University of Zurich[UZH]

MASTER THESIS – Communication Systems Group, Prof. Dr. Burkhard Stiller

# PMD-Track: Portable Medical Devices Tracking

*Julius Willems*
*Zurich, Switzerland*
*Student ID: 16-920-217*

Supervisor: Dr. Bruno Rodrigues, Eder Scheid, Prof. Dr. Burkhard Stiller
Date of Submission: September 07, 2022

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

Master Thesis
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmuehlestrasse 14, CH-8050 Zurich, Switzerland
URL: http://www.csg.uzh.ch/

# Abstract

The recent COVID pandemic posed a challenge to healthcare systems worldwide, exposing limited resources and inefficiency in managing PMDs (Portable Medical Devices). This thesis reports the prototypical implementation of an new and cost-efficient type of asset tracking architecture, enabling hospitals to efficiently locate their resources and maximize the usage of PMDs in emergency situations, such as pandemics or natural disasters, when a sudden increase in demand may be anticipated. PMD-Track's key novelty lies in the use of staff's smartphones to replace expensive stationary gateways, typically required in traditional inventory tracking systems. As employees approach tagged PMDs in their day-to-day, their smartphone updates the location of spotted PMDs in a real-time inventory system, providing room-level localization data with up to 95% accuracy. Experiments were evaluated by contrasting two techniques (fingerprinting and multilateration) based on two different test locations in Zurich.

# Acknowledgments

First of all, I would like to thank *Prof. Dr. Burkhard Stiller* for the opportunity to perform research in this thesis.

Furthermore, I would like to thank my supervisors *Dr. Bruno Rodrigues* and *Dr. Eder Scheid* for their continuous assistance. During the thesis they helped me find solutions to my problems and improved the final written work.

# Contents

# Chapter 1

# Introduction

This chapter provides an introduction to the reasons for the choice of this topic as a Master Thesis. It goes over the motivation of the Master Thesis (1.1), specifies its goals (1.2), and covers a brief outline what will be covered (1.3).

Critical care unit facilities were severely strained in the event of a pandemic like the COVID-19, which hindered the effectiveness of healthcare systems. It revealed various flaws in healthcare systems throughout the world as a result of the unexpected rise in patient load, with authors assessing inefficiencies to respond effectively during times of crisis [3, 38]. Among these flaws, it was found that mechanical ventilators and supervision equipment (such as EKG, capnography monitors, infusion pumps, or pulse oximeters) were not adequately managed on hospital property and that, in the majority of cases, hospitals were understaffed.

Under increasing demands, traditional inventory management systems do not offer a workable solution to control the location and usage of these portable resources, known as Portable Medical Devices (PMD), in real-time [12, 46]. This includes real-time tracking systems, which are oftentimes too expensive for monitoring PMDs, once the installation of gateways is required throughout the hospital's infrastructure to receive localization data [1, 9, 28, 33]

To simplify communication and to access internal hospital services, healthcare institutions, most notably hospitals, typically provide their personnel with cellphones. PMD-Track's primary innovation is to leverage the staff's cellphones to replace costly, fixed gateways or base stations. In comparison to conventional systems, a gateway-less tracking solution offers significant cost benefits in terms of hardware needs. As staff typically move throughout the hospital and cover large parts of its premises in their day to day, once they pass by or come in range, the location of tagged PMDs can be automatically updated in the inventory.

PMD-Track's solution entails a mobile application that acts as a mobile gateway and updates a backend service with the location of recently spotted tags (*i.e.* PMD tracked devices) and displays information (quick visualization in the smartphone's front-end) concerning their position in real-time. From a research point of view, this work designs, implements, and compares two competing localization algorithms commonly used in indoor localization in two different test locations with the goal of finding an optimal trade-off between localization accuracy and associated deployment costs.

In this regard, PMD-Track enables hospitals to make efficient use of their PMDs in emergencies, such as a pandemic or possible natural disasters, where a sudden increase in demand can be experienced. Thus, the PMD-Track design will result in a higher chance of saving lives in such situations by allowing the efficient usage of critical PMDs. Henceforth, a significant opportunity for a cost-effective operation of the proposed solution is that hospitals' smartphones in use by their employees will function as mobile gateways.

The remainder of this thesis is organized as follows. Chapter 2 overviews fundamentals. While Chapter 3 describes the design of the solution, Chapter 4 touches implementation details. Chapter 5 covers the evaluation and Chapter 6 closes with considerations.

## 1.1 Motivation

The pandemic exacerbated the underlying and long existing challenge of equipment management in hospitals. To cope with inefficiencies and to ensure sufficient availability of medical equipment, hospitals acquire excess capacities resulting in cost-overheads and asset utilization rates below 50% - while experts consider a utilization rate of 80% feasible [20]. The goal of this thesis is to provide a prototypical implementation of a lightweight and cost-efficient real-time locating system to manage and increase the usage of critical and expensive hospital inventory.

## 1.2 Goals

PMD-Track's solution foresees the following goals to be met:

- **Research**: Explore the scope of indoor tracking and positioning technologies to provide a low-cost, room-level asset tracking system based on BLE that allows the near real-time tracking of PMDs.

- **Engineering**: Develop an asset tracking approach based on BLE that requires minimal infrastructure and can be used in a large-scale deployment. The thesis should produce a working prototype, which can be contrasted with the thesis' goals.

In this regard, PMD-Track enables hospitals to make efficient use of their PMDs in emergencies, such as a pandemic or eventual natural disasters, where a sudden increase in demand can be experienced. Thus, the PMD-Track design will result in a higher chance of saving lives in such situations by allowing the efficient usage of critical PMDs.

## 1.3 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 overviews fundamentals. While chapter 3 describes the design and chapter 4 covers the implementation. Chapter 5 contains the evaluation and chapter 6 finishes withe considerations.

# Chapter 2

# Fundamentals

Positioning and tracking of assets or people in indoor environments have great potential in various settings such as healthcare, retail, logistic, etc. and have been a popular research area for nearly a decade. Due to the absence of a Global Positioning System (GPS) signal in indoor environments, an array of alternative technologies emerged, such as BLE (Bluetooth Low Energy), WiFi, RFID (Radio Frequency Identification), UWB (Ultra-WideBand), VLS (Visible Light Communication), to name a few.

PMD-Track combines *(i)* indoor positioning with *(ii)* real-time asset tracking based on BLE. Thus, the differential of the approach lies in the indoor use of smartphones as mobile data sinks (*i.e.* replacing gateways) to update the real-time location of tagged assets in the inventory.

## 2.1 Background

### 2.1.1 Bluetooth Low Energy

BLE is a widely adopted wireless technology for personal area networking with a range up to 100m in Line-of-Sight (LOS) situations and operates on 40 channels in the unlicensed 2.4 GHz ISM frequency band. Maintained and developed by the Bluetooth Special Interest Group (Bluetooth SIG), BLE is designed for very low power data transmission of up to 2Mbit/s, thus, making it one of the most popular technologies in Internet of Things (IoT) applications [47] [13] [40]. This trend seems to continue as recent estimations predict 7 billion BLE enabled devices to be shipped annually by 2026 [36].

BLE co-exists independently from Bluetooth Classic and supports point-to-point, broadcast, and mesh communication topologies. Of the 40 2 MHz-wide channels, 37 are used for data transmission and 3 are used for advertising and discovery of BLE enabled devices. Data transmission is based on the Generic Attribute Profile (GATT) - a specification that defines standardized profiles for sending and receiving small amounts of data. For example, a wearable device (*e.g.* Smartwatch) might transmit the measured heart rate and the device's battery level using the Heart Rate Profile and Battery Service Profile.

Apart from it is point-to-point communication capabilities, BLE is also a popular technology choice for localization and proximity scenarios - as witnessed by the emergence of contact tracing apps during the pandemic [15] [2]. At the core of such localization and proximity scenarios is the ability to measure the spatial distance between BLE devices. To this end, tiny, low-cost, and always-on devices, called 'beacons', that continuously broadcast advertising packets can be deployed. The receiver of the advertising packet is able to estimate the distance to the emitting device based on the Received Signal Strength (RSS) (*c.f.* section 2.1.2). In 2019, Bluetooth SIG published version 5.1 of the standard and introduced improved localization and proximity capabilities such as Angle of Arrival (AoA) and Angle of Departure (AoD) detection which are intended to significantly improve RSS-based localization [36].

While the 5.1 standard was released 3 years ago and most Smartphones currently on the market are equipped with a 5.1 compatible chip [19], Apple and Google do not yet support AoA / AoD detection in their BLE APIs [8] [11]. This restricts mobile applications from adopting the new technology and thus limits its expansion - restricting solution providers to traditional RSS-based approaches.

### 2.1.2 Received Signal Strength

Received Signal Strength - RSS is a power measurement of a radio signal, performed at the receiver's antenna. It is usually expressed in decibel milliwatt (dBm) unit and expresses the signal strength - the closer the sender to the receiver, the higher the RSS and vice versa. The non-linear relationship between RSS and spatial distance is exploited by the (log-distance) Path-Loss Model:

$$d = 10^{\frac{L_P - L_O - v}{10 * \alpha}} \tag{2.1}$$

With distance approximation $d$ in meters, the measured RSS $L_p$ at the antenna, the calibrated RSS $L_0$ at 1m distance, a Gaussian random variable $v$ representing signal noise, and the parameter $\alpha$ representing the distance-power gradient [25].

Figure 2.1 depicts the relationship expressed in equation 2.1 by plotting the true measurements (blue dots) and the fitted Path-Loss model (black line). From this we can observe that *(i)* the variance within the measurements is high, meaning that they don't align well with the fitted line and *(ii)* the variance seems to increase with distance, meaning the larger the distance - the larger the variance [39].

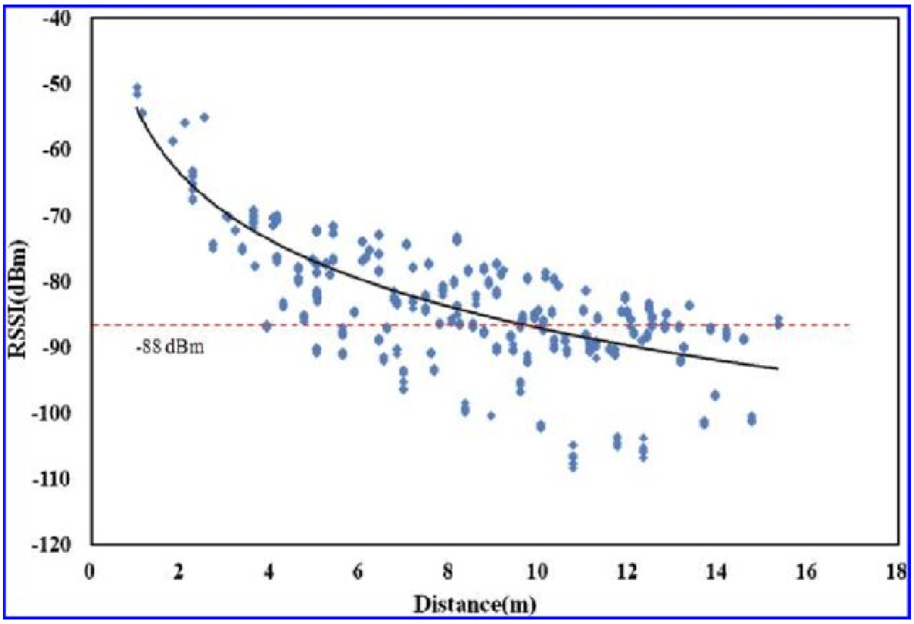


Figure 2.1: RSSI vs. Distance
[39]

Responsible for the noisy RSS signal - expressed by the high variance - is the RF signal interference present in indoor environments. Walls, people, and obstacles typically present inside buildings cause multipath and shadowing effects on the signal. This results in the RSS alone being an unprecise predictor for distance with a large margin of error - which can have a high impact in indoor settings. Thus, the signal is typically de-noised by the means of filtering (*e.g.* Kalman Filtering, Median Filtering) [28] as can be seen in figure 2.2.

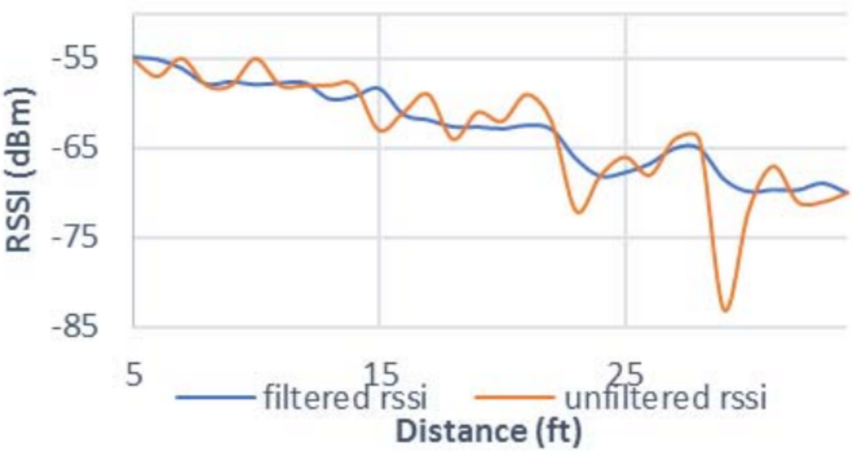


Figure 2.2: Filtered vs. Unfiltered RSSI
[31]

### 2.1.3 Fine Time Measurements

Ranging based on RSS is error prone due to signal interference (*c.f.* section 2.1.2) and Ultra-Wide Band (UWB) was long one of few technologies that achieved sub-meter accuracy in indoor environments [43].

With the IEEE 802.11-2016 specification, WiFi time of flight measurements or Fine Time Measurements (FTM) are introduced for distance calculations between an 802.11-2016 compatible AP and station [22]. The calculation is based on the Round-Trip-Time and the speed of the radio signal. The station initiates the process by submitting an FTM request to the AP. Subsequently, the AP sends a Ping to the station at $t_1$. Upon receipt, the station records the time of arrival $t_2$ and sends back a Pong at $t_3$ to the AP which receives it at $t_4$. With these four timestamps, the RTT can be calculated as can be seen in figure 2.3

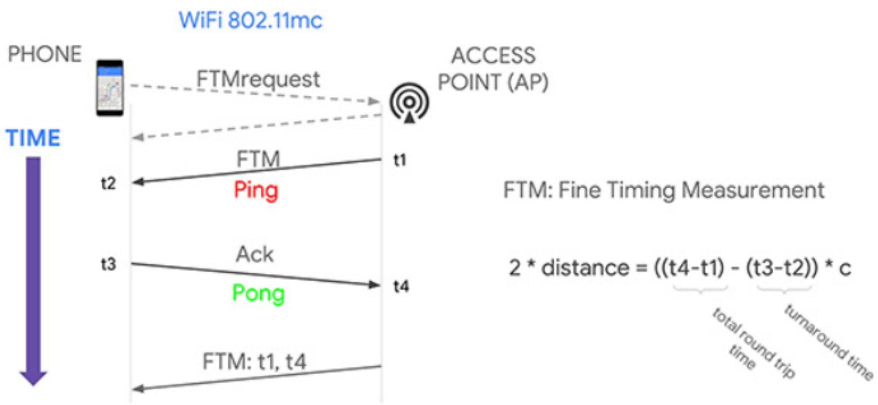


Figure 2.3: Calculation of RTT
[21]

It has been shown that FTM achieves a ranging accuracy of 1-2 meters in favorable settings rendering high potential in indoor localization scenarios [5]. At the time of writing, certain Smartphone models come with a compatible 802.11mc chip [44] and Google started to support WiFi FTM ranging as of Android 9 while Apple has not introduced support yet [21].

### 2.1.4 Inertial Measurement Units in Smartphones

Inertial Measurement Units (IMU) are small sensors commonly embedded in Smartphones. Among those sensors are typically accelerometer, gyroscope, and magnetometer used for various purposes such as step counting or finding the magnetic north in navigation Apps. The accelerometer senses motion by measuring linear acceleration in $m/s^2$ in each of the $x, y, z$ axis. Gyroscopes measure rotational orientation in $x, y, z$, and magnetometers measure the magnetic field of the earth [23].

Based on the information from the IMUs and an initial, known position, the user's trajectory and position can be calculated. However, small measurement errors or sensor biases propagate into the location prediction over time which requires the application of filtering techniques to reduce measurement noise.

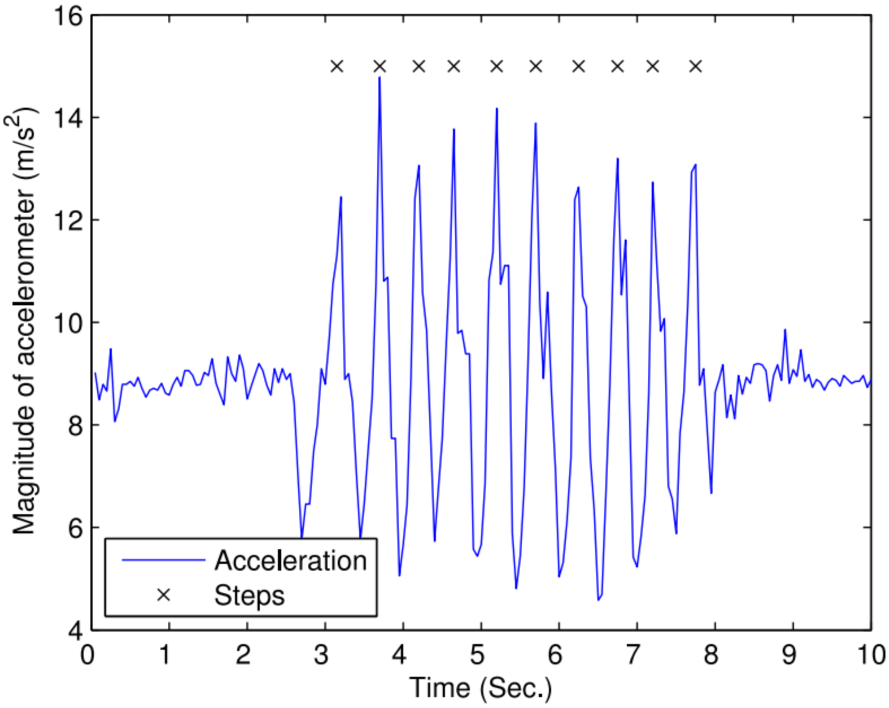


Figure 2.4: Acceleration pattern for 10 steps
[45]

Figure 2.4 illustrates the accelerometer pattern recorded while making 10 steps. Taking the person's stride length into consideration, the traveled distance can be approximated [29].

### 2.1.5 Trilateration and Mulitlateration

The position of an object can be determined by trilaterating the distance measurements of two know points. In two-dimensional space, the distance between a know point $P_i(x_i, y_i)$ and $P(x, y)$ is defined as follows:

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}, i = 1, 2 \tag{2.2}$$

Solving equation 2.2 will determine possible x and y coordinates of point P. Figure 2.5 shows an example of trilateration in twodimensional space. As shown, the solution to this trilateration problem is not deterministic for all points, which are not co-linear with P1 and P2.

