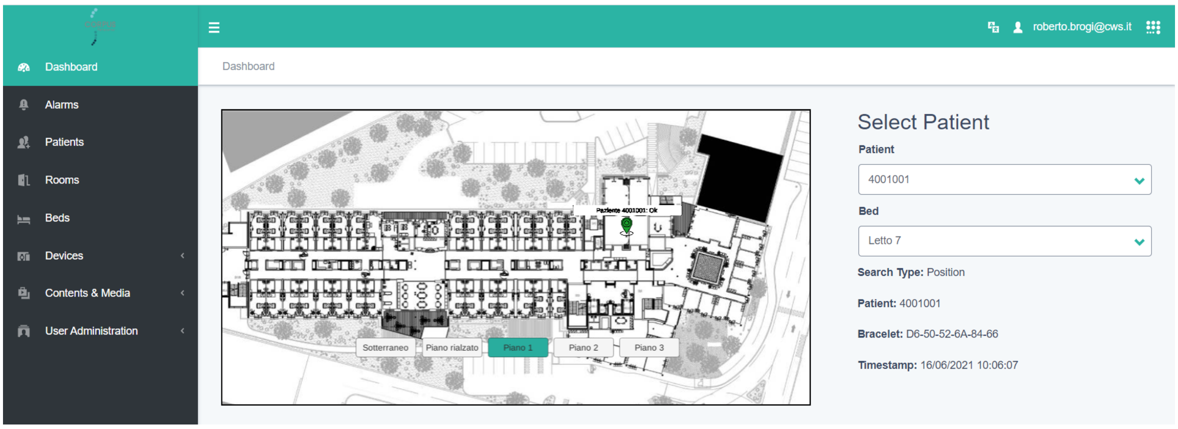


Figure 7.4: The final dashboard of the indoor positioning system.

As reported in Figure 7.4, In the dashboard, we can search the patient position. In addition, this dashboard can have separate rules for each patient to notify nurses in contravention of the rules. For instance, if the patient moves from Zone one to Zone two between 12 am and 5 am, the nurses should be notified of this movement. It also reduces the risk to patients because they are well controlled in all the conditions and rules that nurses have registered for patients.

Chapter 8

Conclusions

Nowadays, many companies take advantage of using Indoor positioning systems to improve their progress. Still, above this, hospitals are one of the main buildings in which implementing indoor positioning with the help of Artificial Intelligence (AI) are vitally and essentially recommended; therefore, in this thesis, I want to discuss and answer how indoor positioning systems along with artificial intelligence can assist the doctors and reduce the probable risks for the patients.

By utilising a bracelet containing a Bluetooth sensor, antenna and Raspberry pi (s), we tend to receive the RSSI signal. After cleaning the data, transfer it to the machine learning model. In the machine learning section, the primary location will be sent to artificial intelligence to avoid errors. Last but not least, we can predict the final location of the patient.

Starting from the Proposed Indoor Positioning System, the solution implemented, and the results, the following can be concluded:

8.1 Technical conclusions

- In the proposed indoor positioning system, the desired behaviour of the system has been implemented using two intelligence algorithms (i.e., machine learning and artificial intelligence). However, the experimental results confirm that the indoor positioning system does not work alone by utilising machine learning algorithms. Therefore Artificial Intelligence with machine learning helps us to have a more accurate positioning.

- Due to the Final Result of the system and the positioning of patients in the hospital, and the movement of patients, the accuracy of the positioning system in each patient room is acceptable (4 meters per room)

- on the examinations performed in Chapter Experimental Results (4.2) and the results we achieved, we were able to avoid some of the problems that may transpire in the future for the system. Unfortunately, though some system events are unpredictable, it is impossible to solve them or mitigate the error in the system.

One of these problems is when the bracelet breaks. If the bracelet breaks and misses, positioning is impossible, or positioning is done with a high error. As mentioned in the testing chapter, we prevented this from happening due to the systematic review and the last time we received the data.

- According to the experiments performed in Section Experiments on Raspberries (4.3), Raspberries work well due to their economical cost compared to antennas, plus in places where the positioning system can not predict or predict with high error (black areas), they can be installed in that area. Therefore, install to have a better positioning.