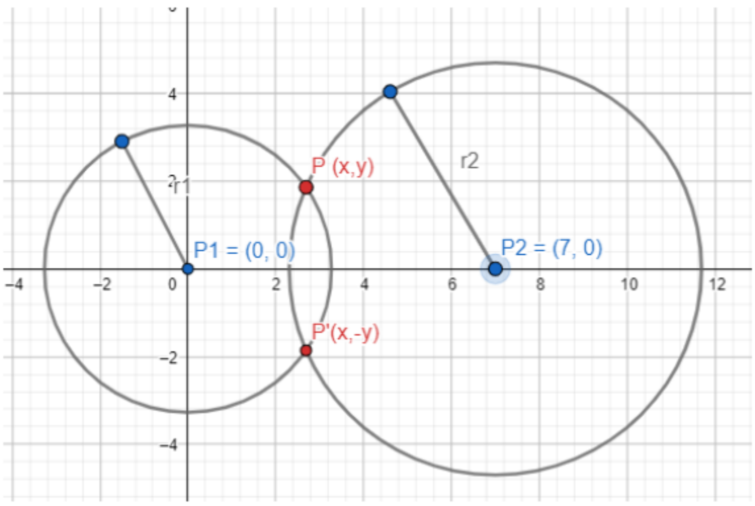


Figure 2.5: 2D Trilateratoin
[41]

To determine the deterministic position of $P(x, y)$, a third know point is required. Similarly, in cases of higher dimensions, $k + 1$ points are required [41].

### 2.1.6 Fingerprinting

Fingerprinting is based on pattern recognition and consists of an *(i)* offline or training phase and an *(ii)* online or operational phase. In *(i)*, a site survey is conducted where, in areas of interest, measurements are collected. These measurements are called fingerprints and are ideal as uniquely as possible to each area. A typical example of fingerprinting is WiFi fingerprinting where the RSS of different Access Points (AP) are measured at specific points. The collected fingerprints are labeled with the location they belong to and stored in a fingerprint database.

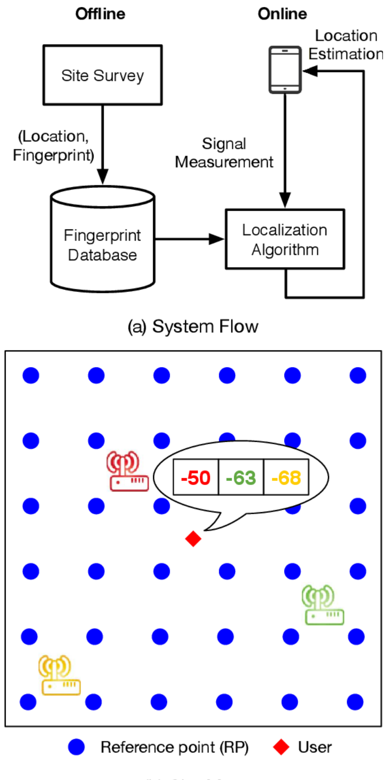


Figure 2.6: Fingerprinting
[18]

In *(ii)*, a user can record the fingerprint at its current location and query it against the fingerprint database. The database will return the user's location by pattern matching the received fingerprint with the closest matching one from the database.

While the approach is fairly popular and delivers good accuracy, it comes with several challenges. One, changes in the environment (*e.g.* AP is moved) impact the site's fingerprints with a decreasing effect on accuracy. Thus, research focused on integrating multiple data sources (*e.g.* WiFi, cellular network, magnetic field, ambient audio) to improve fingerprint stability [42] [29]. Second, to conquer the costly and labor-intensive site survey process, new approaches were developed to reduce the required effort [45] [10]. And lastly, Machine Learning (ML) and statistical methods were developed to improve the pattern matching performance [37] [35].

## 2.2 Related Work

In this section, we explore related work which is key to the development of PMD-Track and that serves as background information for the reader. We divide the related work on indoor localization based on their approach. First, we look at RSS-based ranging approaches which are fundamental for most of the related work. Second, we touch on recent Time of Flight (ToF) based ranging approaches, enabled by 802.11-2016 and UWB, and finally, we cover the more exotic acoustic-based approach for indoor localization.

### 2.2.1 Asset Tracking

Bisio, Sciarrone, and Zappatore [6] developed a crowd-sourced asset tracking solution for the construction industry based on the integration of RFID and BLE technology. Assets (*e.g.* materials, tools) are stored in warehouses and are used outdoors on construction sites. Warehouses are equipped with RFID readers, construction site workers are equipped with smartphones, and assets are tagged with both RFID and BLE tags. If an asset leaves the warehouse, its RFID tag is scanned, and if a construction site worker passes by the asset in the field at close range, the BLE tag's RSS values are scanned by a mobile application installed on the worker's smartphone. With the knowledge of the RFID scanner's location and the smartphone's GPS coordinates, the approximate location of a scanned asset can be determined.

In the context of asset tracking for industry 4.0 applications, [7] analyzed on a theoretical level the aging time and detection probability for BLE advertisement packets in dependence on different density situations and scan periods. The aging time describes the time between two consecutive detections of the same BLE tag. The detection probability captures the probability of detecting all BLE tags at least once.

Lee et al [24] developed a prototypical asset tracking system based on Raspberry Pis as receivers and BLE beacons as transmitters. The topology requires the installation of 4 Raspberry Pis in the corners of each area to act as a data sink and perform multilateration.

In [30] the authors present another BLE-based trilateration system based on stationary gateways acting as receivers. Their key contribution is comprised of using radical centers to resolve situations of non-intersecting trilateration circles.

### 2.2.2 Indoor Positioning

#### 2.2.2.1 RSSI-based approaches

PMD-Track follows a similar approach with a crowdsourced tracking network of mobile applications based on BLE where the challenge is the localization of an asset relative to a moving and dynamic reader device. However, the author's system is not designed for indoor localization and requires two tags per asset as well as the installation of RFID readers inside buildings which increases both costs of ownership and complexity for such a system. [6]

[28] employs a trilateration algorithm using Bluetooth RSS values and the smartphone's sensors data (*e.g.* accelerator and gyroscope) to improve the accuracy of the position of devices using the iBeacon protocol. [28] applies a Kalman Filter (KF) to estimate the real-time trajectory of the device. Although the accuracy of the iBeacon positioning was improved, the approach cannot be directly employed in PMD-Track as it relies on several (*i.e.* 10) iBeacons for a 44 m x 17 m area, fixed stations to sample RSS values, and smartphone sensors. In contrast, PMD-Track reduces the number of required hardware and relies solely on RSS values measured from devices' encounters.

[9] proposes an accurate Indoor Positioning System based on BLE using frequency diversity, trilateration, and KF. In the approach, the tag position is calculated based on the trilateration of 3 RSSI sniffer devices and the tag sending RSSI values. One shortcoming is that it relies on four beacons per room, and each receiver gateway costs around 120 Euros (*i.e.* 500 Euros per room).

[32] provides an algorithm that finds optimal beacon configuration for a given floorplan. The algorithm selects the configuration with the minimum number of required beacons that still ensure *unique localizability* in a predefined area. With this approach, the authors are able to localize with two instead of three or more beacons which results in 22-60% less beacons installed. This approach can be an interesting option to reduce the cost of ownership in a system such as PMD-Track.

[1] proposed a real-time tracking solution based on Arduino to track Bluetooth devices at a maximum range of 10 m. The Arduino board also contains a GSM (Global System for Mobile Communications) antenna that periodically transmits data collected to a user's smartphone. [1] raises two concerns that PMD-Track solve: the use of several static sinks to collect data from BLE tags and the limited mobility, which, combined with the short-range, makes the solution ineffective to track PMD objects.

In [42], the authors propose a low-cost, BLE based room-localization system focusing on the monitoring of patients in their home (*e.g.* apartment, house). The system is based on BLE beacons, installed in each room and a smartphone, carried by the monitored subject. RSS fingerprints were collected within the surveillance area where they collect samples for about 30 seconds in each room. Evaluated in two scenarios, the system achieves 93.75% accuracy in room estimation.

The requirement of one beacon per room and a training phase of 30 seconds per room represents a major challenge considering PMD-Track's focus on large-scale deployments in facilities with hundreds of rooms, thus, making it a bad fit for the envisioned solution.

Various works have contributed to reduce the site survey effort involved in fingerprinting. Liu, et al. [29] contributions are two-fold. *(i)* Focusing on lowering the burdensome survey effort of creating a dense fingerprint database, while maintaining a constant level of accuracy and *(ii)* keeping the fingerprint database always up to date using a crowd-sourced update mechanism. *(i)* is achieved by creating an upfront, minimal WiFi RSS-based fingerprint of each room. Subsequently, in the online phase, the user's initial location (room) is estimated by WiFi RSS ranging and a lookup in the fingerprint database. Once the initial location is determined, IMU measurements from a smartphone are fed into a particle filter together with the building's floor plan. The particle filter removes "impossible" particles (*e.g.* movement through walls) and predicts the user's trajectory. *(ii)* is achieved by re-taking WiFI RSS fingerprints and label them with the user's calculated position. Based on IMU measurements and floor plan constraints, the authors are able to perform room-level localization at the cost of WiFi fingerprinting each room of the buiding.

Carrera et al. [10] reduce the effort of the site survey process by fusing WiFi radio signals, magnetic field readings, and floor plan information into an advanced particle filter. Fingerprinting integrates WiFI RSS measurements as well as geomagnetic field readings and requires only the current room label which considerably reduces site survey effort.

A discriminative learning-based landmark detection algorithm uses room-level landmark fingerprints in combination with RSS ranging and an advanced particle filter to predict the user's location.

Following a similar line of argumentation, reducing site survey efforts are truly beneficial. However, at large scale facilities it still remains a core challenge in the context of PMD-Track.

In [45], Wu, Yang, and Liu remove the requirement for an active site survey process and propose LiFS, a novel approach, leveraging humans moving inside the building during their day-to-day activities to passively construct a WiFi fingerprint.

During the setup phase, the building floor plan is divided into a grid of areas and a distance matrix is derived, representing the walking distance (*i.e.* number of steps) between each pair of areas. Further, in a fully passive way, participants perform their day-to-day activities and move through the building while a mobile application on their smartphone collects WiFi RSS fingerprints and records the walking distance (*i.e.* number of steps) between two consecutive fingerprints. Fingerprints from all participants are merged and a distance matrix expressing their pairwise distance (*i.e.* number of steps between them) is constructed. Finally, by means of multi-dimensional scaling, the floor plan distance matrix, as well as the fingerprint distance matrix, are correlated, resulting in location-labeled fingerprints.

The authors present a novel and efficient way to conduct site surveys - potentially for large areas. Thus, representing an attractive opportunity for PMD-Track. However, the scalability aspects of this approach remain to be determined.

Subsequent works concentrate on online phase of fingerprinting and propose various ML approaches to improve fingerprint detection. Sabanci et al. [37] provide an accurate evaluation of different ML algorithms applied to a room classification task based on WiFi RSS fingerprints collected in a location with 4 rooms and 7 APs. Six different ML algorithms, such as a Neural Network, Extreme Learning Machine, K-Nearest Neighbours, Support Vector Machines, and Naive Bayes were evaluated of which K-NN achieved the highest accuracy with 97.98%.

In [35] Roy et al. address the challenges of device heterogeneity and their impact on WiFi RSS measurements, naturally occurring in crowdsourced deployments. Their contributions show that state-of-the-art ML algorithms achieve 71.78% classification accuracy on a dataset recorded by a heterogeneous set of smartphones. Further, they developed an ensemble classifier embracing the device heterogeneity which manages to improve classification accuracy up to 91.46%.

#### 2.2.2.2 Time-of-Flight based approaches

Biehl, Girgensohn, and Patel [5] investigate the potential of the recently introduced IEEE 08.11-2016 FTM on room-level localization which has been sparsely covered in the research community so far. In a series of experiments, the authors installed BLE beacons and WiFi RTT routers at the same locations in an office building and collected ranging scans for both BLE and WiFI RTT with the corresponding ground-truth coordinate locations. Contributing to several findings, the authors conclude that WiFi RTT achieves a 27.35% improvement in root mean square error in coordinate location estimation. When comparing room-level localization, RTT-location outperforms BLE by an 11.59% difference in F1 score. As their results show, accuracy goes down as the density of installed beacons (BLE, WiFi RTT) drops. While in the case of BLE, the drop is contributed by the well-studied RSS issues, in the case of WiFi RTT, the results indicate that multi-path effects remain a significant design issue for FTM-based approaches.

Their results clearly indicate the opportunities presented by WiFi RTT and with more and more devices adopting the standard, its application in real-world deployments such as PMD-Track becomes more attractive.

In the domain of high-precision, LoS tracking, UWB shows great potential thanks to its centimeter-level accuracy. Djosic et al. [14] investigate how UWB performs in complex indoor environments with partial or even non-LoS connections available. Based on ToF ranging measurements, trilateration is used to position the node while a fingerprinting-based algorithm is used to provide additional context in cases of an insufficient number of LoS measurements. The key aspect of their work is that fingerprints are not labelled with locations but rather with distances to a set of pre-defined reference points which are then used in trilateration. Their findings include that UWB represents an effective alternative in room identification with an accuracy of more than 95%. Further, the system achieved comparable accuracy in cases of 2 instead of 3 LoS connections, making UWB deployments more attractive as the number of anchor points might be reduced.

UWB is still not broadly available on mobile devices with only a few flagship phones supporting it at the time of writing [27]. Thus, its application in PMD-Track is not (yet) feasible due to the costs involved in a large-scale deployment.

#### 2.2.2.3 Acoustic based approaches

Finally, among the more exotic approaches are acoustic-based works. Guo et al. [17] and Rossi et al. [34] propose EchoLoc and RoomSense, approaches using acoustic responses to a chirp emitted by a smartphone. The system is based on the location-specific features, captured in the acoustic response or echo of a chirp and utilizes fingerprinting to learn these response patterns. While achieving high accuracy and low localization errors, these approaches are not feasible in the context of PMD-Track due to the rather small coverage area of the fingerprint (less than 1m) which would translate into a high site survey effort.

### 2.2.3 Discussion

Based on the initial goals defined in section 1.2, we discuss the related work presented in this chapter and assess their suitability for our system.

WiFi RTT and UWB-based technology are still in an early stage and are not as ubiquitous and low-cost as Bluetooth. While these technologies are designed for high-accuracy localization, our goal of room-level accuracy is much simpler and thus, high accuracy can be traded off with complexity and price. Hence, we don't consider it a fitting technology for the use case at hand.

As for the acoustic-based approaches, they have the advantage of requiring no additional infrastructure. However, they see little adoption in practice, and their relatively small coverage area makes them an impractical option for large-scale deployments. Further, it is not clear how changes and noise in the environment affect their localization performance.

Within the RSSI-based approaches, fingerprinting has become a popular choice in indoor environments where LoS connections are not guaranteed and triangulation-based approaches suffer. Despite the advantages, a labor-intensive site survey is always required a *priori* and, thus, represents a challenge for large-scale deployment. Various efforts have been made to reduce the site survey process - some are even completely passive. Finally, other works have focused on algorithms for improved fingerprint recognition and accuracy.

# Chapter 3

# Design

This chapter introduces PMD-Track and describes its design considerations and requirements on an abstract level. Implementation details are presented in more detail at a later point in chapter 4.

## 3.1 Design Considerations

**Accuracy**: PMD-Track's target application is the tracking of PMDs within hospitals or healthcare facilities. A key characteristic of these buildings is their segmentation into a high number of rooms. Thus, to be considered useful, room-level tracking is required.

**Heterogeneous Smartphones**: Deployed as a crowd-sourced application, smartphone brands and models may be heterogeneous and certain limitations may be experienced in the reception of BLE signals or in the usage of system resources.

**Real-Time Location Data**: Near real-time or fresh data is crucial for applications such as PMD-Track. Outdated information can lead to inefficient processes and wasted resources which can ultimately lead to mistrust of users against the system. Thus, it is crucial to manage the user's expectations by communicating the staleness of information.

**Ease of use**: An easy-to-use system allows a quicker user onboarding and benefits the product's reputation. Furthermore, costs can be reduced in user training and support.

**Privacy**: From a user perspective it is important that the person's privacy carrying the smartphone is not violated. Thus, it is crucial that the user's location is not tracked or can't be inferred from other data.

**Security**: Hospitals often implement strict IT-security policies. Thus, minimal friction and touch points with existing IT operations are needed such as the connectivity services through an existing WiFi infrastructure.

## 3.2 Requirements

From the goals of the thesis (*c.f.* section 1.2), we can derive the following requirements:

- **R1** The system shall have room-level tracking accuracy.
- **R2** The system shall be able to track assets in near real-time.
- **R3** The system shall not rely on static gateway devices.
- **R4** The system shall expose the location information through a REST API.

## 3.3 Architecture

The following sections cover PMD-Track's architecture and design considerations. Figure 3.1 illustrates the high-level architecture overview and serves as a reference to guide through the following sections

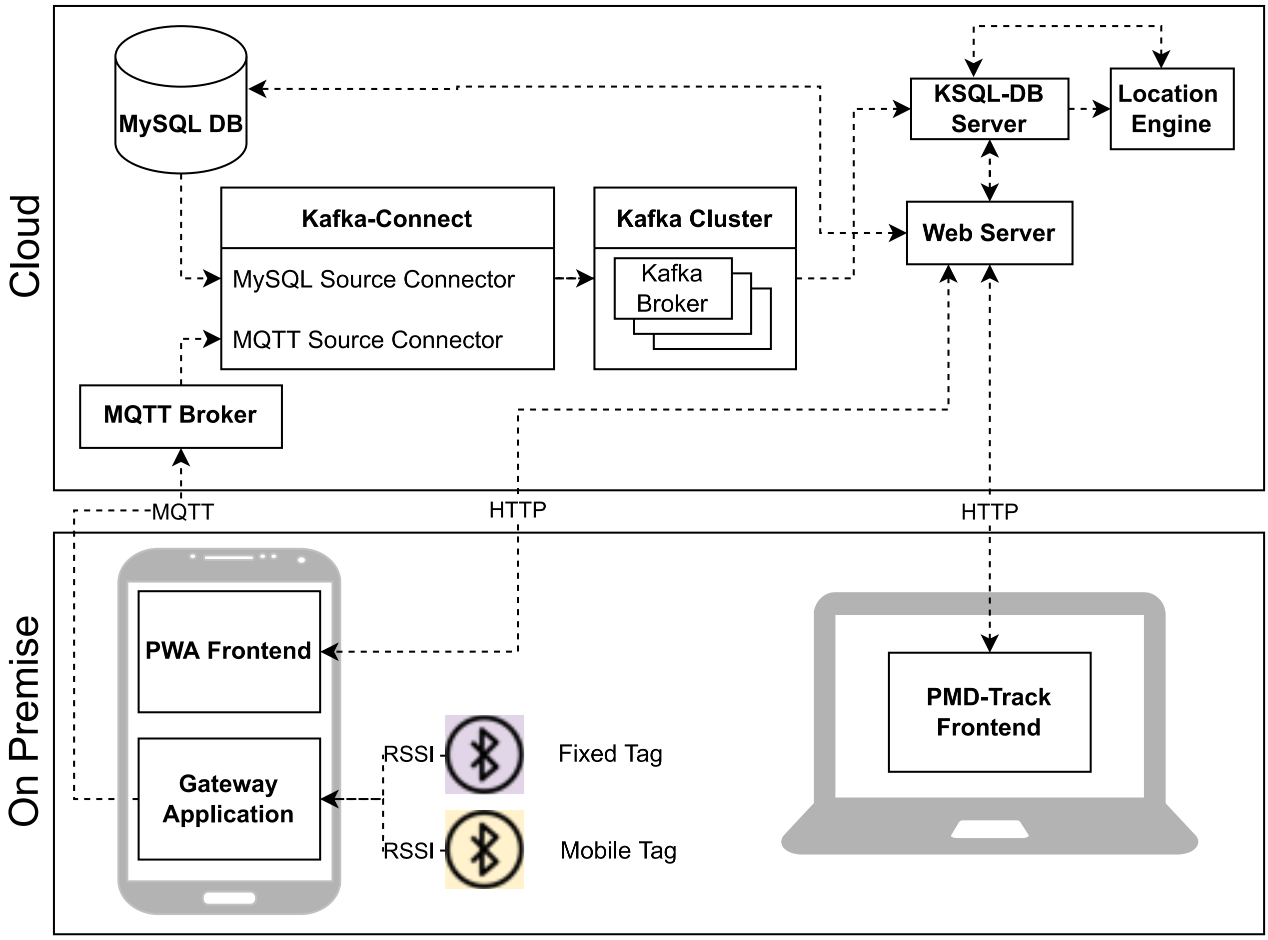


Figure 3.1: PMD-Track Architecture

- **MySQL DB**: Store static inventory data
- **Kafka-Connect**: Integrate various data sources with stream processing framework
- **Kafka Cluster**: Runs Kafka instance
- **KSQL-DB**: Database to query streams (*e.g.* last location of asset)
- **Location Engine**: Predicts asset location in real-time based on raw measurements
- **Web server**: Exposes REST API to query assets and their current location
- **MQTT Broker**: Facilitates on-prem to cloud communication
- **PWA Frontend**: Progressive Web App displaying asset information
- **PMD-Track Frontend**: Dashboard displaying asset information
- **Gateway Application**: Beacon scanning application
- **Fixed / Mobile Tag**: BLE beacons installed either in fixed locations or on PMDs

### 3.3.1 BLE Beacons

BLE beacons are small, inexpensive, and battery-powered devices emitting a periodic BLE signal to advertise their presence to nearby devices. Through its MAC address and RSSI value, a beacon is uniquely identifiable and the distance between it and the receiving device can be approximated by means of the Path-Loss-Model. A common application scenario is proximity detection where an application or device is scanning for beacons in its perimeter to establish a geospatial context associated with the scanned beacons. In the case of PMD-Track, two types of BLE beacons are distinguished. *(i)* **mobile** beacons are attached to the assets one wants to track and are moving along with them. Their MAC address is associated with a specific piece of inventory. *(ii)* **fixed** beacons are installed in predefined areas of the building and thus do not change their location. Their purpose is to serve as reference points for determining the current location of the scanning device within the building. Being able to locate the scanning device allows inferring the location of assets it has detected in its vicinity.

### 3.3.2 Gateway Application

Traditional asset tracking solutions rely on the installation of static gateways that are scattered throughout the building to track tagged assets. A key requirement and novelty of PMD-Track is the absence of such a requirement by leveraging the mobility of the employee workforce. Typically equipped with a smartphone, hospital staff moves through the building and thereby covers a wide area of the hospital's premises in their day-to-day. The gateway application is a BLE tracking software installed on the employee's smartphones that detects BLE beacons that are in proximity. The app runs passively and in the background of the smartphone without requiring any user interaction, thus,

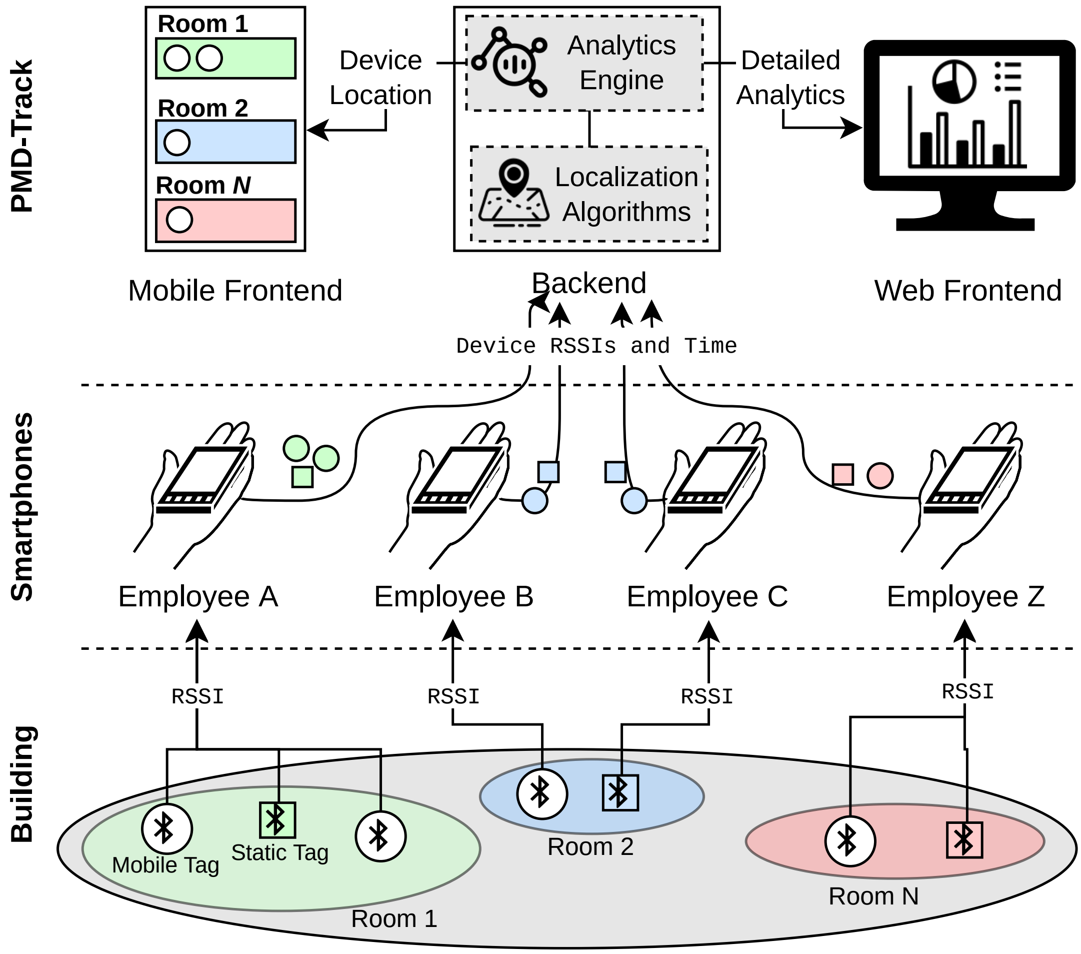


Figure 3.2: PMD-Track Beacon scanning

employees are not disturbed in their activities. It scans both mobile and fixed BLE beacons in range and relays the timestamped information to a cloud service as illustrated in figure 3.2. Reliable and uninterrupted operation of the gateway application is therefore of utmost importance in order to provide up-to-date information.

### 3.3.3 Frontend views

Information about inventory location is displayed through two user interfaces (UI). Field workers are provided a mobile application to quickly query and locate tracked assets. It is important to note that the mobile UI application is a separate app and is not integrated into the existing gateway application that also runs on the smartphone. The reason for this separation is two-fold. *(i)* Apps with the simple purpose of displaying information to users can efficiently be built by leveraging cross-platform frameworks to reduce development effort. On the other hand, the gateway application requires a native implementation as accessing the smartphone's hardware resources (*i.e.* BLE scanning in the background) requires access to OS-level libraries to acquire necessary permissions and to ensure continuous background operation. *(ii)* A separate UI app allows users to consume asset location information without opting in on the BLE tracking due to privacy concerns.

Apart from the mobile application, a web-based dashboard provides asset location information in a single view, suitable for larger displays such as desktops.

### 3.3.4 Communication between gateway and cloud

MQTT is a lightweight, publish-subscribe, machine-to-machine communication protocol. Its low resource consumption makes it a popular choice for IoT applications in resource-constrained environments. In PMD-Track, MQTT is used to facilitate the communication between the gateway application and the cloud service. The gateway application connects to the MQTT broker and publishes aggregated BLE scan results on a predefined topic in regular intervals (*i.e.* every minute). Downstream, a consumer application can also connect to the broker and subscribe to the topic and process the received messages.

The messages on the MQTT topic form a continuous stream of events, each representing a device encounters between the gateway (*i.e.* smartphone) and a BLE beacon it has recently scanned - termed *BleScanEvent*. Its properties include a *client_id* identifying the gateway application, a *mac_address* associated with the BLE beacon, its *rssi* value, and a *timestamp* when it was detected. These messages are fed into a stream processing framework by a connector application that subscribes to the MQTT topic and connects to the stream processing framework to forward the messages on a dedicated stream. Client applications can then consume, aggregate, and act on those events to either derive state or emit new events.

### 3.3.5 Streaming and static data processing

The previous section highlighted how BLE readouts from the gateway application are pushed to the cloud. In this section, the process of handling the continuous stream of BLE readout data and enriching or filtering it with static data is explained. The *BleScanEvents* received in the cloud are small bits of information from various data sources, depending on how many gateway applications are currently running. There is no guarantee that eventually all BLE beacons will be scanned after a certain time as the gateway only detects what currently is in range. If a beacon is lost or never comes into range of a gateway, this beacon will never be detected, thus, no *BleScanEvent* will ever be emitted in the system. In the context of asset tracking, the detection of the absence of an asset is equally important as the detection of its presence. Hence, the need for storing a predefined, stable list of inventory as a base data set, occurs.

Storing a list of inventory can be achieved with a variety of proven technologies. In PMD-Track, a relational databases stores a table of BLE beacons along with their type (*i.e.* *mobile*, *fixed*) and their *mac_address*. Similarly, a table stores a list of active gateway applications along with their *client_id* and human-readable *name*. Finally, a table stores room information about the building.

To make the static information available in the stream processing framework, a connector application periodically fetches the information from the relational database and feeds it into a dedicated stream. Serving as the base data set, a new, enriched stream joins *BleScanEvents* with the static beacon data from the relational database, filtering out non-inventoried beacons.

### 3.3.6 Location Engine

The location engine receives a stream of enriched *BleScanEvents* and produces a stream of positioned *mobile* beacons associated with a location (*i.e.* room). Thus, the asset associated with the *mobile* beacon has been located in the given room.

Room level positioning can be achieved using different localization algorithms - each with its advantages and disadvantages (*c.f.* section 2.2). The remainder of this work focuses on the implementation and comparison of a fingerprinting-based versus multilateration-based localization algorithm with the goal to choose the best-suiting algorithm for the use case of PMD-Track. For the sake of simplicity, the implementation and evaluation of the two approaches will be done outside PMD-Track's architecture. Eventually, however, the location engine is the component where the chosen algorithm will be implemented.

The following sections describe the process of implementing and evaluating the two localization algorithms in an office environment as well as an apartment.

#### 3.3.6.1 Data Preparation

Preparing a single data set per test location is a fundamental step for the subsequent model training and comparison of the two algorithms. The following steps for data gathering and preparation have been done in both test locations.

Figure 3.3 depicts the floor plan of the office test location and figure 3.4 the one of the apartment.

Each room, labeled with a capital letter (A-Z), is equipped with a *fixed* BLE beacon, denoted by the MAC address below the letter of the room. The beacons are installed in the center of the ceiling of each room and the mapping between the room label and MAC address is being stored.

After preparing the test location with beacons, the data set can be gathered. In this process, a person visits each room and takes a certain amount of RSSI samples of beacons that are in range within that room. Recording the RSSI samples is done using a Smartphone app. The user enters the room label where the sampling takes place and the app starts to perform repeated BLE scans. Within each scan - which only lasts a few seconds - the app records the MAC address and RSSI value of the *fixed* beacons it detected. At the end of each scan, the detected beacons and RSSI values are stored in a new data point along with the room label. To make sure RSSI samples are not only taken at a single position within the room, the person moves throughout the room while collecting the samples as can be seen in figure 3.5

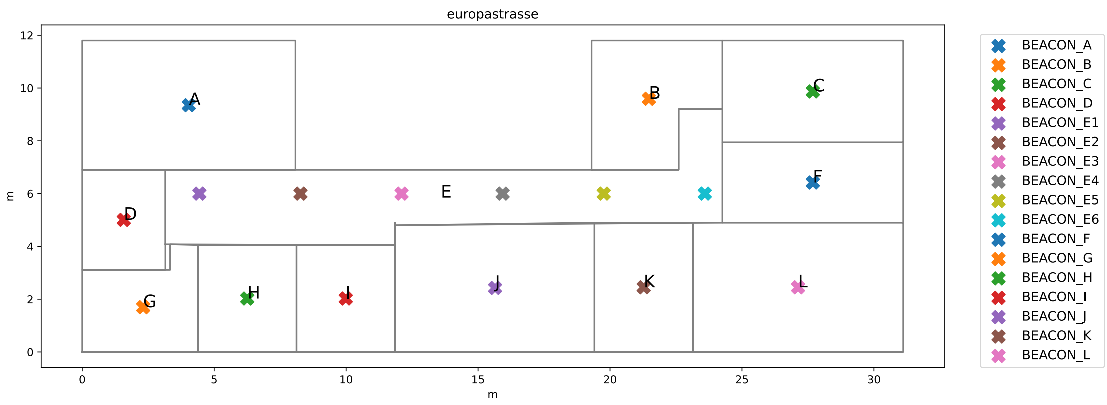


Figure 3.3: Floor plan office

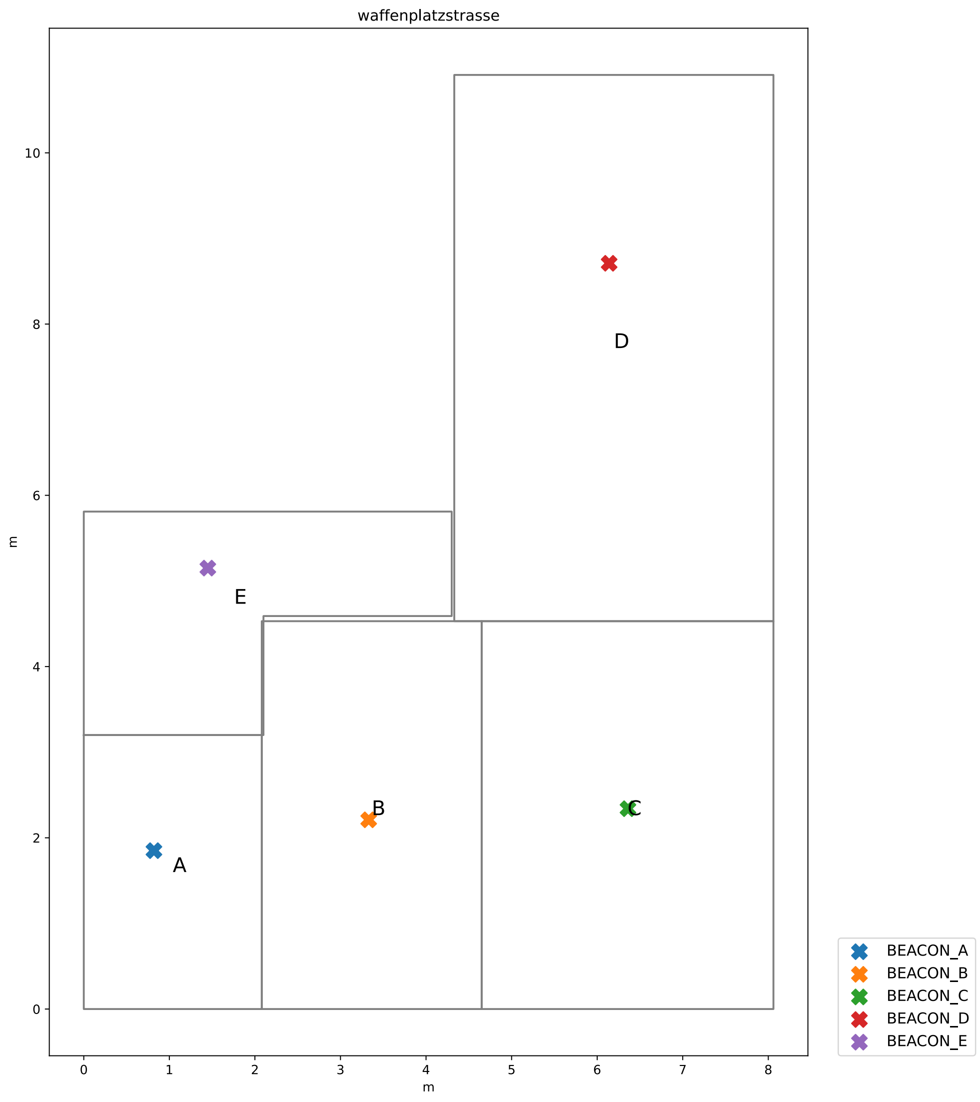


Figure 3.4: Floor plan apartment

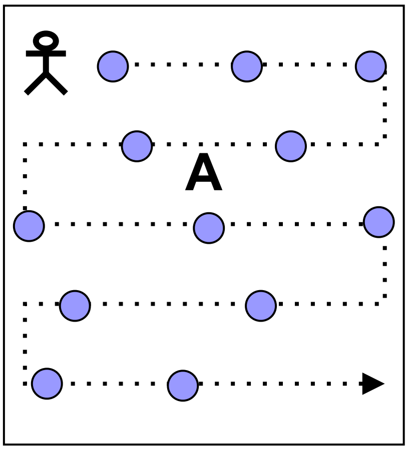


Figure 3.5: Collecting fingerprints at different positions within a room

Table 3.1: Data Set

| **BEACON_A** | **BEACON_B** | **...** | **BEACON_L** | **Room** |
|---|---|---|---|---|
| -65 | -70 | | -99 | A |
| ⋮ | ⋮ | | ⋮ | ⋮ |
| -99 | -75 | | -65 | L |

Once 1'000 data points are collected within the same room, the person stops the app and moves to the next room where the procedure is repeated. Repeating this for $k$ rooms yields the final data set of $k * 1'000$ rows and $k+1$ columns as one column accounts for the room label. As the data collection is terminated after a fixed number of measurements, the final data set is balanced by containing an equal number of samples per class/room. Table 3.3 shows an example of the final data set.