- To use the RSSI values as data for machine learning algorithms, the RSSI values cannot be used without intermediate processing steps to mitigate the non-linearity of the measured values and cleaning data. The outcomes of the analyses confirmed that the RSSI level changes with distance, the geometrical orientation of the sensors, and environment characteristics. As mentioned in the Experimental chapter, it has a significant effect on the RSSI signal of the environment.

- Having successfully developed the RNN model, trained and evaluated the model, based on the experiment results, we can conclude that RNN produces better accuracy compared to others.

8.2 Future works

- One of the problems in this system is that if one of the antennas breaks down and a new antenna needs to be replaced, the model of machine learning needs to be rebuilt with a different ID and the train data label changed. In the future, we can design a system that will not have this problem if we replace it, or we will face fewer problems. It is crystal clear that in the system we designed since the divisions are insignificant in case of failure in any part, and there is no need to change in other parts.

- According to the experimental results, we will receive different RSSI signals if we rotate the Raspberry pi (s) or Antenna (s). This difference increases the error rate in the positioning section.

- Future research directions include upgrading artificial intelligence systems to decrease errors and increase system accuracy.

- Reduce time between two antenna records to reduce computation time and increase system real-time.

- The information of all patients stored in the database with their movement time and the places where they moved. Therefore, it is clear to invent a system that provides doctors with data on patient movement during the day or month in data analysis. However, the doctor can better diagnose whether the patient moves during the day or whether the patient sleeps well during the night.

Appendix A

Code & Pseudocode

As reflected in the code (A) is a sample format of the training part that is stored in Elasticsearch based on each room number and then based on time and beacon id. With this process, all the information about each room can be easily obtained.


```
[
  {
    "Location": "1",
    "Data":
        [
      {
        "timestamp": 1502984373838719200,
         "bt_mac":"User1",
        "Data_antenna":
                [
          {
            "ap_mac": "antenna1",
            "bt_rssi": -48
          },
          {
            "ap_mac": "antenna2",
            "bt_rssi": -94
          },
          {
            "ap_mac": "antenna3",
            "bt_rssi": -90
          },
          {
            "ap_mac": "antenna4",
            "bt_rssi": -86
          }
        ]
      },
          {
        "timestamp": 150298437383876360,
```

```
    "bt_mac":"User1",
   "Data_antenna":
          [
     {
       "ap_mac": "antenna1",
       "bt_rssi": -40
     },
     {
       "ap_mac": "antenna2",
       "bt_rssi": -93
     },
     {
       "ap_mac": "antenna3",
       "bt_rssi": -56
     },
     {
       "ap_mac": "antenna4",
       "bt_rssi": -36
     }
   ]
  }
 ]
},
{
 "Location": "2",
 "Data":
   [
  {
   "timestamp": 1502984373836719220,
    "bt_mac":"User1",
   "Data_antenna":
          [
     {
       "ap_mac": "antenna1",
       "bt_rssi": -40
     },
     {
       "ap_mac": "antenna2",
       "bt_rssi": -64
     },
     {
       "ap_mac": "antenna3",
       "bt_rssi": -36
     },
```

```
          {
            "ap_mac": "antenna4",
            "bt_rssi": -26
          }
        ]
      }
    ]
  }
]
```


Listing A.1: sample format of the training section

Data extraction function from Elasticsearch for Training section.

```
% location: Room number required to extract data

from elasticsearch import Elasticsearch

def get_data(config,location):


    try:
        es = Elasticsearch([{'host': config['elasticsearch']['host'],

        'port': config['elasticsearch']['port'],

         'http_auth': (config['elasticsearch']['user'],
         config['elasticsearch']['passwd'])}])

        All=es.search(index=index_elastic, body={   "query" : {
               "match_all" : {}
           },
          })

        data = es.search(index=config['elasticsearch']['index'], body={"query": {
            "query_string": {
             "query": "\"1-1-2\"",
              "fields": [ "location" ]
           }
             },"size": 30000})
```

```
        data = es.search(index=config['elasticsearch']['index'], body={"query": {
            "match": {
                "location": location
            }
        },"size": 30000}
                        )

        return data,400

    except Exception as e:
        logger.info('We have a problem to connect to the ElasticSearch {}'.format(e))
        data = "ElasticSearch_error"
        status=404
        return data,status
```