The RSSI value captures the distance relationship between the beacon and the smartphone. Due to environmental noise, not every beacon might be captured in a data point. This results in the sparsity of the final data set (*i.e.* cells being null). As fingerprinting and multilateration-based approaches can't operate on sparse input data, two imputation strategies are applied to fill missing observations. In the first case, a certain beacon might never be detected at all for a given class. Considering the floor plan in figure 3.3, the data points for room $A$ might not have any records of the beacon in room $L$ as it is too far away and thus out of range. In this case, the column of beacon $L$ for room $A$ is set to the constant value of -200 to indicate an out-of-range beacon.

Table 3.2: Fixed value imputation

| **BEACON_J** | **BEACON_J (imputed)** | **Room** |
|---|---|---|
| null | -200 | A |
| null | -200 | A |
| null | -200 | A |
| null | -200 | A |

In the second case, RSSI values are only partially absent due to shadowing or other signal interference effects. Considering again the floor plan in figure 3.3, rows might exist for room $A$ where the RSSI value of the beacon of the adjacent room $D$ is missing. In this case, the conditioned mean of beacon $D$ given room $A$ is used to impute the missing observations. $value = mean(D|Room = A)$.

Table 3.3: Column mean imputation

| **BEACON_D** | **BEACON_D (imputed)** | **Room** |
|---|---|---|
| -65 | -65 | A |
| null | -64.67 | A |
| -62 | -62 | A |
| -67 | -67 | A |

Applying the described imputation strategy for each room eventually yields a non-sparse data set.

#### 3.3.6.2 Model Training

Once the data set is complete and all missing values have been imputed, the two models can be defined, fitted, and evaluated. Predicting the room label given a set of RSSI measurements is a typical classification task. In the case of the fingerprinting model, a k-nearest neighbors (kNN) classifier is employed. On the other hand, an adapted implementation of a multilateration algorithm is presented to approximate the room based on geometric computations.

**kNN** is a non-parametric, supervised learning method and a popular choice for RSSI-based fingerprinting [4, 16, 26]. When used for classification, an unseen data point is classified by a majority vote of its neighboring data points, with the data point being assigned to the class most common among its k nearest neighbors. K is typically chosen to be a small, odd number to avoid a deadlock situation.

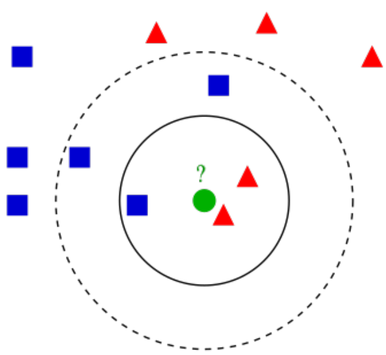


Figure 3.6: kNN example

In the example shown in figure 3.6, the unseen green data point is classified as a red triangle for $k = 3$ or blue square for $k = 5$. kNN uses distance-based metrics to determine neighboring data points. In case of the collected RSSI values, the data set is split into a disjoint training and test data set with the dimensions $cn \times k + 1$ and $(1 - c)n \times k + 1$ for a train-test-split coefficient $c$ (*i.e.* c=0.2). The training phase consists of storing the k-dimensional training samples and their associated class labels. In the classification phase of an unlabeled vector $q$, its k nearest neighbors are calculated from the training data set, based on the chosen distance metric. Eventually, the class label of $q$ is determined based on the majority of its nearest neighbor's class labels.

$$class(q) = majority(k\_min(\parallel q - v_i \parallel))\forall v_i \in train$$

In the case of PMD-Track, the train-test-split was set to 0.2, $k$ was set to 7, and $l_2$ norm (euclidean) was used as the distance metric.

**Multilateration** based localization determines the position of an object by calculating the intersecting point based on the distance to at least 3 known reference points. This geometric-based localization yields a geospatial position with x and y coordinates. In non-line-of-sight conditions, determining the distance based on the RSSI value is error-prone due to environmental signal interference. Further, in the case of PMD-Track, instead of an exact position, room-level granularity suffices. The following sections cover the design of an adapted multilateration algorithm that predicts the room label instead of coordinates in combination with floor plan information.

Distance measurements might not be accurate due to noise in the RSSI signal. Thus, rather than a deterministic intersection point between the ranges of three known points, an overlapping intersection area might exist. Overlaying this intersection area with floor plan information yields coverage areas on a room-level basis. The room with the largest coverage area is considered to be the most probable current location of the asset. The following sections describe different localization scenarios on a case-by-case basis and how they are resolved to a room label prediction.

The first case handles the situation when the smartphone and the fixed beacon are in close proximity such that they can be considered in the same room. This is the case for strong RSSI values above -70 dBm which translates to a range of up to approximately 2 meters.

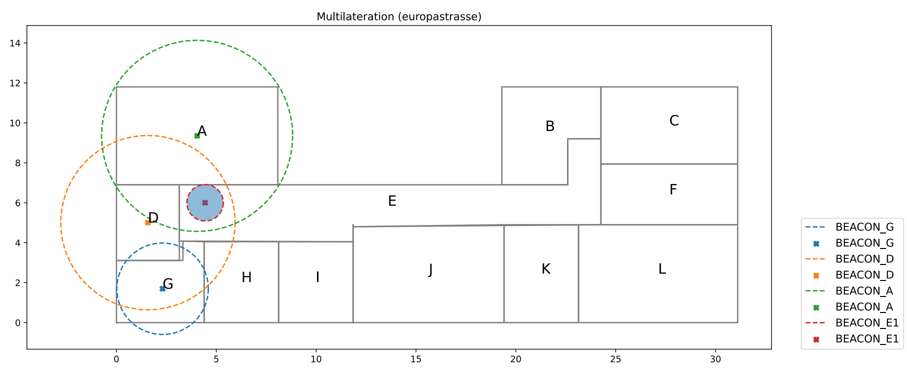


Figure 3.7: Multilateration: close proximity

Figure 3.7 illustrates a situation where fixed beacons from rooms A, D, G, and E are detected by the smartphone. The radii of the dotted circles are calculated based on the path-loss model and indicate the distance measured from the smartphone to each one of them. As can be seen from the plot, the beacon in room E (hallway) is in close proximity and in this case within the range of 2 meters. Thus, the predicted room label in this situation is E.

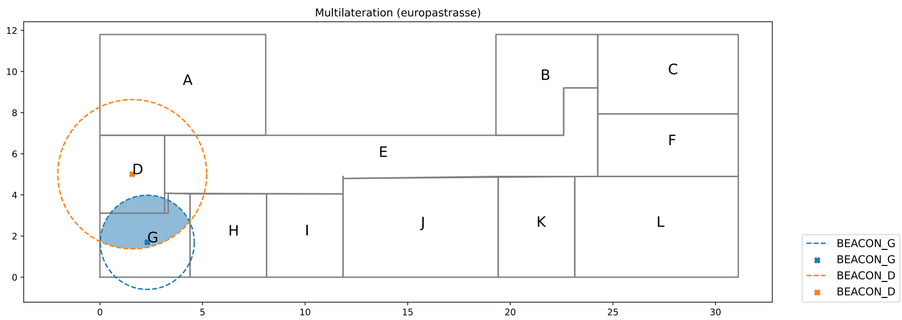


Figure 3.8: Multilateration: Single intersection

If no beacon is within the range of a 2-meter radius, the intersection area of the detected fixed beacons is calculated. In figure 3.8, the intersection area of beacon D and G is filled in blue. Overlaying the intersection area on the building floor plan shows that rooms D, G, and H are covered. The room with the largest coverage area is room G and is therefore the predicted room label.

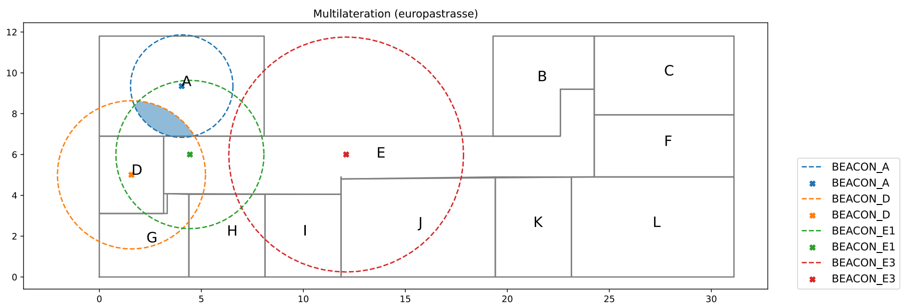


Figure 3.9: Multilateration: Multiple intersections, max intersection cardinality

In case multiple beacons are detected, multiple intersection areas might exist. Figure 3.9 shows such a situation where beacons A, D, E1, and E3 are detected with their respective intersection areas. To characterize different intersection areas, the concepts of *intersection set* and *intersection cardinality* are introduced. Given an intersection $i$, a set of origin shapes can be defined as the intersection set $s_{inter}(i)$ whose elements yield the intersection $i$. For example, given two circles $c_1$ and $c_2$ that produce intersection $i$, the intersection set is defined as $s_{inter}(i) = \{c_1, c_2\}$. Given an intersection set $s$, the intersection cardinality is simply defined as $\parallel s \parallel$. In case of the previous example, $\parallel s_{inter}(i) \parallel = \parallel \{c_1, c_2\} \parallel = 2$.

Coming back to the situation shown in figure 3.9, one can observe that the intersection of beacons A, D, and E1 has the highest cardinality. *i.e.*

$$\exists s_{inter}(i) \mid \parallel s_{inter}(i) \parallel > \parallel s_{inter}(j) \parallel \forall j \in intersections \wedge i \neq j$$

Analogously to the previous case, once the intersection with the highest cardinality is determined, the intersection area is overlayed on the floor plan and the room with the largest coverage area is predicted as the room label. In this case, it's room A.

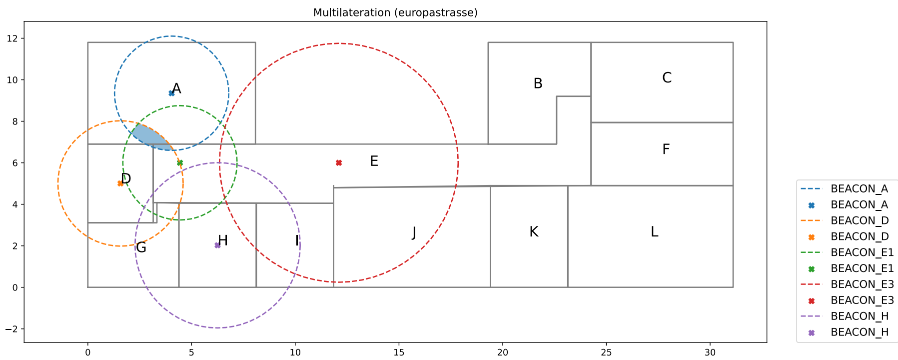


Figure 3.10: Multilateration: Multiple intersections, min sum of radii

As shown in the previous example, multiple intersections with different intersection cardinalities may occur. In case there exist multiple intersections with the same maximum cardinality, another selection criteria is applied to determine the intersection to be overlaid on the floor plan. To this end, the notion of *radii sum* is introduced. Given an intersection set $s_{inter}(i) = \{c_1, c_2, ..., c_n\}$, the radii sum is defined as the sum of the individual radii $r_{c_k}$

$$R_i = \sum_{k=1}^{n} r_{c_k}$$

Such a situation is depicted in figure 3.10 where there exist three intersections with cardinality 3 ($\{A, D, E1\}$, $\{D, E1, H\}$, $\{H, E1, E3\}$). Based on the fact that RSSI noise increases with distance, a strong RSSI signal is more stable and accurate than a weak signal. The radius of the dotted circles inversely correlates to the signal strength (*i.e.* the stronger the RSSI signal, the smaller the radius). Thus, the smaller circles provide a more accurate indication on the actual distance than larger circles. On this premise, the intersection set $\{A, D, E1\}$ is considered the most reliable intersection to be used for room label prediction. Overlaying it on the floor plan reveals that room A has the largest coverage area.

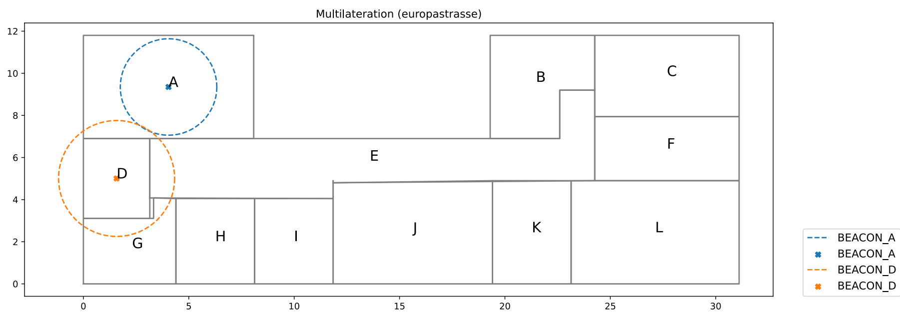


Figure 3.11: Multilateration: No intersection

Finally, a situation might occur where no two beacon signals overlap. This case is illustrated in figure 3.11 where the ranges of beacon A and D don't overlap or touch at any point. In this case, the beacon closest to the smartphone is considered to predict the room label. In the situation at hand, the room predicted is A as beacon A has a stronger RSSI signal and is thus closer to the smartphone.

# Chapter 4

# Implementation

This chapter covers the implementation details of PMD-Track.

## 4.1 Gateway Application

The gateway application is designed and implemented to be run on Android Smartphones. As the app does not offer any user interaction but runs in the background, its reliable and continuous operation is crucial for the system's data intake.

The Android operating system (as well as iOS) enforces multiple privacy and resource policies, that apps are required to adhere to, in order to maintain the user's privacy and enhance device performance. This represents an important challenge in the case of PMD-Track as its gateway application relies on a persistent internet connection as well as a continuous scan of BLE devices. Additionally, apps that are not in the foreground are assumed to be less important to users and thus are facing drastic limitations such as the deprivation of resources.

To mitigate the risk of the gateway application being killed by the operating system, a set of Android-specific precautions have to be taken. Firstly, a foreground service has to be launched to indicate the app's background operation. Secondly, location permissions have to be set up correctly in order to get access to background BLE scans. Further, the BLE scan has to be configured with a set of scan filters (c.f. code block below) as unfiltered BLE scans are considered potentially harmful to user privacy, and thus, some manufacturers such as Samsung have decided to prohibit unfiltered scans in the background.

```
List<ScanFilter> filters = List.of(
    new ScanFilter.Builder()
            .setManufacturerData(65535, new byte[]{}) // Custom manufacturer
            .build());

ScanSettings settings = new ScanSettings.Builder()
    .setScanMode(ScanSettings.SCAN_MODE_LOW_LATENCY)
    .build()

bluetoothLeScanner.startScan(filters, settings, leScanCallback);
```

On the other hand, a wake lock has to be acquired in order to prevent the phone from entering doze mode, in which the network adapter is turned off temporarily. Finally, the app needs to be whitelisted from the battery optimization to avoid being killed after a specific period. Despite these precautions, vendor-specific customizations of the Android operating system might still render a threat to the app's stable operation.

## 4.2 Kafka Stream Processing

Apache Kafka has been chosen as a stream processing framework to facilitate the communication of services and to enable the processing of BLE scan results. For the sake of simplicity, the Kafka infrastructure scaling has been kept at a minimum with single cluster and single partition deployment. The yaml extract below shows the configuration used for running a single Kafka container.

```
kafka:
    image: confluentinc/cp-kafka
    container_name: kafka
    depends_on:
      - zookeeper
    ports:
      - "9092:9092"
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka:9092,PLAINTEXT_HOST://localhost:29092
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
      KAFKA_GROUP_INITIAL_REBALANCE_DELAY_MS: 0
      KAFKA_TRANSACTION_STATE_LOG_MIN_ISR: 1
      KAFKA_TRANSACTION_STATE_LOG_REPLICATION_FACTOR: 1
```

To integrate the MySQL database as well as the MQTT broker with the Kafka stream processing framework, Kafka-Connect, a free and open-source component of Confluent's Kafka suite has been chosen. Different data sources can be included by using the service's exposed REST API. The below configuration files describe the MySQl and MQTT data source configuration respectively.

```
{
  "name": "mysql-source",
  "config": {
    "connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",
    "connection.url": "jdbc:mysql://mysql:3306/cliniq",
    "connection.user": "root",
    "connection.password": "root",
    "mode": "bulk",
    "table.whitelist": "beacons,readers",
    "topic.prefix": "mysql_"
  }
}
```

```
{
  "name": "mqtt-source",
  "config": {
    "connector.class": "io.confluent.connect.mqtt.MqttSourceConnector",
    "tasks.max": 1,
    "mqtt.server.uri": "tcp://<BROKER_IP_ADDRESS>:1883",
    "mqtt.topics": "dataloc/cliniq/ble-batch-scan-results",
    "kafka.topic": "ble_batch_scan_results",
    "value.converter": "org.apache.kafka.connect.converters.ByteArrayConverter",
    "confluent.topic.bootstrap.servers": "kafka:9092",
    "confluent.topic.replication.factor": 1
  }
}
```

## 4.3 Location Engine

The analysis and comparison of the localization algorithms has been done in a python-based Jupyter Notebook. For the implementation of the kNN classifier, Scikit-learn, a popular library for machine learning tasks has been used. The below code snippet was used to train and test the kNN model on a specific set of features.

```
def train_test_model(df, selected_features, selected_classes, ...):
  knn = KNeighborsClassifier(n_neighbors=7)

  # Select features & classes
  df_ = df[df['class'].isin(selected_classes)][[*selected_features, 'class']]

  # Run multiple epochs to average out model accuracy
  score_list = []
  train_df, test_df = None, None
  for epoch in range(max_iter):

    # Only use a subset of the data to train / test the model.
    if n_samples:
      df_ = df.groupby('class', as_index=False, group_keys=False)
              .apply(lambda x: x.sample(n_samples))

    train_df, test_df = train_test_split(
                            df_,
                            test_size=test_size,
                            shuffle=True,
                            random_state=random_state)

    knn.fit(g[selected_features], train_df['class'])
    score = knn.score(test_df[selected_features], test_df['class'])
    score_list.append(score)

  return np.mean(score_list)
```

For the implementation of the multilateration-based localization algorithm, the shapely library was used to perform geometric computations such as the calculation of intersections or the discretization of floor plans into a set of shapes. The latter was performed in a manual process, where the floor plan was discretized into a set of polygons where each element of an array represents a corner, as can be seen in the JSON file below.

```
{
  'A': Polygon([(0,6.90), (0,11.80), (8.08, 11.80), (8.08,6.90)]),
  'B': Polygon([(19.30,6.90), (19.30,11.80), (24.26,11.80), (24.26,9.20), ...],
  'C': Polygon([(24.26,7.94), (24.26,11.80), (31.11,11.80), (31.11,7.94)]),
  'D': Polygon([(0,3.11), (0,6.90), (3.15,6.90), (3.15,3.11)]),
  'E': Polygon([(3.15,4.08), (3.15,6.90), (22.60,6.90), (22.60,9.20), ...]),
  'F': Polygon([(24.26,4.90), (24.26,7.94), (31.11,7.94), (31.11,4.90)]),
  'G': Polygon([(0,0), (0,3.11), (3.33,3.11), (3.33, 4.08), ...]),
  'H': Polygon([(4.39,0), (4.39, 4.05), (8.12, 4.05), (8.12, 0)]),
  'I': Polygon([(8.12,0), (8.12,4.05), (11.85,4.05), (11.85,0)]),
  'J': Polygon([(11.85,0), (11.85,4.80), (19.41,4.90), (19.41,0)]),
  'K': Polygon([(19.41,0), (19.41,4.90), (23.14,4.90), (23.14,0)]),
  'L': Polygon([(23.14,0), (23.14,4.90), (31.11, 4.90), (31.11, 0)]),
}
```

The computationally expensive calculation of possible feature combinations has been parallelized on multiple CPU cores to decrease the processing time.