Listing A.2: Extraction function from Elasticsearch in the Training section.

```
Data cleaning function

import pandas as pd

% location: Room number required to extract data

def clean(config,location):

    list_antenna=[]
    list_bt_rssi=[]
    list_time=[]
    data=get_data(config,location)

    for l in range(len(data[0]['hits']['hits'])):
        if data[0]['hits']['hits'][l]['_source']['location']==location:
#             print(data[0]['hits']['hits'][l]['_source']['location'])
            f_data=data[0]['hits']['hits'][l]['_source']['Data']
    #    print('The number of data that we have in Elastic search: ',len(f_data))


            for i in range(len(f_data)):
                for j in range(len(f_data[i]['antennas'])):
                    list_antenna.append(f_data[i]['antennas'][j]['ap_mac'])
                    list_bt_rssi.append(f_data[i]['antennas'][j]['bt_rssi'])
                    list_time.append(f_data[i]['timestamp'])
```

```
#     return list_antenna, list_bt_rssi, list_time, data

    df = pd.DataFrame(list(zip(list_antenna, list_bt_rssi)),
                      columns =['antennas', 'bt_rssi'])

    # unique
    lable=list(df.antennas.unique())
    lable.append('timestamp')


    df_new = pd.DataFrame(columns = lable)

    for i in range(len(list_antenna)):
        dic=dict.fromkeys(lable)
        try:
            dic[list_antenna[i]]=int(list_bt_rssi[i].split('-')[1])
            dic['timestamp']=int(list_time[i])
            df_new = df_new.append(dic, ignore_index=True)
        except:
            pass

    df_new=df_new.fillna(0)
    df_finaly = pd.DataFrame(columns = lable)

    #df_new.shape[0]
    def avrage_no_zero(input_list):
        sum_list=sum(input_list)

        #if the list is 0 dont calculate (Because there is an error divided by zero)
        if sum_list==0:
            return 0
        else:
            zero=input_list.count(0)
            return sum_list/(len(input_list)-zero)


    def avrage(frame):
        lable_frame=list(frame.columns)
        list_finaly=[]
        for i in range(len(lable_frame)):
#           print (i)
            ar=avrage_no_zero(list(frame[lable_frame[i]]))
            list_finaly.append(ar)
```

```
#    print (list_finaly)
     a_series = pd.Series(list_finaly, index = frame.columns)

     return a_series

for i in range(0,df_new.shape[0],3):
#    print (i)
     df_help = pd.DataFrame(columns = lable)
     if i ==0:
#         print ('first',i)
#         print ('last',i+3)
          df_help=df_new.loc[i:i+3]
          df_finaly = df_finaly.append(avrage(df_help), ignore_index=True)
     else:
#         print ('first',i+1)
#         print ('last',i+3)
          df_help=df_new.loc[i+1:i+3]
          df_finaly = df_finaly.append(avrage(df_help), ignore_index=True)

return df_finaly,data
```

Listing A.3: Data cleaning function in Training section.

```
import pika
import logging.config
class RabbitQueue:

    def __init__(self, config):

        try:
            logging.config.fileConfig('./logging_positioning.conf')
            logger = logging.getLogger('positioningLogger')

            credentials = pika.PlainCredentials(config['RabbitMQ']['user'],
            config['RabbitMQ']['pwd'])
            parameters=pika.ConnectionParameters(config['RabbitMQ']['host']
            ,int(config['RabbitMQ']['port']), '/',credentials)
            connection = pika.BlockingConnection(parameters)

            self.channel = connection.channel()
            self.channel.exchange_declare(exchange=
            config['RabbitMQ']['excange_name'], exchange_type='topic',
            durable='true')
            result = self.channel.queue_declare(queue='',
```

```
                exclusive=True)

            self.queue_name = result.method.queue

            self.flag=True
            # print(self.flag)

        except Exception as e:
            # print('We have a problem to connect the Rabbit %s' % e)
            logger.info('We␣have␣a␣problem␣to␣connect␣the␣Rabbit{}'.format(e))
            self.flag = False
            self.channel=1

    def get_queue(self):
        return self.queue_name

    def get_channel(self):
        return self.channel,self.flag
```