## 4.4 Frontend

Both the mobile and web-based frontend application have been adapted for the pilot study. Google Maps was embedded to display the location of tracked PMDs on a map of Zurich. Figures 4.1 and 4.2 show the mobile and desktop UI applications.

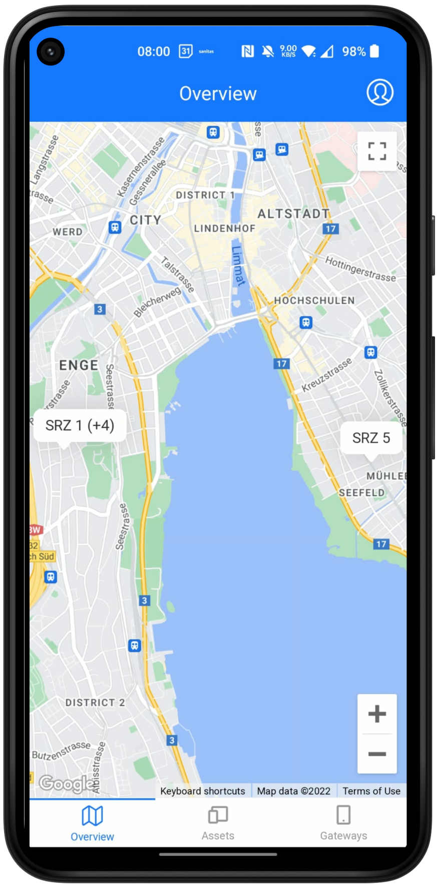


Figure 4.1: Mobile Application

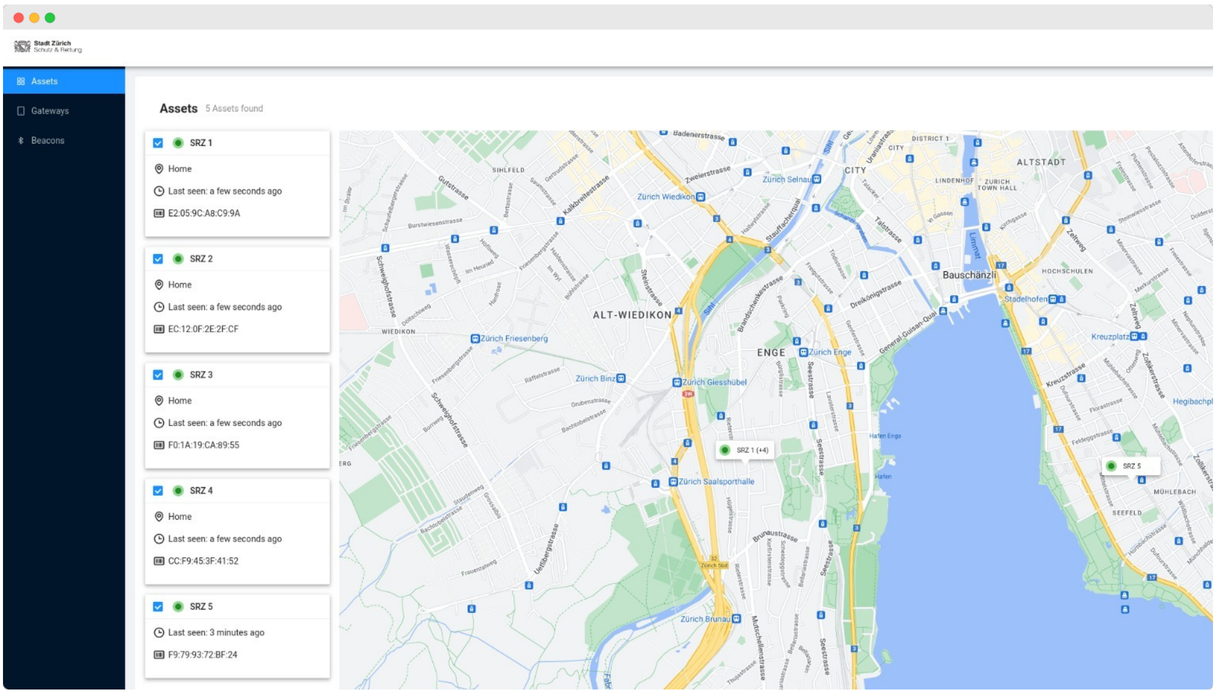


Figure 4.2: Desktop Application

# Chapter 5

# Evaluation

This chapter describes the experiments conducted to validate the chosen approach and to contrast the results with the thesis goals and requirements.

The same evaluation was performed in two test locations independently - an office and an apartment building. For both test locations, an initial data set was gathered on which both the fingerprinting and multilateration-based approaches were trained and evaluated.

The goal of the evaluation is the analysis and comparison of the model accuracy under different input feature configurations with the goal of minimizing the feature input space (*i.e.* number of beacons) while maximizing model accuracy. A column in the data set corresponds to a certain beacon or feature. As the initial data set was gathered with one beacon per room, training and evaluating the model with different beacon configurations can be achieved by simply considering a subset of the columns (*i.e.* beacons or features) at a time.

The findings of the accuracy analysis are combined with a set of economic considerations to support the decision-making process when implementing an indoor localization solution. Further, a prototypical implementation of the system has been deployed and evaluated within the scope of a pilot project at the emergency department of the city of Zurich. Finally, a discussion wraps up the evaluation by contrasting the outcomes with the thesis goals.

## 5.1 Model accuracy vs number of beacons

An input feature combination is a subset of beacons the model is trained on. The following section compares the accuracy of the competing models under possible input feature combinations. The evaluation is conducted against all classes/rooms of the data set. As an evaluation metric, accuracy was used as it is a common choice with non-skewed data. Both data sets are well balanced with respect to the class members as can be seen in figures 5.1 and 5.2 which show the number of samples per class.

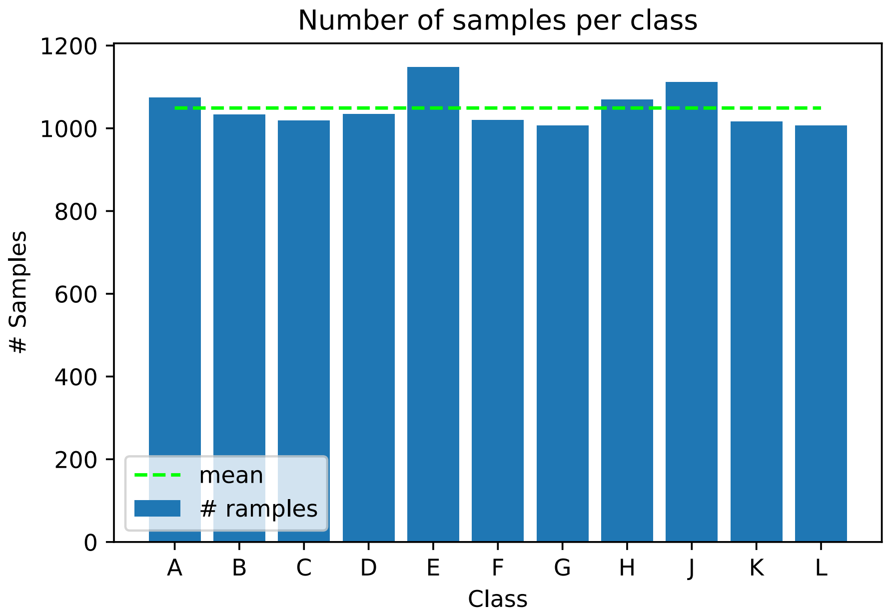


Figure 5.1: Office test location

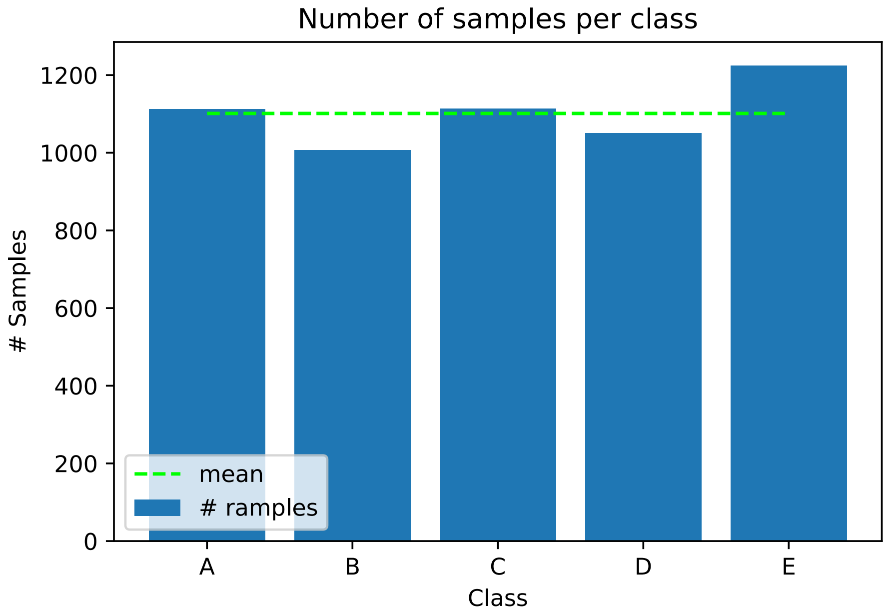


Figure 5.2: Apartment test location

To evaluate the accuracy of a specific input feature combination, the data set is first partitioned into a training and testing set with an 80% and 20% split respectively. Subsequently and in the case of kNN, the model fits on the training set. Finally, the accuracy is calculated on the testing set by computing the number of correctly classified samples divided by the total number of samples in the testing set. Cross-validation is performed to achieve a more reliable accuracy score by repeatedly training and testing the model on different train-test-splits.

The remainder of this section describes the results obtained by evaluating the models on all possible feature input combinations. As mentioned above, a feature combination is a subset of the total number of features. Thus, there are feature combinations of length $1, 2, ..., d$ where $d$ is the number of available features. Performing this evaluation is computationally expensive due to the high number of combinations.

$$combinations = \sum_{i=1}^{d} \frac{d!}{i!(d-i)!}$$

For the office and apartment test location this yields a total of 65'535 and 31 combinations respectively. In the following analysis, accuracy scores are grouped by the number of features involved and averaged.

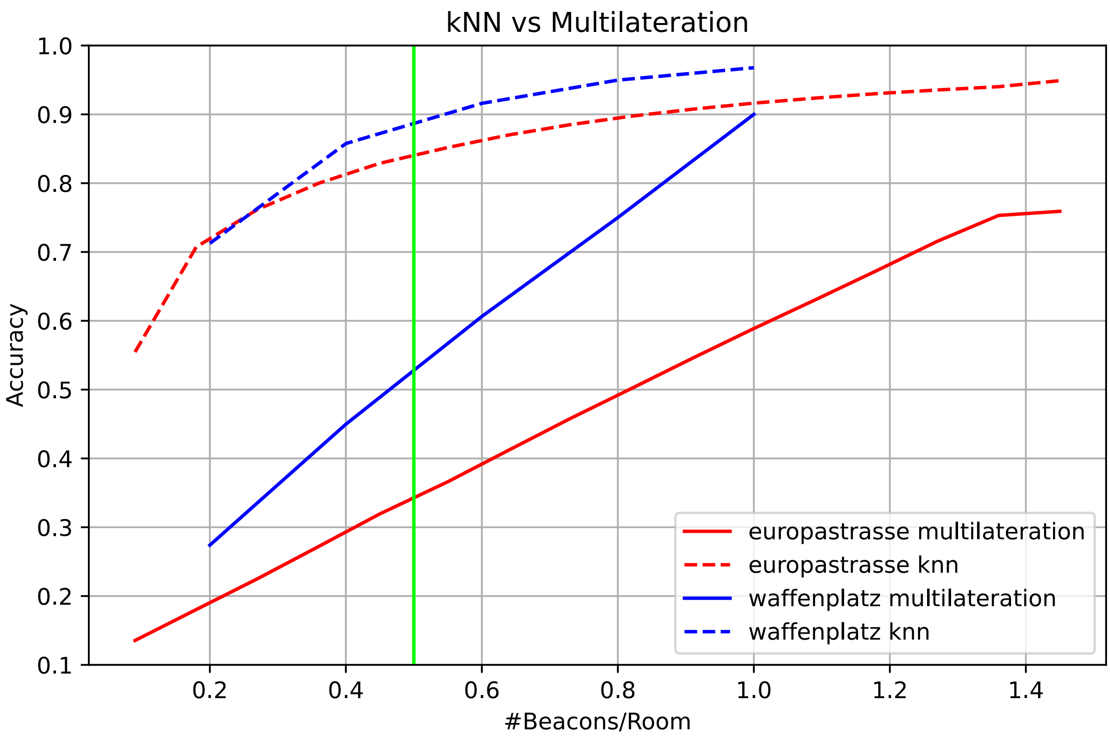


Figure 5.3: Accuracy threshold

Figure 5.3 shows accuracy on the y-axis and the number of beacons per room ratio on the x-axis. For example, the green vertical line indicates the accuracy score for a beacon per room ratio of 0.5. For the office test location, this means that equipping half of all available rooms with beacons yields an accuracy of 35% and 83% on average for the multilateration-based and kNN-based approaches respectively. It is worth noting that the office test locations are equipped with 16 beacons and have 11 rooms, which explains the beacon-per-room ratio of up to 1.4. Based on this interpretation, one can easily observe that kNN performs considerably better than multilateration for all beacon per room ratios and across test locations. Further, it appears that accuracy for the fingerprinting-based approach grows logarithmically whereas the multilateration-based approach has a more linearly-shaped growth.

The box plots in figures 5.4, 5.6, 5.5, and 5.7 show the distribution of accuracy within a group of a certain number of features. A wide distribution indicates that the choice of beacons has a high impact on the accuracy whereas a narrow distribution suggests that the choice is less relevant concerning achieved accuracy.

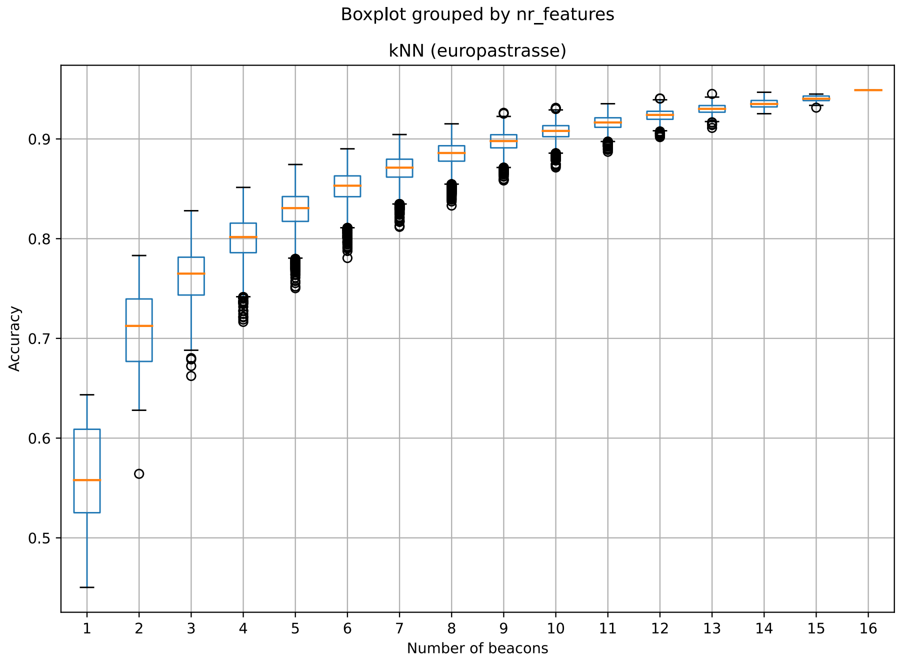


Figure 5.4: Accuracy box plot kNN (office)

For example, in Figure 5.4 for the office test location and the kNN model, one can observe that installing 3 beacons in arbitrary rooms, yields a median accuracy of approximately 77% with Q1 and Q2 quartiles around 75% and 78% respectively. It is straightforward to observe that as the number of beacons increases, the accuracy distributions get narrower. Alternatively, for a low number of beacons (1-3), the choice of beacons seems to have a high impact as can be seen by the wider distributions.

This observation is consistent with the evaluation of the kNN model in the apartment location with the exception of beacon configurations of size 1 (c.f. figure 5.5. For the multilateration model in the office test location, accuracy distributions seem more similar in terms of variance across different numbers of beacons (c.f. figure 5.6).

For the apartment, the distributions follow a similar pattern as with the kNN-based model (c.f. figure 5.7).

## 5.2 Model accuracy vs placement of beacons

As a next step, an interesting question is to analyze the significance of individual beacons with respect to accuracy to better understand what beacons contribute more to accuracy than others. Combining this with floor plan information might reveal geospatial patterns as to where to place beacons on a floor plan to achieve maximum accuracy. To this end, the

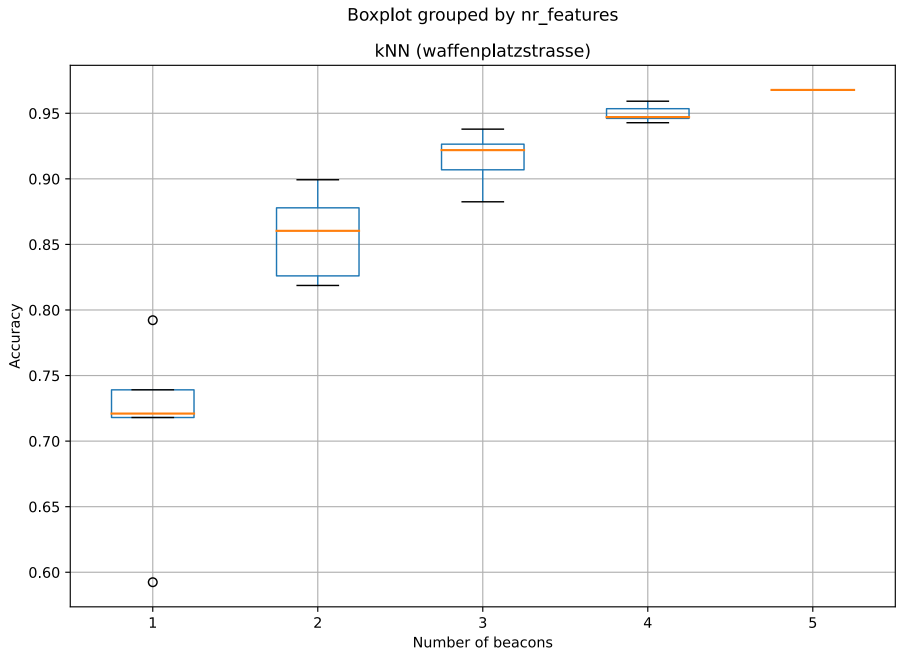


Figure 5.5: Accuracy box plot kNN (apartment)

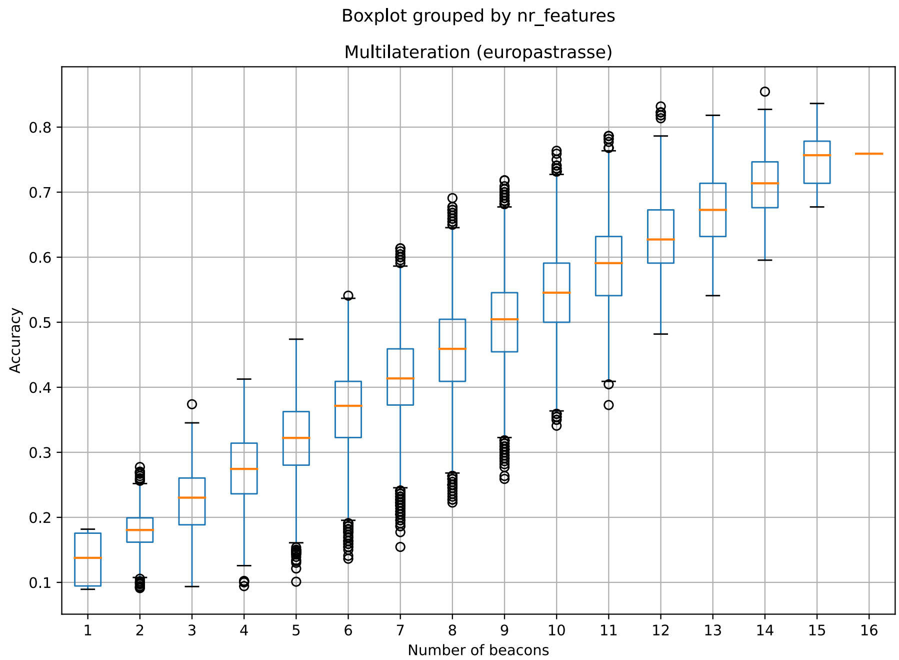


Figure 5.6: Accuracy box plot multilateration (office)

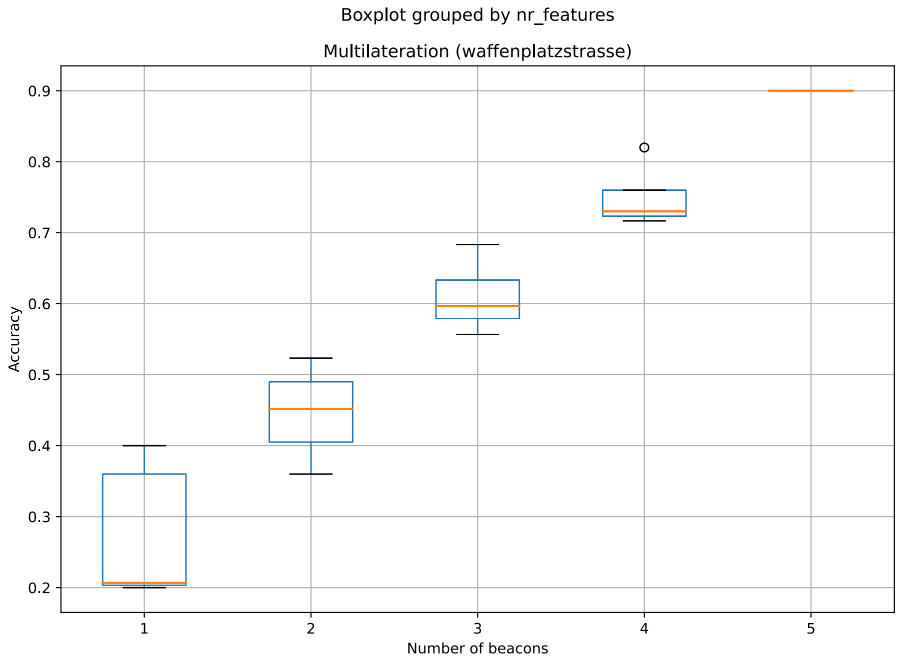


Figure 5.7: Accuracy box plot multilateration (apartment)

frequency is analyzed with which beacons occur in the top-ranking beacon configuration of each feature space length. In this sense, for each group of specific feature-length (*e.g.* all combinations of length 3), the combination is selected which achieves the best accuracy score. Repeating this for all feature lengths and counting the occurrence of each beacon yields the bar plots in figures 5.8 and 5.9.

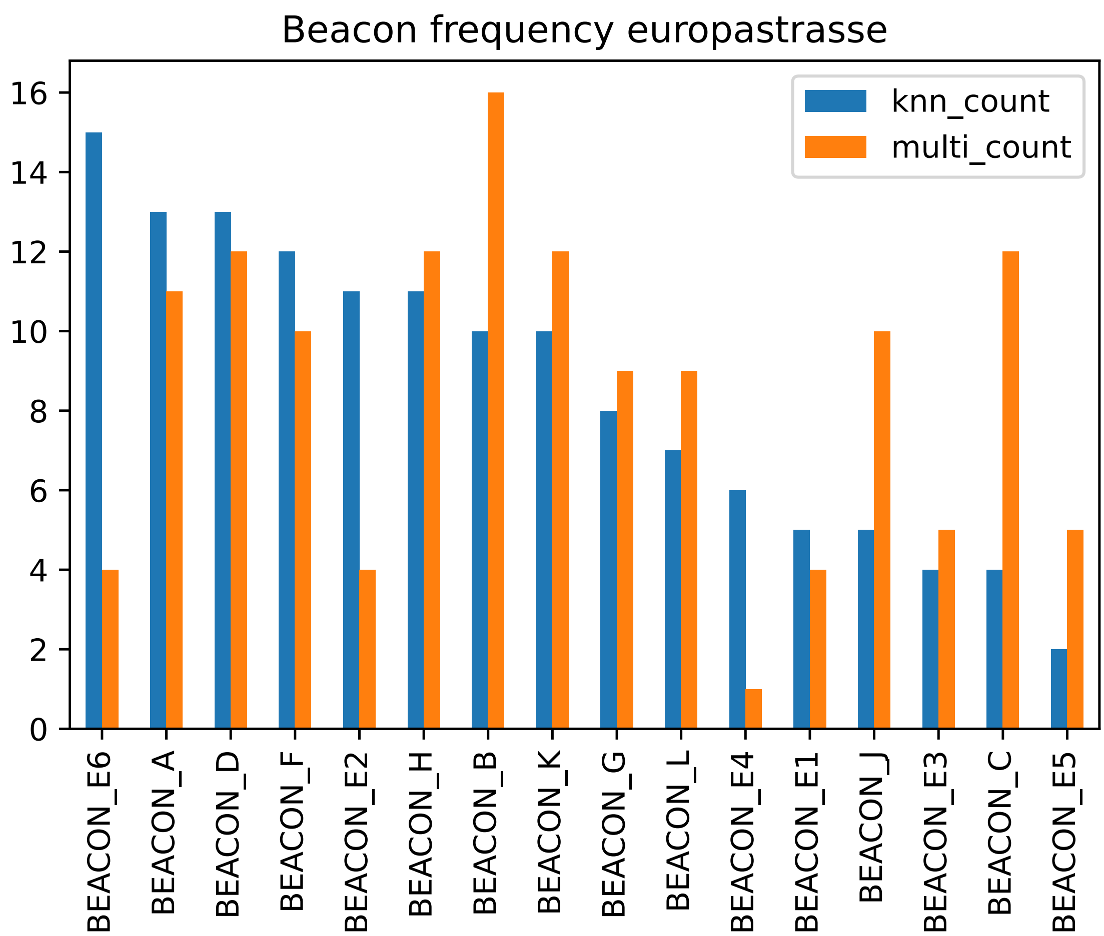


Figure 5.8: Beacon frequency office

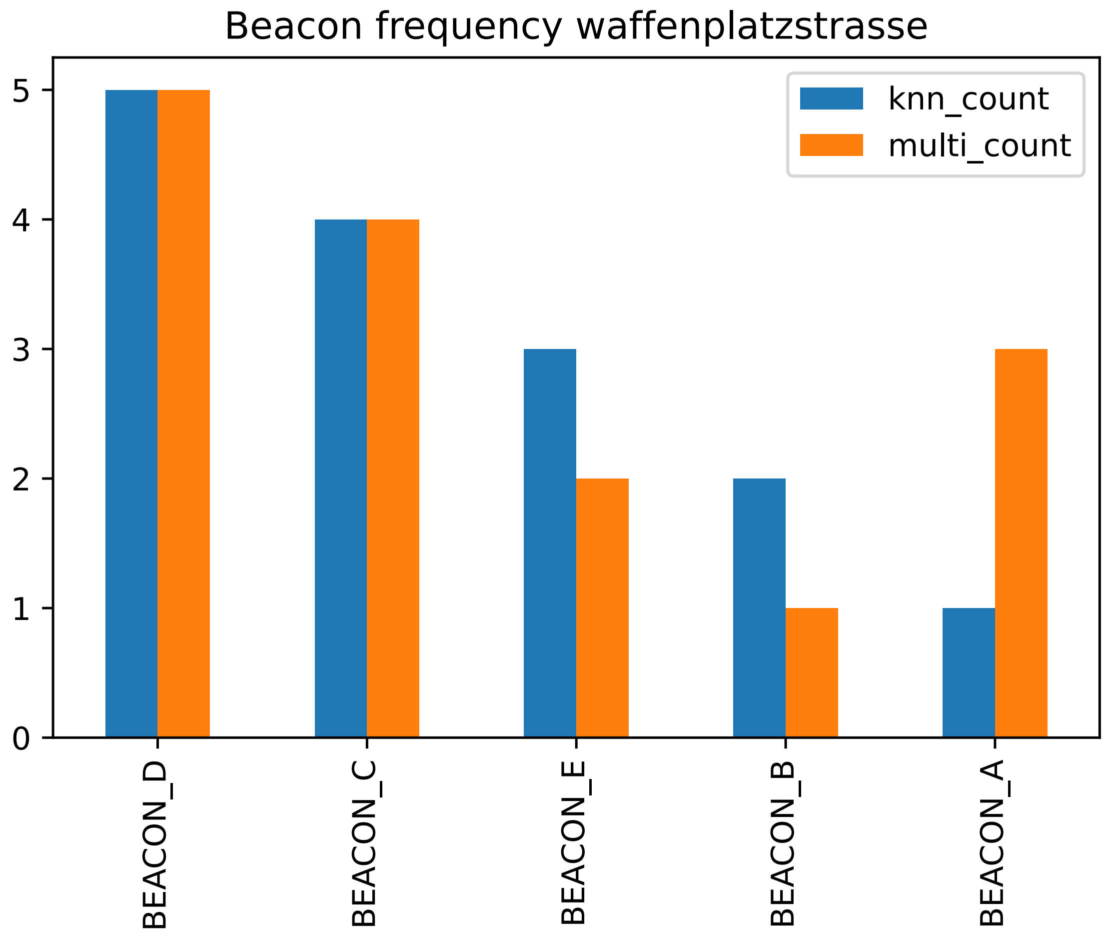


Figure 5.9: Beacon frequency apartment

While the beacon frequency of kNN and multilateration-based approaches are almost identical in the apartment, the office test location shows another picture. For certain beacons, the frequency count for both models is in a similar range and may be off by a count of 1 to 2. However, there are beacons for which the discrepancy in the frequency count is higher. For example, most beacons in the hallway (E6, E2, E4, and E5) are more important for the fingerprinting-based approach than for the multilateration-based one.

Figure 5.10 shows the position of the fixed beacons overlaid on the floor plan where the size of the beacon corresponds to its frequency count.

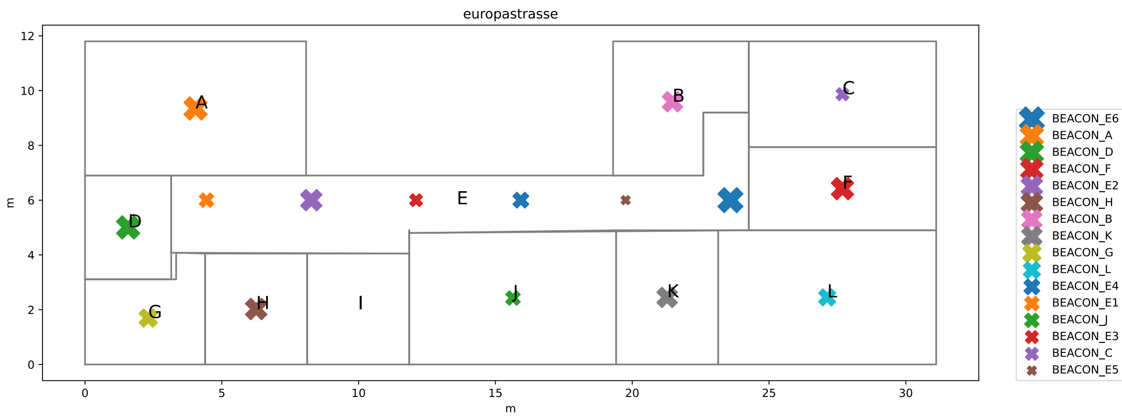


Figure 5.10: Beacon frequency office (kNN)

It can be seen that beacons D, E6, and F are all aligned horizontally and vertically centered. Further, if beacon A is also considered, it seems that two pairs of beacons emerge, one on the left side (D, A) and one on the right side (E6, F).

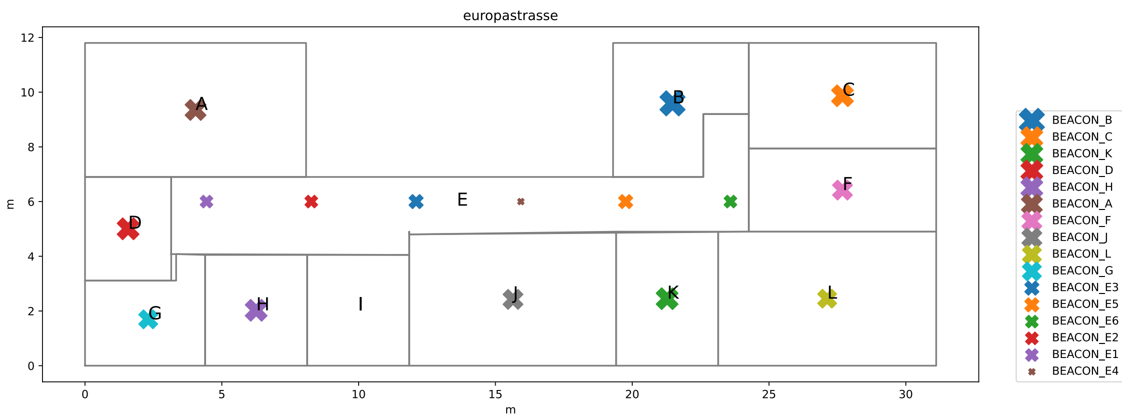


Figure 5.11: Beacon frequency office (multilateration)

On the other hand, figure 5.11 shows the same map but for the multilateration-based approach where another pattern emerges. In this case, the beacons in the center of the floor plan (along the hallway) are the least important ones. Instead, beacons in the outer rooms seem to have higher importance.

The described observations indicate general-pattern candidate locations for each localization approach. In the case of fingerprinting-based localization, beacons might be installed in pairs across a horizontally or vertically centered line with respect to the floor plan. Conversely, beacons might be best placed on the outermost, border rooms of the floor plan. Despite those observations, it is important to highlight the qualitative nature of these results. In the case of the apartment, the beacon frequency analysis was not conducted due to the low number of beacons (only 5 beacons).

## 5.3 Model accuracy vs training size

Despite the number and position of fixed beacons, the model training size has an influence on the accuracy of the fingerprinting-based model. The following sections analyze the model accuracy of the kNN localization model subject to the training size. Similar to the previous evaluation, the model was repeatedly trained and tested on different training data sizes ranging from 20 to 200 samples per room/class. The motivation for this analysis is to reduce the overhead induced by collecting samples for the training database.

Figure 5.12 depicts a heatmap showing the number of beacons on the y-axis and training size on the x-axis.

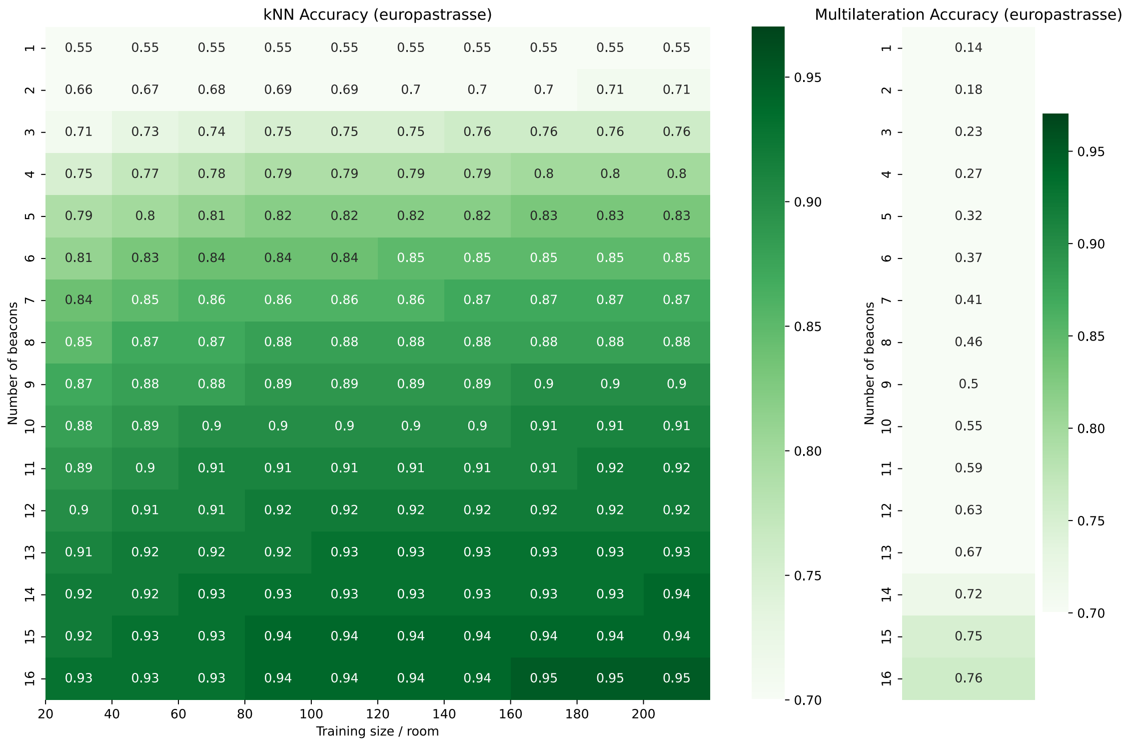


Figure 5.12: Accuracy scores for office

Training size has a positive effect on model accuracy as can be seen by the increasing accuracy score. Further to notice is that the positive effect decreases with an increasing number of beacons. For example, the accuracy of a 3-beacon model can be increased by 5% when using a training set size of 200 instead of 20. In contrast, with a 16-beacon model, the accuracy can only be improved by 2%. Figure 5.13 illustrates this behaviour in a 3D contour plot. Looking at the first derivative of the accuracy vs training size relationship confirms that trend and additionally shows the diminishing marginal effect of additional training samples (c.f. Figure 5.14).