Listing A.4: RabbitQueue.

```
import redis
import logging.config
class RedisCache:

    def __init__(self, config):
        try:
            logging.config.fileConfig('./logging_positioning.conf')
            logger = logging.getLogger('positioningLogger')

            host = config['Redis']['host']
            # print(host)
            pwd = config['Redis']['pwd']
            port = int (config['Redis']['port'])
            db = int (config['Redis']['db'])
            ssl = bool(int(config['Redis']['ssl']))

            self.redis_instance = redis.StrictRedis(host=host, port=port,
            db=db, password=pwd, ssl=ssl)

            self.flag=True
            self.redis_instance.ping()

        except Exception as e:
```

```
            logger.info('We␣have␣a␣problem␣to␣connect␣the␣Redis:{}'.format(e))
            # print('We have a problem to connect the Redis: %s' % e)
            self.flag=False

    def get_redis_instance(self):
        return self.redis_instance,self.flag
```

Listing A.5: Redis cache.

```
from .conection.Flag_redis import set_flag_communication
import json
import time
import pika
################################################################################
##                                  send_RabbitMQ
##
##                  function for write the message on the RabbitMQ
##
##                  bt_mac=tag id for send the data to the RabbitMq
##
##      timestamp=the time that we have(it is for Elasticsearch or the execution time)
##
##      status=the status of the message that we have to publish it
##
##      location=the location that predict or the other comment
##
##      config=the configuration information
##
##      r=the r connection of the redis
##
##      rq=RabbitQueue(config)
##
##      routing_key_send=routing_key_send_pos or routing_key_send_inf
##
################################################################################

def send_RabbitMQ(bt_mac,timestamp,status,location,config,r,rq,routing_key_send,logger):

    #the format of the writing on the message Broker
    logger.info('The␣send_RabbitMQ␣is␣active␣for␣this␣user:{}'.format(bt_mac))
    result_data={
            "bt_mac":bt_mac  ,
            "timestamp": timestamp,
            "status":status,
```

```
            "location":location
            }


    try:
        channel, flag_rabbit = rq.get_channel()
        channel.basic_publish(exchange=str(config['RabbitMQ']['exchange_name']),
        routing_key=str(routing_key_send), body=json.dumps(result_data))


        logger.info('The␣routing␣key␣for␣this␣mac␣id␣:␣{}'.format(bt_mac)+'␣is
␣␣␣␣␣␣␣␣:{}'.format(str(routing_key_send)))

        logger.info('The␣data␣that␣send␣to␣the␣RabbitMQ␣is:{}'.format(result_data))

        # set_flag_communication(user_bt, flag)
        # flag==1 it means we are communicate to the Rabbit
        # set_flag_communication(user_bt, flag,r):
        set_flag_communication(bt_mac, 1, r,logger)
        logger.info('We␣set␣the␣flag␣on␣the␣RedisCache␣that␣we␣communicate␣to
␣␣␣␣␣␣␣␣the␣RabbitMQ␣for␣this␣mac␣id␣:␣{}'.format(bt_mac))

    except Exception as e:

        logger.info('We␣have␣problem␣on␣RabbitMQ␣for␣this␣mac␣id:
␣␣␣␣␣␣␣␣{}'.format(bt_mac))

        # def write_log_file_room(bt_mac,error_type,status,location):
        # set_flag_communication(user_bt, flag)
        # flag==0 it means we are not communicate to the RabbitMQ
        # set_flag_communication(user_bt, flag,r):
        set_flag_communication(bt_mac, 0, r,logger)
        logger.info('We␣set␣the␣flag␣on␣the␣RedisCache␣that␣we␣can␣not
␣␣␣␣␣␣␣␣communicate␣to␣the␣RabbitMQ␣for␣this␣mac␣id␣:␣{}'.format(bt_mac))

        # print("We set the flag on the RedisCache that we can not communicate
        to the RabbitMQ")
␣␣␣␣␣␣␣␣#␣write_log_file_room(bt_mac,status,location,config):
␣␣␣␣␣␣␣␣just_write_log␣(bt_mac,int(time.time()),␣1000,␣'RabbitMq_not_connect',␣config,l
␣␣␣␣␣␣␣␣just_write_log(bt_mac,timestamp,status,␣location,␣config,␣logger)


def␣just_write_log(bt_mac,␣timestamp,␣status,␣location,␣config,␣logger):
```

```
    logger.info('The just_write_log is active for this user:{}'.format(bt_mac))

    result_data = {
        "bt_mac": bt_mac,
        "timestamp": timestamp,
        "status": status,
        "location": location
    }

    path_log = config['Save']['log_path']
    try:

        with open(path_log, 'a') as outfile:
            json.dump(result_data, outfile)
            outfile.write("\n")

        logger.info('We write the message on the log file for this mac id :
        {}'.format(bt_mac))
        logger.info('The data that write on the log file for this mac id :
        {}'.format(bt_mac)+' is :{}'.format(result_data))

        return True

    except IOError:
        logger.info('We can not open the log file for this mac id : {}'.format(bt_mac))

        return False
```