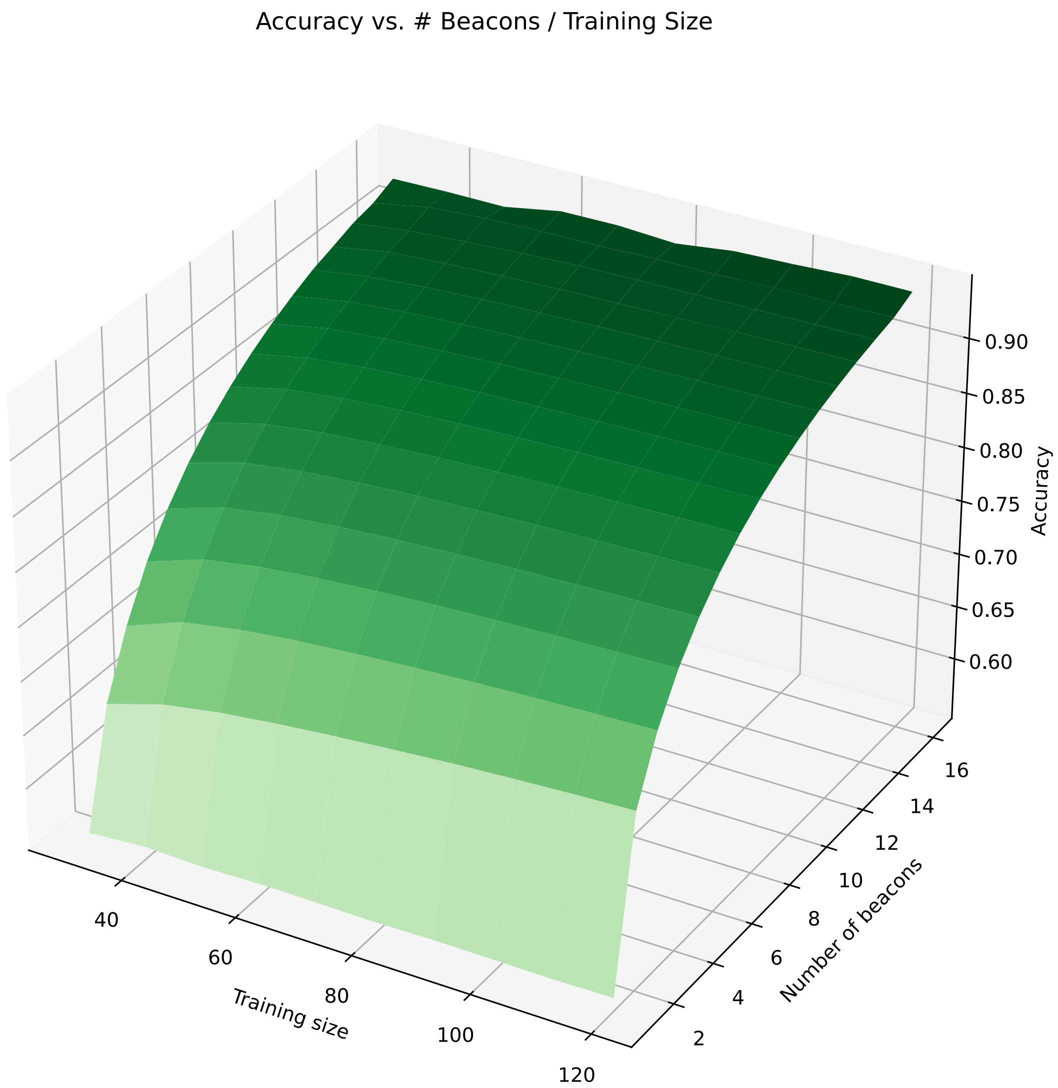


Figure 5.13: Accuracy curve for office

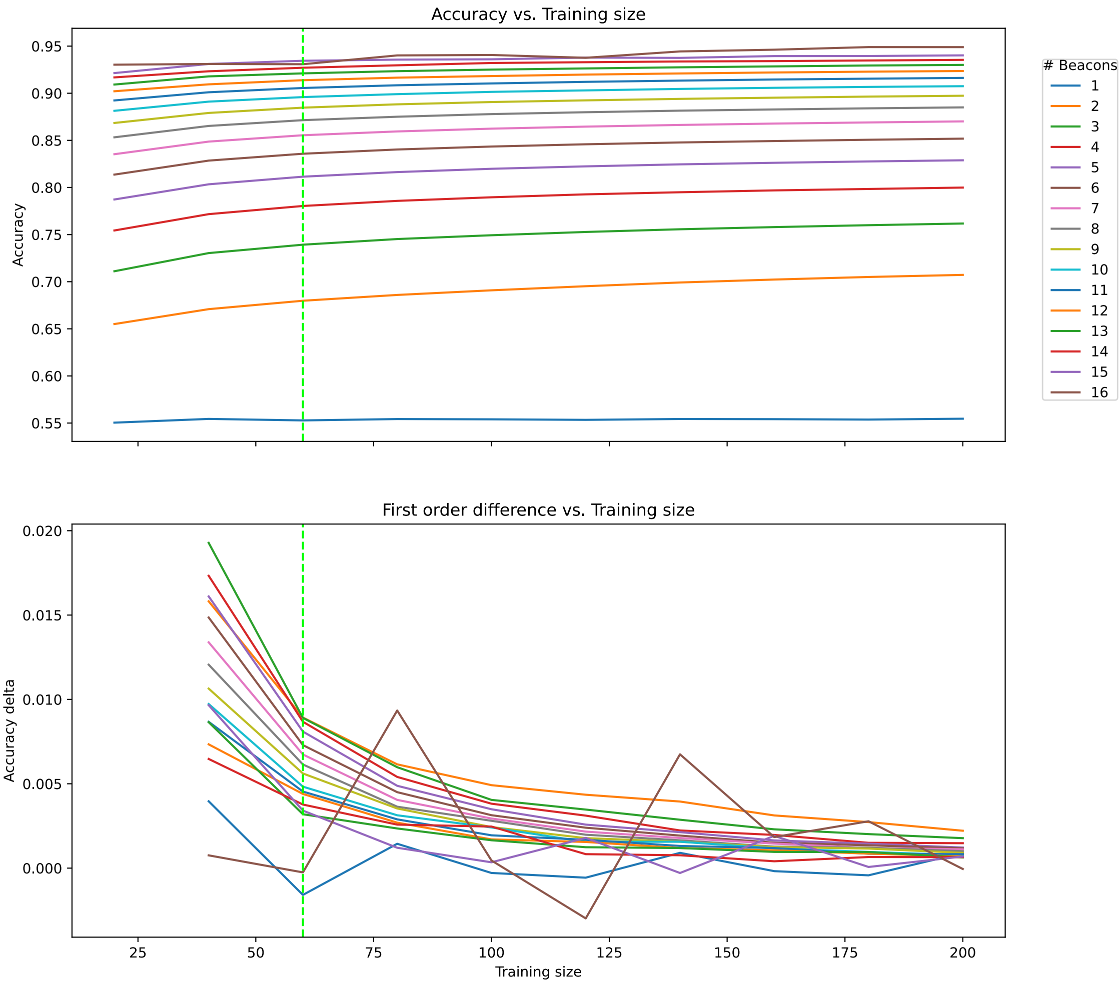


Figure 5.14: Accuracy vs training size (office)

A similar, though less consistent behaviour can be observed for the apartment test location in figures 5.15, 5.16, and 5.17.

Figure 5.15: Accuracy scores for apartment

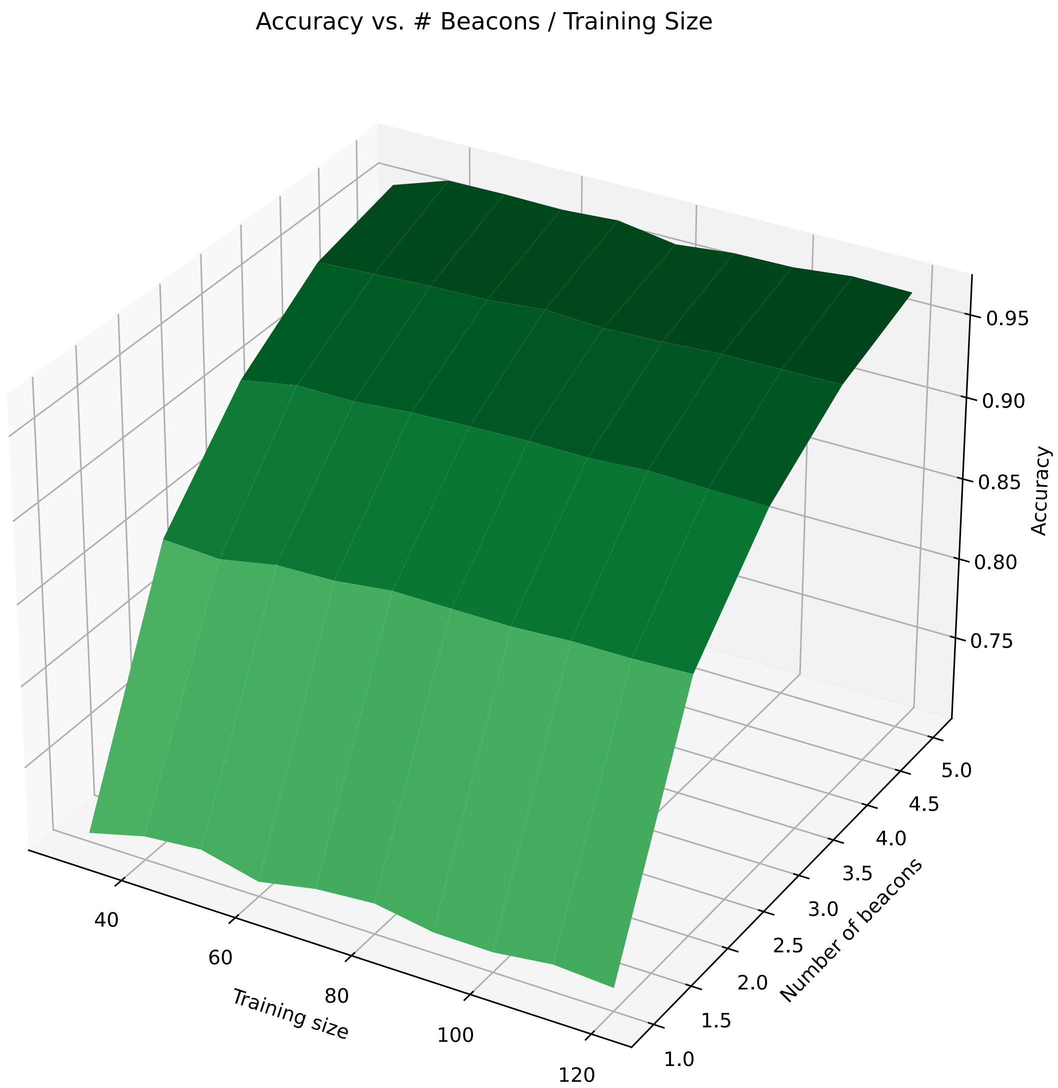


Figure 5.16: Accuracy curve for apartment

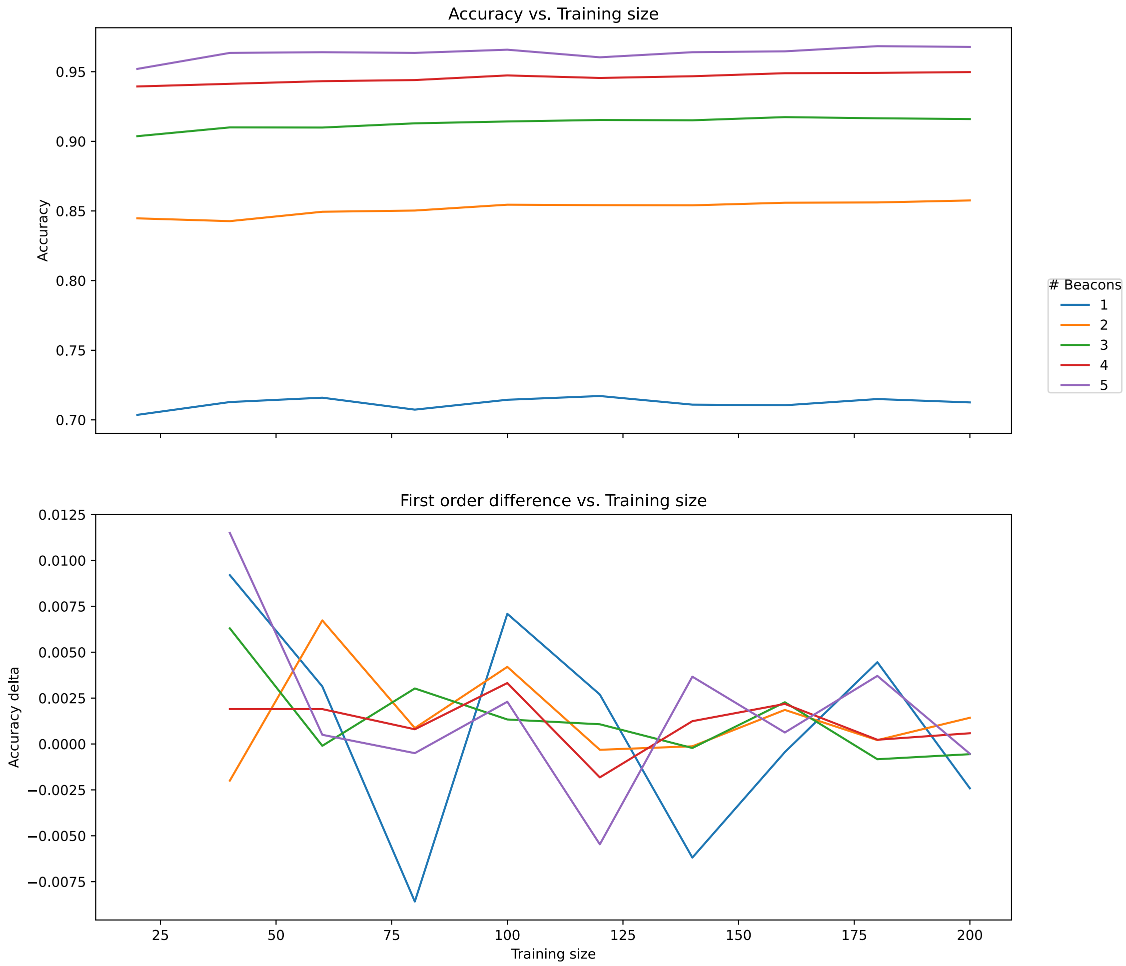


Figure 5.17: Accuracy vs training size (apartment)

## 5.4 Economical Evaluation

With the given analysis on model accuracy, the stage is set for a set of economical considerations regarding the cost of ownership associated with the deployment of each localization algorithm. Table 5.1 specifies a set of assumptions the following economical evaluation is based. First of all, the scenario envisions the deployment in a location with 500 rooms which corresponds to typical hospital size. The installation and labeling of a fixed beacon on the ceiling of a room is assumed to be carried out by a trained field worker with an hourly rate of $30 and an average installation time of 15 minutes. The training time of 15 minutes to collect 200 training samples per room is based on our experience and is also accounted for with an hourly rate of $30. Hardware components such as beacons and required batteries can be purchased at $5 and $2 respectively at the time of writing. Finally, a beacon-to-room ratio of 0.4 (fingerprinting) and 0.8 (multilateration) was chosen to achieve an estimated accuracy of 80% and 50-75% respectively. These numbers are obtained by extrapolating the observations in figure 5.3.

Table 5.2 and 5.3 compare the associated costs with each of the localization models. Setup

Table 5.1: Economical evaluation: parameters

| **Parameter** | **Value** |
|---|---|
| Rooms | 500 |
| Installation time per room | 15 min |
| Fingerprinting per room | 15 min |
| Installation hourly rate | \$30 |
| Fingerprinting hourly rate | \$30 |
| Beacon unit price | \$5 |
| Battery unit price | \$2 |
| Battery lifetime | 1 year |
| beacon-room-factor-knn | 0.4 |
| beacon-room-factor-multi | 0.8 |

costs include the installation of beacons and the collection of the fingerprint database. It is worth noting that training overhead is roughly at par with the costs for the additional beacons needed in the multilateration case. Thus, at first sight, the two approaches seem comparable with a cost advantage for the multilateration model. However, when looking at yearly costs, the picture changes when facing maintenance overhead. Recurring expenses for batteries and labor costs for battery replacement work drive the yearly maintenance costs. In this case, fingerprinting-based localization has an edge due to the smaller hardware footprint as compared to multilateration which yields only half of the maintenance costs.