Listing A.6: Function for writing the message on the RabbitMQ.

Appendix B

Experimental Data

```
% A piece of the Training data record for room number "4-1-7"

{
  "_index": "cambium_training",
  "_type": "_doc",
  "_id": "4-1-7",
  "_version": 8526,
  "_score": 0,
  "_source": {
    "temp_bt_mac": "CD-9F-CA-F3-5A-EC",
    "Data": [
      {
        "antennas": [
          {
            "bt_rssi": "-85",
            "ap_mac": "00-00-00-04-01-06"
          }
        ],
        "bt_mac": "CD-9F-CA-F3-5A-EC",
        "key": "1613741702CD9FCAF35AEC",
        "timestamp": 1613741700
      },
      {
        "bt_mac": "C9-A8-71-14-6B-A3",
        "antennas": [
          {
            "bt_rssi": "-78",
            "ap_mac": "00-00-00-04-03-01"
          },
          {
            "bt_rssi": "-85",
            "ap_mac": "BC-E6-7C-3B-1D-22"
          }
```

```
        ],
        "key": "1613741703C9A871146BA3",
        "timestamp": 1613741703
      },
      {
        "bt_mac": "DC-60-CB-91-8C-25",
        "antennas": [
          {
            "bt_rssi": "-81",
            "ap_mac": "00-00-00-04-03-01"
          },
          {
            "bt_rssi": "-80",
            "ap_mac": "00-00-00-04-01-06"
          },
          {
            "bt_rssi": "-85",
            "ap_mac": "58-C1-7A-87-61-CC"
          }
        ],
        "key": "1613741710DC60CB918C25",
        "timestamp": 1613741709
      },
      {
        "bt_mac": "D1-79-85-79-A6-38",
        "antennas": [
          {
            "bt_rssi": "-86",
            "ap_mac": "00-00-00-04-01-06"
          },
          {
            "bt_rssi": "-86",
            "ap_mac": "BC-E6-7C-3B-1D-36"
          }
        ],
        "key": "1613741707D1798579A638",
        "timestamp": 1613741706
      },
      {
        "bt_mac": "FD-B2-9C-DF-3B-94",
        "antennas": [
          {
            "bt_rssi": "-75",
            "ap_mac": "00-00-00-04-01-06"
```

```
            },
            {
              "bt_rssi": "-82",
              "ap_mac": "00-00-00-04-03-01"
            }
          ],
          "key": "1613741692FDB29CDF3B94",
          "timestamp": 1613741691
        },
        {
          "bt_mac": "F9-A9-F0-C5-7F-D9",
          "antennas": [
            {
              "bt_rssi": "-85",
              "ap_mac": "58-C1-7A-87-61-CC"
            }
          ],
          "key": "1613741700F9A9F0C57FD9",
          "timestamp": 1613741700
        },
        {
          "bt_mac": "D7-69-4D-87-15-86",
          "antennas": [
            {
              "bt_rssi": "-79",
              "ap_mac": "BC-E6-7C-3B-1D-22"
            }
          ],
          "key": "1613741693D7694D871586",
          "timestamp": 1613741691
        },
            {
              "bt_rssi": "-89",
              "ap_mac": "00-00-00-04-01-04"
            },
            {
              "bt_rssi": "-85",
              "ap_mac": "BC-E6-7C-3B-1D-22"
            }
          ],
          "key": "1613741713CF16325F1D85",
          "timestamp": 1613741712
        },
        {
```

```
          "bt_mac": "FE-DF-30-B6-3A-E6",
          "antennas": [
            {
              "bt_rssi": "-80",
              "ap_mac": "00-00-00-04-03-01"
            }
          ],
          "key": "1613741713FEDF30B63AE6",
          "timestamp": 1613741712
        },
        {
          "bt_mac": "CD-B9-9D-C3-B5-2C",
          "antennas": [
            {
              "bt_rssi": "-75",
              "ap_mac": "00-00-00-04-01-06"
            },
            {
              "bt_rssi": "-92",
              "ap_mac": "00-00-00-04-03-01"
            },
            {
              "bt_rssi": "-88",
              "ap_mac": "BC-E6-7C-3B-1D-22"
            }
          ],
          "key": "1613741713CDB99DC3B52C",
          "timestamp": 1613741712
        },
      "temp_key": "1613741702CD9FCAF35AEC",
      "bt_uuids": "476C6F62-616C-2D54-6167-000000000000",
      "location": "4-1-7"
    }
}
```



```
% one record of data for cambium_realtime_testing
{
  "_index": "cambium_realtime_testing",
  "_type": "_doc",
  "_id": "1624721579%{bt_mac}",
  "_version": 20,
  "_score": 0,
  "_source": {
```

```
      "timestamp": 1624721579,
      "antenna_mac": "58-C1-7A-87-62-38",
      "tags": [
        "_split_type_failure"
      ],
      "temp_bt_mac": "%{bt_mac}",
      "antennas": [
        [
          {
            "ap_mac": "58-C1-7A-87-62-38"
          }
        ],
        [
          {
            "ap_mac": "58-C1-7A-C5-B1-66"
          }
        ],
        [
          {
            "ap_mac": "BC-E6-7C-40-DE-04"
          }
        ],
        [
          {
            "ap_mac": "BC-E6-7C-40-BB-B2"
          }
        ],
        [
          {
            "ap_mac": "BC-E6-7C-3B-1C-5A"
          }
        ],
        [
          {
            "ap_mac": "BC-E6-7C-3B-3E-88"
          }
        ],
        [
          {
            "ap_mac": "58-C1-7A-C5-6C-F4"
          }
        ],
        [
          {
```

```
                "ap_mac": "BC-E6-7C-3B-1D-52"
            }
        ],
        [
            {
                "ap_mac": "BC-E6-7C-3B-3F-56"
            }
        ],
        [
            {
                "ap_mac": "BC-E6-7C-3B-1D-34"
            }
        ],
        [
            {
                "ap_mac": "58-C1-7A-C5-B1-F6"
            }
        ],
        [
            {
                "ap_mac": "BC-E6-7C-3B-40-34"
            }
        ],
        [
            {
                "ap_mac": "58-C1-7A-87-6F-42"
            }
        ],
        [
            {
                "ap_mac": "BC-E6-7C-3B-1D-F0"
            }
        ],
        [
            {
                "ap_mac": "BC-E6-7C-3B-1D-EA"
            }
        ],
        [
            {
                "ap_mac": "58-C1-7A-87-5F-28"
            }
        ],
        [
```

```
        {
          "ap_mac": "BC-E6-7C-32-97-80"
        }
      ],
      [
        {
          "ap_mac": "58-C1-7A-87-68-6A"
        }
      ],
      [
        {
          "ap_mac": "BC-E6-7C-3B-1D-4E"
        }
      ],
      [
        {
          "ap_mac": "BC-E6-7C-3B-1C-72"
        }
      ]
    ],
    "temp_key": "1624721579%{bt_mac}",
    "ble_discovered_clients": [
      {
        "bt_uuids": "",
        "bt_rssi": -76,
        "bt_mac": "3A-D2-28-EB-ED-8F",
        "bt_timestamp": 1624721573
      },
      {
        "bt_uuids": "",
        "bt_rssi": -76,
        "bt_mac": "00-7C-2D-EF-5B-81",
        "bt_timestamp": 1624721573
      }
    ]
  },
  "sort": [
    0,
    1624721579
  ]
}
```

Bibliography

Aguilera, Dr.Pablo (**july** 2017). "Modern Wireless Indoor Positioning 2/3". **in**: url: https://www.linkedin.com/pulse/modern-wireless-indoor-positioning-23-state-art-pablo-aguilera-bonet/.

Al-Ammar (2014). "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms". **in**: url: https://www.semanticscholar.org/paper/Comparative-Survey-of-Indoor-Positioning-and-Al-Ammar-Alhadhrami/35c7389554c815372bfd227f3d76867efa72c09f.

Alhadhrami, Suheer (2014). "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms". **in**: url: https://www.semanticscholar.org/paper/Comparative-Survey-of-Indoor-Positioning-and-Al-Ammar-Alhadhrami/35c7389554c815372bfd227f3d76867efa72c09f.

aws.amazon (2019). "Logstash". **in**: url: https://aws.amazon.com/elasticsearch-service/the-elk-stack/logstash/.

Bitbrain (**may** 2017). "All about Indoor Positioning Systems (IPS)". **in**: url: https://www.bitbrain.com/blog/indoor-positioning-system.

cambiumnetworks (2021). "cnPilot e600 Wi-Fi Access Point". **in**: url: https://www.cambiumnetworks.com/products/wifi/cnpilot-e600-wi-fi-access-point/.

Cumino, Chiara (**march** 2021). "Un sistema di posizionamento indoor per la clinica MELITTAKLINIK". **in**: url: https://www.cws.it/case-study/un-sistema-di-posizionamento-indoor-per-la-clinica-melittaklinik/.

elastic (2019). "What is an Elasticsearch index?" **in**: url: https://www.elastic.co/what-is/elasticsearch.

Estimote (2019). "What are Broadcasting Power, RSSI and other characteristics of a beacon's signal?" **in**: url: https://community.estimote.com/hc/en-us/articles/201636913-What-are-Broadcasting-Power-RSSI-and-other-characteristics-of-a-beacon-s-signal-#:~:text=The%20beacon's%20range%20is%20technically,of%20up%20to%2015%20m.&text=Beacons%20do%20not%20broadcast%20constantly..

Farid, Zahid (**may** 2013). "Recent Advances in Wireless Indoor Localization Techniques and System". **in**: url: https://www.hindawi.com/journals/jcnc/2013/185138/.

Global-Tag (2021). "PROTECTION CASE BEACONY – BLE Smart Beacon with silicone case". **in**: url: https://www.global-tag.com/portfolio/ble-watchcase-beacony/.

Guo, Hang (2016). "A New Indoor Positioning Method Based on Zigbee System". **in**: url: https://www.ion.org/publications/abstract.cfm?articleID=14570.

He, Suining (2016). "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons". **in**: url: http://www.cs.ust.hk/~gchan/papers/COMST16_IP.pdf.

Lanjudkar, Pankaj (**december** 2018). “Indoor Positioning and Indoor Navigation (IPIN) Market Expected to Reach $ 43,511 Million by 2025”. **in**: url: https://www.alliedmarketresearch.com/press-release/indoor-positioning-and-indoor-navigation-ipin-market.html#:~:text=According%20to%20a%20recent%20report,positioning%20and%20indoor%20navigation%20(IPIN).

— (**march** 2020). “Indoor Positioning and Navigation Market - Forecast (2020 - 2025”. **in**: url: https://www.researchandmarkets.com/reports/4531980/indoor-positioning-and-navigation-market.

Saleem, Aqsa (**june** 2019). “INDOOR PROFESSOR ASSTANT PROFESSOR COMPUTER SYSTEM ENGINEERING”. **in**: url: https://michigansciencecenter.net/indoor-positioning-system-final-year-project-thesis-supervised-by/.

unixtimestamp (2015). “Epoch Unix Timestamp Conversion Tools”. **in**: url: https://www.unixtimestamp.com/.

wikipedia (2020a). “Message broker”. **in**: url: https://en.wikipedia.org/wiki/Message_broker.

— (2020b). “RabbitMQ”. **in**: url: https://en.wikipedia.org/wiki/RabbitMQ.

Yogyakarta (2017). “Recurrent Neural Networks Model for WiFi-based Indoor Positioning System”. **in**: url: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8267833.