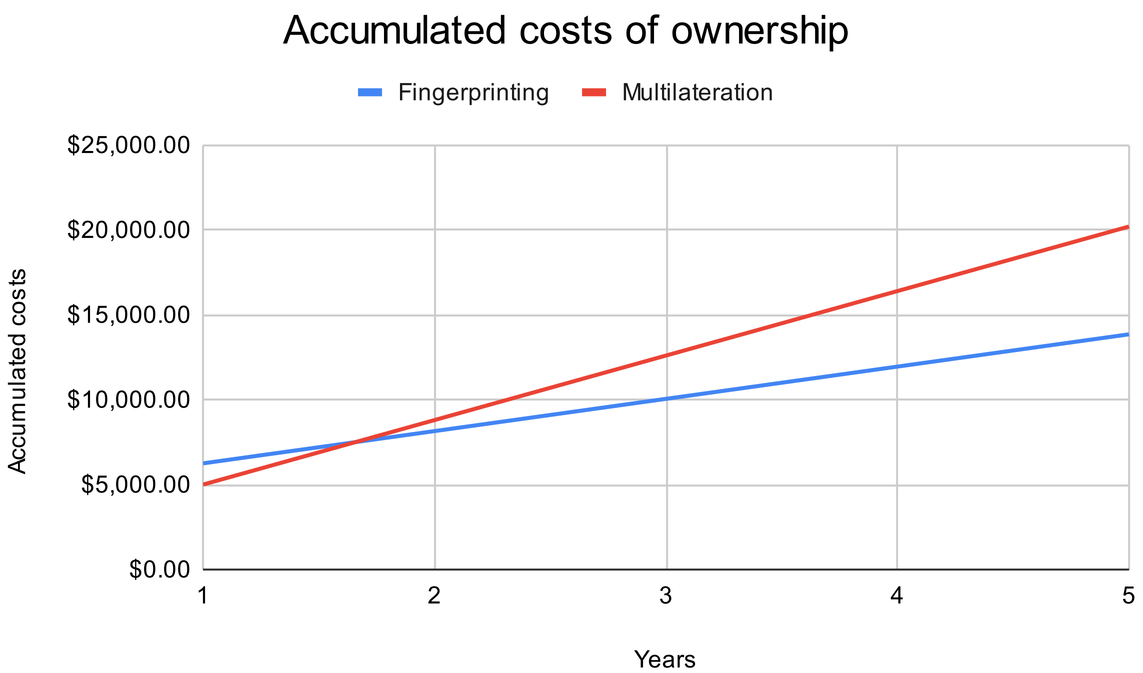


Figure 5.18: Accumulated costs of ownership

Figure 5.18 outlines the accumulated costs of ownership over a period of 5 years. It is

Table 5.2: Economical evaluation: fingerprinting

| **Description** | **Units** | **Price/Unit** | **Total** |
|---|---|---|---|
| BLE Beacons | 200 | $5.00 | $1,000.00 |
| Installation of BLE Beacons | 200 | $7.50 | $1,500.00 |
| kNN Fingerprinting | 500 | $7.50 | $3,750.00 |
| **Setup Costs** | | | **$6,250.00** |
| Coin cell battery | 200 | $2.00 | $400.00 |
| Battery replacement work | 200 | $7.50 | $1,500.00 |
| **Recurring Costs (Yearly)** | | | **$1,900.00** |

Table 5.3: Economical evaluation: multilateration

| **Description** | **Units** | **Price/Unit** | **Total** |
|---|---|---|---|
| BLE Beacons | 400 | $5.00 | $2,000.00 |
| Installation of BLE Beacons | 400 | $7.50 | $3,000.00 |
| kNN Fingerprinting | 0 | $7.50 | $0.00 |
| **Setup Costs** | | | **$5,000.00** |
| Coin cell battery | 400 | $2.00 | $800.00 |
| Battery replacement work | 400 | $7.50 | $3,000.00 |
| **Recurring Costs (Yearly)** | | | **$3,800.00** |

noteworthy that the fingerprinting-based approach recovers the higher setup costs in less than 2 years and has a cost advantage of roughly 45% over a 5-year period. Thus, it can be concluded that under the current assumptions, the fingerprinting-based approach is more effective both in terms of hardware footprint as well costs. This fact can change once the battery life of beacons drastically improves such that the need for replacement can be reduced considerably.

## 5.5 Privacy & Security

In terms of privacy and security concerns, the separation of the gateway and frontend application allows users to read information from the system without being part of the BLE tracking network. The fact that users can voluntarily opt into the tracking network can be seen as a measure to retain user privacy. However, the system might expose the current location of users running the gateway application and who are in the range of a tracked asset. This issue deserves to be addressed and needs further investigation.

Overall, the system is designed with minimal possible intervention and interference with the existing hospital's infrastructure, except for the employee smartphones that are typically handed to the staff. Attaching beacons to medical devices does not render a problem from a regulatory perspective as the beacons operate in the well-secured ISM band. In this regard, the system can be run in isolation and has no effect on security-relevant aspects of the hospital's operation.

## 5.6 Pilot: Emergency Department Zurich

In parallel to the theoretical evaluation, an adaption of the described system has been deployed at the emergency department of the city of Zurich where it is used to track assets of the logistics division. The system was adapted to track assets outdoors through the use of the smartphone's GPS location as a reference point. Within the context of this pilot, the feasibility of the architecture and performance have been successfully validated.

## 5.7 Discussion

This work provides the design and prototypical implementation of a new type of architecture for an indoor asset tracking system. By leveraging smartphones, it has been demonstrated that the system does not rely on stationary gateways as opposed to traditional RTLS while maintaining room-level accuracy.

PMD-Track's key strength lies in its lightweight hardware requirements, which yield multiple benefits such as a simpler setup process, less maintenance overhead, and most importantly, lower costs of ownership, as compared to existing tracking solutions. It has been shown that a fingerprinting-based approach in combination with a specific beacon placement configuration achieves room-level tracking accuracy of over 90% with a minimal number of fixed beacons involved. This allows solution providers to make better cost-accuracy trade offs, depending on the use case requirements. Further, elements in the pre-processing phase such as the number of training samples and the chosen imputation strategy can be optimized as they have a determining impact on the overall model accuracy.

Conversely, the distributed nature of PMD-Track introduces additional challenges. Regardless of the localization method used, the system relies on a critical mass of users participating in the distributed tracking of PMDs. If the number of users doesn't suffice to cover the entire premises on a regular basis, ”blind zones” without coverage may occur where no up-to-date information is available due to staff not passing through. In this case, the isolated deployment of stationary gateways might still be necessary in specific areas.

# Chapter 6

# Final Considerations

This chapter concludes the thesis by summarizing the scope, its goals, and achievements in Section 6.1. Section 6.2 presents the findings and Section 6.3 provides an outlook on future work.

## 6.1 Summary

In this work, a prototypical, gateway-free, and distributed RTLS has been implemented and validated by a pilot study. In the scope of this thesis, two popular localization algorithms have been implemented and evaluated in two test locations. The evaluation has been performed by collecting a base data set in either of the test locations where the accuracy of both localization algorithms has been evaluated under various input configurations. Finally, the accuracy evaluation has been contrasted with a set of economical considerations regarding the deployment and operating costs of the envisioned solution.

For both test locations, it could be shown that the accuracy of the fingerprinting-based approach surpassed the one of the multilateration-based localization algorithms by 15-45% in non-line-of-sight conditions. Further, it has been shown that fingerprinting-based localization achieves roughly the same level of accuracy with half the number of fixed reference beacons. In light of these findings, deployment and running costs of the two alternatives have been analyzed and while the multilateration-based approach has a slight edge in terms of deployment costs, running costs for the fingerprinting-based approach are considerably lower such that after a 5-year period, total costs are roughly 45% lower for the fingerprinting localization. Further, the distributed RTLS architecture and real-time processing has been successfully validated in a pilot study.

## 6.2 Conclusions

Data pre-processing is a main driver for model accuracy. In PMD-Track, a straightforward data collection and imputation strategy has been pursued and acceptable accuracy results have been obtained. Further analysis and improvements of the *(i)* data collection and *(ii)* data imputation process could potentially help to de-noise the data set and thus contribute to higher accuracy scores for both localization algorithms.

The data collection for both test locations has been performed by sequentially fingerprinting each room. While this approach worked well in the case of only a few rooms, it will eventually render a bottleneck for large scale roll-outs. In the latter case, the fingerprint collection could be parallelized and distributed to reduce the overall data collection time.

While the kNN model worked considerably well with a standard configuration of $k = 7$ and euclidean distance metric, different hyper parameters could evaluated which might further improve model accuracy.

Finally, the beacon placement has a significant impact on model accuracy. Repeating the analysis in different test locations could help to uncover beacon placement patterns best suited for the respective localization algorithm and floor plan.

## 6.3 Future Work

This work has focused on the experimental evaluation of two localization algorithms. The data collection and pre-processing mechanism could be further improved by *(i)* parallelizing the fingerprint collection and by *(ii)* de-noising the RSSI data to potentially obtain better accuracy scores. While the kNN model is a popular choice due to its simplicity and good performance, other ML classifiers could be explored and evaluated on the same data set. Another approach is to investigate the potential of reinforcement learning where users could be asked to provide feedback whether the asset was truly located at the predicted location. As a next step, the best performing classifier needs to be implemented and integrated into PMD-Track's architecture, specifically the Location Engine.

Finally, an indoor pilot needs to be conducted in a hospital-like setting to evaluate whether the proposed system is capable of tracking the inventory in real-time.

# Bibliography


[1] Hannah AS Adjei et al. "Developing a Bluetooth based Tracking System for Tracking Devices using Arduino". In: *2020 5th International Conference on Computing, Communication and Security (ICCCS)*. IEEE. 2020, pp. 1–5.

[2] Nadeem Ahmed et al. "A survey of COVID-19 contact tracing apps". In: *IEEE access* 8 (2020), pp. 134577–134601.

[3] Paul Barach et al. "Disruption of Healthcare: Will the COVID Pandemic Worsen non-COVID Outcomes and Disease Outbreaks?" In: *Progress in Pediatric Cardiology* 59 (2020), p. 101254.

[4] Jingxue Bi et al. "An adaptive weighted KNN positioning method based on omnidirectional fingerprint database and twice affinity propagation clustering". In: *Sensors* 18.8 (2018), p. 2502.

[5] Jacob T Biehl, Andreas Girgensohn, and Mitesh Patel. "Achieving accurate room-level indoor location estimation with emerging IoT networks". In: *Proceedings of the 9th International Conference on the Internet of Things*. 2019, pp. 1–8.

[6] Igor Bisio, Andrea Sciarrone, and Sandro Zappatore. "A new asset tracking architecture integrating RFID, Bluetooth Low Energy tags and ad hoc smartphone applications". In: *Pervasive and Mobile Computing* 31 (2016), pp. 79–93.

[7] Igor Bisio et al. "Exploiting context-aware capabilities over the internet of things for industry 4.0 applications". In: *Ieee network* 32.3 (2018), pp. 101–107.

[8] *Bluetooth Low Energy nbsp;: nbsp; Android developers*. URL: https://developer.android.com/guide/topics/connectivity/bluetooth/ble-overview.

[9] Vicente Cantón Paterna et al. "A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering". In: *MDPI Sensors* 17.12 (2017), p. 2927.

[10] JLV Carrera et al. "Discriminative Learning-based Smartphone Indoor Localization. arXiv 2018". In: *arXiv preprint arXiv:1804.03961* ().

[11] *Core Bluetooth*. URL: https://developer.apple.com/documentation/corebluetooth.

[12] Davide Dardari, Pau Closas, and Petar M Djurić. "Indoor Tracking: Theory, Methods, and Technologies". In: *IEEE Transactions on Vehicular Technology* 64.4 (2015), pp. 1263–1278.

[13] Rémi Dekimpe et al. "A battery-less BLE IoT motion detector supplied by 2.45-GHz wireless power transfer". In: *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE. 2018, pp. 68–75.

[14] Sandra Djosic et al. “Fingerprinting-assisted UWB-based localization technique for complex indoor environments”. In: *Expert Systems with Applications* 167 (2021), p. 114188.

[15] *Exposure notifications: Helping fight covid-19.* URL: `https://www.google.com/covid19/exposurenotifications/`.

[16] Xingbin Ge and Zhiyi Qu. “Optimization WIFI indoor positioning KNN algorithm location-based fingerprint”. In: *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. IEEE. 2016, pp. 135–137.

[17] Dongfang Guo et al. “Demo Abstract: Infrastructure-Free Smartphone Indoor Localization Using Room Acoustic Responses”. In: (2021).

[18] Suining He and S-H Gary Chan. “Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons”. In: *IEEE Communications Surveys & Tutorials* 18.1 (2015), pp. 466–490.

[19] *Home.* URL: `https://www.gsmscore.com/model-finder/bluetooth-support/bluetooth-5.1/`.

[20] Ruslan Horblyuk et al. “Out of control: little-used clinical assets are draining healthcare budgets: little-used mobile clinical equipment is a significant expense for healthcare organizations, but cutting costs involves more than inventory reduction”. In: *Healthcare Financial Management* 66.7 (2012), pp. 64–69.

[21] Berthold KP Horn. “Indoor positioning using time of flight with respect to Wi-Fi access points”. In: *URL: http://people. csail. mit. edu/bkph/ftmrtt_errors (visited on 06/10/2019)* (2020).

[22] “IEEE Standard for Information technologyâTelecommunications and information exchange between systems Local and metropolitan area networksâSpecific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. In: *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (2016), pp. 1–3534.

[23] Adafruit Industries. *Adafruit LSM303AGR accelerometer magnetometer - Stemma QT Qwiic.* URL: `https://www.adafruit.com/category/521`.

[24] Carman Ka Man Lee et al. “A bluetooth location-based indoor positioning system for asset tracking in warehouse”. In: *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE. 2019, pp. 1408–1412.

[25] Xinrong Li. “RSS-based location estimation with unknown pathloss model”. In: *IEEE Transactions on Wireless Communications* 5.12 (2006), pp. 3626–3633.

[26] Xiaomei Liang, Xuerong Gou, and Yong Liu. “Fingerprint-based location positoning using improved KNN”. In: *2012 3rd IEEE international conference on network infrastructure and digital content*. IEEE. 2012, pp. 57–61.

[27] *List of UWB-enabled mobile devices.* 2022. URL: `https://en.wikipedia.org/wiki/List_of_UWB-enabled_mobile_devices`.

[28] Liu Liu et al. “Real-time Indoor Positioning Approach using iBeacons and Smartphone Sensors”. In: *MDPI Applied Sciences* 10.6 (2020), p. 2003.

[29] Yang Liu et al. “Indoor localization using smartphone inertial sensors”. In: *2014 11th Workshop on Positioning, Navigation and Communication (WPNC)*. IEEE. 2014, pp. 1–6.

[30] Sumedh Naik et al. “Real Time Asset Tracking using BLE Beacons”. In: *2019 Global Conference for Advancement in Technology (GCAT)*. IEEE. 2019, pp. 1–4.

[31] Adrian Ozer and Eugene John. "Improving the accuracy of bluetooth low energy indoor positioning system using kalman filtering". In: *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2016, pp. 180–185.

[32] Niranjini Rajagopal et al. "Beacon placement for range-based indoor localization". In: *2016 international conference on indoor positioning and indoor navigation (IPIN)*. IEEE. 2016, pp. 1–8.

[33] Bruno Rodrigues et al. "BluePIL: a Bluetooth-based PassIve Localization Method". In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE. 2021, pp. 28–36.

[34] Mirco Rossi et al. "RoomSense: an indoor positioning system for smartphones using active sound probing". In: *Proceedings of the 4th Augmented Human International Conference*. 2013, pp. 89–95.

[35] Priya Roy et al. "JUIndoorLoc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity". In: *Wireless Personal Communications* 106.2 (2019), pp. 739–762.

[36] Blog Post RTLS and Location-Based Services: Five Trends To Keep An Eye On In 2022 Looking back. *Bluetooth technology overview*. URL: `https://www.bluetooth.com/learn-about-bluetooth/tech-overview/`.

[37] Kadir Sabanci et al. "Wifi based indoor localization: application and comparison of machine learning algorithms". In: *2018 XXIIIrd International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory (DIPED)*. IEEE. 2018, pp. 246–251.

[38] Amalesh Sharma, Sourav Bikash Borah, and Aditya C Moses. "Responses to COVID-19: The Role of Governance, Healthcare Infrastructure, and Learning from past Pandemics". In: *J of Business Research* 122 (2021), pp. 597–607.

[39] Meimanat Soleimanifar et al. "Applying received signal strength based methods for indoor positioning and tracking in construction applications". In: *Canadian Journal of Civil Engineering* 41.8 (2014), pp. 703–716.

[40] Petros Spachos and Konstantinos N Plataniotis. "BLE beacons for indoor positioning at an interactive IoT-based smart museum". In: *IEEE Systems Journal* 14.3 (2020), pp. 3483–3493.

[41] Nicolas Spielmann and Simon Tuck Stiller. "WiFi-based Crowd Safety System". In: ().

[42] Thomas Tegou et al. "A low-cost room-level indoor localization system with easy setup for medical applications". In: *2018 11th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE. 2018, pp. 1–7.

[43] Roy Want, Wei Wang, and Stan Chesnutt. "Accurate indoor location for the IoT". In: *Computer* 51.08 (2018), pp. 66–70.

[44] *Wi-Fi location: Ranging with RTT nbsp;: nbsp; Android developers*. URL: `https://developer.android.com/guide/topics/connectivity/wifi-rtt#supported-phones`.

[45] Chenshu Wu, Zheng Yang, and Yunhao Liu. "Smartphones based crowdsourcing for indoor localization". In: *IEEE Transactions on Mobile Computing* 14.2 (2014), pp. 444–457.

[46] Andrew C Yao and John G Carlson. “The Impact of Real-time Data Communication on Inventory Management”. In: *International Journal of Production Economics* 59.1-3 (1999), pp. 213–219.

[47] Chaoshun Zuo et al. “Automatic fingerprinting of vulnerable ble iot devices with static uuids from mobile apps”. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 1469–1483.

# Abbreviations

| | |
|---|---|
| 2D | Two dimensions |
| 3D | Three dimensions |
| AoA | Angle of Arrival |
| AoD | Angle of Departure |
| AP | Access Point |
| API | Application programming interface |
| BLE | Bluetooth low energy |
| CCS4DT | Cloud Correlation Service for Data Tracking |
| CLEAR | Classification of events, activities and relationships |
| cm | Centimetre |
| COVID | Coronavirus disease |
| CPU | Central Processing Unit |
| CRUD | Create, read, update, and delete |
| DB | Database |
| ETL | Extract, transform, load |
| EPC | Electronic Product Code |
| FTM | Fine Time Measurements |
| GATT | Generic Attribute Profile |
| GB | Gigabyte |
| GPS | Global Positioning System |
| ID | Identifier |
| IEEE | Institute of Electrical and Electronics Engineers |
| IMU | Inertial Measurement Unit |
| IoT | Internet of Things |
| IP address | Internet Protocol address |
| JSON | JavaScript Object Notation |
| KF | Kalman Filter |
| KNN | K-Nearest Neighbor |
| LoS | Line of Sight |
| MAC | Media access control |
| ML | Machine Learning |
| MOT | Multi object tracking |
| PMD | Portable medical device |
| RAM | Random access memory |
| RDBS | Relational database system |
| REST | Representational state transfer |

| | |
|---|---|
| RFID | Radio Frequency Identifiers |
| RSS | Received signal strength |
| RTT | Round Trip Time |
| s | Seconds |
| SIG | Special Interest Group |
| TSDB | Time series database |
| ToF | Time of Flight |
| UUID | Universally unique identifier |
| UWB | Ultra-WideBand |
| VLS | Visible Light Communication |

# List of Figures

# List of Tables

# Appendix A

# Contents of the CD

The enclosed CD contains the following content and links: