UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

Strategic Cost Reduction in Indoor Positioning Systems Using Signal Propagation Modeling Techniques

Yuri Freitas Assayag

Manaus - AM
April 2025

Yuri Freitas Assayag

Strategic Cost Reduction in Indoor Positioning Systems Using Signal Propagation Modeling Techniques

Thesis presented to the Post-graduate Program in Informatics of the Institute of Computing of the Federal University of Amazonas in partial fulfillment of requirements for the degree of Doctor in Computer Science.

Advisor

Horácio A. B. Fernandes de Oliveira, D.S.c

UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP

Manaus - AM
April 2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

A844s Assayag, Yuri Freitas
Strategic Cost Reduction in Indoor Positioning Systems Using Signal Propagation Modeling Techniques / Yuri Freitas Assayag. - 2025.
101 f. : il., color. ; 31 cm.

Orientador(a): Horácio Antonio Braga Fernandes de Oliveira.
Tese (doutorado) - Universidade Federal do Amazonas, Programa de Pós-Graduação em Informática, Manaus, 2025.

1. Indoor Positioning. 2. Bluetooth-Low-Energy. 3. Path-loss Model. 4. Fingerprint. 5. Trilateration. I. Oliveira, Horácio Antonio Braga Fernandes de. II. Universidade Federal do Amazonas. Programa de Pós-Graduação em Informática. III. Título

Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

FOLHA DE APROVAÇÃO

"STRATEGIC COST REDUCTION IN INDOOR POSITIONING SYSTEMS USING SIGNAL PROPAGATION MODELING TECHNIQUES"

YURI FREITAS ASSAYAG

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Horácio Antônio Braga Fernandes de Oliveira - **Presidente**

Prof. Dr. Juan Gabriel Colonna - **Membro Interno**

Prof. Dr. Felipe Leite Lobo - **Membro Externo**

Prof. Dr. Leandro Nelinho Balico - **Membro Externo**

Manaus, 04 de abril de 2025.

de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Felipe Leite Lobo**, **Usuário Externo**, em 10/04/2025, às 11:19, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Leandro Nelinho Balico**, **Usuário Externo**, em 10/04/2025, às 16:47, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2534080** e o código CRC **ACD34F9D**.

Avenida General Rodrigo Octávio, 6200 - Bairro Coroado I Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181 / Ramal 1193
CEP 69080-900, Manaus/AM, coordenadorppgi@icomp.ufam.edu.br

I dedicate this work to God, who has given me life and continues to bless me each day with the energy and strength to pursue my goals with courage.

Acknowledgements

First and foremost, I would like to express my gratitude to God for the strength to face and overcome challenges, for my health.

I am profoundly thankful to my family, especially my parents, Rosalvo and Karla, for their unwavering support throughout my academic journey and for the sacrifices they made to help me reach this point. I am also eternally grateful to my grandmother Cleide, my uncles Virle and Jocicleide, and my siblings Matheus and Maria Eduarda, who inspire me to become a better person every day. A special thanks to my aunt Cláudia, who sadly passed away during the course of this doctorate. I will always be grateful for everything she did for me, and this thesis is dedicated to her memory.

I would also like to extend my thanks to the Federal University of Amazonas, particularly the Institute of Computing, for the excellent education and dedicated faculty. A special mention goes to my advisor, Professor Horácio Fernandes, for his guidance, availability, and the invaluable opportunities he provided me.

I am grateful for the crucial support from the Coordination for the Improvement of Higher Education Personnel (CAPES) - Finance Code 001, Amazonas State Research Support Foundation - FAPEAM - through the POSGRAD project, Institute for Innovation, Research, and Scientific and Technological Development of Amazonas (IPDEC), Positivo Technologies, and Samsung Electronics da Amazônia Ltda., through agreement No. 003, signed with ICOMP/UFAM, which made this work possible.

Lastly, I would like to thank all my laboratory colleagues for their invaluable contributions throughout the research process, for their assistance with experiments, and for sharing their experiences. I also want to express my gratitude to all my friends who were part of this significant phase of my life.

Resumo

Sistemas de Posicionamento Interno são usados para estimar a posição de dispositivos móveis em ambientes internos. A impressão digital é a técnica mais utilizada devido à sua maior precisão. No entanto, essa técnica requer uma fase de treinamento trabalhosa que mede o indicador de intensidade do sinal recebido em todos os pontos de referência. Por outro lado, IPSs baseados em modelos usam modelos de propagação de sinal para estimar distâncias a partir do RSSI. Portanto, eles não exigem treinamento caro, mas resultam em erros de posicionamento maiores. Para mitigar esse problema, esta tese explora melhorias na modelagem de propagação de sinal como uma alternativa para reduzir os esforços de coleta de dados com foco na precisão do sistema. Três novas abordagens são propostas. SynTra-IPS é uma abordagem híbrida que gera conjuntos de dados de treinamento sintéticos usando um modelo de propagação logarítmica. Algoritmos de aprendizado de máquina processam esses conjuntos de dados e técnicas de fusão de dados aprimoram as estimativas de posição. O ADAM-IPS aprimora a seleção de nós de ancoragem e aplica modelagem de propagação de sinal com fusão de dados para estimar distâncias, eliminando a necessidade de coleta extensiva de conjuntos de dados. O PSO-MIPS utiliza a otimização de enxame de partículas com modelagem de propagação de sinal para refinar a estimativa de posição sem a necessidade de parâmetros predefinidos ou treinamento prévio. Esses métodos foram testados em ambientes reais de larga escala, demonstrando sua eficácia na redução dos requisitos de coleta de dados, mantendo a precisão de localização ideal em comparação com os sistemas de posicionamento interno existentes.

Palavras-chaves: Sistema de Posicionamento Interno; Bluetooth de Baixa Energia; Modelo de Perda de Caminho; Impressão Digital; Trilateração.

Abstract

Indoor Positioning Systems (IPSs) are used to estimate the position of mobile devices in indoor environments. Fingerprinting is the most used technique because of its higher accuracy. However, this technique requires a labor-intensive training phase that measures the Received Signal Strength Indicator (RSSI) at all Reference Points (RPs) locations. On the other hand, model-based IPSs use signal propagation models to estimate distances from RSSI. Thus, they do not require expensive training but result in higher positioning errors. To mitigate this problem, this thesis explores improvements in signal propagation modeling as an alternative to reduce data collection efforts focusing on system accuracy. Three novel approaches are proposed. SynTra-IPS (Synthetic Training Indoor Positioning System) is a hybrid approach that generates synthetic training datasets using a log-distance propagation model. Machine learning algorithms process these datasets, and data fusion techniques enhance position estimates. ADAM-IPS (Adaptive Model Indoor Positioning System) improves anchor node selection and applies signal propagation modeling with data fusion to estimate distances, eliminating the need for extensive dataset collection. PSO-MIPS (Particle Swarm Optimization - Model-based Indoor Positioning System) uses particle swarm optimization with signal propagation modeling to refine position estimation without requiring predefined parameters or prior training. These methods were tested in large-scale real-world environments, demonstrating their effectiveness in reducing data collection requirements while maintaining optimal localization accuracy compared to existing indoor positioning systems.

Keywords: Indoor Positioning Systems; Bluetooth Low Energy; Path-loss Model; Fingerprint; Trilateration, .

List of Figures

List of Tables

Contents

1 Introduction

In this chapter, we present the motivation behind choosing the topic of this thesis, a brief overview of the study area, the proposed objectives and the main contributions of our work.

1.1 Motivation

Location-based services have undergone remarkable transformations over time, consistently striving to enhance the accuracy with which people and objects can be located. In the past, the Global Positioning System (GPS) revolutionized navigation by enabling accurate outdoor positioning through satellite signals. However, GPS faces inherent limitations in indoor environments, such as inside buildings, where satellite signals are obstructed or weakened, resulting in reduced accuracy (Zheng et al., 2022). This challenge has encourage the study and development of indoor positioning systems, driven by the growing reliance on location-based technologies in our lives and the need for seamless navigation across all environments.

While GPS laid the foundation for outdoor geolocation, indoor positioning systems represent the next evolutionary step, extending navigation capabilities to enclosed spaces such as schools, shopping malls, airports, and large buildings. In corporate settings, these systems can be leveraged to monitor people flow, optimize space utilization, and enhance asset management. For instance, in hospitals, indoor positioning enables the tracking of medical equipment or the monitoring of patient and staff movements. In critical scenarios, such as natural disasters or dangerous environments, these systems can assist robots in navigating efficiently. As a result, indoor positioning has become indispensable for streamlining processes, improving user experiences, and unlocking new business opportunities.

Advancements in technology have given rise to a variety of solutions based on Wi-Fi, Bluetooth, and other emerging technologies. Consequently, numerous indoor positioning systems have been proposed in the literature, each aiming to improve localization accuracy and user experience. Nevertheless, no single system has yet emerged as a universal standard, as each approach comes with its own set of advantages and drawbacks. Factors such as accuracy, cost, and complexity vary significantly depending on the specific application and scenario, highlighting the need for continued innovation and adaptation in this field.

1.2 Background

Initially, indoor localization techniques primarily relied on Wi-Fi networks, which were already widely deployed in many environments. However, with the rapid evolution of Internet of Things (IoT) technology and the growing demand for energy, and cost-efficient solutions, Bluetooth-Low-Energy (BLE) has emerged as a compelling and, in many cases, superior alternative to Wi-Fi for indoor localization applications (Huang et al., 2019). BLE offers a range of advantages that make it an attractive choice for such systems, including significantly lower power consumption, reduced costs, ease of implementation (without requiring complex configurations), and enhanced scalability, enabling the seamless integration of hundreds of devices within a given environment.

Both Wi-Fi and BLE devices utilize Received Signal Strength Indicator (RSSI) as a key metric to measure the signal strength between two devices. RSSI, measured in decibels (dBm), is commonly used to estimate the distance between a transmitter and a receiver, as well as to determine relative positioning in indoor environments (Sadowski and Spachos, 2018). A stronger signal (higher RSSI) typically indicates that the device is closer to the transmitter, while a weaker signal (lower RSSI) suggests greater distance. However, RSSI-based localization is not without its challenges, as it is susceptible to various environmental factors such as obstacles, interference, device orientation, and other dynamic conditions, all of which can complicate accurate positioning in indoor settings (Fang and Chen, 2020).

Despite these challenges, advanced techniques have been developed to mitigate the limitations of RSSI, transforming it into a powerful tool for indoor applications. Among the most prominent approaches, two main categories stand out: fingerprint-based and model-based. These methods have proven effective in addressing the inherent complexities of indoor environments, paving the way for more reliable and accurate localization solutions.

1.2.1 Fingerprint-based IPS

In the fingerprint-based approach, the scenario is divided into distinct reference points (or cells), where the signal strengths (RSSI) from multiple transmitters are systematically collected. This data is then stored in a database, forming a "fingerprint map" that links each reference point to a unique RSSI pattern. This initial stage, known as the calibration or offline phase, is critical for establishing a reliable foundation for subsequent localization (Kim et al., 2018).

Once the fingerprint map is created, user localization takes place during the online or operational phase. In this phase, when a device requires positioning, the RSSI values between the mobile device and the fixed devices (known as anchor nodes) in the environment are measured and compared against the patterns stored in the database from the calibration phase. The device's location is estimated by identifying the closest match between the measured RSSI pattern and the pre-recorded patterns in the database. To enhance the accuracy of this estimation, various algorithms can be employed during this stage.

While this technique delivers high accuracy and excels in characterizing environments under static and controlled conditions, it is not without its challenges. Constructing the training dataset can be a time-consuming and labor-intensive process, particularly in medium to large-scale environments (Ullah et al., 2020). Additionally, the technique is highly sensitive to environmental changes, such as alterations in wall arrangements or the introduction of new obstacles. Such changes can render the reference map obsolete, necessitating the collection of new data and recalibration of the system. Despite these limitations, the fingerprint-based approach remains a powerful method for indoor localization in stable environments.

1.2.2 Model-based IPS

In model-based IPSs, signal propagation models are employed to map the relationship between the RSSI and the distance between the mobile device and the anchor nodes, thereby eliminating the need for the extensive data collection required in fingerprinting methods. Additionally, model-based IPSs offer greater flexibility, as they allow for adjustments to the model based on environmental conditions without the necessity of recalibrating an entire database Ullah et al. (2020). In this approach, the RSSI values measured between a mobile device and fixed nodes are converted into estimated distances. Using these distance estimates, the position of the mobile device can be calculated through algorithms such as Maximum Likelihood Estimation (MLE) and Least Squares (LS), which are both computationally efficient and well-suited for systems with limited resources.

However, a significant challenge in this approach lies in fine-tuning the propagation model to account for the impact of environmental obstacles. This requires accurately defining parameters such as the path loss exponent and the attenuation caused by obstacles, which can be difficult to estimate with accuracy. Despite this limitation, model-based IPSs remain a robust and adaptable solution for indoor positioning, particularly in dynamic environments where flexibility and efficiency are paramount.

1.3 Hypothesis

A method for mobile device localization that utilizes techniques involving multiple parameter values in the signal propagation model, integrated with data fusion and optimization algorithms, will achieve greater efficiency in both system deployment time and localization accuracy, surpassing traditional IPSs as model-based or fingerprint-based.

1.4 Objectives

This thesis aims to develop and test new approaches for locating mobile devices in indoor environments, focusing on using signal propagation models to reduce the time, effort, and cost associated with data collection, while also achieve a level of accuracy that meets the demands of most real-world indoor applications. The specific objectives of this work are described as follows:

- Investigate wireless communication technologies, such as BLE, that can provide reliable RSSI information for mobile device localization.
- Design and implement positioning algorithms based on the log-distance signal propagation model, utilizing the RSSI of BLE devices as input, while considering prior knowledge of the positions of fixed anchor nodes in the environment.
- Propose a novel method for selecting the optimal anchor nodes that enhance the position calculation.
- Analyze and optimize the performance of the proposed approaches, aiming to reduce potential localization errors and enhance the reliability and robustness of the systems.
- Conduct extensive testing and evaluations of the proposed approaches in a real-world closed environment scenario, comparing the achieved accuracy with existing approaches.

1.5 Publications

- Y. Assayag, H. Oliveira, E. Souto, R. Barreto and R. Pazzi, "Indoor Positioning System Using Synthetic Training and Data Fusion." in **IEEE Access**, 2021.

https://doi.org/10.1109/ACCESS.2021.3105188

- Y. Assayag, H. Oliveira, E. Souto, R. Barreto and R. Pazzi, "Adaptive Path Loss Model for BLE Indoor Positioning System," in **IEEE Internet of Things Journal**, 2023.
  https://doi.org/10.1109/JIOT.2023.3253660
- Y. Assayag, H. Oliveira, E. Souto, R. Barreto and R. Pazzi, "A Model-Based BLE Indoor Positioning System Using Particle Swarm Optimization," in **IEEE Sensors Journal**, 2024.
  https://doi.org/10.1109/JSEN.2024.3352535
- Y. Assayag, H. Oliveira, M. Lima, J. Junior, M. Preste, L. Guimaraes and E. Souto, "Indoor environment dataset based on RSSI collected with bluetooth devices," in **Data in Brief**, 2024.
  https://doi.org/10.1016/j.dib.2024.110692.
- Y. Assayag, H. Oliveira, E. Souto, R. Barreto, R. Pazzi and M. Carvalho, "Efficient exploration of indoor localization using genetic algorithm and signal propagation model," in **Computing**, 2025.
  https://doi.org/10.1007/s00607-024-01391-x

1.6 Thesis Overview

The approaches described from the execution of this thesis are described in Chapters 2, 3, and 4.

In the first approach, described in Chapter 2, Syntra-IPS differs from model-based solutions that convert real RSSIs into estimated distances, our solution converts real distances from the map into synthetic RSSIs, which allows it to take into consideration the walls of the scenario, among other things. When compared to fingerprint-based IPSs, most solutions either try to reduce the training using sensors, data analysis, and crowdsourcing or try to reduce the dataset to improve performance. Our solution eliminates the real-world training part of the fingerprint technique and replaces it with synthetic datasets. In addition, this approach is the first to explore the log-distance signal propagation model to generate several synthetic fingerprint datasets and apply data fusion from several datasets to provide an improved position estimation in indoor localization.

In ADAM-IPS, described in Chapter 3, we do not use a fixed number of anchor nodes and we choose the anchor nodes' combination that favors the position computation through the Least square algorithm. Results from performance evaluations show that this approach outperforms the works that use all anchor nodes and fixed parameters for the propagation model. This approach innovates by exploring the log-distance signal propagation model to map different distances according to the variation in model parameter values and use data fusion from different estimated positions to improve the final positioning estimate.

Finally, in the last approach described in Chapter 4, we eliminate the need to collect data in the environment. Instead, in MIPS-PSO, we only require prior knowledge of the positions where anchor nodes are fixed. Additionally, unlike model-based solutions with fixed parameters, we use the signal propagation model with different parameter values, allowing us to model the signal in different regions of the scenario, with different particles. Therefore, we propose a new method based on PSO that uses the signal propagation model to get the best particle closest to the real mobile device position. In summary, we innovate by exploring the use of the signal propagation model together with PSO to estimate the mobile device position with higher accuracy.

Our main contributions are summarized as follows:

1. Our solution uses several synthetic datasets to characterize the signal in different regions of the scenario, without the need for complex real-world data gathering from the environment.

2. We propose a new data fusion strategy that combines the positioning estimates by fingerprint-KNN using all synthetic datasets, into a single, more accurate position that outperforms approaches that use just a single synthetic dataset.

3. We propose another data fusion technique to combine different estimated positions

(based on different model parameters) into a single, more accurate position that outperforms approaches that use only fixed parameters in a log-distance model.

4. We propose a new algorithm to choose the best set of anchor nodes that benefit the position computation based on the collinearity of the anchors' coordinates.

5. We innovate by exploring the use of the signal propagation model together with PSO to estimate the mobile device position with higher accuracy.

6. Through a large number of real-world experiments, we verify the efficiency and effectiveness of the proposed solutions. Our results show that the approaches can achieve a competitive localization accuracy compared to state-of-the-art IPSs such as model-based IPSs, IPSs using a single synthetic dataset, and even traditional fingerprint-based IPSs with real training.

2 Indoor Positioning System using Synthetic Training and Data Fusion

In this chapter, we propose SynTra-IPS (Synthetic Training Indoor Positioning System), a hybrid approach between a fingerprint and a model-based IPS that uses synthetic, simulated datasets combined with data fusion techniques to eliminate the cost of fingerprint collection. In our solution, we use the scenario map, with the positions of known anchor nodes and the log-distance signal propagation model, to generate several synthetic, model-based, fingerprint training datasets. In the online phase of our solution, the positions estimated by the several synthetic datasets using K-Nearest Neighbors (KNN) are combined using data fusion techniques into a single, more accurate position. We evaluated the performance of our Syntra solution in a real-world, large-scale environment using mobile devices with technology, and we compared our solution to classic approaches from the literature. Our results show that Syntra can locate mobile devices with an average error of only 2.36 m while requiring no training in the real world environment.

2.1 Introduction

Positioning systems can be defined as the process of finding the position of a target in outdoor or indoor environments (Youssef and Agrawala, 2005). Today, one of the most known positioning systems is the Global Navigation Satellite System (GNSS), which includes the Global Positioning System (GPS), that is able to locate devices in outdoor environments, where there is a line-of-sight among the device and the satellites. On the other hand, Indoor Positioning Systems (IPSs) focus on locating mobile devices in indoor environments, where GNSS can not provide a good accuracy (Zhang et al., 2017). Currently, there is a lot of research to propose new methods and technologies that increase the accuracy of the IPSs, motivated by the high complexity of indoor environments (Cheng et al., 2020; Sadowski and Spachos, 2018).

The main technology used in IPSs is based on local radio signals, and the position can be estimated using the Time of Arrival (ToA) (He et al., 2012), Time Difference of Arrival (TDoA) (Schreiber and Bajer, 2016), Angle of Arrival (AoA) (Fascista et al., 2017), and the Received Signal Strength Indicator (RSSI) (Sadowski and Spachos, 2018). The RSSI being the most frequently used due to its high availability since most devices with wireless communication, such as or Wi-Fi, already comes with this feature. Wi-Fi is a wireless communication technology widely available in different places such as malls and airports, which means no additional hardware and deployment requirements for indoor localization. On the other hand, BLE has also been widely used in indoor localization due to its low power consumption, allowing it to be used by energy-constrained devices such as smartwatches while also being available in most smartphones.

Most IPSs can be classified into model-based and fingerprint-based. Model-based IPSs estimate the positions based on the distance between the mobile device and the Anchor Nodes (ANs), which are fixed devices with known positions (Assayag et al., 2020). For this, the RSSI values are converted into distances using a path loss signal propagation model, the most known being the Log-Distance model. Then, the position computation is done using, for instance, the least-squares technique. However, due to the high complexity of indoor environments and the high RSSI variation, this conversion is not always done realistically (Li et al., 2018).

Fingerprint-based IPSs (Guo et al., 2018) are known to be more accurate and popular. This method is divided into two phases: offline and online. In the offline phase, also known as training, several evenly spaced Reference Points (RPs) are distributed in the environment. For each RP, several RSSI values between a mobile device and the anchor nodes need to be collected. They are then stored in a dataset along with the position where the signals were collected. In the online phase, the mobile device that we want to locate sends an advertising packet that is received by the anchor nodes that estimate the RSSI and send them to a server. The server compares these RSSI measurements to the ones in the dataset to estimate the mobile device position. This can be done using machine learning techniques such as the KNN (Moghtadaiee et al., 2019; Wang et al., 2020). Although fingerprint-based IPSs are more accurate, the fingerprint collection on the offline phase is very time-consuming and laborious. Moreover, the fingerprint dataset is unable to adapt to future changes in the environment, requiring a new fingerprint collection, which makes their implementation unfeasible in large-scale locations. Thus, the main challenge of this method is how to reduce the need for a real fingerprint collection.

In this work, we propose SynTra-IPS (Synthetic Training Indoor Positioning System), a hybrid approach between a fingerprint and a model-based IPS. In the offline phase of our solution, we use a log-distance propagation model with different parameters to generate several synthetic training datasets that reflect the RSSIs in the different RPs of the environment under different propagation conditions. In the online phase, we execute the KNN in all synthetic datasets to locate a signal from the mobile node. Then, we use data fusion techniques to combine all of the estimated positions into a single, more accurate position. To evaluate the performance of our solution, we implemented a real-world, large-scale testbed using mobile devices with BLE technology. Our results show that Syntra can locate mobile devices with a average error of only $2.36\ m$. As we will show, this is a better accuracy when compared to model-based solutions, getting close to a complete fingerprint-based solution, but without the need for any real-world, laborious training.

Our main contributions are summarized as follows:

1. Our solution uses several synthetic datasets to characterize the signal in different regions of the scenario, without the need for complex real-world data gathering from the environment.

2. We propose a new data fusion strategy that combines the positioning estimates by KNN using all synthetic datasets, into a single, more accurate position that outperforms approaches that use just a single synthetic dataset.

3. Through a large number of real-world experiments, we verify the efficiency and effectiveness of the proposed solution. Our results show that the system can achieve competitive localization accuracy compared to state-of-the-art IPSs such as model-based IPSs, IPSs using a single synthetic dataset, and even traditional fingerprint-based IPSs with real training.

The rest of the chapter is organized as follows. In the next section, we show our related work. Section 2.3 presents Syntra, our proposed IPS solution. Section 2.4 shows our real-world testbed and experimentation methodology. In Section 2.5, we show and discuss the results of the performance evaluation. In Section 2.6, we discuss the applicability and limitations of our solution. Finally, Section 2.7 presents our conclusions and future work.

2.2 Related Work

To estimate the position of a mobile device in an indoor environment is a complex task since the electromagnetic signal sent by devices does not have a deterministic behavior (Jung et al., 2011). In order to make IPS more robust and accurate, many techniques and algorithms are proposed in the literature. Most solutions can be classified into model-based and fingerprint-based.

In model-based IPSs, the signals measured between the mobile device and at least three anchor nodes are used to estimate the distances. For this, signal propagation models such as the Log-distance (Assayag et al., 2020) and Two-ray Ground Reflection Model (TGRM) (Mohammed El Amine and Ouslim, 2015) are used. In Lazaro et al. (2010), the authors use the frequency diversity in the wireless channel to reduce the multipath effect on the distance estimation. Similarly, in Fang and Chen (2020), the authors propose the Optimal Multi-channel Trilateration Positioning Algorithm (OMCT) to find the global optimal parameter values and prevent the algorithm from falling into local optimum. Thus, the focus is reducing the multipath effects to increase system accuracy. In Li et al. (2016b), it is proposed to use assistant nodes and an adaptive Kalman filter to assist and improve the distance estimation. However, the experiments did not consider the complexity of the environment, such as walls and other obstacles, which can result in lower accuracies.

In Shi et al. (2020), the authors propose a model-based IPS that uses the k-means algorithm to separate the RSSI into three groups, where each group receives different filters that allow the propagation model to make more stable distance estimations. Sadowski and Spachos (2018) compare the performance of a distance-based IPS using four dominant technologies: Wi-Fi, BLE, LoRaWAN, and ZigBee. The evaluation metrics were system accuracy and energy consumption, and the results show that Wi-Fi and BLE have advantages over other technologies. The model-based IPS mentioned above uses a fixed path loss exponent to characterize the signal behavior in all regions of the scenario, which is not a suitable solution for large-scale scenarios. In Assayag et al. (2020), the authors have performed a smaller training to find different path loss exponents that characterize each region of the scenario. The results show that using dynamic model parameters decreases the positioning error.

However, despite efforts to improve accuracy, indoor environments are complex, making it difficult to estimate distances only by analyzing the RSSI. As result, several proposed IPSs are fingerprint-based. Fingerprint-based IPS can use several machine learning algorithms to estimate the mobile device position, such as decision trees, random forest, KNN, and deep learning. In (Praveen Kumar et al., 2019) the authors describe the main machine learning algorithms that can be used for localization.

A known work that uses this approach is the RADAR (Bahl and Padmanabhan, 2000), which combined empirical measurements in the proper environment with a signal propagation model to estimate the target location. Similarly, in Youssef and Agrawala (2005), the reduction of the empirical data needed by RADAR motivated the solution. The authors use clustering techniques to reduce computational requirements. Torres et al. (2016) proposed a fingerprint-based IPS for home monitoring. Their solution showed that it is possible to get a precise positioning at the room level with no extra access point, an accessible solution for home monitoring.

Unlike the works that use the conventional fingerprint, crowdsourcing-based approaches use the user's movements to generate the radio map and reduce the effort to implement the system. In Chen et al. (2016), the authors presented a location algorithm that uses a few RSSI's measured by users in real-time to update the dataset with no complete training. Similarly, Niu et al. (2015) developed a crowdsourcing-based IPS, called WicLoc, which builds the fingerprint dataset by recording user movements, as well the RSSI, achieving room-level location accuracy. However, these systems require many sensors such as an accelerometer and gyroscope to get users' movement. Although our work is not based on crowdsourcing, our solution can be used without effort to generate immediate results through the synthetic dataset, and crowdsourcing can be used to improve the result based on the real users' data.

Other solutions use a virtual dataset generated by mathematical models to reduce the training effort. In Maher and Malaney (2009), it is proposed a method to create the dataset in real-time using an optimized ray-tracing algorithm. Similarly, the authors in Kubota et al. (2013) proposed a new method for interpolating the RSSI using a path loss model containing wall attenuation. However, these methods require the material type of the walls, which is hard to get. In Nowicki and Wietzykowski (2017), the authors use a deep neural network to reduce the radio map generation workload by learning the data distribution. Similarly, Kim et al. (2018) propose a new architecture to reduce the dimension of the resource space and thus reconstruct the radio map using a deep neural network. However, training a neural network requires a lot of labeled, trained data.

Ali et al. (2017) explores the floor plan and wall map of the environment to assistant the signal propagation model and generate the simulated training base. The experimental results show that by using the floor plan information and environmental parameters, it is possible to achieve significant positioning accuracy. In Moghtadaiee et al. (2019), it is proposed a method that requires only a few RPs to reconstruct a denser training dataset. The method uses a signal propagation model based on zone and interpolation to generate the RSSI. In Jung et al. (2011), the proposed solution requires little training to learn the model parameters and then generates extra RSSI values in new, virtual RPs. With the common goal of reducing training workload, in Qiang et al. (2019), the authors introduce the Hierarchical Positioning Algorithm (HPA). This algorithm

Table 1 – Comparison of different indoor positioning systems.

Solution	Type	Training	Obstacle Count	Data Fusion
(Kubota et al., 2013)	fingerprint model-based	synthetic	no	no
(Qiang et al., 2019)	fingerprint-based	synthetic	yes	no
(Li et al., 2016b)	fingerprint model-based	real	no	no
(Bahl and Padmanabhan, 2000)	fingerprint-based	real	no	no
(Niu et al., 2015)	crowdsourcing	real	no	no
(Ali et al., 2017)	fingerprint-based	synthetic	yes	no
(Assayag et al., 2020)	model-based	real	no	no
SynTra	fingerprint model-based	synthetic	yes	yes

creates several sub-dataset with different densities in virtual RPs. However, the author uses only a sufficiently small number of fingerprints, with the same path loss exponent in all RPs.

In Table 1, we show the comparison between the main works mentioned in this section. The positioning error of the mentioned works depend on the way the experiment was carried out (real or simulation), as well as the size of the scenario and the algorithms used to estimate the positioning.

2.2.1 Discussion

Our proposed approach differs from all of the above solutions. First, differently from model-based solutions that convert real RSSIs into estimated distances, our solution converts real distances from the map into synthetic RSSIs, which allows it to take into consideration the walls of the scenario, among other things. When compared to fingerprint-based IPSs, most solutions either try to reduce the training using sensors, data analysis, and crowdsourcing or try to reduce the dataset to improve performance. Our solution completely eliminates the real-world training part of the fingerprint technique and replaces it with synthetic datasets. In particular, in the more directly related works Qiang et al. (2019) and Ali et al. (2017), the authors present calibration-free positioning techniques, which exploit the floor plan/wall map of the environment for the construction of RSSI maps, calculating the path loss of the signals using a signal propagation model. However, in this case, the authors generate only a single synthetic dataset to represent the signal behavior in the environment, with the same path loss exponent in all RPs. In our work, we use several synthetic datasets combined using the proposed data fusion techniques to improve the accuracy.

To the best of our knowledge, no existing work considered exploiting the log-distance signal propagation model to generate several synthetic fingerprint datasets. In addition, this article is the first to apply data fusion from several datasets to provide an improved position estimation in indoor localization. The details of our proposed solution are described in the next section.

2.3 SynTra-IPS Architecture

In this section, we present our proposed SynTra-IPS. Like most IPS solutions, Syntra is composed of two phases: offline and online. In the next sections, we present the details of both phases.

2.3.1 Phase 1: Datasets Construction

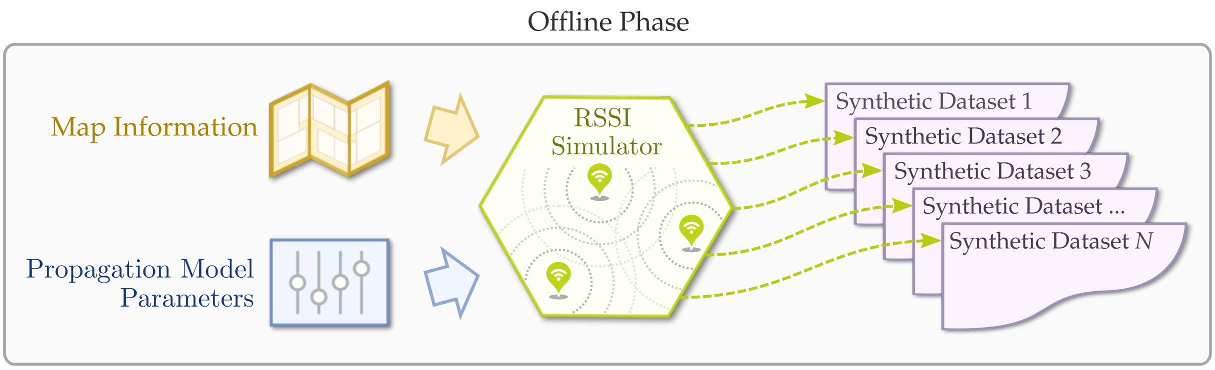

Figure 1 – Offline phase of our Syntra architecture: map information, as well as a set of propagation model parameters, are used as an input to a log-distance-based RSSI simulator that outputs a number of synthetic training datasets.

Figure 1 shows an overview of the offline phase. In this phase, we use the map information and an RSSI simulator to obtain several synthetic training datasets generated by a log-distance model with different propagation parameters. Thus, some datasets will eventually be better to characterize the scenario than others.

In this phase, we assume that an area A contains a set of n ANs with previously known positions, and then we measure the signal strength at m RPs for their neighboring ANs. The RPs are evenly separated, and their positions $(X_i,\ Y_i)$, $i\ =\ (1, 2, 3, ..., m)$ are also known. Thus, for each RP, we have vectors of received signal strengths defined as: RP_i=$(RSSI_1, RSSI_2, RSSI_3, ..., RSSI_m, Label_i, X_i, Y_i)$, where RSSI$_m$ is the received signal strength from the n^{th} AN, Label$_i$ is the RP$_i$ identification, and (X_i, Y_i) is the RP position.

A fingerprint dataset is composed of several measurements of signals at the RPs, and we associate the signals with their real locations. Usually, this dataset is created based on a real-world training step to collect the signals at each RP. However, as mentioned, this is an intensive laborious step, especially in medium to large-scale scenarios, since it requires several days to collect all of the data. Also, there is a need to re-create a new dataset when some scenario characteristics change.

Thus, to eliminate the cost of collecting fingerprints, we propose an RSSI simulator to create synthetic fingerprint datasets based on map information and virtual reference points that match the distribution of real RPs. Our goal is to get the information of the scenario through the floor plan of the building and, thus, reduce the hard step to create the signal map. The RSSI simulator was developed by our research group. As input, it requires the real ANs' positions, the location of the RPs, and the location and dimensions of the rooms' walls. As an output, our RSSI simulator generates a set of synthetic datasets using different parameters for the signal propagation model, in our case, the log-distance (Cantón Paterna et al., 2017). These signals, for each RP, can be computed as:

$$RSSI_n = PL_{d0} - 10\eta \log\left(\frac{d}{d_0}\right) - \sum_i L_i + X_\sigma \tag{2.1}$$

where $RSSI_n$ is signal strength received from the n^{th} AN, d is the distance between the RP and AN$_n$, PL_{d0} is the RSSI value measured at distance d_0 (usually $1\ m$), η is the path loss exponent, i.e., a signal loss rate related to the environment, $\sum_i L_i$, is the attenuation constants in dB for the quantity of walls between the RP and AN_n and, finally, X_σ is a zero-mean Gaussian random variable that models the RSSI variation (Huang et al., 2019). When establishing the parameter values in Equation 2.1, it is possible to get several synthetic signal values for all of the RPs, and then create a single, synthetic training dataset.

However, it is known that signal propagation in indoor environments is subject to several challenges since obstacles can cause high signal variation. Thus, different areas of the scenario may have different parameters in the propagation model that best characterizes the signal's behavior. Considering that we have four main parameters in the log-distance model (PL_{d0}, η, L_i, and X_σ), we can establish different values for each parameter and create T different training datasets with all values combinations that can be tuned to make it fit nearly any regions of the environment. For instance, combining parameters values of PL_{d0} = {-55, -60}, η = {2.5, 3.0, 4.0}, L_i = {2, 3}, and X_σ = {1,2,3}, it is possible to get 36 different synthetic datasets. Some of them will perform better in different areas of the scenario.

Therefore, the result of the offline phase of our SynTra-IPS is a set of synthetic datasets with different parameter values for the propagation model. In the next section, we will show how to combine the results of these synthetic datasets to estimate the users' positions.

2.3.2 Phase 2: Estimating Positions

In the online phase, the RSSI values of a mobile device are used to estimate its position. Figure 2 shows an overview of the online phase of our SynTra-IPS. In this phase, the positions estimated by the several synthetic datasets, using KNN, are combined using data fusion techniques to form a single, more accurate position.

The online phase starts when a mobile device sends a packet. This packet will be received by several ANs that will be able to estimate the RSSIs. These RSSIs are sent to a central location server where the synthetic datasets, computed in the offline phase, are stored.

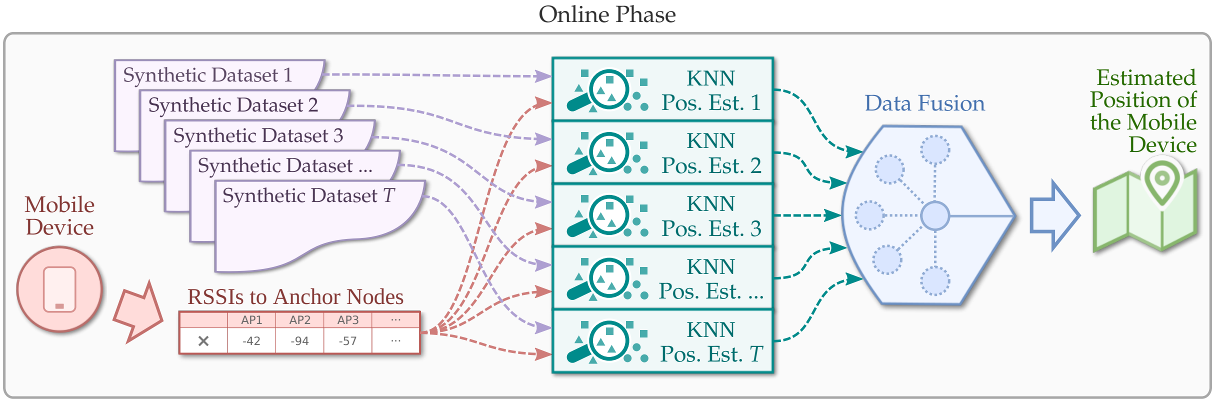

Figure 2 – Online phase of our Syntra architecture: positions estimated by several synthetic datasets, using KNN, are combined using data fusion techniques to form a single, more accurate position.

In the next step, for each synthetic dataset, we find the synthetic sample that best matches the real-world RSSIs samples from the mobile device. For this, several machine learning techniques can be used. In Syntra, we used the KNN algorithm, one of the most popular techniques used in fingerprint-based IPSs. KNN uses the Euclidian distance as a similarity measure to find the dataset sample that is most similar to the real-world RSSIs. Thus, the position estimation of the mobile device, using that specific synthetic dataset, is the same position as the RP from that closest sample.

After executing the KNN for each one of the T synthetic datasets, we will have T possibly different position estimations, each one with its own accuracy, depending on how close the propagation model parameters of the synthetic dataset matches the characteristics of the real-world area the mobile device is located.

Finally, the last step of the online phase is how to combine all of these T position estimations into a single, more accurate one. For this, we use data fusion techniques. Data fusion allows us to combine data from several sources in such a way that the accuracy of the resulting estimation is higher than any of the individual sources. For our Syntra solution, we proposed and evaluated the performance of four data fusion techniques, which will be explained in the next paragraphs.

2.3.2.1 SynTra Voting

In the first data fusion technique, we used a simple majority voting mechanism to determine the best position. Thus, we consider the final position to be the one that was most often chosen among the T predictions. Here, voting is done using the point identification label. We will refer to this variation of our solution as Syntra Voting.

2.3.2.2 SynTra Dist

In the second data fusion technique, called Syntra Dist, we use the euclidian distances between the matched sample and the k-nearest samples as a measurement of accuracy. Thus, for each of the T synthetic datasets, instead of having only the estimated position, we will also have this local distance information for each k-nearest samples to indicate how accurate the estimated position is.

Thus, given a position estimation and the computed distance for each one of the T synthetic datasets, we will choose the position estimation from the dataset with the lowest global distance. This approach considers that if the distance value is low, we will have more chance of choosing a synthetic dataset in which the propagation parameters more closely resembles that of the real-world region where the mobile device is located.

2.3.2.3 Syntra Avg

In the third data fusion technique, called Syntra Avg, we simply get the average position (X, Y) among all of the T position estimations, as shown in Equation 2.2:

$$(X, Y) = \frac{\sum_{i=0}^{T} (X_i, Y_i)}{T} \tag{2.2}$$

where (X, Y) is the final estimated position, (X_i, Y_i) is the position estimated using the i^{th} synthetic dataset, and T is the number of synthetic datasets. This is a simple approach that considers that, on average, the several position estimations from the T synthetic datasets are in nearby regions. However, this approach is sensitive to outliers, i.e., estimated positions with higher errors due to the unrealistic propagation parameters from their synthetic datasets.

2.3.2.4 Syntra WAvg

Finally, the last data fusion approach, called Syntra WAvg, is a combination of Syntra Dist and Syntra Avg. It tries to solve the outliers problem of Syntra Avg using the distance metric, used in Syntra Dist, as weights.

First, in order to invert the distances so the higher the better, we need to compute the sum of the weights as follows:

$$sumDist = \sum_{i=0}^{T}(maxDist - dist_i) \tag{2.3}$$

where $maxDist$ is the maximum distance identified among all of the T predictions, and $dist_i$ is the distance value among the k neighbors in the i^{th} synthetic dataset. Thus, the final position can be computed as follows:

$$(X,Y) = \sum_{i=0}^{T}(\frac{maxDist - dist_i}{sumDist}) * (X_i, Y_i) \tag{2.4}$$

Therefore, the final position is computed using the weighted average positions from all of the T estimated positions, prioritizing the ones with shorter distances, and reducing the outliers influence.

2.4 Experimental Testbed

This section presents our experimentation methodology and real-world testbed. The results of the performance evaluation will be discussed in Section 2.5.

2.4.1 System Environment

To evaluate the performance of Syntra, we conducted an experiment in a real, large-scale environment with an area of 720 m^2 with 15 anchor nodes distributed throughout the area. The test scenario consists of 15 spaces (11 rooms plus 3 halls), as shown in Figure 3, in which each space is covered by at least one anchor node. The anchor nodes are fixed on the ceiling in locations where it was somewhat convenient to connect them to the power supply.

Figure 3 – Real-world experimentation testbed: 11 rooms, 3 halls, and 15 anchor nodes. Gray dots represent the 150 reference points.

Even though our solution does not require any real-world training, we still need to define reference points for the generation of the synthetic datasets. Thus, we separated the environment into 150 different reference points, evenly spaced 2 m apart from each other. Finally, combined with the floor plan information, shown in Figure 3, we have all of the required information to generate the synthetic datasets. We can then apply the signal propagation model described in Equation 2.1 to simulate RSSI values at all RPs.

2.4.2 Synthetic dataset Parameters

Indoor environments are complex structures hard to be modeled by a single signal propagation model since different areas have different signal characteristics caused by the diversity in layouts and obstacles that cause multipath and reflections (Cantón Paterna et al., 2017). Even during the day, these signals can vary due to crowd mobility. The log-distance propagation model has parameters that require calibration to generate simulated signals that are mostly similar to real-world signal behavior.

To represent the signal in the different areas, it would need several parameter values for the log-distance model, but performing the calibration of the parameters is costly, and the effort to do so would be equivalent to performing a real collection in all RPs. Therefore, we use a range of values for the parameters, with common values to be found in IPS (Bullmann et al., 2020; Röbesaat et al., 2017; Sadowski and Spachos, 2018; Shi et al., 2020). In this case, we would avoid the effort of carrying out an extensive experiment to calibrate those parameters.

Instead of creating just one synthetic dataset with fixed, averaged parameters to model the whole scenario, our proposed Syntra generates several synthetic datasets with different signal parameters in such a way that eventually one of the datasets will be better than the others to represent a specific region. Thus, given the possible values that each of the parameters of the log-distance model can assume, the combination of these parameters generates several synthetic datasets. Table 2 shows several possible values for these parameters and the resulting number of possible combinations, which is the total number of synthetic datasets.

Table 2 – Combination of synthetic datasets that can be generated by different parameters values for the log-distance model.

Model Parameters				**Possible Combinations**
PL_{d0}	η	L_i	X_σ	
55,60	3.5, 4.0, 5.5	2,3	1,2,3	36
50,55,60	3.5, 4.0, 4.5	2,3	1,2,3	52
50,55,60	3.5, 4.0, 4.5	2,3,4	1,2,3	81
50,55,60	3.5, 4.0, 4.5	2,3,4,5	1,2,3	108
50,55,60,65	3.5, 4.0, 4.5	2,3,4,5	1,2,3	240
50,55,60,65	3.5, 4.0, 4.5	2,3,4,5,6	1,2,3,4	400

In 2, similar to values of Equation 2.1, PL_{d0} represents the possible values for the RSSI at $1\ m$, η represents the values for the path loss exponent, L_i, the values for wall losses, and X_σ, the RSSI variations. For instance, in the first row, by combining all of the possible values for these parameters, it is possible to generate 36 different synthetic datasets. One issue with this combination is the rapid increase of the number of datasets, which results in a higher processing cost on the online phase. Using the combinations in the last row, for instance, would result in 400 different synthetic datasets.

Figure 4 shows a simulation using just one dataset generated by the following

Figure 4 – Signal characterization of the environment using the signal propagation model as executed by the RSSI simulator.

parameter values: $PL_{d0} = -60$, $\eta = 4.5$, $L_i = 4\ dBm$, and $X_\sigma = 1$. In this figure, we can see how a single synthetic dataset represents the signal behavior in the scenario.

2.4.3 Experimental Methodology

To validate our proposed solution, we performed a real, laborious RSSI collection at the same reference points described in the previous section and depicted in Figure 3. During the experiments, the anchor nodes received BLE advertising packets sent by beacons at a 1 Hz rate. Beacon nodes are mobile devices that we will estimate the positions in the online phase and they operate with a single, small, and long-lasting battery. For the experiment, we used 11 different beacons to diversify the RSSI behavior.

Thus, at each RP, the signal values among beacons and anchor nodes are estimated and sent to a central server that then stores the data in a real-world fingerprint dataset. During the experiments, the highest communication range observed between beacons and ANs was 25 m, even though at this distance, most packets are not received.

After the training process, the fingerprint dataset had 15.000 samples (signal measurements) from different beacons at 150 RPs. Again, it is important to emphasize that this is an exhaustive process and it is unnecessary for our proposed solution, being performed only for evaluation purposes to be compared to real-world data. Figure 5 depicts the average RSSI values from the real signal propagation in our test environment.

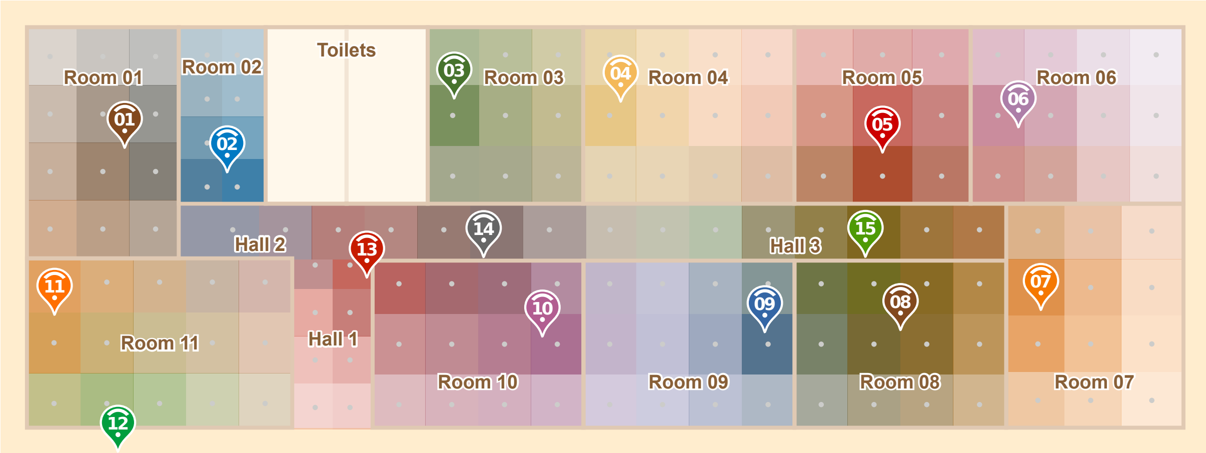

Figure 5 – Signal characterization of the scenario based on the measurements made empirically.

2.5 Performance Evaluation

We evaluated the performance of Syntra in three different aspects. First, we analyzed the impact of the number of dataset combinations on the positioning error. Second, we evaluated the performance of the data fusion techniques. Finally, we compared the performance of our Syntra solution to traditional approaches found in the literature. In all of the experiments, we used a fixed value of 10 for the k parameter of KNN since it had the best results even though the difference from other k values was not significant.

2.5.1 Datasets Combinations

A key aspect of our proposed Syntra solution is the number of synthetic datasets generated by combining the propagation model parameters. To evaluate the impact of the number of datasets on the system performance, we executed our solution using the different combinations of parameter values specified previously in Table 2. Thus, we executed Syntra using small combinations composed of only 36 datasets up to larger combinations of 400 datasets.

Table 3 shows the average positioning error obtained when using these different datasets combinations for all of the data fusion techniques (Voting, Dist, Avg, and WAvg) as well as the result of the best, single dataset. The first thing we can notice when focusing on the last column of the table is that the WAvg data fusion technique resulted

Table 3 – Impact of the number of datasets on the average positioning error for the different data fusion approaches, highlighting the best result.

Number of Datasets	Best Dataset	Data Fusion Technique			
		Voting	*Dist*	*Avg*	*WAvg*
36	2.84 m	3.04 m	2.85 m	2.63 m	2.53 m
52	2.84 m	2.94 m	2.84 m	2.43 m	2.38 m
81	**2.84 m**	**2.93 m**	**2.85 m**	**2.42 m**	**2.36 m**
108	2.84 m	3.01 m	2.90 m	2.44 m	2.37 m
240	2.84 m	3.18 m	2.85 m	2.54 m	2.41 m
400	2.84 m	3.10 m	2.89 m	2.51 m	2.41 m

in the smallest average error and that it was effective in reducing the error from $2.84\ m$ (without data fusion) to $2.36\ m$. Also, we can see that the combination with 81 datasets (highlighted row) resulted in the best performance. Finally, for any combination with more than 36 datasets, the positioning error does not change significantly, ranging from $2.36\ m$ to $2.41\ m$, actually increasing slightly for a higher number of datasets (e.g., 400 datasets resulted in $2.41\ m$).

To better understand the behavior of the positioning error for each of the individual datasets without using data fusion, Figure 6 shows the error resulted from each of the 81 synthetic datasets. Each bar corresponds to a specific dataset, i.e., a specific combination of parameters for the propagation model. As we can see, the error obtained by the individual datasets can vary a lot, depending on how well the propagation model parameters represent the real-world scenario. For example, some bases have an error of almost $5\ m$, while others have the smallest error of $2.84\ m$. However, as we will see in the next section, we can reduce even more this error by using data fusion

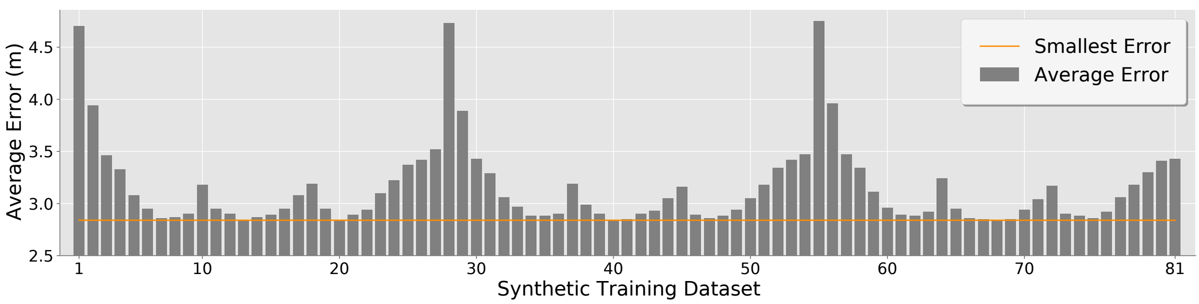

Figure 6 – Average positioning error for each of the 81 individual, synthetic datasets; each bar corresponds to a specific dataset, i.e., a specific combination of parameters for the propagation model. The orange line highligths the smallest error among the datasets.

2.5.2 Data Fusion Technique

Another key aspect of our proposed Syntra solution is to combine the positions estimated by the several synthetic datasets into a single, more precise position. For this, we have proposed four different data fusion techniques: Voting, Dist, Avg, and WAvg. In this section, we evaluate their performance. For this, we used the combination of the 81 synthetic datasets highlighted previously in Table 3 with propagation model parameters detailed in Table 2.

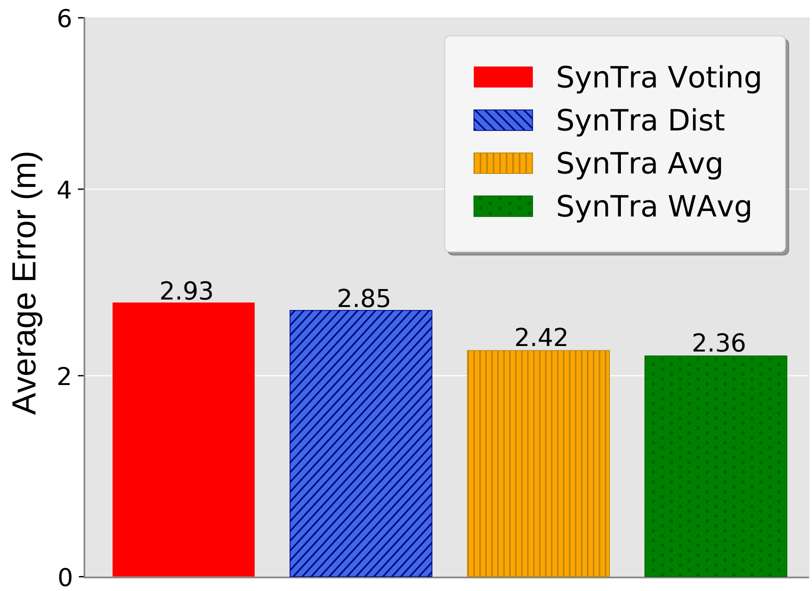

Figure 7 – Comparison of the average positioning error for the different data fusion techniques.

Figure 7 shows the average error resulted when using each data fusion techniques. As we can see, Syntra WAvg resulted in an average error of $2.36\ m$, being the most accurate technique, followed by Syntra Avg with $2.42\ m$. The worst result observed was $2.93\ m$, from Syntra Voting. In the last section, we saw that by using only a single synthetic dataset, without data fusion, we could get an average error of $2.84\ m$ in the best dataset. Thus, only Syntra Avg and WAvg really resulted in a better solution than any of the individual datasets, with Syntra Dist being very close. However, it is important to note that in a real-life application, we do not which of the individual datasets would be the best without doing the laborious real-world training.

Figure 8 shows the distribution of the positioning errors. As we can see, in the

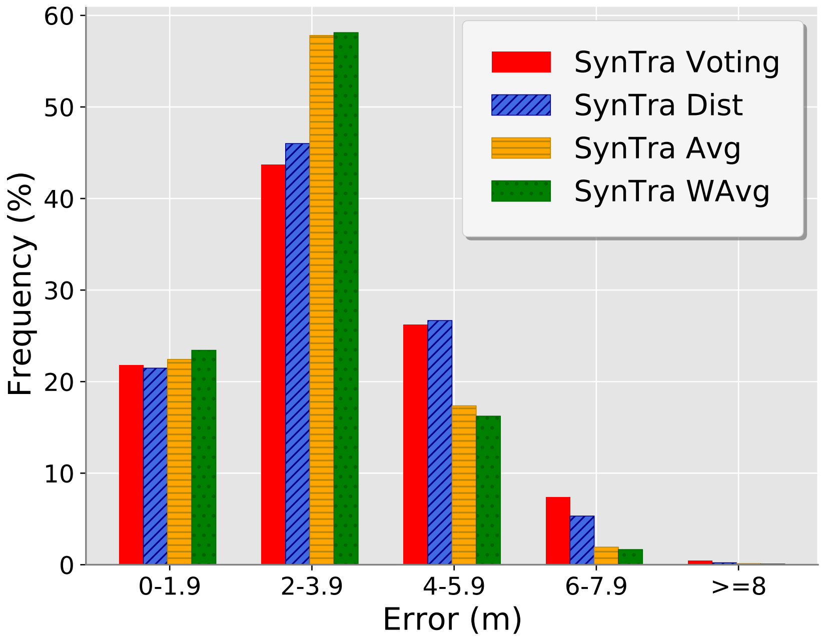

Figure 8 – Error distribution of the estimates positions for the different data fusion techniques.

case of Syntra WAvg, almost 60% of the errors are between $2\ m$ and $4\ m$, while more than 20% are less than $2\ m$. Figure 9 presents the cumulative error of the position estimations for each of the data fusion techniques. The sharper the curve, the better since most of the estimations have smaller errors. As we can see, Syntra WAvg were able to achieve the lowest errors, having almost 85% of the estimations with an error smaller than $4\ m$.

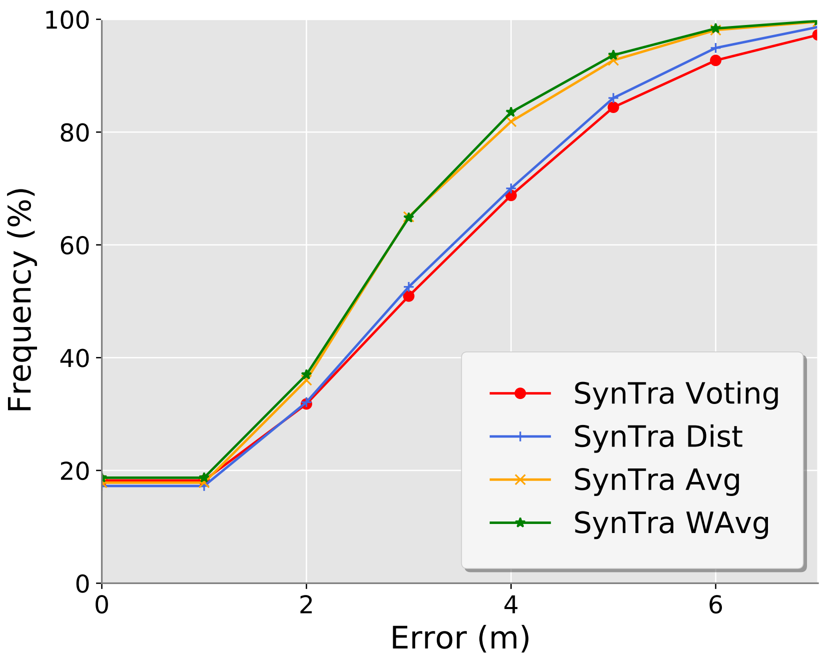

Figure 9 – Cumulative error of the position estimations for the different techniques.

The main reason Syntra Avg and WAvg resulted in the best solutions is that they use all of the 81 predictions from the synthetic datasets to compute their positions. In these solutions, the final estimated position is taken by averaging the coordinates from all predicted positions in each synthetic dataset.

In the case of Syntra Avg, the estimated position can be affected by outliers caused by datasets with unrealistic propagation model parameters. As can be seen in Figure 6, the accuracy is different according to the use of each offline synthetic dataset. In this figure, the orange line highlights the dataset with the smallest average error, in this case, $2.84\ m$, while other datasets resulted in a $4.75\ m$ average error.

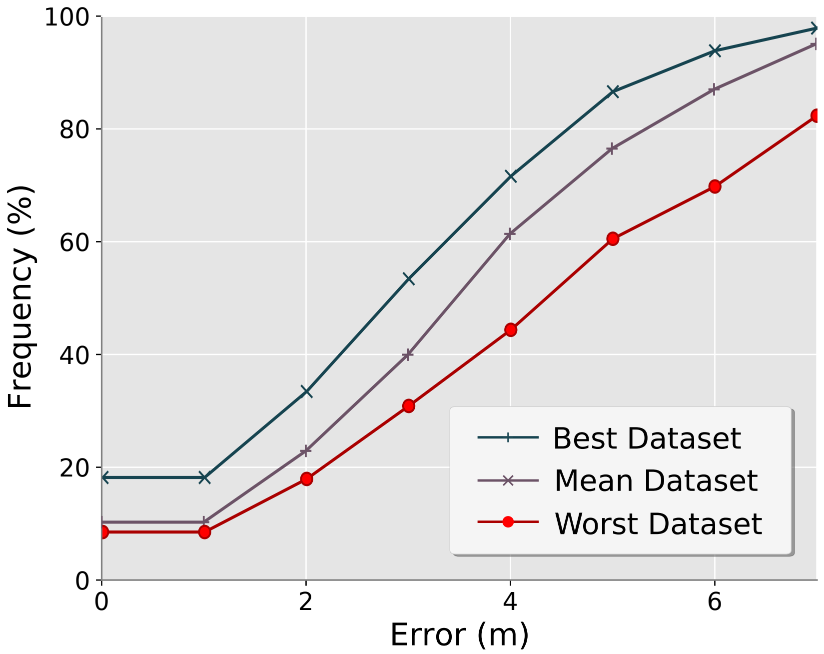

Figure 10 – The cumulative error for the individual datasets with smallest (2.84 m), mean (3.52 m), and largest (4.75 m) positioning error.

To better observe and compare the behavior of some specific datasets, Figure 10 shows the cumulative error of 3 synthetic datasets that resulted in the smallest, average, and largest positioning errors. In this figure, the best dataset is generated by the parameters $PL_{d0} = -55$, $\eta = 4$, $L_i = 3\ dBm$, $X_\sigma = 3$, while the mean dataset is generated the parameters $PL_{d0} = -50$, $\eta = 3.5$, $L_i = 2\ dBm$, $X_\sigma = 3$, and finally, the worst dataset is generated by the parameters $PL_{d0} = -60$, $\eta = 4.5$, $L_i = 4\ dBm$, $X_\sigma = 1$. We can see that just by varying some parameter values, the average positioning error is very different. In the best dataset, more than 70% of the position estimations resulted in errors lower than $4\ m$, while in the worst dataset, only 40% of the estimations were lower than the same error.

For this reason, the Syntra WAvg is proposed to penalize outliers and benefit from estimates closer to the real position. For this, we use a quality measurement based on the Euclidean distance from the estimated sample to the real-world sample, as explained in Sections 2.3.2.2 and 2.3.2.4. Then, we use all of the 81 predictions to estimate the final beacon position, penalizing the predictions more distant.

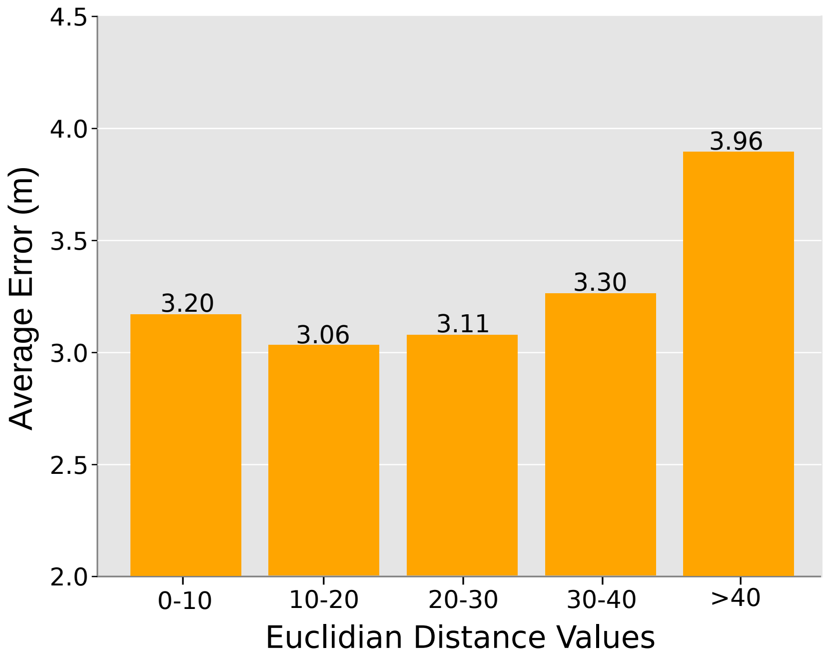

Figure 11 – Average positioning error by Euclidean Distance. Higher Euclidean distances can be used to identify higher positioning errors. Higher distance values can increase the positioning error by almost 1 m.

To better visualize how the Euclidian Distance between the estimated sample and the real-world sample can predict outliers, Figure 11 shows the average positioning error by this distance. As we can see, even though this metric is not able to indicate how small an error will be, it can indeed identify the position estimations with higher errors.

2.5.3 Comparison With Other Solutions

In this section, we compare the performance of our Syntra WAvg to traditional IPS approaches from the literature. We analyze the results provided by the different approaches in our scenario. The evaluated approaches are:

1. Model-based: a multilateration-based solution that uses the log-distance propagation model, as in (Fang and Chen, 2020; Shi et al., 2020).

2. Best Dataset: a fingerprint-based IPS using our best, single synthetic dataset, similar to (Ali et al., 2017).

3. Real Training: a fingerprint-based IPS with a complete, laborious training of the whole area, as in (Bahl and Padmanabhan, 2000; Torres et al., 2016; Youssef and Agrawala, 2005).

Model-based IPSs require only minimal training to estimate an unknown position. This training is required for finding the log-distance model parameters that allow the estimation of distances between the mobile devices and the anchor nodes through the measured signal strengths. In this evaluation, we used the log-distance model with the parameters $PL_{d0} = -55$, and $\eta = 4.2$. We chose these values based on the signal samples collected during our training, and we confirmed those were the best possible values, resulting in the smallest positioning errors. The positioning estimate was done using the least-squares algorithm.

For the Best Dataset, we carried out an experiment to find the best log-distance model parameters for a single synthetic dataset. This solution represents the one proposed in (Ali et al., 2017), as mentioned in the related work section. However, in this case, the authors generate only a single synthetic dataset to represent the signal behavior in the whole environment. To be fair in our comparisons, we consider this single dataset to be the best synthetic dataset generated by the propagation model. However, as mentioned earlier, finding such parameters remains a challenge and requires real-world training. In addition, for large-scale scenarios, this approach is not ideal to characterize the signal behavior in the different regions of the scenario.

Finally, to evaluate the performance of the traditional fingerprint using a Real Training, we separated our real-world data collection into training and testing, in which the measurements from 8 beacons were used to train the model, and the measurements from the other 3 beacons were used for testing. The KNN algorithm with the parameter

$K = 10$ was used to find the reference point with signals most similar to those measured in the online phase. This approach can be seen as the best-case scenario since we gathered real-world RSSI data from the experimented area. The main goal of our solution is to get as close as possible to this approach but without requiring the laborious training phase.

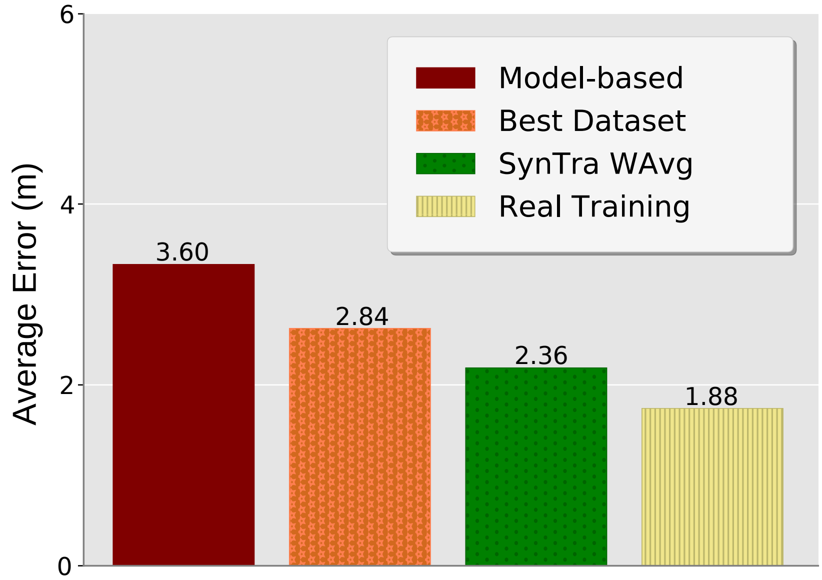

Figure 12 – Average error of the evaluated methods: model-based IPS, fingerprint with the best, single synthetic dataset, conventional fingerprint with a real training dataset, and our proposed solution.

Figure 12 shows the average positioning error of the evaluated approaches. As we can see, the average error for the model-based solution is $3.60\ m$, being the highest error among all approaches. The main reason for this is that the signal transformation into distance using a signal propagation model with fixed parameters is unreliable. In addition, the high RSSI variance, which is natural in indoor environments, makes this task even more complex. Hence, fingerprint-based techniques are most widely used since they result in lower positioning errors.

Still in Figure 12, we can see that the positioning error decreased when we used only one synthetic dataset generated by the best parameter values, resulting in $2.84\ m$. However, as mentioned earlier, although there is no need to transform the signal into the distance, in a real-world application, we would not know which of the several datasets would result in the smallest error without requiring a real-world training phase.

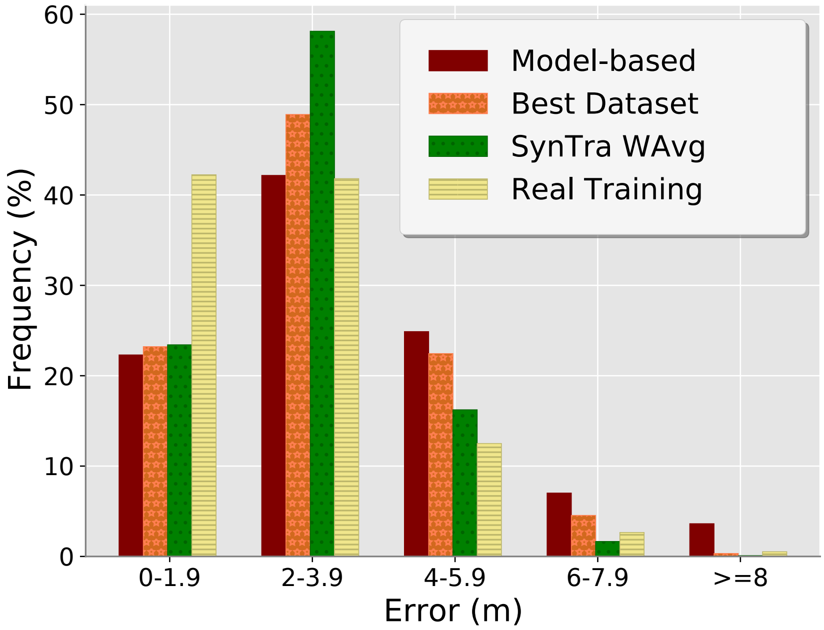

Figure 13 – Error distribution of the evaluated IPS techniques.

Our proposed Syntra WAvg, resulted in better position estimations than the previous approaches with an average of $2.36\ m$, being almost 20% lower than the best, single synthetic dataset and 35% better than a model-based solution. Figure 13 shows that our approach contains a higher number of measurements with lower positioning errors when compared to these approaches, behind only the fingerprint with real-world training, which is the best case possible.

Finally, the fingerprint technique with a Real Training dataset resulted in the lowest average error among all of the mentioned techniques. The main reason is that this technique uses the training dataset with signal measurements from the real-world environment. Thus, despite the signal propagation model being able to adjust its parameters to generate synthetic signals similar to real-world signals, the propagation channel has complex characteristics in indoor environments. Thus, an approximation of this signal behavior is the maximum that we can achieve.

Figure 14 shows that the fingerprint with Real Training dataset has about 55% of the position estimations with an error smaller than $2\ m$, followed by our approach with almost 40% of the estimations. On the other hand, in our Syntra WAvg, more than 80% of the estimations have an error smaller than $4\ m$, almost the same as the fingerprint with Real Training. Thus, our solution was able to get close to the best-case scenario,

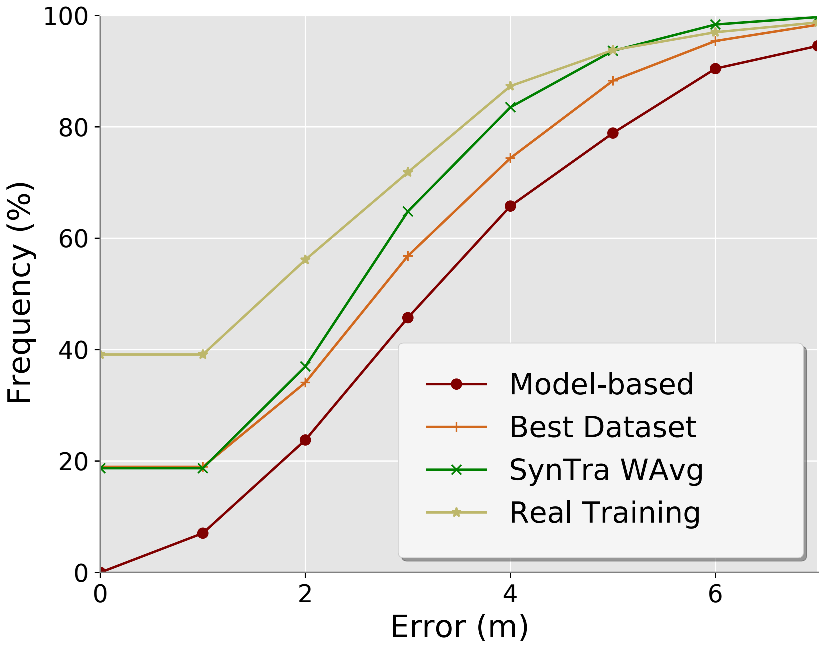

Figure 14 – Cumulative error of the position estimations.

with a difference of only $0.48\ m$, but without requiring any real-world training. As mentioned earlier, a possible solution to bring our solution closer to the real world would be to use crowdsourcing to supplement synthetic datasets with real data and obtain a hybrid solution.

Table 4 – Table with average error per room comparing the different approaches, highlighting the smallest mistakes compared to our approach.

ROOM	*SynTra Voting*	*SynTra Dist*	*SynTra Avg*	*SynTra WAvg*	*Model-based IPS*	*Best Dataset*	*Real Training*
Room 01	3.54	3.60	**2.89**	2.95	3.72	3.48	**2.70**
Room 02	3.67	3.39	3.26	**3.16**	3.90	3.21	**2.02**
Room 03	2.77	3.27	**2.33**	2.43	4.02	2.88	**2.21**
Room 04	2.68	3.34	**2.27**	2.38	3.59	2.74	2.40
Room 05	2.60	2.88	**2.14**	2.23	3.54	2.39	2.20
Room 06	3.29	3.38	2.45	**2.44**	4.29	3.10	**2.31**
Room 07	4.13	2.99	2.95	**2.68**	2.75	3.57	**2.07**
Room 08	3.14	2.80	2.28	**2.05**	3.07	3.23	**1.80**
Room 09	3.16	2.62	2.71	**2.60**	**2.60**	3.20	**2.03**
Room 10	3.74	3.43	**2.81**	2.93	3.23	3.68	**1.86**
Room 11	2.90	2.86	3.12	**2.84**	3.20	2.91	**1.62**
Hallway 1	1.49	1.36	1.31	**1.30**	4.49	1.4	**0.84**
Hallway 2	1.44	**1.40**	1.66	1.44	3.34	1.25	**0.57**
Hallway 3	**0.77**	1.19	0.87	0.92	5.69	0.91	**0.74**
Average	2.93 m	2.85 m	2.42 m	2.36 m	3.60 m	2.84 m	1.88 m

To better understand the behavior of the errors throughout the evaluated scenario, in Table 4, we separate the average error per room obtained by each approach. To facilitate comparison, we use values in bold. In this table, we can see, as expected, that the fingerprint with real training was the one with the lowest errors per room. However, we can see that the accuracy per room varies a lot according to the approach used due to factors such as the number of reference points, anchor nodes coverage, and obstacles.

Model-based IPS are the ones that result in the highest average error per room, with high errors mainly in hallways, rooms 2, 3, and 6. This happens because in these rooms, the 3 anchor nodes with the strongest signals usually form a linear organization, which makes positioning calculation difficult by least-squares algorithms. On the other hand, in almost every room, our Syntra Avg and Syntra WAvg data fusion solutions had the lowest average error compared to the model-based IPS, and best individual dataset. In this case, the largest average error obtained by Syntra WAvg was $3.16\ m$ in room 2, still resulting in position estimates close to the real position in the same room.

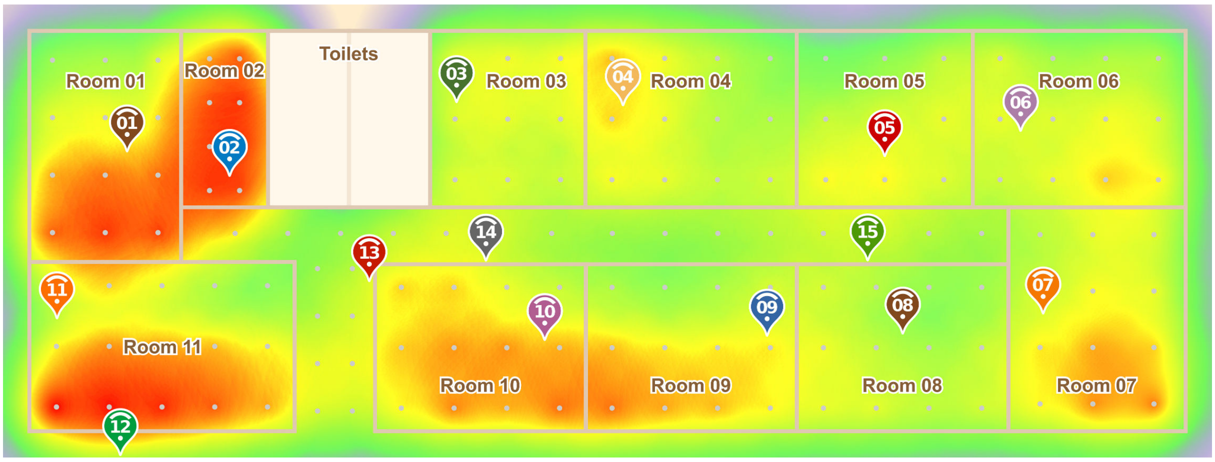

Figure 15 – Heat-map of the average errors for all test points in the scenario.

In order to better visualize our solution, Figure 15 shows a heat-map of the SynTra WAvg errors in the whole scenario. In this heat-map, we can see problematic regions of the scenario, such as rooms 1, 2, and 11. In these rooms, the worst performance is due to the positions and lack of anchors coverage. In our scenario, the anchor nodes were fixed in locations where it was somewhat convenient to connect them to the power supply. Thus, increasing the density of anchor nodes and centralizing them in the rooms is an alternative to reduce the positioning error.

2.5.4 Computational Costs Analysis

In this part, we discuss the computational costs of our solution. The most significant and sensitive part is the position estimation that uses a fingerprint dataset. This dataset is composed of s samples measurements for each m RPs and n different anchor nodes. Therefore, the dataset total size is $(s * m * n)$.

During the online phase, signals from a mobile device are used to estimate its position. As mentioned earlier, the algorithm used for this process is KNN. The complexity of KNN depends on the size of the input dataset (Praveen Kumar et al., 2019). Thus, in the traditional fingerprint method, the cost of estimating the mobile device position is $O(s * m * n)$. However, our approach creates T different synthetic datasets. In this case, a mobile device will be classified by KNN into T different datasets. Thus, the complexity of our approach is greater, when compared to the traditional fingerprint, since it involves one more variable T, thus being $O(T * s * m * n)$. There is still the data fusion cost, but it is at most $O(T)$ and, thus, can be ignored.

As we can see, our proposed solution requires a higher processing load compared to traditional fingerprint systems that run KNN only once. This is a key aspect since it limits the number of position estimations per second. However, since it is possible to combine several samples to be classified at the same time using vector implementations of KNN, this limitation can be eased when running in parallel on different CPU cores or even using Graphics Processing Units (GPUs) on a dedicated IPS server.

2.6 Discussion

Fingerprint-based IPSs have an extensive training phase that collects signal strengths at different reference points to create a fingerprint radio map. This technique does not require any prior knowledge of the scenario for radio map creation. However, this radio map needs to be re-created in the presence of changes in the scenario, such as changes in the walls and insertion of new obstacles, making it unfeasible to be maintained for large scenarios. On the other hand, to reduce training cost, our solution requires a small effort to get the floor plan information, and it also needs prior knowledge of the anchor nodes' positions to generate the synthetic training dataset. Despite this effort, the located mobile devices could be displayed on the same map of the area, which could be used as the floor plan.

One can argue that the datasets generated by the RSSI simulator do not represent the real behavior of the signal for the whole scenario. However, when we generate a set of synthetic datasets using different parameters for the signal propagation model, we try to approximate the real signal distribution dynamically in different regions of the system. Some datasets can result in the best estimations in some areas, while other datasets will have better results in other areas. The proposed data fusion techniques try to combine the best estimations into a single solution. Also, in our experiments, we only considered 2D environments. For more complex environments, such as multiple floors, the log-distance model can be easily extended to also include the higher loss from the floors and ceilings.

In our experiments, to perform the comparison with the traditional fingerprint, we used 150 different RPs $2\ m$ apart from each other. It is known that increasing the density of RPs, decreases the average error at the cost of increasing the workload needed for the fingerprint collection. Our solution can create an unrestricted number of virtual RPs, and generate denser datasets, possibly lowering the average error. However, in this case, the performance evaluation could only be done by simulation, which would not be ideal to represent the real scenario. Another important issue is regarding the mobile devices. In our experiments, we used 11 different devices but with the same hardware from the same manufacturer. However, when using mobile devices with different hardware, such as different smartphones, the signal behavior can vary. We believe that our use of several propagation parameters, combined with data fusion, might consider these hardware differences, resulting in better results than traditional fingerprint-based IPSs. In this work, this aspect was not evaluated and will be studied in future works.

2.7 Conclusion

In fingerprint-based IPSs, building the training dataset in the offline phase is an expensive and complex task. To reduce the effort of data collection, we propose and evaluate a new fingerprint-based IPS, that uses a signal propagation model to generate several synthetic training datasets. We propose four new techniques for the online phase that use data fusion from the position estimations obtained through the different synthetic datasets to estimate a single, more precise position.

Our experiments in a real-world scenario, show two significant contributions: (1) the use of several synthetic datasets to characterize the signal in different regions of the scenario without the need for complex data gathering from the environment, and (2) the use of data fusion techniques to compute the final position of the mobile device. Our performance evaluation shows that Syntra resulted in an average error of $2.36\ m$, being almost 20% lower than the best, single synthetic dataset, 35% better than a model-based solution, and only $0.48\ m$ from a traditional fingerprint-based IPS, the best-case scenario.

In future works, we intend to experiment with other signal propagation models for the RSSI simulation. We also intend to evaluate the performance of different machine learning algorithms other than KNN. Finally, we intend to propose and evaluate the performance of other data fusion techniques and crowdsourcing.

3 Adaptive Path Loss Model for BLE Indoor Positioning System

In this chapter, we propose the ADAM (Adaptive Model) Positioning System, a model-based IPS that chooses the best Anchor Nodes (AN) to benefit the positioning computation and uses different parameters for the log-distance model to represent the signal in different regions and conditions of the scenario. Then, we estimate a single, more precise position using a data fusion technique. Our proposal does not require training nor prior knowledge of the best parameters for each region. We evaluated the performance of our proposed system in a real-world, large-scale environment using Bluetooth-based mobile devices. Our results clearly show that ADAM can locate mobile devices with an average error of 2.93 m in relation to the real position, which is 23% better than literature-based models using fixed parameters for the entire environment.

3.1 Introduction

The Global Navigation Satellite System (GNSS), which includes the Global Positioning System (GPS), is today the most widely used location system and has a high impact on locating devices in outdoor environments. However, because of interference from buildings and the lack of line-of-sight, GPS proved to be ineffective for use in indoor environments (Zhang et al., 2017). Thus, as an important Internet of Things (IoT) based application, Indoor Positioning Systems (IPSs) aim at locating devices or people in these indoor environments such as hospitals, shopping malls, and schools. In an indoor environment, several interferences cause a challenge to IPSs, demanding increasingly robust solutions. Therefore, several techniques are proposed using different technologies and information sources, resulting in different precision, cost, and complexity depending on the application and scenario.

IPSs mainly use radio frequency-based technologies such as WiFi (Mendoza-Silva et al., 2021), Bluetooth (Wang et al., 2016), WLAN (Ullah et al., 2020), and Radio Frequency Identification (RFID) (Sasikala et al., 2021). Among these technologies, Bluetooth Low Energy (BLE) has been widely used in IPS because of its low power consumption, small hardware size, and lower-cost compared to WiFi access points (Cantón Paterna et al., 2017). Several methods are used to estimate positions such as Time of Arrival (ToA) (He et al., 2012), Time Difference of Arrival (TDoA) (Schreiber and Bajer, 2016), Angle of Arrival (AoA) (Fascista et al., 2017), and Received Signal Strength Indicator (RSSI) (Bullmann et al., 2020). Methods based on ToA, TDoA, and AoA are more accurate than RSSI-based methods. However, they require high precision components or special antennas, making the system expensive and complex for several applications. On the other hand, RSSI-based methods have the advantage of low complexity, cheaper hardware, and simple algorithms.

Many IPSs can be classified into two categories: model-based (Li et al., 2018; Yang et al., 2020) and fingerprint-based (Moghtadaiee et al., 2019; Wang et al., 2020,1). Fingerprint-based IPS are divided into two phases: offline and online. During the offline phase, also known as the training phase, several Reference Points (RPs) are used to collect RSSI data in different regions of the scenario to create a signal map database. In the online phase, the RSSI measured in real-time are compared to the signal database to estimate the mobile device position based on a matching algorithm, which chooses the RP with signals most similar to those measured in the online phase. Although it has high accuracy, this method has several limitations that make it impractical in some locations. One of the main limitations is the high labor cost required to establish the initial training database which can quickly become outdated with environmental changes such as new walls, furniture, and anchors' positions (Zhang et al., 2017).

In a model-based IPS, a signal propagation model is used to map the distances among a mobile device and several anchor nodes, which are fixed in the scenario, and their positions are known in advance. The positioning accuracy is highly dependent on a good choice of the path loss exponent, used by the propagation model to characterize the signal in the environment. After mapping all the distances, this method uses at least three anchors to find the target's position through Least square or maximum likelihood. Compared to fingerprinting, model-based IPS requires much fewer data and deployment efforts (Zhang et al., 2017).

In this chapter, we propose the ADAM (Adaptive Model) Positioning System, a novel model-based IPS. First, we establish the system coordinates and store the anchors' information, such as their identifications and positions. Then, we propose an algorithm to choose, based on the organization of their coordinates, the best anchor nodes that are mostly non-collinear to benefit the position computation using Least square. Finally, we use the log-distance signal propagation model with different parameter values to map the distances between the mobile device and the best anchors, avoiding the exhaustive process of data collection/training. By using different model parameters, we have different position estimates that best represent different regions of the building floor plan. We then use data fusion techniques to combine all of the estimated positions into a single, more accurate position. We evaluated the system's performance in a real-world, large-scale scenario using Bluetooth-based devices. Our results show that the ADAM Position System can locate mobile devices with an average accuracy of 2.93 m around their real position, without the training effort required by fingerprint-based techniques and is 23% better than a model-based IPS with fixed parameters.

Our main contributions are summarized as follows:

1. We propose a new algorithm to choose the best anchor nodes that benefit the position computation based on the collinearity of the anchors' coordinates.

2. Our solution uses different values for the log-distance parameters to expand the characterization of the signal in different regions, allowing us to map different distances among the devices.

3. We use data fusion techniques to combine different estimated positions (based on different model parameters) into a single, more accurate position that outperforms approaches that use only fixed parameters.

4. Through several experiments carried out in a real-world, large-scale environment, we verified that the ADAM Positioning System proved to be competitive compared to IPS based on a state-of-the-art model, with no complex data collections in the environment to estimate the best model parameters.

This Chapter is organized as follows. In the next section, we show our related work. Section 3.3 introduces the system model. Section 3.4 shows our real-world testbed and experimentation methodology. In Section 3.5, we show and discuss the results of the performance evaluation. Section 3.6 presents a discussion of the work. Finally, Section 3.7 presents our conclusions and future work.

3.2 Related Work

In recent years, several solutions have been proposed to create efficient, cost-effective IPSs. In this section, we review the relevant RSSI-based IPSs in the literature, analyzing and comparing them to our approach.

Most proposed solutions use model-based or fingerprint-based techniques. Fingerprint-based techniques separate the scenario into different training points to collect signals and then create a fingerprint database or radio map. The performance of this technique depends on the number of reference points, the number of packets collected, and the matching algorithm (Mendoza-Silva et al., 2021). The best-known work that uses this technique, called RADAR (Bahl and Padmanabhan, 2000), combines empirical measurements with signal propagation modeling to determine the user's location.

In Wang et al. (2020), the authors use three fingerprint databases collected at different distances to compare eight positioning algorithms. The method used by Wang et al. (2020) and Bahl and Padmanabhan (2000), has two major drawbacks: first, it is challenging to create the database as it requires collecting data from multiple locations, and second, updating the database because of changes in the environment is a time-consuming and labor-intensive task. These issues make it difficult for the method to be widely adopted.

To minimize the cost of collecting signals, Wu et al. (2016) use virtual reference points based on the log-distance model. Similarly, in Zhang et al. (2017), the authors use a path loss model to create the fingerprint database. The work has shown that a decrease in the amount of data collection, although financially beneficial, leads to a decline in the precision of location identification. Specifically, the ability to locate mobile devices is limited to room-level resolution, as opposed to locate the devices within the room.

Signal propagation models can reduce the effort to implement IPSs. In Onofre et al. (2016), the authors implement an adjusted distance curve to measure the distance among BLE tags using a log-distance model. The curve fitting technique employed eliminates the requirement for many data. However, it should be noted that they performed the experiments in a small, 13m room with no obstacles. Given the highly dynamic nature of RF signals, the derived distance curve may not be generalized to large-scale environments.

Moghtadaiee et al. (2019) propose a zone-based path loss propagation model. Experiments show that the zone-based model decreases the average error by 26% when compared to conventional path loss models (Cantón Paterna et al., 2017; Wu et al., 2016; Zhang et al., 2017). However, the methodology of this work presupposes that the optimal path loss exponent value is already known.

Yang et al. (2020) proposed a new RSSI-based trilateration algorithm. The authors preprocess data using a Gaussian filter to reduce the RSSI variation and the path loss exponent is estimated through a Least Squares Curve Fit (LSCF). Similarly, the authors in Cengiz (2021) propose to increase the number of anchor nodes and use line fitting algorithms to improve location estimation and map RSSI to distance. In these works, the authors adopt a strategy with fixed parameter values of the propagation model to describe the signals in all regions of the scenario. This method, however, is not deemed optimal for large-scale scenarios, due to the signal behavior in different regions of the scenario. Also, the previously cited works tend to solely focus on increasing the number of anchor nodes without considering the impact of their spatial organization on the final positioning result. Finally, Shi et al. (2020) proposed a positioning system that adjusts the path loss exponent based on filters applied to the collected RSSI data. The main disadvantage of Shi et al. (2020) is the need for calibration, which involves model parameter adjusting. This task is comparable to the fingerprint offline phase, which requires significant effort.

3.2.1 Discussion

Our proposed solution eliminates the need for data collection in the environment by requiring only the predefined coordinate information of the anchor nodes. Additionally, our solution uses advanced resources by combining a set of multiple parameter values for the log-distance signal propagation model, enabling dynamic signal mapping in different regions of the scenario. This approach is distinct from other works based on the signal propagation model. A data fusion technique is employed to integrate the optimal results from each region into a single, more precise position estimate. Furthermore, we do not use a fixed number of anchor nodes and we choose the anchor nodes' combination that favors the position computation through the Least square algorithm. Results from performance evaluations show that our work outperforms the works that use all anchor nodes and a fixed parameters for the propagation model.

Therefore, this work innovates by exploring the log-distance signal propagation model to map different distances according to the variation in model parameter values and use data fusion from different estimated positions to improve the final positioning estimate.

3.3 ADAM IPS Architecture

In this section, we present our ADAM Positioning System architecture, which can be divided into two phases. In phase 1, we use the scenario's map information to store the identification and coordinates of the anchor nodes. In phase 2, we use the log-distance signal propagation model to map different distances between the mobile device and the best anchor nodes through RSSI. Finally, we compute the mobile device position

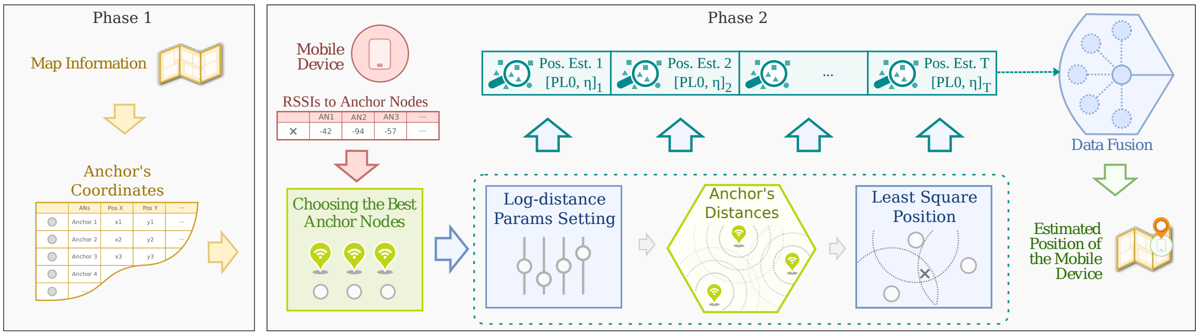

Figure 16 – Architecture of our ADAM Indoor Positioning System.

using data fusion from sub-estimates returned by the Least square algorithm. Based on the ADAM architecture system in Figure 16, in summary, the steps of our solution for estimating the mobile device positioning are:

Phase 1:

1. Fix the n anchor nodes in the scenario (walls or ceiling);
2. Get the scenario floor plan and create a 2D virtual map;
3. With the help of the floor plan and a measuring tape, store the identifications (ID) and coordinates (x, y) of all anchor nodes.

Phase 2:

1. Get RSSI from a mobile device (beacon) to anchor nodes (receivers);
2. Choose the best anchor nodes to be used in the positioning computation, based on the three nodes with the highest RSSI and far from being collinear;
3. Use the signal propagation model with different parameters to get different distances between the mobile device and the anchor nodes chosen in the previous step;
4. Perform residual computation to identify estimates considered outliers and remove them from the final positioning computation;
5. Perform data fusion of the position estimations from the previous step and combine them into a single, more accurate target position.

The details of each step are explained in the following sections.

3.3.1 Phase 1 - Anchor Nodes Information

In model-based IPSs, it is common to require collecting some data in the environment to fit propagation models, causing an extra workload that increases with the size of the scenario. Our proposed solution does not require the effort of prior training, requiring only information from the anchor nodes, which are common in this type of system and easy to get through the building floor plan. A floor plan of a building is available in legally approved buildings and contains all information on dimensions and room layout.

To simplify the explanation, we assume a $2D$ area with a set of n anchor nodes, with their known coordinates $(X_i,\ Y_i)$, $i\ =\ (1, 2, 3, ..., n)$, distributed across the scenario ensuring a good signal coverage. With a tape measure and the floor plan area, we can store in a database the identifications and coordinates where the anchor nodes are fixed (ceiling or wall of the rooms). In this way, in the first phase of the system, we store the information in a Table 5.

Table 5 – Anchor node identifications and coordinates.

Anchor$_{id}$	**Pos$_x$**	**Pos$_y$**
AN$_1$	4.5m	13.0m
AN$_2$	8.0m	9.5m
AN$_3$	16.5m	15.0m
...	...	...
AN$_n$	X$_n$	Y$_n$

Therefore, in this phase, we created a database containing the anchor node information, storing their coordinates and their respective identifications. Storing this information in this phase is required since in the next phase, which is when we estimate the mobile device position, we need to choose the best anchor nodes to be used in the positioning computation, as explained in Section 3.3.2.1. Thus, the only prior knowledge that the online phase requires is the coordinates of the anchor nodes.

3.3.2 Phase 2 - Mobile Device Position

In this phase, the RSSI values are used to map the distances between anchor nodes and the mobile device. The mobile device (beacon) sends BLE advertising packets at a 1 Hz rate, which are received by anchor nodes fixed in the scenario. Then, the anchor nodes measure the RSSI through the BLE advertising packet, which is the crucial information used to estimate the mobile device's position on the map. As soon as the anchor nodes process the packet and measure the RSSI, they send the following information to the server: its identification, BLE packet timestamp, mobile device identification, and finally, the RSSI. The server gathers this information to create an RSSI vector of all anchor nodes, through the same packet, which will be used in the next steps. An RSSI vector example can be seen in Table 6:

Table 6 – Example of RSSI values among anchor nodes.

Anchor$_1$	**Anchor$_2$**	**Anchor$_3$**	**Anchor$_4$**	**Anchor$_5$**	**Device**
-61	-68	-74	-71	-81	Beacon 1

When there is no communication due to long distances or excessive obstacles, we use the value -105 to represent no signal. Furthermore, based on some preliminary experiments, we ignore RSSI values smaller than -91 dBm, as in these cases, signal interference hampered the distance mapping. With the information from the anchor nodes and their respective RSSI values, we can use this data in the next steps of the system.

3.3.2.1 Choosing the Best Anchor Nodes

In this subsection, we discuss how we choose the best anchor nodes that will be used in the position computation. As mentioned in the previous subsection, the main information used by the positioning computation is the RSSI. During the positioning computation, we perform an RSSI mapping of each anchor node to a distance, using Equation 3.2. The most common way in model-based IPS is to use information from all anchors. However, two problems can arise from this. First, weak RSSIs are related to enormous distances or many obstacles between devices, so when we map weak RSSIs to a distance, the error is usually high. Thus, the ideal is to use only the anchor nodes with the strongest RSSI. Second, to estimate the position closest to the real device position, we use the Least square algorithm, which will be detailed in the next section. So, when we use anchors with collinear coordinates, the Least square algorithm decreases the estimated position accuracy, which may be the case when we choose only the nodes with strong RSSIs. A possible solution would be, during the system deployment, to prevent the nearby nodes from having a collinear organization.

However, most times we do not have this freedom because of the impossibility of fixing it at the ideal points, for example due to lack of electrical installation, or because of the high anchor nodes density to cover all environments. Therefore, in this part, we seek to find the best anchor node combination that is far from being collinear. The first step is to sort the RSSI vector (Table 6) in order of the strongest RSSI. The result is show in Table 7.

Table 7 – Ordered RSSI

Anchor$_1$	**Anchor$_2$**	**Anchor$_4$**	**Anchor$_3$**	**Anchor$_5$**	**Device**
-61	-68	-71	-74	-81	Beacon1

Next, we separate them into sets of three anchor nodes, as we can see in Table 8:

Table 8 – Set of anchor nodes

Anchor Nodes	**Collinearity Factor**
{Anchor$_1$, Anchor$_2$, Anchor$_4$}	0.92
{Anchor$_1$, Anchor$_2$, Anchor$_3$}	**0.20**
{Anchor$_1$, Anchor$_2$, Anchor$_5$}	0.73
...	...
{Anchor$_4$, Anchor$_3$, Anchor$_5$}	0.64

So, we use a collinearity filter based on the anchor coordinates to find the collinearity factor among anchor nodes. For each set of three anchor nodes, we use their coordinates in a linear regression function to get the line that represents the average of all coordinates. The next step is to compute the difference between the respective coordinates for this line. Thus, we used the determination coefficient (score) to get the anchors' organization. This score is a value that ranges from 0 to 1, and the closer to 1, the more collinear the organization is. The closer to 0, the further the anchors are from being collinear. Therefore, based on some preliminary experiments, we use a threshold of 0.5 to accept the anchors while rejecting combinations with a result higher than the threshold. In Section 3.5.2, we further evaluate this threshold in our experiments. If a set with 3 anchor nodes passes the check, then they can be used in the position computation. Otherwise, we test for another set and then run the test again. We repeat these steps until we find the anchors with the strongest RSSI and the least collinearity.

However, if all tests fail, we choose the combination of three anchors with the strongest RSSI values and the least collinearity among all options. In Figure 17, we can see an illustration of the linearity filter.

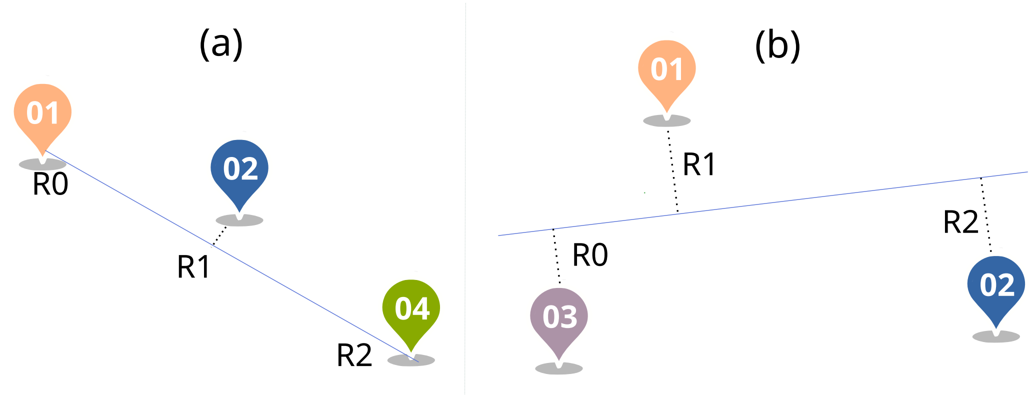

Figure 17 – Collinearity between 3 anchor nodes. In (a) the nodes are organized close to collinear, i.e, collinearity factor close to 1. In (b) the nodes are far from being collinear, i.e, collinearity factor close to 0.

3.3.2.2 Distance Mapping and Position Computation

After choosing the best anchor nodes according to the filter mentioned in the previous step, we will use their respective RSSIs to map the distance between the best anchors and the mobile device. For this, we use the log-distance signal propagation model, which models the signal strength in relation to the distance using a logarithmic equation. Equation 3.1 (Röbesaat et al., 2017) describes the signal propagation model:

$$RSSI_n = PL_{d0} - 10\eta \log_{10} \frac{d}{d_0} \tag{3.1}$$

where PL_{d0} is the RSSI reference value measured at distance $d0$, the parameter η is the path loss exponent, which indicates the signal behavior as it propagates in the environment. Finally, d is the distance among the devices. In this case, Equation 3.1

uses distance to model the RSSI, but we can use it inversely to map the RSSI at an approximate distance from the target to each anchor, through an adaptation described in Equation 3.2 (Röbesaat et al., 2017):

$$d_n = 10^{\frac{PL_{d0} - RSSI_n}{10 \times \eta}} \tag{3.2}$$

where $RSSI_{dn}$ is the RSSI measured by the n anchor node through the packet received by the mobile device, and $dist_n$ is the distance mapping between them.

So, by choosing the best values in the parameters PL_{d0} and η, we can get the closest distance to the real distance for each anchor. However, this requires a parameter that can vary depending on the region the signal was sent from. After mapping all the distances, we estimate the mobile device position using Least square. This algorithm finds a point on the map that minimizes the distance to at least three anchors, where each anchor has a circle around it, with a radius equal to the distance. Considering n ($n > 3$) anchor nodes, and their distances, the target's position (x, y) can be estimated using Equation 3.3:

$$\begin{cases} {d_1}^2 = (x_1 - x)^2 + (y_1 - y)^2 \\ {d_2}^2 = (x_2 - x)^2 + (y_2 - y)^2 \\ \dots \\ {d_n}^2 = (x_n - x)^2 + (y_n - y)^2 \end{cases} \tag{3.3}$$

Thus, by subtracting the n-th equation from the first, the system can be linearized to $Ax = b$ (Huang et al., 2019):

$$A = \begin{bmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_2 - x_n) & 2(y_2 - y_n) \\ \dots & \dots \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{bmatrix}; \tag{3.4}$$

$$b = \begin{bmatrix} {x_1}^2 - {x_n}^2 + {y_1}^2 - {y_n}^2 + {d_1}^2 - {d_n}^2 \\ {x_2}^2 - {x_n}^2 + {y_2}^2 - {y_n}^2 + {d_2}^2 - {d_n}^2 \\ \cdots \\ {x_{n-1}}^2 - {x_n}^2 + y_{n-1}^2 - {y_n}^2 + d_{n-1}^2 - {d_n}^2 \end{bmatrix} \tag{3.5}$$

The Least square (3.6) is used to find the coefficient that minimizes the squared error. For this, the pseudo-inverse is the most used approach to solve the Least square problem for linear systems with Equation $Ax = b$ (Huang et al., 2019; Wu et al., 2019). Here, when b is not in the interval of A, then there is no solution to the system, but it is still desirable to find an x that minimizes the Euclidean distance for vector b. Thus, we can derive the pseudo-inverse matrix as a solution to the Least square problem. Therefore, the mobile device position can be obtained as follow:

$$X = (A^T A)^{-1}(A^T b) \tag{3.6}$$

3.3.2.3 Parameter Adjustment

The only environment-dependent variables for distance mapping are PL_{d0} and η. Because of the internal environment suffering a high variation in the RSSI, caused by multiple paths and various obstacles, it is difficult to map the distances using the RSSI. Although some works use fixed values for these parameters, this is not a good method for representing the signal in large-scale scenarios. By reason, we used a set with C different values for each of the mentioned parameters, which when combined, tend to result in a distance mapping closer to the real one. For example, considering the parameter setting being $PL_{d0} = [p_1, p_2, ..., p_i]$ and $\eta = [n_1, n_2, ..., n_j]$, we can combine the values of PL_{d0} with the values of η in Equation 3.2 to get $C = i \times j$ different distance mappings for each anchor. That way, instead of having just one circle around itself, there will be C different circles with a radius equal to the distances mapped by the parameter combinations.

Unlike approaches that use only fixed values, where only a single position estimate is obtained, our positioning computation is done using the different distances mapped according to combinations of parameter values. Among different estimated positions in C, many estimated positions will be closer to the real position. In the next step, we will get all of the position estimations and combine them into a single, more accurate position.

3.3.2.4 Data Fusion

In this step, we compute the mobile device position using all the position sub-estimations. As explained in before Section, when using the combinations of values for the propagation model parameters (PL_{d0} and η) in Equation 3.2, we perform the mapping from RSSI to distance, which will be used in the final position computation. Each distinct combination of PL_{d0} and η results in C_i positioning estimate. For example, using the set of $PL_{d0} = \{-40, -45, -50, -55\}$ and $\eta = \{3.5, 4.0, 4.5, 5.0, 5.5\}$, the number of different estimates is 20. However, although most of the estimates result in positions close to the real position, some parameter combinations result in positions considered outliers. To detect those positions that can negatively affect the result, we use a filter based on the residual computation. For example, considering an RSSI vector in Table 7 for ($Anchor_1, Anchor_2$, and $Anchor_4$), and $C_1 = (PL_{d0} = -50, \eta = 4.5)$, then we use Equation 3.2 to map the RSSIs vector in distance.

With the respective RSSIs mapped into distance using C_1 model parameters, we use Least squares (3.6) to estimate the target position (P_1). The next step is to check the distance between the C_1 estimated positioning (P_1) and each anchor nodes coordinates. Thus, we can verify the difference between the distance mapped by RSSI and the distance from P_1 for each anchor node. This filter, described in Equation 3.7, is performed for each of the C different estimates.

$$residual_{C_i} = \frac{\sum_{j=1}^{n} \sqrt{euclidian(P_i, AN_j) - d_j)^2}}{n} \tag{3.7}$$

where P_i is the (x_i, y_i) coordinate of each position estimated in C_i, AN_j is the j-th anchor node coordinate (x_j, y_j), previously known in phase 1, and d_j is the distance estimated based on the parameters combination used to find the position P_i. Finally, n is the number of anchors chosen as described in 3.3.2.1. Table 9 shows a good example result:

Table 9 – Residual values among anchor nodes.

Informations	**Anchor$_1$**	**Anchor$_2$**	**Anchor$_4$**
RSSI	−61	−68	−71
Distance (d)	3 m	4.3 m	5.7 m
euclidian(P, AN)	2.5 m	3.3 m	5 m
Anchor Residual	0.5 m	1 m	0.7 m

Therefore, if the residual computation for the current estimated position in C is greater than the threshold 5.0, this estimation is considered an outlier and will not be used in the final position computation. Otherwise, we store the estimated point coordinate and repeat the verification test for the other estimations in C. At the end of all checks, we have a dynamic list with the best estimated C' positions using the different parameter combinations. Finally, the final position of the mobile device is computed by averaging all the remaining C' estimates, resulting in a single, more accurate position. Although the average is highly affected by outliers, we use the filter based on the residual computation described in Equation 3.7 to detect the estimates considered outliers and exclude them from the final mobile device positioning, reducing the outliers' impact on the average of all position computations. The final position of the mobile device is computed as follows:

$$finalPosition = \frac{\sum_{i=1}^{C'} (X_i, Y_i)}{C'} \tag{3.8}$$

3.4 Experimental Testbed

This section presents details of our real-world testbed and the method to collect data. We will discuss the performance evaluation in Section 3.5.

3.4.1 System Environment

To evaluate the performance of our proposed ADAM Positioning System, we implemented the proposed solution on the second floor of a school building, covering an area of $720m^2$. The layout of the building consists of 11 classrooms and 3 hallways, as shown in Figure 18. The deployed infrastructure is based on anchor nodes with BLE technology. In Figure 18, we can see that at least one anchor node covers each space by room.

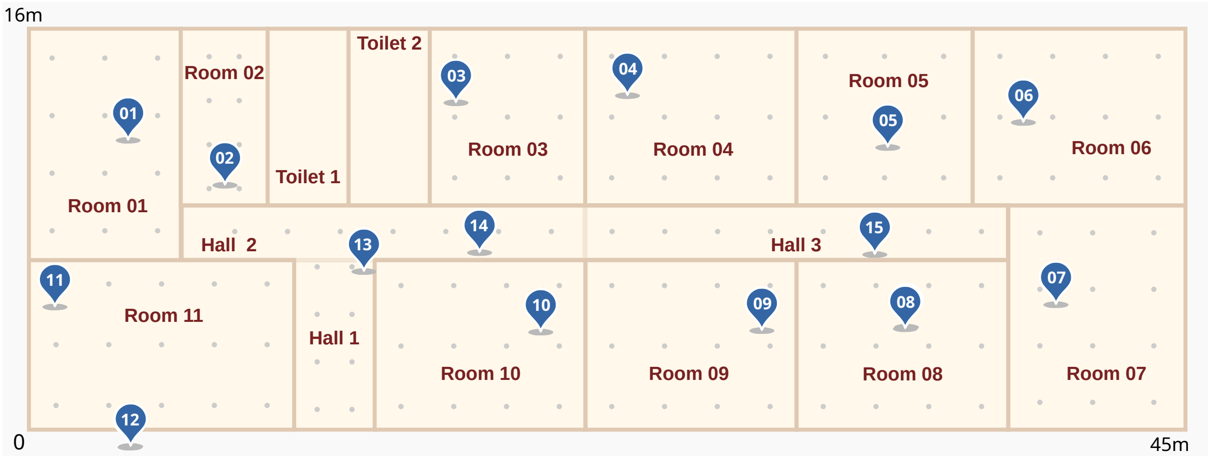

Figure 18 – Experimentation scenario consisting of 11 rooms, 3 hallways, 15 anchor nodes and 150 test points. Gray dots represent the 150 test points.

We distributed 150 test points across the scenario, spaced approximately 2 m apart from each other to collect RSSI samples. Decreasing the spacing among test points helps to reduce error in fingerprinting-based IPS, as more different data will be in the training phase. As our approach is model-based, we do not require prior training and we use data collection for evaluation purposes only. Therefore, a spacing of 2 m among test points results in an average of 12 test points per classroom, an amount needed to assess what region the mobile device is located. The data collection took place over 5 days covering different times of the day, and different climates.

To define the test points positions and the anchor nodes positions, we used the floor plan that contains the size of the respective room dimensions. Thus, we defined a coordinate system that uses the lower left corner of the map (room 11) as the origin of the system (X_0, Y_0) and we used an image software (called Inkscape) to generate an image containing the points properly spaced in the room. The image with the test points' positions is used by our application to help the data collection described in the next section. Finally, by knowing the real anchors' positions and all of the test points' positions, we have the necessary information to collect data in the environment to evaluate the accuracy of our IPS.

3.4.2 Experimental Methodology

To measure the system accuracy, we performed an intensive and time-consuming work to collect the RSSIs at the 150 test points shown in Figure 18. To diversify the RSSI behavior, we used 11 different beacons to collect signals at all test points. Beacon nodes are the mobile devices whose positions will be estimated. They operate with a single, small, long-lasting battery. The premise of the hardware architecture was not to rely on the Wi-Fi infrastructure of the building. Therefore, we use Bluetooth Low Energy (BLE). The anchor nodes are sniffers that monitor the network and capture the BLE advertising packets from the beacons and then send the informations to the server via a 900 MHz long range communication. Finally, the server builds the model and contains information on the coordinates of the anchor nodes and the scenario map. Therefore, our experimental methodology aims to be a passive location system, in which we do not require active participation from users nor the need for installing or running any software on the client's side.

At each test point, we performed a data collection, storing 100 RSSI measurements among the beacons and the anchor nodes. In general, 15.000 signal samples were stored for performance evaluation. To collect the data in the correct position, we use a tape-measure to correctly space the test points in each classroom, and we use our collection application to mark the current position on the map, automatically storing the real position of the test point, according to the previously established coordinate system.

It is important to highlight that our solution does not require this exhaustive step of data collection, and we perform it only to evaluate the performance of our solution in a real scenario. Thus, for each of the signal samples stored, we computed the device mobile estimated position and used the Root Mean Square Error (RMSE) as an evaluation metric. The RMSE calculates the positioning error using the estimated position by our model and the real position where the measurement was taken.

3.4.3 Signal Propagation Model Parameters

Signal propagation models represent the signal behavior in indoor environments. However, this behavior is complex to model, especially if it is a medium or large-scale environment, because of shadowing, scattering, and multi-path fading (Jung et al., 2011). Thus, correctly choosing the parameters values in the signal propagation model is important for the model to represent the signal behavior in the different regions of the scenario. In this way, the mobile device positioning estimation will be more accurate when the values chosen for the parameters of the signal propagation model represent the signal path loss in the environment. Thus, to map the RSSI to a distance between the mobile device and the anchor nodes depends strongly on the path loss exponent.

However, to find out which are the best parameter values for each region would require the need to collect data at all test points, which is unfeasible for certain locations that have time and people restrictions to collect the data. To avoid this exhausting step for collecting data, we used a set of 20 different values for the main parameters of the log-distance (PL_{d0} and η), as mentioned in Table 10, which are commonly found in real-world scenarios.

Table 10 – Possible combinations of values for the log-distance parameters.

Model Parameters		**Possible Combinations**
$PL0_{d0}$	η	
{-40}	{5.0}	1
{-40,-45}	{4.0, 4.5, 5.0}	6
{-40,-45,-50}	{3.5, 4.0, 4.5, 5.5}	12
{-40,-45,-50,-55}	{3.5, 4.0, 4.5, 5.0, 5.5}	20

In small scenarios with Line-of-Sight (LoS) up to $150\ m^2$, it is common for log-distance to be used with fixed parameters to represent the signal throughout the environment, as shown in the first line of Table 10. Thus, estimating the mobile device position using the conventional method degrades the accuracy as the scenario increases in size, as more regions have different signal behavior. On the other hand, we used a set of 20 different values for the main parameters of the log-distance, with common values to be found in IPS (Röbesaat et al., 2017; Sadowski and Spachos, 2018; Shchekotov, 2015; Shi et al., 2020), in such a way that eventually a combination of values will be better than the others to represent different regions. Thus, given the values that each parameter can take, the combination of these parameters can lead to a different estimated position. Therefore, we use the values in the last row of Table 10 to generate different distance estimates among devices and combine them all to generate a single position that is more representative of the real position.

3.5 Performance Evaluation

In this section, we present our performance evaluation through experiments in a real indoor environment. First, we evaluated the effect of different parameter values on performance. Next, we show the impact of anchor node choices. Finally, we compared the performance of our ADAM Positioning System to the traditional model-based IPSs found in the literature.

3.5.1 Parameters Evaluation

Because of the differences in signal behavior at different regions, choosing the best parameter values that characterize the signals and allowing the estimation of distances among the devices is an exhausting task and mainly impractical depending on the environment. Therefore, the key point of our solution is to use several log-distance parameter values. In this way, we evaluated the parameter values used in our approach, compared to the best individual values empirically found that characterize the environment.

Based on our experiments, we have the real information from where measurements were taken and we know in advance the fixed position of the anchor nodes, so we can get the real distance between all test points and the anchor nodes. Based on this information, we can obtain the best parameter values of the propagation model (PL_{d0} and η) to map RSSIs in distance. This step was followed only to find the best parameter values and compare them with our solution.

For the evaluation, we define a range from -40 to -60 for the parameter PL_{d0}, varying 5 dBm and a range from 3.0 to 6.0 for the parameter η. We can see in Table 11 that when we use individual values in each parameter to model the signal behavior in the whole scenario, the average error is 3.60 m. In addition, we verified which individual parameters resulted in the lowest average error per room.

Table 11 – Comparison of different approachs.

Rooms	**Fix Params** [-55,4.2]	**Bests** [PL_{d0},η]	**ADAM**
Room 1	3.72 m	2.20 m [-45, 5.5]	2.43 m
Room 2	3.90 m	1.70 m [-55, 3.8]	2.50 m
Room 3	4.02 m	2.68 m [-60, 3.3]	2.69 m
Room 4	3.59 m	3.12 m [-55, 4.4]	3.14 m
Room 5	3.54 m	3.00 m [-55, 4.0]	3.01 m
Room 6	4.29 m	3.20 m [-40, 5.0]	3.48 m
Room 7	2.75 m	2.84 m [-50, 4.0]	2.88 m
Room 8	3.07 m	1.98 m [-40, 4.6]	2.30 m
Room 9	2.60 m	3.24 m [-45, 4.4]	2.88 m
Room 10	3.23 m	3.80 m [-45, 4.6]	3.94 m
Room 11	3.20 m	3.20 m [-40, 4.8]	2.84 m
Hall 1	4.49 m	4.06 m [-40, 4.8]	3.41 m
Hall 2	3.34 m	2.79 m [-45, 5.0]	2.91 m
Hall 3	5.69 m	2.02 m [-55, 4.6]	1.93 m
Mean	*3.60 m*	*2.84 m*	*2.93 m*

In the third column of Table 11, we can see that different rooms resulted in different parameter values that allowed them to get the lowest average error. It is important to note that it was only possible to find these individual fixed values because we performed data collection on the entire scenario, as described in 3.4.2. On the other hand, our solution uses 20 different values combinations for the log-distance parameters to represent all environments, thus allowing us to estimate 20 different positions for the same measurement, combining them all in a single and more accurate position using data fusion.

We can observe in the last column of Table 11, that by using our parameters set PL_{d0} and η, it was possible to obtain an average error of 2.93 m, without knowing previously what are the best values by classroom. Despite the average error being 0.09 cm higher than that obtained when we used the best parameters found in each classroom, as seen in the third column of Table 11, we eliminate the need to collect signal samples at test points to find the individual values to parameters. Nevertheless, in some classrooms like room 3, room 4, room 5, room 11 and hallways, the average error of our solution was equal to or less than the approach using only the best individual values of PL_{d0} and η found based on our experiment, showing that using a range of values for the signal propagation model parameters is a good alternative to avoid the cost of finding the best parameters values per room.

3.5.2 Choosing the Best Anchor Nodes

The main information we have to estimate the mobile device position is the RSSI measured between the device and the anchor nodes. However, using only RSSI among all nearby anchors is not advisable for a good location, as depending on the organization, some anchors can affect the positioning estimate. During the experiments, we noticed that when the anchors' coordinates with RSSI for the device are collinear, the positioning accuracy decreases, causing considerably larger errors. To understand the impact of the anchors' organization on the system accuracy, we used the collinearity filter to get the linearity coefficient of the anchor nodes.

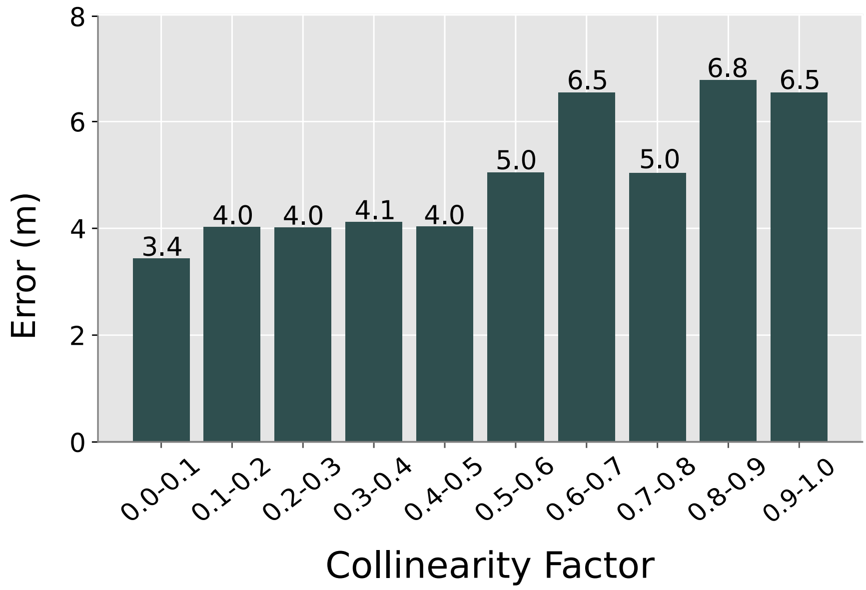

Figure 19 – Average error when varying the set of anchor nodes. The value 0 means the coordinates are far from collinear and 1 means collinear organization.

Figure 19 shows that for anchor nodes with a collinearity coefficient between 0 and 0.5, i.e., non-collinear, the average error was up to 4 m. On the other hand, when the coefficient is greater than 0.5, i.e., more collinear, the mean error increases up to

6.8 m. Therefore, we used the threshold of up to 0.5 as a filter to choose the anchors with the best organization to be used in the positioning computation.

3.5.3 Comparison with Other Solutions

In this section, we perform a comparative analysis of our approach and three different model-based IPS. As our proposed solution, these variations use log-distance to model the signal behavior. However, they use fixed values for the model parameters. For comparison, all approaches use the same database, with RSSIs collected in the same scenario and we confirm that the fixed values used were the best possible and resulted in the smallest errors. Approaches evaluated are:

1. Using 3 anchors with the highest RSSI values
2. Using 4 anchors with the highest RSSI values
3. Using all anchors

The above approaches are commonly found in the literature (Cantón Paterna et al., 2017; Fang et al., 2015; Huang et al., 2019; Wang et al., 2013), varying the number of nodes considered in the positioning computation to improve the model accuracy.

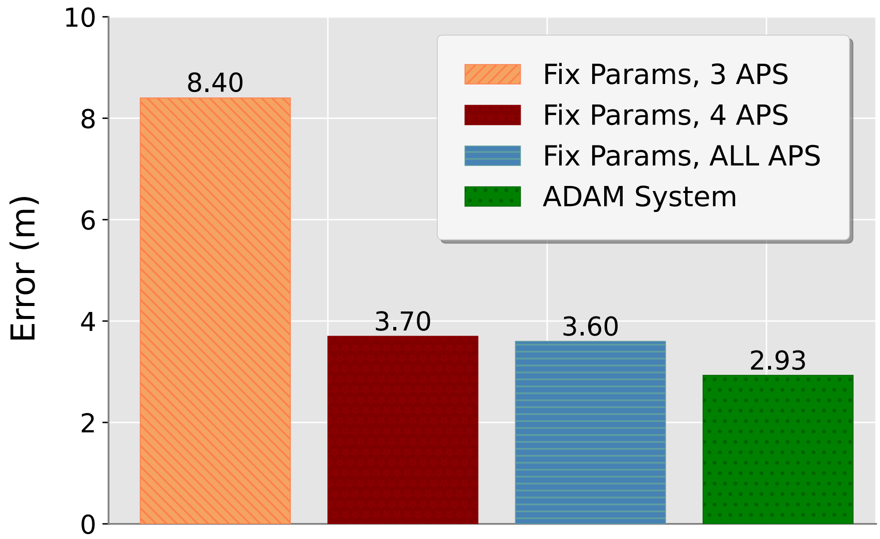

Figure 20 – Using the 3 anchor nodes with the strongest RSSI and fixed log-distance parameters is the worst approach, while our solution was the one that achieved the lowest average error, resulting in 2.93 m.

Figure 20 shows that our solution had the lowest average error compared to all approaches, resulting in an average error of 2.93 m, 23% smaller than the second-best solution that uses all anchor nodes and log-distance with fixed values for $PL0$ and η. Still, in Figure 20, we can see that the approach that uses the three anchors with the strongest RSSI results in the worst average error, reaching 8.40 m. This is mainly because of regions that are covered by three nearby anchor nodes with collinear organization. Thus, when we add one more anchor node in the position computation, the probability of a collinear organization decreases, consequently decreasing the error to 3.70 m, which still is 26% higher than our approach. The larger average errors occur because the other approaches are more vulnerable to dynamic environmental factors.

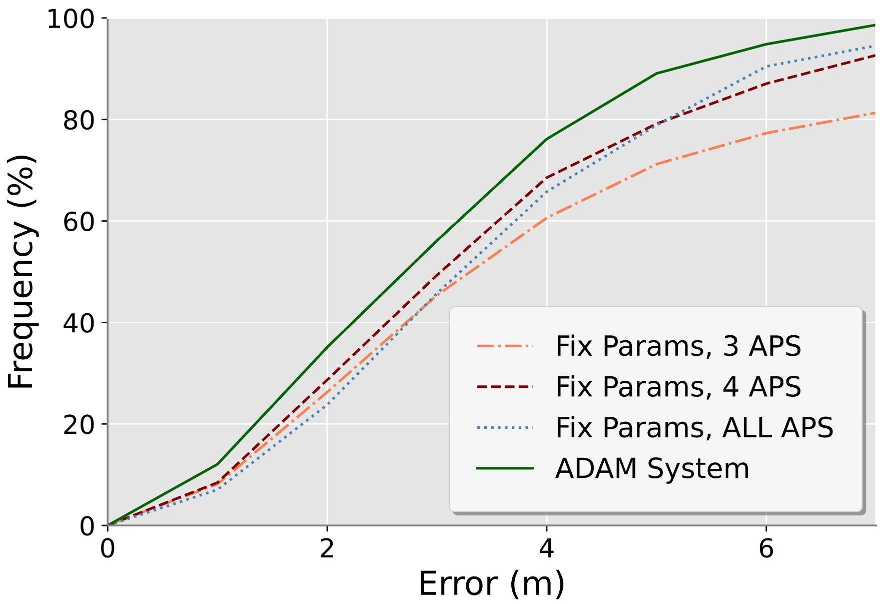

Figure 21 – Cumulative error among approaches. Our solution has the fastest growing curve, showing that most data have small errors.

From the results measured in Figure 21, the green curve of our solution grew faster, showing that our system achieves significantly better accuracy than the other approaches, resulting in a higher data frequency with smaller errors when compared to other algorithms.

In Figure 22, we see that nearly 73% of the collected samples contain an average error between $0-4$ m when localized using our solution, which is a remarkably excellent result when compared to the other approaches. Furthermore, all samples found with our model have an average error smaller than 8 m, unlike the solution that uses the 3 closest

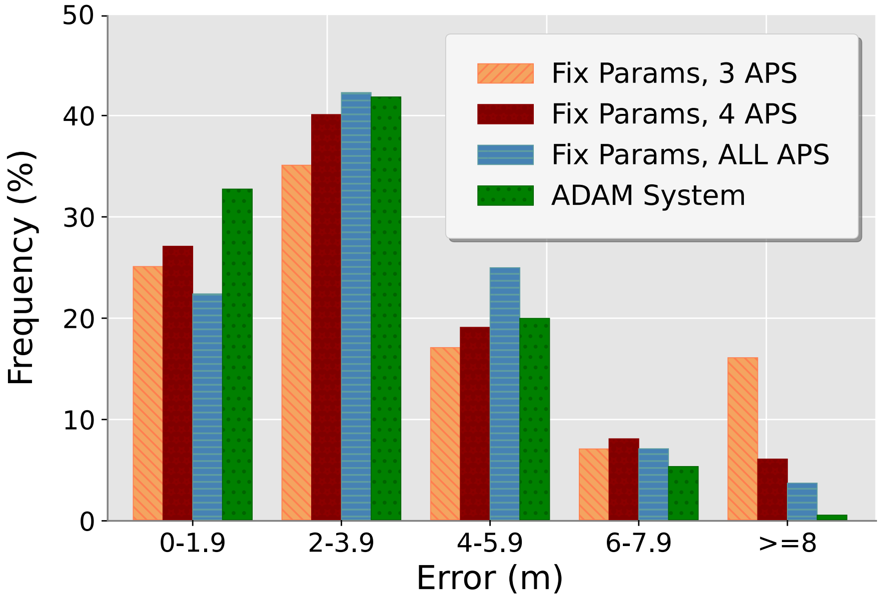

Figure 22 – Error distribution among approaches. Our solution has 73% of samples with positioning errors less than 4 m.

anchor nodes, which has 15% of the samples with an average error above 8 m, followed by the second solution that uses 4 anchors, which has 5% of the samples. Finally, the last solution with all the anchor nodes but also with fixed parameters resulted in 3% of the samples with errors greater than 8 m. As mentioned before, some regions are difficult to locate and especially do not allow choosing the anchor nodes used in the positioning computation. This is mainly because of the coverage of a few anchors in these regions, causing an increase in the average location error. A solution for these cases is to increase the anchors' coverage in the scenario. Therefore, as expected, our proposal presents greater reliability in the mobile device positioning and surpasses the compared approaches that use fixed values for the log-distance parameters and that do not consider the anchors' organization.

3.6 Discussion

In indoor positioning systems, the estimated position of the mobile device can be used in two different ways: ubiquitous and client-server. In the client-server approach, the mobile device is equipped with a graphical interface and uses a location-based application to show the users their locations on a map through requests to the server. The server returns the estimated location based on the RSSIs between the mobile device and the anchor nodes, which the application displays on the device screen. On the other hand, in the ubiquitous approach, the mobile device may take the form of wristbands or BLE tags that are configured to continually transmit BLE packets.

Being simple devices without graphical interfaces, this type of approach does not allow the mobile device to receive its location. We chose this approach to allow the administration of a school to monitor the times that students enter and exit the school, the number of students present during classes, and the areas within the school where students are located. It is important to note that each approach presents its benefits and drawbacks, and the choice of which to employ is based on the intended use of the positioning system. However, it is noteworthy that our proposed method can be used in both of the above-mentioned approaches since the main aim of this work is to innovate the position computation process to reduce the average error of the estimations.

To minimize the training cost, our solution requires a minimal effort to get floor plan information and prior knowledge of the positions of anchor nodes. Despite this requirement, the mobile devices can be accurately located and displayed on the same map of the area, which serves as the floor plan.

Some technical challenges need to be addressed. In our experiments, we used a homogeneous set of 11 devices from the same manufacturer. However, when using mobile devices with dissimilar hardware, such as various smartphones, variations in signal behavior may be observed. We hypothesize that the utilization of multiple propagation parameters with data fusion can improve performance when compared to traditional fingerprint-based IPSs. However, this aspect was not evaluated in this study and will be the subject of future research.

Additionally, our experiments were limited to 2D environments. In more complex environments, such as multi-level buildings, the log-distance model would need to be extended to account for the increased signal loss from floors and ceilings. Last, although we employed the log-distance propagation model in this study, our solution is adaptable to other propagation models.

3.7 Conclusion

In this chapter, we proposed a new model-based IPS, the ADAM Positioning System, which introduces a method for mapping the RSSI to the approximate distance between the anchor nodes and the mobile device. We present two significant contributions to the area. First, an algorithm to determine the anchor nodes' collinearity to reduce the impact of the nodes' organization and choose the anchors that benefit the positioning computation. Our second contribution is the use of different values for the log-distance model ($PL0$ and η) that adapt to the environment, in order to get different distances among the devices, which can be combined using data fusion and resulting in a single, more accurate estimation of the mobile device position.

Our proposed solution provides a significant improvement over IPSs that use fixed values to model the signal across the entire scenario. Also, our solution does not require any training effort like other IPS solutions. For the performance evaluation, we conducted all experiments in a real-world, large-scale scenario, with an infrastructure composed of BLE devices. Our results show that using the proposed ADAM Positioning System, the average error was 2.93 m, which is 23% better when compared to the conventional approach that maintains a single value in the log-distance parameters. In addition, we offer a good location estimate with minimal labor costs.

For future work, we aim at investigating the regions that got higher errors by exploring better parameter combinations. Furthermore, although the solution uses the log-distance model, we intend to evaluate the use of other signal propagation models.

4 A Model-based BLE Indoor Positioning System using Particle Swarm Optimization

In this chapter, we propose a new indoor positioning system that combines Particle Swarm Optimization (PSO) with signal propagation models to improve the accuracy of mobile device positioning. The PSO algorithm is used to optimize the position estimation process by generating different particles in the map, while the signal propagation model is used to model the attenuation and reflection of wireless signals in each particle. Our MIPS-PSO system does not require any prior training nor any knowledge of the best parameters of the signal propagation model. We evaluated the performance of our system using data collected in a real indoor environment with Bluetooth-Low-Energy (BLE) devices. Our results show that the MIPS-PSO achieves an average error of 2.57 m, an improvement of 40% when compared to a traditional trilateration, model-based IPS.

4.1 Introduction

With the proliferation of mobile computing and the Internet of Things (IoT), Location-based Services (LBS) have become increasingly popular. The most widely used technology in positioning applications is the Global Navigation Satellite System (GNSS), which includes the Global Positioning System (GPS). However, in complex indoor environments or areas with many buildings, where there is no direct line of sight between the mobile device and the satellite, the accuracy of GNSS is weakened (Chen and Zou, 2017).

In response to this challenge, Indoor Positioning Systems (IPSs) have been developed to address the need for accurate positioning in indoor environments such as parking lots, hospitals, museums, and schools where GNSS is inadequate.

Indoor positioning systems use wireless communication technologies, which vary in terms of their range, noise interference, and availability on different hardware types. Among the technologies that use wireless signals, the most popular ones are Radio Frequency Identification (RFID) (Wang et al., 2017), Acoustic Chirp (Gabbrielli et al., 2023), Bluetooth-Low-Energy (BLE) (Sadowski and Spachos, 2018), Ultra-wideband (UWB) (Cai et al., 2018), and Wi-Fi (Chen and Zou, 2017). BLE has gained significant popularity in indoor positioning research owing to its extensive range, affordability, and widespread availability in mobile devices, particularly those with power limitations (Sadowski and Spachos, 2018). For these reasons, we chose to use this technology in our work.

Within the context of BLE-based IPSs, our categorization involves distinguishing between fingerprint-based and model-based approaches. The inputs to model-based IPSs are usually Time of Arrival (ToA) (He et al., 2012), Time Difference of Arrival (TDoA) (Xiong et al., 2023), Angle of Arrival (AoA) (Fascista et al., 2017). ToA and TDoA require precise synchronization between communication devices, while AoA requires specialized antenna arrays resulting in high equipment costs. On the other hand, the inputs to fingerprint-based IPSs are usually RSSI, which is also used in model-based IPS. RSSI is the most common solution due to its simplicity and availability in various wireless devices, eliminating the need for expensive hardware. However, RSSI is susceptible to environmental interference, affecting accuracy in indoor environments with many obstacles. Therefore, RSSI-based methods require techniques to minimize the RSSI variations. Today, it is still an open challenge to efficiently explore all measurements to improve user position estimation accuracy.

Fingerprint-based IPSs have two distinct phases to estimate the device position, the offline phase, and the online phase. During the offline phase, the environment is mapped into various Reference Points (RPs), and signal samples are collected at each RP to create a database containing a fingerprint map. In the online phase, a machine learning algorithm is used to estimate the device's position based on the data collected in the offline phase. However, this approach requires significant time for offline data collection, making the method impractical for large-scale scenarios. Additionally, the accuracy of this technique is affected by changes in the environment that may alter signal behavior (He et al., 2012).

In contrast to fingerprint-based IPSs, model-based IPSs do not require extensive data collection at RPs. These systems rely on information from fixed nodes in the environment, called anchor nodes, and utilize a signal propagation model to characterize signal behavior in the location environment. This model directly depends on the RSSI values measured in the environment (Li et al., 2018). Location computation in model-based IPSs is carried out by optimization algorithms that use the signal propagation model to map RSSI to the distance between the devices involved in communication. However, they do require some model parameter values that are usually obtained by caring out some real-world experiments.

Reviewing the existing literature, some studies rely on deterministic techniques, like fingerprints, for consistent positioning. Nevertheless, it is important to highlight the advantages of using stochastic methods (Caceres Najarro et al., 2020; Jean and Weiss, 2014; Xiong et al., 2023; Xiong and So, 2023), especially in navigating the intricacies of system implementation. Stochastic approaches offer optimization solutions, particularly when deterministic methods pose challenges due to high implementation costs. Bearing these factors in mind, our choice is to embrace a stochastic approach in formulating our proposed solution.

This chapter presents a novel Model-based Indoor Positioning System (MIPS) that uses Particle Swarm Optimization (PSO) to locate mobile devices accurately, called MIPS-PSO. Firstly, information about the IDs and coordinates of all fixed anchor nodes in the environment is stored. Next, the PSO algorithm is applied to get the closest position to the target. To do this, different particles are generated at random positions in the scenario. Each particle uses the signal propagation model with different model parameters to get an RSSI vector, based on the distance between the particle and the anchor nodes. The cost function is then used to select the particle that has the most similar signals to the mobile device's signals by comparing the generated RSSI vectors with the measured RSSIs in the real environment. The parameters of all other particles are updated according to the best particle from the current iteration and, then, change their positions based on these new parameters. These steps are repeated until all particles are quickly converged to the best-estimated position for the mobile device. This solution was implemented and tested in a real-world environment, and the results were compared to different variations of model-based IPS. Our results clearly show that the proposed solution can locate mobile devices with an error of 2.57 m, which satisfies the requirements in most real-world indoor applications.

Our main contributions are summarized as follows:

1. A hybrid approach is introduced, which combines particle swarm optimization and the log-distance signal propagation model to enable the implementation of indoor positioning systems in scenarios where data collection is challenging or even unfeasible.

2. The log-distance model, when employed with diverse parameter values, facilitates an enhancement in positioning accuracy, thereby obviating the necessity for the exhausted collection of real-world data.

3. The adaptive method, employing particle swarm optimization, integrates multiple particles, each exhibiting distinct BLE wireless signal behaviors. This method allows for the characterization of signals across various regions within the scenario, identifying the particle's position with signal characteristics more akin to those detected by the mobile device. Consequently, this method more precisely estimates the position of the target device.

4. Experiment results conducted within a real-world environment demonstrate the superior accuracy of our approach in comparison to classical techniques, which directly infer distance and employ fixed parameters for the propagation model.

The rest of the Chapter is organized as follows. In the section 4.2, we show the related work. Section 4.3 introduces our proposed system model. Section 4.4 shows our real-world testbed and experimentation methodology. The obtained results are then shown in Section 4.5. Finally, Section 4.6 presents our conclusions and future work.

4.2 Related Work

Currently, there are several techniques and algorithms proposed in the IPS literature aiming at making systems more accurate and robust. Among the algorithms used, we can find several works that use Particle Swarm Optimization (PSO) in indoor positioning systems. PSO is an algorithm inspired by the social behavior of animals seeking food (Chen and Zou, 2017). In this algorithm, animals are treated as particles that share their experiences and information so that the whole group can move toward the best solution.

When using RSSI as the main source of information, techniques can be classified into two groups: those that are model-based and those that are fingerprint-based.

To estimate the location of a mobile device, fingerprinting techniques use algorithms such as K-Nearest Neighbors (KNN) (Bahl and Padmanabhan, 2000), Support Vector Machine (SVM) (Zheng et al., 2022), Random Forest (Guo et al., 2018), and Neural Networks (NN) (Cheng et al., 2020). This technique compares the RSSI of the mobile device with the stored fingerprints in a database to determine its position. The main disadvantage of this technique is being easily affected by environmental changes, requiring the creation of a new database whenever a change occurs in the scenario that modifies the RSSI behavior. RADAR (Bahl and Padmanabhan, 2000) was one of the first IPS systems to use fingerprinting with the KNN algorithm, measuring the similarity of fingerprints through the Euclidean distance of samples.

Li et al. (2016a), introduce an IPS that utilizes the Affinity Propagation (AP) clustering algorithm and an optimized Artificial Neural Network (ANN) achieved through particle swarm optimization. The PSO-ANN algorithm, known for its efficiency in both training and prediction, aims to significantly reduce processing time in both offline training and online localization phases. Similarly, in Zheng et al. (2022), a robust location model is employed, leveraging the swarm optimization algorithm to determine the optimal location estimate. Furthermore, in Cheng et al. (2020), the authors tackle the internal positioning problem using an ANN and optimize its parameters with a particle swarm intelligence algorithm.

In Shan et al. (2020), the researchers utilized the nearest algorithm to pinpoint the initial region of the mobile device. Subsequently, they applied particle swarm optimization iteratively to get the optimal location within that approximate region. Conversely, in Shih and Liang (2018), the authors propose various innovative approaches employing machine learning algorithms and indexing methods to enhance indoor positioning accuracy. These approaches include a modified swarm algorithm and genetic algorithms. Lastly, in the work detailed in Wang et al. (2017), the authors devised an Improved PSO algorithm (IMPSO) for determining optimal connection weights and significantly optimizing the parameters of a Feedforward Neural Network (IMPSO-FNN).

On the other hand, there are model-based IPSs, which use signal propagation models to represent the signal behavior and eliminate the need for extensive training in the environment to collect signal samples (fingerprints). The most common solutions are based on trilateration, which uses three anchor nodes as a reference. Therefore, the signals with higher intensity for the three anchors are used in the signal propagation model to map the distances between devices, forming three circles. When more than three nodes are used, the process is called multilateration. The calculation of the mobile device's position is performed through algorithms such as Least Square (LS) or Maximum Likelihood Estimation (MLE), which minimizes the distance in relation to all circles and allows the estimation of the approximate position of the mobile device.

The study conducted by Sadowski and Spachos (2018) compares the performance of various technologies used in model-based IPSs and demonstrates that Wi-Fi and BLE have advantages over other technologies, mainly in terms of communication range and energy consumption. In Yang et al. (2020), the authors conducted experiments to determine the best value of the path loss exponent of the log-distance model, based on the RSSIs between all anchor nodes, which allowed for better mapping of RSSI to distance. However, this approach requires a significant effort to measure the exponent of all anchor nodes. To reduce the experimental costs to determine the set of fixed values for the signal propagation model, the authors in the paper Assayag et al. (2023) proposed a model-based IPS with dynamic parameters for the signal propagation model, resulting in a decrease in the average positioning error with the least squares algorithm. In Guo et al. (2019), the authors employ particle swarm optimization with KNN to determine the optimal set of anchor nodes for use in the positioning calculation. In Kuang et al. (2018), the authors introduce two hybrid 3D positioning algorithms. Initially, they employ the classical least squares algorithm to obtain approximate estimates of the geographic position of a destination node. Subsequently, particle swarm optimization is employed to refine these initial estimates. Similarly, in Xia et al. (2021), the authors initiate the positioning process by utilizing the maximum likelihood estimation. Following this initial positioning, all range information is integrated into the PSO to calculate a more precise location.

In Cai et al. (2018), the authors introduce the Ensemble Learning Particle Swarm Optimization (ELPSO) algorithm designed for real-time indoor localization using ultra-wideband technology. The feasibility of ELPSO is showcased through its application in a 2D and 3D UWB indoor localization system, revealing promising outcomes. Additionally, in Guo et al. (2019), the PSO algorithm is employed for simulating parameter estimation in indoor settings. This application involves fitting the signal attenuation curve, effectively eliminating low-quality experimental data and yielding a model that accurately matches the signal attenuation curve. In Li et al. (2018), the researchers use a Particle Swarm Optimization with Neural Network to train the model for estimating distances using RSSIs, ultimately minimizing positioning errors. Similarly, in another work Chen and Zou (2017), the authors put forward an enhanced Wi-Fi indoor positioning approach. This method incorporates an improved unlicensed Kalman filter and utilizes particle swarm optimization to minimize measurement errors and enhance the overall accuracy of positioning.

4.2.1 Discussion

Our proposed solution differs from all the previously mentioned works in several ways. Firstly, we completely eliminate the need for collecting data in the environment, instead, we only require prior knowledge of the positions where anchor nodes are fixed. Additionally, unlike model-based solutions with fixed parameters, we use the signal propagation model with different parameter values, allowing us to model the signal in different regions of the scenario. Lastly, we propose a new method based on PSO that uses the signal propagation model to obtain the best particle closest to the actual position of the mobile device. In summary, we innovate by exploring the use of the signal propagation model together with PSO to estimate the position of the mobile device with higher accuracy. Further details on our solution will be presented in the Section 4.3.

4.3 MIPS PSO Architecture

In this section, we present the architecture of our IPS-PSO. The system can be divided into two phases, as shown in Figure 23.

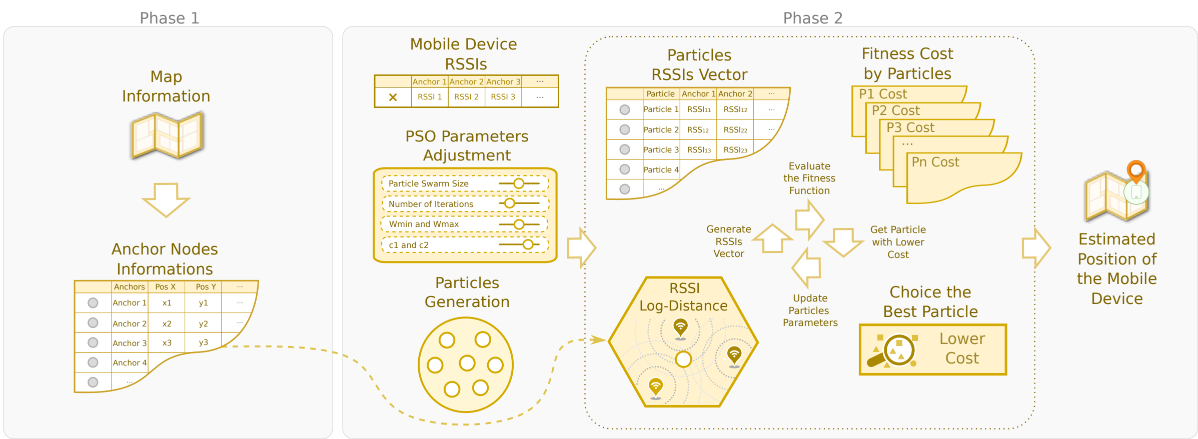

Figure 23 – The framework of the MIPS-PSO based on the positioning system.

During the first phase, we use a scenario map to record the identification and coordinates of anchor nodes. These are the only information previously needed for our solution, and will be used in the next phase. In the second phase, PSO is implemented to produce a group of particles in the scenario, each with different parameter values in the log-distance signal propagation model. The PSO algorithm executes in rounds, where in each round, the particle with the best global performance is chosen as a reference, and all other particles walk towards the best. After a predefined number of executions, the mobile device position is estimated as the same position as the current best-particle. Based on the MIPS-PSO architecture system in Figure 23, in summary, the key steps of our approach to estimate the positioning of the mobile device include:

Phase 1 - Floor Plan Information:

1. Fix the anchor nodes (n) in the scenario (walls or ceiling);
2. Get the scenario floor plan and create a $2D$ virtual map;
3. Assuming that the locations of the anchor nodes are denoted by $\boldsymbol{a}_i = [x_i, y_i]$, $i = 1, 2, 3, ..., n.$, store the anchor nodes positions in a database.

Phase 2 - Mobile Device Position:

1. Get RSSI vector measured from a mobile device to anchor nodes: $\boldsymbol{D} = [RSSI_1, RSSI_2, RSSI_3, ..., RSSI_n]$.
2. Configure the range (min, max) of the particle parameters (x, y, PL_{d0}, and η) and initialize the PSO with p particles randomly in the scenario.
3. Use the signal propagation model ($RSSI_p^n = PL_{d0} - 10\eta \log_{10} \frac{d}{d_0}$), to get RSSI vector of the each particle, where $RSSI_p^n$ is the RSSI of the particle $\boldsymbol{p}$ to anchor node n. The step (3) of the Phase 1 is used to calculate the distance d between the particle position and the n-th anchor node position.
4. Apply the fitness function to identify the similarity between the RSSI vectors of the particles and the RSSI vector of the mobile device: $\boldsymbol{F}(\hat{\boldsymbol{x}}_p) = \sum_{i=1}^{n} \sqrt{(RSSI_{\boldsymbol{p}i} - RSSI_{\boldsymbol{D}i})^2}$.
5. Select the particle with the most similar RSSIs to the mobile device among all particles $argmin\ [\boldsymbol{F}(\hat{\boldsymbol{x}})]$, also known as $\boldsymbol{g}$best.
6. Update the positions and parameters of all particles toward the $\boldsymbol{g}$best particle. The iteration process to move the particles towards the optimal solution at each time

k is $\boldsymbol{V}_i(k+1) = w\boldsymbol{V}_i(k) + c_1 r_1[\boldsymbol{b}_i(k) - \boldsymbol{p}_i(k)] + c_2 r_2[\boldsymbol{b}_g(k) - \boldsymbol{p}_i(k)]$, where $\boldsymbol{V}i(k)$ represents the velocity of a particle $\boldsymbol{p}$.

7. Repeat the steps 3, 4, 5 and 6 for k iterations to approximate the estimated position to the real position. When the PSO algorithm iterates a sufficient number of times, the position of the best particle from the last iteration is used to estimate the final position of the mobile device: $\boldsymbol{Final\ Position} = \boldsymbol{g}best_{(x,y)}$.

The specific procedure is summarized in Algorithm 1:

Algorithm 1 Algorithm for positioning

Require:
initialize constants PL_{d0}, η, w_{min}, w_{max}, c_1, c_2, particle size and, number of generations;
Ensure:
randomly initialize the particle positions $p_{(x,y)}$;
randomly initialize the particle properties PL_{d0} and η within the limited range;
randomly initialize the particle velocities v_i;
for each generation **do**
  for each particle p in the swarm **do**
    compute the RSSI values using Equation 4.1;
    compute the fitness value using Equation 4.2;
    if $fit_{(i)} \leq fit_{(best)}$ **then**
      best particle = p_i;
    end if
  end for
  update parameters using Equations 4.3, 4.4, and 4.5;
end for
estimated position of the target device based on position of the best particle;

The Sections 4.3.2 and 4.3.1 provide more details of each step.

4.3.1 Phase 1 - Floor Plan Information

In fingerprint-based IPS, it is common to require a workload to create a database with signal information at different reference points. However, this step requires a considerable amount of time, especially in medium to large-scale scenarios. On the other hand, our approach eliminates the need for signal collection in the environment and only relies on information about the scenario obtained from the building floor plan. In this phase, we assume that n anchor nodes are fixed in a $2D$ area have their coordinates ($\boldsymbol{a}_i = [x_i, y_i]$, $i = 1, 2, 3, ..., m$) previously known and ensure good signal coverage throughout the environment. Using the building floor plan, we store in a database the identifications and installation positions of all anchor nodes, as presented in Table 12.

Table 12 – Anchor node informations.

Anchor$_{id}$	**Position$_{(x,y)}$**
Anchor Node$_1$	[x_1, y_1]
Anchor Node$_2$	[x_2, y_2]
Anchor Node$_3$	[x_3, y_3]
Anchor Node$_4$	[x_4, y_4]
...	...
Anchor Node$_n$	[X$_n$,Y$_n$]

Storing this information is important because in the next phase, when we estimate the mobile device, we use the positions of all anchor nodes to map the RSSIs to distances using a signal propagation model. Therefore, all necessary information can be obtained in advance through the building floor plan. In this work, all coordinates used are in relation to the origin of the system $(x = 0, y = 0)$ declared as the lower left corner of the map, as illustrated in Figure 24, explained in Section 4.4.1.

4.3.2 Phase 2 - Mobile Device Position

This phase is responsible for estimating the mobile device position. The architecture of our system mainly involves two types of devices: one transmitter and several receivers. The mobile device acts as the signal transmitter, while the receivers are called anchor nodes. During this phase, when the mobile device sends a BLE packet, each packet includes a timestamp and mobile identification. Upon receipt by fixed anchor nodes in the environment, information such as RSSI is obtained, crucial for positioning computation and directly influencing the final result.

Due to the limitations of wireless signals, only the closest anchor nodes to the mobile device can receive the transmitted packets. Consequently, when an anchor node receives a packet, it promptly forwards it to a server that compiles all measurements from anchor nodes into a single vector using the mobile device identification and packet timestamps. When the server detects that an anchor node has not received the packet, the value is automatically replaced with -110, indicating a lack of communication. Once the RSSIs of mobile device have been collected, the server will compile this data into a vector $\boldsymbol{D} = [RSSI_1, RSSI_2, RSSI_3, ..., RSSI_n]$. The vector's size will be proportional to the number of anchor nodes positioned in the scenario.

Table 13 shows an example of an RSSI vector obtained by sending packets from a mobile device, where columns with abbreviation "AN" (Anchor Node) mean the RSSI values for the respective anchor nodes.

Table 13 – RSSI values get through a BLE packet.

Mobile Device	**Pkt**	**AN_1**	**AN_2**	**AN_3**	**AN_4**	**AN_5**
$Device_1$	1	−83	−62	−71	−110	−78

4.3.2.1 Particle Generation

Our solution uses the PSO algorithm, which is a powerful approach for solving non-linear optimization problems, and it is based on random populations that walk towards the best global solution of a system. In this step, we initialize the particle swarm size with p particles randomly positioned within the scenario using the PSO algorithm. The experiments aimed at determining the best number of generated particles are described in Section 4.5.1.

Table 14 – Particle Properties

$Particle_{id}$	**Properties**
$Particle_1$	([32.9, 3.70], -53, 4.2)
$Particle_2$	([18.9, 13.1], -57, 3.6)
$Particle_3$	([27.0, 10.6], -51, 3.8)
$Particle_4$	([18.0, 6.00], -60, 4.7)
...	...
$Particle_p$	([X_n,Y_n]), PL_{d0}, η)

Each particle represents a mobile device scattered in the environment, with its own RSSI vector representing the signal at its position. The goal is that in each iteration of the PSO algorithm, the particles move towards the probable position of the target, called the best global. To do this, we use the log-distance propagation model to generate the RSSI vector of each particle. Each particle has properties such as its position on the map (x, y), PL_{d0}, and η. The PL_{d0} and η parameters are part of the log-distance propagation model and are affected by environmental factors. According to common values to be found in IPS (Chen and Zou, 2017; Sadowski and Spachos, 2018; Shih and Liang, 2018), the PL_{d0} parameter varies between -50 and -60, while the η parameter varies between 3.5 and 5. In Table 14, it is possible to visualize an example of a particle database with different parameters. These parameters are used to generate the RSSI vectors, as we will explain in the next section.

4.3.2.2 Signal Propagation Model

As mentioned earlier, we use the log-distance signal propagation model to describe the characteristics of the BLE signal in different areas of the scenario. As a result, each particle will have a distinct RSSI vector, the same size as the mobile device's RSSI vector, which depends on the parameters used in the propagation model, as well as the distance between the particle's position and the anchor positions. The log-distance signal propagation model is represented by Equation 4.1.

$$RSSI = PL_{d0} - 10\eta \log_{10} \frac{d}{d_0} \tag{4.1}$$

where PL_{d0} represents the RSSI measured at a reference distance $d0$, while the parameter η is known as the path loss exponent, indicating the propagation of the signal in the environment. In turn, d represents the distance between the particle position and the position of an anchor node. The values of PL_{d0} and η are obtained from the properties of each particle, as can be seen in Table 14, while d is calculated using the Euclidean distance between the particle position and an anchor node position, based on the information stored in the first phase, as shown in Table 12.

The result of Equation 4.1 is the expected RSSI (dBm) between the particle and an anchor node. During the generation of the RSSI vector for a particle, depending on the parameters employed in the propagation model, the generated RSSI values may surpass -90. However, values exceeding -90, such as -95, -99, or -100, fail to provide an accurate representation of RSSI to distance mapping. To address this, we have opted to replace all RSSI values generated above -90 with the value -110, denoting that the particle failed to establish communication with the anchor node. Consequently, the highest value within the RSSI range in our system remains fixed at -90. Thus, depending on the parameters used in the signal propagation model, the particle closest to the real position of the mobile device tends to have RSSI values similar to the RSSI vector collected physically in the scenario, as described in Table 15.

Table 15 – RSSI vector of each particle

Particle$_{id}$	**AN$_1$**	**AN$_2$**	**AN$_3$**	**AN$_4$**	**AN$_5$**
Particle$_1$	-52	-110	-62	-89	-74
Particle$_2$	-68	-56	-110	-51	-85
Particle$_3$	-77	-65	-76	-110	-74
Particle$_4$	-59	-110	-86	-74	-68
...	...	...	...	...	...
Particle$_p$	RSSI$_1$	RSSI$_2$	RSSI$_3$	RSSI$_4$	RSSI$_5$

4.3.2.3 Fitness Function

To identify the particles that have the most similar RSSIs to the mobile device, it is necessary to apply an appropriate fitness function to the RSSI vectors of all particles. Because the particles are located at different positions on the map and have different parameters applied to the signal propagation model, the signal behavior at each position tends to be different, as can be seen in each row of Table 15.

In this step, our goal is to compute the discrepancy between the real RSSI vector, measured through the packet sent by the mobile device, and the particle RSSI vector generated using the log-distance model. The variation of the log-distance parameters in each particle serves to minimize the inevitable impact caused by interference that affects the RSSI. Thus, a particle's fitness function is computed using Equation 4.2.

$$\boldsymbol{F}(\hat{\boldsymbol{x}}) = \sum_{i=1}^{n} \sqrt{(RSSI_{\boldsymbol{P}i} - RSSI_{\boldsymbol{D}i})^2} \tag{4.2}$$

where $\hat{\boldsymbol{x}}$ stands for the RSSI vector of a particle, n represents the total number of anchor nodes, and i is the current anchor node used to assess the RSSI differences. $RSSI_{\boldsymbol{P}i}$

indicates the value of the RSSI measured by the particle for anchor node$_i$. Finally, $RSSI_{\boldsymbol{D}i}$ represents the value of the real RSSI measured by the mobile device for the same anchor node. Therefore, in case the parameters used are not representative enough, the RSSI will be significantly different, leading to a high value in the fitness function. Conversely, a low value indicates that the parameters used by the particle resemble the real-world behavior of the measured signal.

The objective of the fitness function is to assess the disparity between the RSSI of an individual particle and the RSSI of the mobile device. Its purpose is to identify the particles with an RSSI vector that closely aligns with the real RSSI values measured by the mobile device, thereby determining the most optimal particles. For each iteration of the algorithm, the particles keep tracking the position with minimum cost.

Table 16 – Fitness value of each particle

Particle$_{id}$	**AN$_1$**	**AN$_2$**	**AN$_3$**	**AN$_4$**	**AN$_5$**	**Fitness**
Particle$_1$	31.0	43.0	9.0	21.0	4.0	103.0
Particle$_2$	15.0	6.0	34.0	59.0	7.0	116.0
Particle$_3$	**6.0**	**3.0**	**5.0**	**0.0**	**4.0**	**18.0**
Particle$_4$	24.0	43.0	15.0	36.0	10.0	123.0
...	...	...	...	...	...	
Particle$_n$	Cost$_1$	Cost$_2$	Cost$_3$	Cost$_4$	Cost$_5$	...

Table 16 presents an example of the values generated by the fitness function between the RSSI vectors from Table 15 and the RSSI vector measured by the mobile device, as shown in Table 13. For instance, the RSSI measured by the mobile device for AN$_1$ is -83 dBm, while the RSSI of Particle$_3$ for the same anchor node is -77 dBm, resulting in a difference of 6 dBm. By performing this calculation for all anchor nodes, Particle$_3$ has a fitness value of 18, being the lowest among the illustrated particles and, therefore, the best global particle so far.

4.3.2.4 Position Estimation

In the previous step, we used the fitness function to select the particle with the most similar RSSI vector to the mobile device among all particles, also known as $\boldsymbol{g}best$. However, we cannot simply consider this particle as the estimated position of the mobile device because positions are randomly generated on the map, and the selected best particle may not be close enough to the real mobile device position. For this reason, we run several iterations of the algorithm to approximate the estimated position to the real position. In each iteration, the positions and parameters of all particles are updated toward the best particle from the previous iteration, according to an updated equation that determines the particles' velocity.

To perform the update process, the stored data is updated at each time k, updating the velocity and position of each particle in all dimensions. The iteration process to move the particles towards the optimal solution is a follow:

$$\begin{aligned}\boldsymbol{V}_i(k+1) = w\boldsymbol{V}_i(k) + c_1 r_1[\boldsymbol{b}_i(k) - \boldsymbol{p}_i(k)]+ \\ c_2 r_2[\boldsymbol{b}_g(k) - \boldsymbol{p}_i(k)]\end{aligned} \tag{4.3}$$

$$\boldsymbol{p}_i(k+1) = \boldsymbol{p}_i(k) + \boldsymbol{V}_i(k+1) \tag{4.4}$$

$$w = w_{max} - k\frac{w_{max} - w_{min}}{K} \tag{4.5}$$

where $\boldsymbol{V}_i(k)$ represents the velocity of a $\boldsymbol{p}_i$ during the k-th iteration, k denotes the current iteration number, and K is the total number of iterations; w falls within the range of w_{min} to w_{max} and is referred to as the inertia factor, $c1$ and $c2$ are acceleration coefficients that adjust the search rate. The choice of values for these parameters will be explored in Section 4.5.1. The parameters $r1$ and $r2$ are randomly generated from a normal distribution. Additionally, $\boldsymbol{b}_i$ denotes the position given the best fitness value of the i-th particle in the k-th iteration (best local), while $\boldsymbol{p}_i$ represents the particle parameter configuration in the current iteration i. Finally, $\boldsymbol{b}_g$ means the position of the particle among all the particles in the k-th iteration (best global, as known $gbest$).

Therefore, we computation the update of particle parameters. If a particle has a high cost, it means that it is far from the $\boldsymbol{g}best$, and thus its (x, y) coordinates need to be updated towards the position of the $\boldsymbol{g}best$ with high velocity. On the other hand, if a particle is close to the position of the $\boldsymbol{g}best$, its velocity is small, and the position is slightly modified. In addition to the position, the parameters PL_{d0} and η are also updated towards the parameters of the $\boldsymbol{g}best$. Thus, in each iteration, the RSSI vectors of the particles are altered, and the fitness function of each vector is modified, allowing new particles to be selected as $\boldsymbol{g}best$. When the PSO algorithm iterates a sufficient number of times, the fitness begins to stabilize and converges to the probable real position of the mobile device. Therefore, the position of the best particle from the last iteration is used to estimate the mobile device position, as shown in Equation 4.6.

$$\boldsymbol{Final\ Position} = \boldsymbol{g}best_{(x,y)} \tag{4.6}$$

4.4 Experimental Testbed

This section presents details of our real-world testbed and the method to collect data. We will discuss the performance evaluation in Section 4.5.

4.4.1 System Environment

We evaluated the real-world performance of our solution on the second floor of a school building covering an area of 720 m^2 (45 m by 16 m). The testing scenario consisted of 11 classrooms and 3 halls, each with a minimum of 1 fixed anchor node. The setup of the test is illustrated in Figure 24, which shows the anchor nodes' positions (in orange) and RPs (in gray) across the entire area.

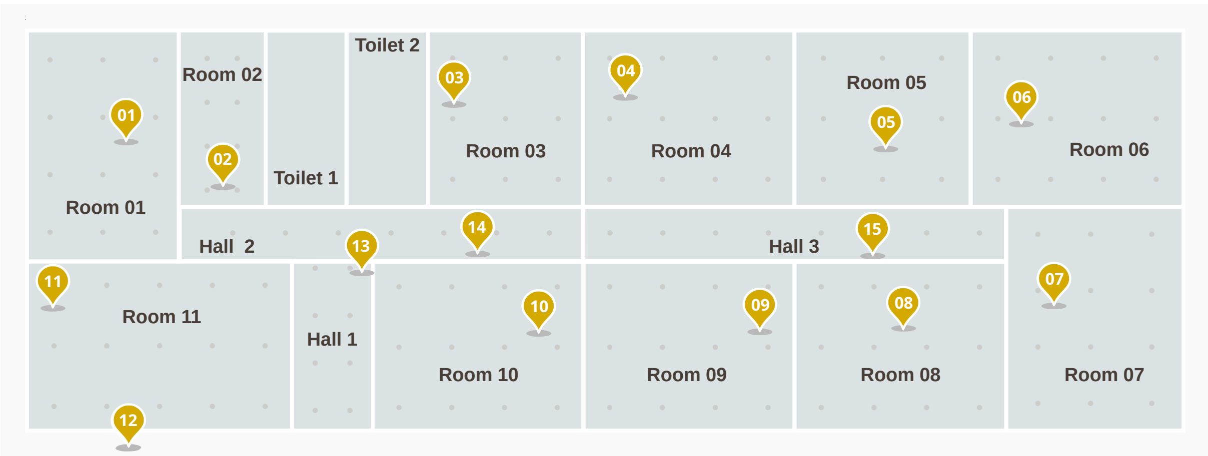

Figure 24 – Experimentation scenario composed of 15 anchors (in orange) fixed to the ceiling of the rooms and, 150 reference points (in gray dots) with a spacing of about 2 m in the rooms.

To cover the entire space, we placed roughly 150 RPs, with a spacing of about 2 m in the rooms. It is important to note that our method does not require pre-existing data and the reference points were placed solely to assess the system's performance.

A crucial piece of information for our solution is knowing the coordinates of the anchor nodes to perform the distance calculation. To do this, initially, we previously have the floor plan, where it is possible to obtain the area of the map. Consequently, we selected the lower-left corner of the map as the system's origin point ($x = 0$, $y = 0$) and utilized graphic software, such as Inkscape, to create an image that accurately represents the spatial layout of the rooms (as shown in Figure 24). In our scenario, the anchor nodes were strategically positioned in locations that offered the most convenient power connectivity. As a result, their coordinates were derived from the floor plan of the scenario. The coordinates for each anchor node are stored within the server, organized in a structure akin to the one depicted in Table 12. This information is used to generate the RSSI vector of the particles through the signal propagation model.

4.4.2 Experimental Methodology

To evaluate the accuracy of our system, we developed a database with 15.000 signal samples, collecting 100 samples at each RP. During the testing, we used 11 different mobile devices to capture various RSSI behaviors in the database, where all devices are equipped with similar hardware. Despite the similar hardware, the RSSI behavior varies because each BLE board possesses its unique characteristics.

All of the mobile devices we used were based on BLE technology and operated on small batteries that allow for long-lasting BLE packet transmission. Additionally, the anchor nodes in the scenario also communicate with the mobile devices through BLE.

Signal samples are gathered from all directions (0^o, 90^o, 180^o, and 270^o) using mobile devices situated in various positions, such as worn as a wrist bracelet, stowed in front and back pockets, and placed in different orientations. The various arrangements contribute to the diversity in RSSI, and all these factors are considered when computing the average system error.

RSSI is vulnerable to interference, fading, and shadowing. To mitigate the effects of environmental shading, we have employed the approach of computing the arithmetic mean during the collection of RSSI values from the mobile device. This involved setting a transmission power interval for BLE advertising packets to 0.1 seconds. By averaging 10 values, our aim is to decrease the volatility of the environment and the system itself, thereby minimizing the impact of interference peaks. Consequently, the user's position is updated every 1 second, in alignment with typical human walking patterns.

After receiving a BLE packet, the anchor nodes transmit it to the server via long-range $900\ MHz$ communication. This allows the server to process the RSSIs and compute the estimated position. In our experiments, the maximum distance over which BLE devices demonstrated communication capability was 25 m. After the packets arrive at the server, the time required for the positioning calculation is on average 85 ms, as shown in the Section 4.5.1, ensuring a virtually instantaneous update period for the user's position. Hence, our experimental approach strives to establish a passive location system, eliminating the necessity for user engagement, with all computations conducted exclusively on the server.

4.5 Performance Evaluation

In this section, we present the performance evaluation of our system, tested in a real-world scenario. Firstly, we evaluate the impact of population size on the PSO algorithm. Then, we compare the performance of our MIPS-PSO with other model-based IPSs.

4.5.1 Parameters Evaluation

In this step, we conducted a series of experiments to evaluate the dynamic performance of our solution by measuring the relationship between positioning error and particle population size. During the testing, we utilized a sequence of 5 iterative generations across different particle quantities. This decision to use 5 iterations was influenced by the consistent observation that, following these 5 cycles, the results tended to reach a stabilized state. This state of stability was identifiable by the particles converging towards a shared focal point.

We started the test with an initial population size of only 5 particles, generated with random positions. This first test resulted in an average error of 5 m, as depicted in Figure 25, which can be explained by the insufficient population size for this scenario. In large scenarios like ours, which contain several rooms, the probability of the 5 particles being far from the real position of the mobile device is high.

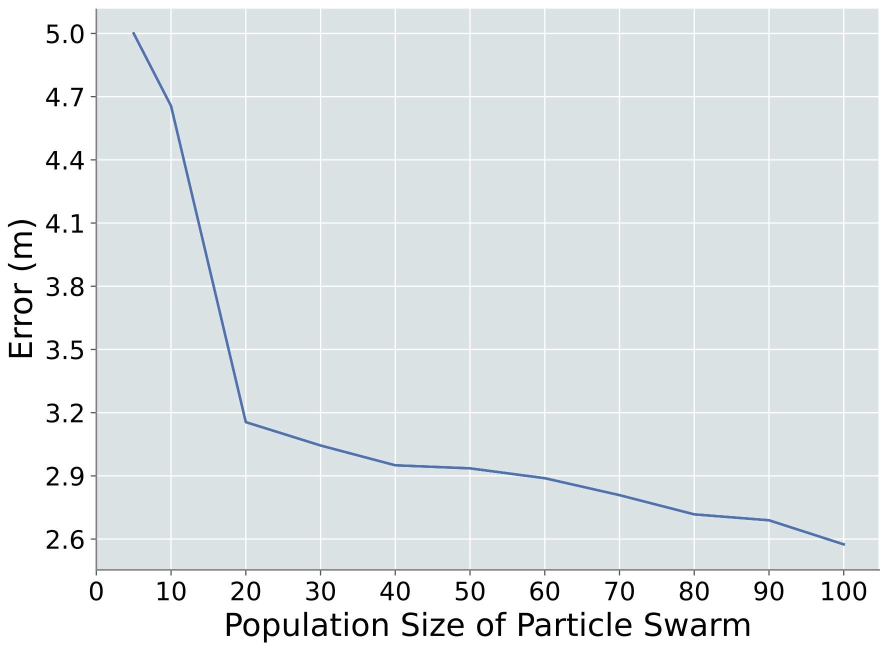

Figure 25 – Average error x population size. As we increase the population, the mean error decreases.

When we increment the population size to 10, the average error decreased to 4.70 m, as we increased the probability of particles being spread across the scenario. Consequently, using a population of 70 particles, the average error was reduced to 2.80 m. The best result obtained during the tests was with a population of 100 particles, resulting in an average error of 2.57 m for all estimates. Emphasizing its significance, it should be noted that the assessment of particle fitness can be independently and efficiently parallelized, allowing for separate calculations during each iteration.

To analyze the average time required to estimate the position of a mobile device, we adjust the population size in our solution within the range of $[10, 100]$ and determine the average time from 20 position estimates. The results are displayed in Table 17. The

Table 17 – CPU times required for implementing the PSO algorithm

Particles Size	CPU Time	Particles Size	CPU Time
10	54.8 ms	60	74.3 ms
20	60.8 ms	70	79.5 ms
30	64.1 ms	80	78.2 ms
40	68.5 ms	90	81.6 ms
50	73.5 ms	100	84.9 ms

localization experiment was done using the same computer with an Intel(R) Core(TM) $i7 - 118000H@2.30$ GHz CPU with 16 cores and 16 GB of RAM.

Figures 26, 27, and 28 illustrate an instance of our solution's execution, involving a total of 30 particles. Figure 26 shows the real position of the mobile device marked with an **X** and the 30 particles generated randomly in the scenario marked with blue dots. Each particle has different values for the parameters PL_0 and η, in addition to the cost associated with it obtained from the fitness function.

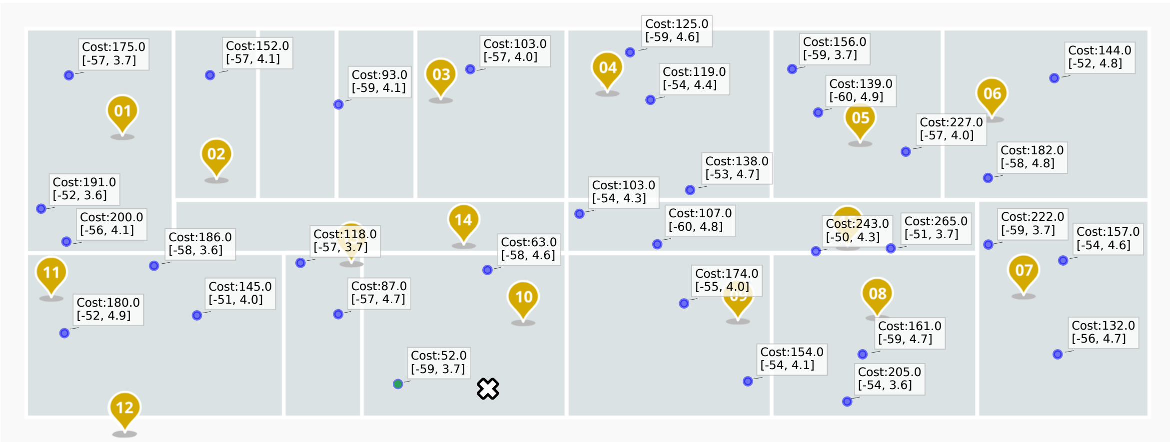

Figure 26 – 30 particles were randomly generated in the scenario, each with their respective costs and parameters PL_0 and η.

After 3 iterations of the algorithm, all particles move towards the position of the best particle chosen in the previous iteration, as shown in Figure 27.

Finally, Figure 28 illustrates the particles arranged very close to the actual position of the mobile device at the end of the 5 iterations, resulting in an error of 1.67 m. However, according to the particles positions generated, the result would be different.

During the particle velocity calculations, choosing appropriate values for PSO parameters is crucial. The parameters such as w, w_{min}, and w_{max} represent the lower and upper limits of the inertia weight. The inertia weight characterizes the influence

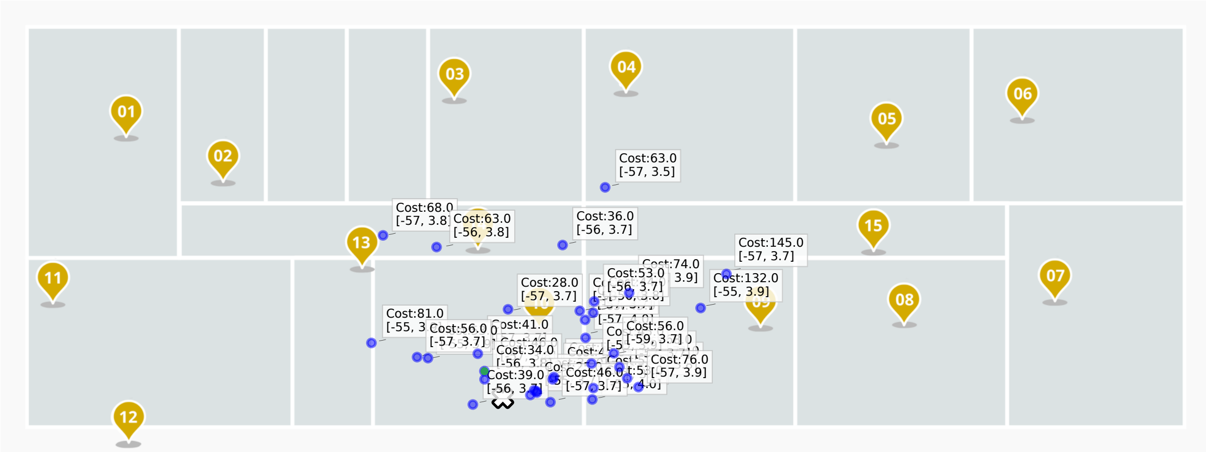

Figure 27 – Particle distribution after 3 iterations.

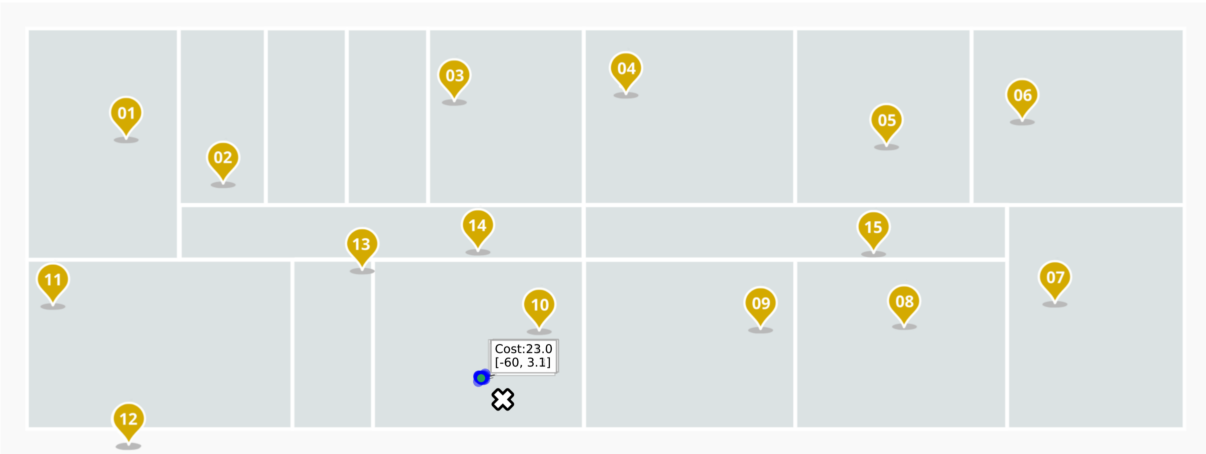

Figure 28 – Final position of the particles after 5 iterations, resulting in an average error of 1.67 m

of the previous generation's velocity on the current generation's velocity. Additionally, $c1$ and $c2$ denote the weight assigned to a particle's next action derived from its own experience and the experience of other particles, respectively.

In this study, we conducted various experiments, varying the velocity in the range $[-1, 1]$, and defining $c1$ and $c2$ in the interval $[0, 1]$, common ranges found in the literature (Cheng et al., 2020; Shan et al., 2020; Shih and Liang, 2018; Wang et al., 2017). With a fixed population size of 100 particles and 5 iterations, and by varying the values of 100 particles and 5 iterations, and varying the values of w_{min}, w_{max}, $c1$, and $c2$ within the defined intervals, we obtained different results for the average error. The highest average error was found using the values $w_{min} = 0.9$, $w_{max} = 0.9$, $c1 = 0.8$, and $c2 = 0.8$, resulting in 3.42 m, while the lowest error was achieved using the values $w_{min} = 0.5$, $w_{max} = 0.6$, $c1 = 0.4$, and $c2 = 0.4$, resulting in 2.57 m. These values enable particles to possess a higher capability for global optimization, facilitating a quicker convergence to the approximate position of the mobile device with minimal iterations.

Table 18 displays the parameters employed in this study, which have been substantiated through the experiments described earlier.

Table 18 – The summarization of MIPS-PSO parameters

Swarm Size	100
Number of Iterations	5
w_{min}	0.5
w_{max}	0.6
$c1$	0.4
$c2$	0.4

4.5.2 Comparison with Other Solutions

The viability of the suggested approach was validated through an experiment carried out in a real-world environment. This experiment involved a comparison between the proposed method and alternative approaches, namely Least Squares (LS) with fixed parameters (Yang et al., 2020), LS with dynamic parameters (Assayag et al., 2023), and KNN-PSO (Guo et al., 2019) algorithms. The results confirmed the feasibility of the proposed method.

The compared solutions also use the log-distance to estimate the position of mobile devices, but the position computation is performed using trilateration with the least squares algorithm. In these instances, the obtained fix-value used in the log-distance model was η=4.2 and PL_{d0}=-55, as determined from the collected data.

Figure 29 illustrates that our solution achieved the smallest error, measuring 2.57 m. This outcome represents a 40% reduction compared to the LS-fixed approach. Within the model-based solutions employed for comparison, LS-Fixed exhibited the highest average positioning error. This outcome was primarily attributed to the least squares algorithm utilizing the log-distance model for estimating the distance between the mobile device and the anchor node, with fixed values in the signal propagation model. Consequently, the algorithm relies on a constant set of values to characterize

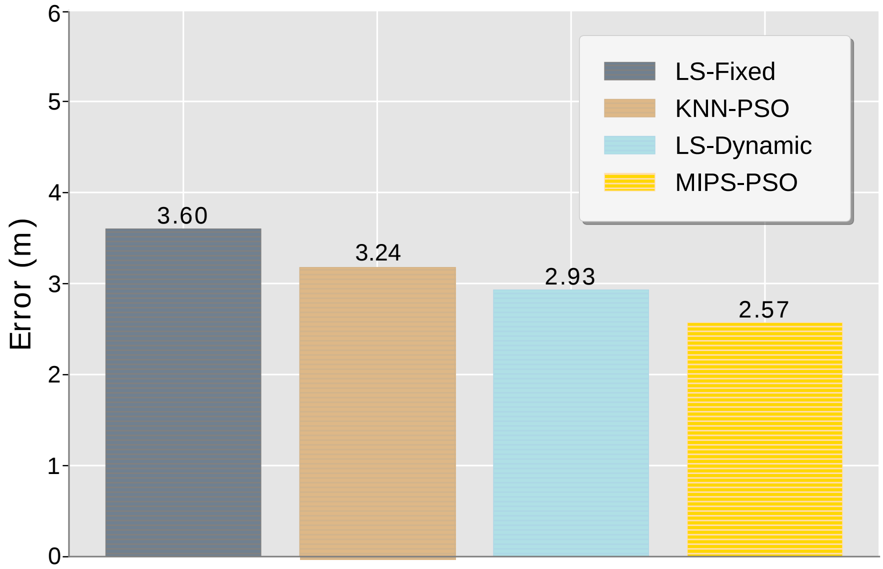

Figure 29 – Mean Absolute Error between approaches. Using LS-Fixed is the worst solution among the comparisons, while our solution resulted in 2.57 m using MIPS-PSO.

signal behavior across the whole environment, a condition that does not accurately represent reality, particularly in larger environments with numerous obstacles.

This problem is mitigated by using different values for the parameters of the log-distance model. As depicted in Figure 29, the average positioning error decreases to 2.93 m when utilizing a dynamic set of values in the signal propagation model, as detailed in (Assayag et al., 2023). In contrast to model-based positioning systems employing trilateration with the LS algorithm, the use of PSO for mobile device position estimation results in an average error of 3.24 m. This represents an 11% improvement compared to LS-Fixed. However, adopting our PSO approach, as opposed to the conventional use of model-based PSO, further reduces the average error from 3.24 m to 2.57 m, signifying a 26% enhancement.

To better understand the comparative analysis, Figure 30 is presented to assist in the understanding of the cumulative error among the solutions. We can observe that the curve of our solution, represented in orange, shows a faster growth when compared to the curves of the other solutions. This means that our system achieves better accuracy with almost 95% of the samples with positioning estimates smaller than 6.0 m. These results are justified by the fact that RSSI is affected by environmental factors that make

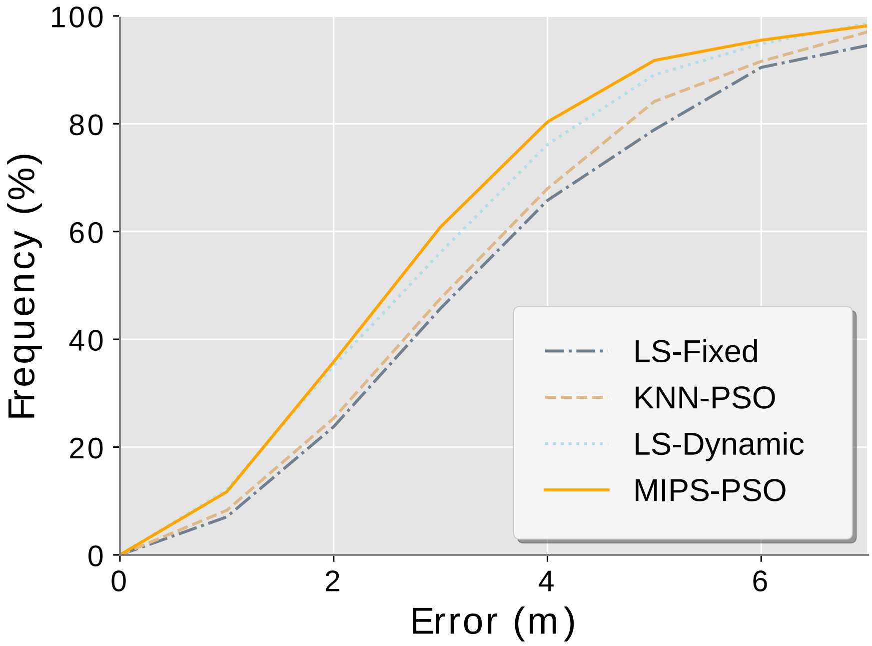

Figure 30 – Cumulative error among approaches. Our solution has the fastest growing curve, showing that most data have small errors.

modeling by signal propagation models difficult, especially in samples collected at reference points with high RSSI variations, which directly impact the mean error.

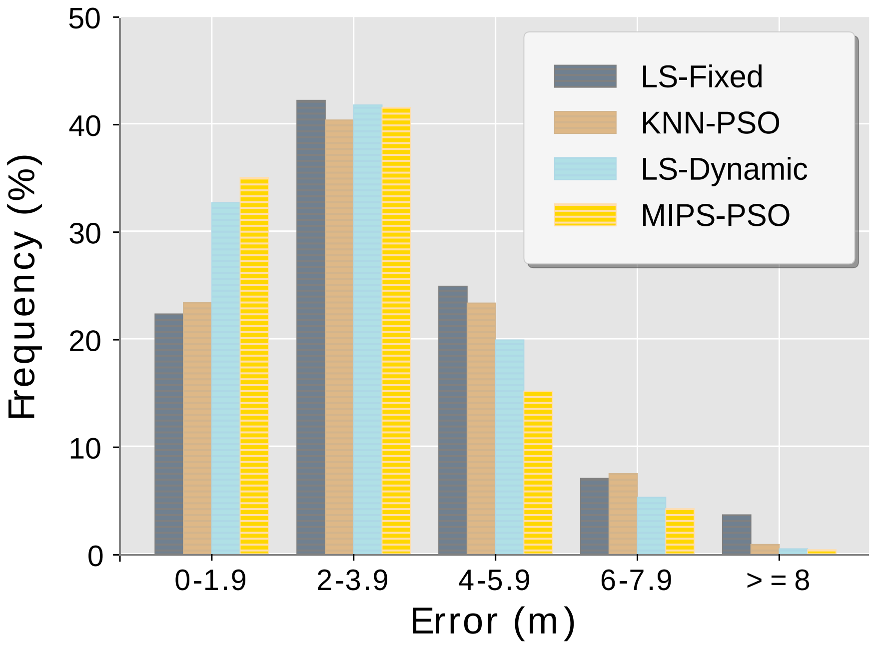

Figure 31 – Error distribution among approaches. Our solution has 75% of samples with positioning errors less than 4 m.

Figure 31 is an extension of the illustration presented in Figure 30 and shows the distribution of the error in relation to the frequency of the samples. Based on this result, we can observe that about 75% of the samples located using our solution had a mean error between $0 - 4$ m. It is important to note that it is very difficult to achieve a mean error of 0 m in indoor positioning systems based on models, due to the inevitable variation in RSSI. Furthermore, it is evident that approximately 25% of samples from the conventional PSO exhibited a mean error exceeding 4 m, whereas only 15% of samples demonstrated the same mean error when employing our positioning solution.

Table 19 allows for a more specific analysis of the positioning performance in each room.

Table 19 – Error by Room.

-	-	*Mean Absolute Error (m)*			
Rooms	*Room Size*	LS Fixed	LS Dynamic	KNN PSO	MIPS PSO
Room 1	52 m²	3.72	2.43	4.35	2.90
Room 2	26 m²	3.9	2.5	4.69	2.97
Room 3	42 m²	4.02	2.69	4.75	2.48
Room 4	60 m²	3.59	3.14	3.21	2.58
Room 5	49 m²	3.54	3.01	4.00	2.76
Room 6	60 m²	4.29	3.48	3.15	3.06
Room 7	63 m²	2.75	2.88	2.97	3.32
Room 8	55 m²	3.07	2.3	2.33	2.46
Room 9	55 m²	2.6	2.88	2.67	2.52
Room 10	55 m²	3.23	3.94	3.26	3.13
Room 11	68 m²	3.20	2.84	3.31	2.69
Hall 1	19.5 m²	4.49	3.41	2.94	1.58
Hall 2	37 m²	3.34	2.91	3.52	1.52
Hall 3	40 m²	5.69	1.93	2.33	1.59
Total	**720 m²**	**3.60 m**	**2.93 m**	**3.24 m**	**2.57 m**

The error per room, indicated in each line of the table, solely accounts for the average positioning estimate error, utilizing the RSSI data obtained from mobile devices at the reference points within each room. Conversely, the total error comprises the average of errors from all 15.000 RSSI data collections conducted at all reference points. It's crucial to emphasize that calculating the average of these room-specific averages is distinct from deriving a global average from the entirety of the 15.000 collections.

We compare our PSO-based solution with the trilateration solution using all anchor nodes with fixed parameters in the log-distance model. We can observe that for almost all rooms, our solution showed a reduction in the mean positioning error, reaching up to a 4 m difference for Hall 3, when we compare our solution with LS-Fixed. As mentioned earlier, certain areas of the scenario are more difficult to characterize using the signal propagation model, mainly due to obstacles in the environment, the arrangement of anchor nodes, and the low density of anchor nodes. One possible solution for these cases would be to increase the number of anchor nodes to improve signal coverage.

4.6 Conclusion

In this Chapter, a novel technique is presented for an Indoor Positioning System (IPS) that uses a model-based approach combined with particle swarm optimization to improve the positioning accuracy. BLE packets sent by a mobile device are used to get RSSI information, and PSO is employed to create a population of particles with RSSI vectors generated by a signal propagation model with different parameters. The particles follow an optimization-focused search strategy that converges to the approximate position of the mobile device in each iteration. To achieve this, a fitness function is used to continuously compare the RSSI vector of the particles with the real RSSI vector of the mobile device. The final mobile device position is estimated by selecting the best particle at the end of all iterations. To evaluate the performance of our solution, we conducted experiments in a real-world environment and our approach was compared to other model-based techniques available in the literature. The results demonstrate that our proposed solution reduces positioning error by up to 40% when compared to traditional LS, with an average error of 2.57 m while not requiring any training nor any model parameters estimation. In future work, other signal propagation models will be explored, and we will explore the use of inertial measurement devices to enhance the seamless positioning ability. Furthermore, we will address the variation of the BLE signal in certain regions of the scenario, contributing new parameters to the signal propagation model.

5 Thesis Discussion

In this chapter we will discuss the main advantages and disadvantages of the proposed approaches.

SynTra-IPS offers a hybrid approach between fingerprinting and model-based positioning, eliminating the need for labor-intensive real-world data collection. Instead, it generates synthetic training datasets using a log-distance propagation model, significantly reducing deployment effort while maintaining high accuracy (average error of 2.36 m). By employing data fusion, it combines multiple position estimates, improving reliability. Additionally, its computational load can be mitigated using parallel processing (GPUs or multi-core CPUs), making it scalable for high-performance environments. However, the main drawback of SynTra-IPS lies in its computational complexity. During the online phase, it must run the KNN algorithm across multiple synthetic datasets, which significantly increases the processing load. This can limit the number of position estimates that can be made per second, especially in real-time applications. Despite the possibility of mitigating this with parallel processing on CPUs or GPUs, the solution still demands a higher processing capacity than more traditional approaches.

ADAM offers a practical solution for environments where data collection is limited or infeasible. It uses a log-distance signal propagation model with adaptive parameters and selects anchor nodes in a strategic way to improve the accuracy of position estimation. By avoiding exhaustive data collection and relying only on minimal inputs such as floor plans and anchor positions, ADAM simplifies deployment and reduces setup time. The system achieves an average error of 2.93 m, which is a notable improvement over fixed-parameter model-based systems. On the downside, ADAM's accuracy is somewhat lower compared to the more complex SynTra-IPS and MIPS-PSO approaches. Additionally, ADAM's performance depends on anchor placement quality—poorly distributed anchors may degrade accuracy. It also requires prior knowledge of the floor plan and anchor positions, which may not always be available. While more efficient than full fingerprinting, it may still struggle in highly dynamic environments where signal conditions change frequently.

MIPS-PSO leverages Particle Swarm Optimization to dynamically refine position estimates, achieving an average error of 2.57 m, 40% better than traditional trilateration. Unlike fingerprint-based methods, it does not require offline training, reducing deployment time. The PSO algorithm efficiently converges to the best estimate by iteratively adjusting particle positions based on signal similarity, making it highly adaptive to varying propagation conditions. Nevertheless, this technique has its challenges. The iterative nature of PSO and the need to evaluate multiple particles in each iteration introduces a substantial computational load. Furthermore, the method's performance may be sensitive to how particles are initialized and how well the algorithm parameters are tuned. In highly dynamic or noisy environments, convergence might take longer, potentially affecting its responsiveness.

Conclusion: Trade-offs and Recommendations:

- SynTra-IPS is ideal when high accuracy (2.36 m) is critical and computational resources are available.
- ADAM is a low-effort solution (2.93 m error) suitable for scenarios where minimal setup is preferred.
- MIPS-PSO (2.57 m error) is best for dynamic environments, provided sufficient processing power is available.

6 Conclusions and Open Problems

This thesis presents our research on indoor positioning systems (IPS), where we have introduced innovative methods that significantly advance the field. Our focus has been on reducing the cost and complexity of implementing IPS while maintaining or enhancing location accuracy. These contributions are particularly relevant given the increasing demand for precise indoor positioning, driven by the rise of IoT devices and location-based services. The proposed techniques address key limitations of existing IPS methods. SynTra-IPS eliminates the need for real-environment training by generating synthetic datasets using a log-distance propagation model. ADAM-IPS improves accuracy through optimized anchor node selection, adaptive path loss models, and data fusion. PSO-MIPS leverages particle swarm optimization to refine position estimates without requiring prior training or fixed parameters.

Our methodology emphasizes rigorous empirical validation, with experiments conducted in large-scale real-world environments. By integrating data fusion techniques, optimization algorithms, and advanced signal propagation models, we have enhanced localization accuracy and reduced implementation effort. Comparative evaluations against state-of-the-art IPS systems demonstrate the competitiveness of our approaches, achieving an average positioning error of up to 2.36 m. The significance of this work is underscored by its publication in prestigious journals such as IEEE Access, IEEE Internet of Things Journal, and IEEE Sensors Journal.

While our research has yielded promising results, utilizing RSSI for location prediction and introducing novel signal propagation techniques, several challenges remain. Although tested in real-world settings, the scalability of our methods to more complex environments, such as multi-story buildings or indoor-outdoor transitions, has not been fully explored. Future work could involve additional experiments or simulations in diverse scenarios to assess the generalizability of our solutions. Moreover, this thesis primarily focuses on technical performance metrics, such as positioning error, without evaluating user experience, practical usability, or addressing security and privacy concerns related to IPS. Additionally, our evaluations were conducted using a single type of mobile device. Testing with a wider range of BLE devices, including different smartphones and BLE chips, would help determine the impact of hardware variability on system performance.

Bibliography

Ali, M., Hur, S., and Park, Y. (2017). LOCALI: Calibration-Free Systematic Localization Approach for Indoor Positioning. *Sensors*, 17:1213. pages 14, 15, 31

Assayag, Y., Oliveira, H., Souto, E., Barreto, R., and Pazzi, R. (2020). Indoor Positioning System Using Dynamic Model Estimation. *Sensors*, 20(24):7003. pages 10, 12, 13, 15

Assayag, Y., Oliveira, H., Souto, E., Barreto, R., and Pazzi, R. (2023). Adaptive path loss model for ble indoor positioning system. *IEEE Internet of Things Journal*, 10(14):12898–12907. pages 71, 87, 88

Bahl, P. and Padmanabhan, V. (2000). RADAR: An In-Building RF-based User Location and Tracking System. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 2, pages 775–784 vol.2. pages 13, 15, 31, 42, 69

Bullmann, M., Fetzer, T., Ebner, F., Ebner, M., Deinzer, F., and Grzegorzek, M. (2020). Comparison of 2.4 GHz WiFi FTM- and RSSI-Based Indoor Positioning Methods in Realistic Scenarios. *Sensors*, 20(16):4515. pages 23, 40

Caceres Najarro, L. A., Song, I., and Kim, K. (2020). Differential evolution with opposition and redirection for source localization using rss measurements in wireless sensor networks. *IEEE Transactions on Automation Science and Engineering*, 17(4):1736–1747. pages 67

Cai, X., Ye, L., and Zhang, Q. (2018). Ensemble learning particle swarm optimization for real-time uwb indoor localization. *Optik*, 2018. pages 66, 71

Cantón Paterna, V., Calveras Augé, A., Paradells Aspas, J., and Pérez Bullones, M. (2017). A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering. *Sensors*, 17(12):2927. pages 17, 22, 40, 43, 60

Cengiz, K. (2021). Comprehensive analysis on least-squares lateration for indoor positioning systems. *IEEE Internet of Things Journal*, 8(4):2842–2856. pages 43

Chen, L., Yang, K., and Wang, X. (2016). Robust Cooperative Wi-Fi Fingerprint-based Indoor Localization. *IEEE Internet of Things Journal*, 3(6):1406–1417. pages 14

Chen, X. and Zou, S. (2017). Improved wi-fi indoor positioning based on particle swarm optimization. *IEEE Sensors Journal*, 17(21):7143–7148. pages 65, 66, 69, 71, 77

Cheng, C.-H., Wang, T.-P., and Huang, Y.-F. (2020). Indoor positioning system using artificial neural network with swarm intelligence. *IEEE Access*, 8:84248–84257. pages 10, 69, 70, 86

Fang, S.-H., Hsu, Y.-T., Shiao, Y., and Sung, F.-Y. (2015). An enhanced device localization approach using mutual signal strength in cellular networks. *IEEE Internet of Things Journal*, 2(6):596–603. pages 60

Fang, X. and Chen, L. (2020). An Optimal Multi-Channel Trilateration Localization Algorithm by Radio-Multipath Multi-Objective Evolution in RSS-Ranging-Based Wireless Sensor Networks. *Sensors*, 20(6):1798. pages 3, 12, 31

Fascista, A., Ciccarese, G., Coluccia, A., and Ricci, G. (2017). A localization algorithm based on v2i communications and aoa estimation. *IEEE Signal Processing Letters*, 24(1):126–130. pages 10, 40, 66

Gabbrielli, A., Fischer, G. K. J., Schaechtle, T., Xiong, W., Schott, D. J., Bordoy, J., Wendeberg, J., Höflinger, F., Schindelhauer, C., and Rupitsch, S. J. (2023). Airborne acoustic chirp spread spectrum communication system for user identification in indoor localization. *IEEE Transactions on Instrumentation and Measurement*, 72:1–15. pages 66

Guo, H., Li, H., Xiong, J., and Yu, M. (2019). Indoor positioning system based on particle swarm optimization algorithm. *Measurement*, 134:908–913. pages 71, 87

Guo, X., Ansari, N., Li, L., and Li, H. (2018). Indoor localization by fusing a group of fingerprints based on random forests. *IEEE Internet of Things Journal*, 5(6):4686–4698. pages 10, 69

He, J., Li, S., Pahlavan, K., and Wang, Q. (2012). A realtime testbed for performance evaluation of indoor toa location system. In *2012 IEEE International Conference on Communications (ICC)*, pages 482–486. pages 10, 40, 66, 67

Huang, K., He, K., and Du, X. (2019). A hybrid method to improve the ble-based indoor positioning in a dense bluetooth environment. *Sensors*, 19(2):424. pages 2, 18, 50, 51, 60

Jean, O. and Weiss, A. J. (2014). Geolocation by direction of arrival using arrays with unknown orientation. *IEEE Transactions on Signal Processing*, 62(12):3135–3142. pages 67

Jung, S., oh Lee, C., and Han, D. (2011). Wi-Fi Fingerprint-based Approaches Following Log-distance Path Loss Model for Indoor Positioning. In *2011 IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals*, pages 1–2. pages 12, 14, 56

Kim, K. S., Lee, S., and Huang, K. (2018). A Scalable Deep Neural Network Architecture for Multi-Building and Multi-Floor Indoor Localization Based on Wi-Fi Fingerprinting. *Big Data Analytics*, 3. pages 3, 14

Kuang, W., Zhang, M., Li, W., Chen, C., and Xia, M. (2018). 3d outdoor positioning based on rssi. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5. pages 71

Kubota, R., Tagashira, S., Arakawa, Y., Kitasuka, T., and Fukuda, A. (2013). Efficient Survey Database Construction Using Location Fingerprinting Interpolation. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 469–476. pages 14, 15

Lazaro, A., Girbau, D., Moravek, P., and Villarino, R. (2010). Localization in Wireless Sensor Networks Using Diversity for Multipath Effects Mitigation. In *European Workshop on Smart Objects: Systems, Technologies and Applications*, pages 1–6. pages 12

Li, G., Geng, E., Ye, Z., Xu, Y., Lin, J., and Pang, Y. (2018). Indoor positioning algorithm based on the improved rssi distance model. *Sensors*, 18(9):2820. pages 10, 40, 67, 71

Li, N., Chen, J., Yuan, Y., Tian, X., Han, Y., and Xia, M. (2016a). A wi-fi indoor localization strategy using particle swarm optimization based artificial neural networks. *International Journal of Distributed Sensor Networks*, 12(3):4583147. pages 69

Li, Q., Li, W., Sun, W., Li, J., and Liu, Z. (2016b). Fingerprint and Assistant Nodes Based Wi-Fi Localization in Complex Indoor Environment. *IEEE Access*, 4:2993–3004. pages 12, 15

Maher, P. S. and Malaney, R. A. (2009). A Novel Fingerprint Location Method Using

Ray-Tracing. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, pages 1–5. pages 14

Mendoza-Silva, G., Costa, A. C., Torres-Sospedra, J., Painho, M., and Huerta, J. (2021). Environment-aware regression for indoor localization based on wifi fingerprinting. *IEEE Sensors Journal*, pages 1–1. pages 40, 42

Moghtadaiee, V., Ghorashi, S. A., and Ghavami, M. (2019). New Reconstructed Database for Cost Reduction in Indoor Fingerprinting Localization. *IEEE Access*, 7:104462–104477. pages 11, 14, 40, 43

Mohammed El Amine, C. and Ouslim, M. (2015). The Implementation of Indoor Localization Based on An Experimental Study of RSSI Using a Wireless Sensor Network. *Peer-to-Peer Networking and Applications*, 9. pages 12

Niu, J., Wang, B., Cheng, L., and Rodrigues, J. J. P. C. (2015). WicLoc: An Indoor Localization System Based on WiFi Fingerprints and Crowdsourcing. In *2015 IEEE International Conference on Communications (ICC)*, pages 3008–3013. pages 14, 15

Nowicki, M. and Wietzykowski, J. (2017). Low-Effort Place Recognition with WiFi Fingerprints Using Deep Learning. *Advances in Intelligent Systems and Computing*, page 575–584. pages 14

Onofre, S., Silvestre, P. M., Pimentão, J. P., and Sousa, P. (2016). Surpassing bluetooth low energy limitations on distance determination. In *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*, pages 843–847. pages 43

Praveen Kumar, D., Amgoth, T., and Annavarapu, C. S. R. (2019). Machine Learning Algorithms for Wireless Sensor Networks: A Survey. *Information Fusion*, 49:1–25. pages 13, 36

Qiang, C., Van de Velde, S., and Steendam, H. (2019). How to get the best out of your fingerprint database: hierarchical fingerprint indoor positioning for databases with variable density. *IEEE Access*, pages 1–1. pages 14, 15

Röbesaat, J., Zhang, P., Abdelaal, M., and Theel, O. (2017). An improved ble indoor localization with kalman-based fusion: An experimental study. *Sensors*, 17(5):951. pages 23, 49, 50, 56

Sadowski, S. and Spachos, P. (2018). RSSI-Based Indoor Localization With the Internet of Things. *IEEE Access*, 6:30149–30161. pages 2, 10, 13, 23, 56, 66, 70, 77

Sasikala, M., Athena, J., and Rini, A. (2021). Received signal strength based indoor positioning with rfid. In *2021 IEEE International Conference on RFID Technology and Applications (RFID-TA)*, pages 260–263. pages 40

Schreiber, R. and Bajer, J. (2016). Software defined radio based receiver for tdoa positioning system. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–5. pages 10, 40

Shan, Z., Zhang, F., Lv, N., and Xiang, W. (2020). Research on indoor positioning algorithm based on neighborhood partitioning. In *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)*, pages 380–384. pages 70, 86

Shchekotov, M. (2015). Indoor localization methods based on wi-fi lateration and signal strength data collection. In *2015 17th Conference of Open Innovations Association (FRUCT)*, pages 186–191. pages 56

Shi, Y., Shi, W., Liu, X., and Xiao, X. (2020). An RSSI Classification and Tracing Algorithm to Improve Trilateration-Based Positioning. *Sensors*, 20(15):4244. pages 13, 23, 31, 43, 44, 56

Shih, C. and Liang, C. (2018). The improvement of indoor localization precision through partial least square(pls) and swarm (pso) methods. In *2018 IEEE Sensors Applications Symposium (SAS)*, pages 1–6. pages 70, 77, 86

Torres, J., Belmonte, O., Montoliu, R., Trilles, S., and Calia, A. (2016). How Feasible Is WiFi Fingerprint-Based Indoor Positioning for In-Home Monitoring? In *2016 12th International Conference on Intelligent Environments (IE)*, pages 68–75. pages 13, 31

Ullah, M., Hossain Rian, S., Al Mamun, A., Das Plabon, S., and Paul, S. (2020). Path loss and received signal power profiling for optimal positioning of access point in indoor wlan. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–5. pages 3, 4, 40

Wang, B., Gan, X., Liu, X., Yu, B., Jia, R., Huang, L., and Jia, H. (2020). A novel weighted knn algorithm based on rss similarity and position distance for wi-fi fingerprint positioning. *IEEE Access*, 8:30591–30602. pages 11, 40, 42

Wang, C., Shi, Z., and Wu, F. (2017). An improved particle swarm optimization-based feed-forward neural network combined with rfid sensors to indoor localization. *Information*, 8(1):9. pages 66, 70, 86

Wang, Y., Yang, Q., Zhang, G., and Zhang, P. (2016). Indoor positioning system using euclidean distance correction algorithm with bluetooth low energy beacon. *2016*

International Conference on Internet of Things and Applications (IOTA), pages 243–247. pages 40

Wang, Y., Yang, X., Zhao, Y., Liu, Y., and Cuthbert, L. (2013). Bluetooth positioning using rssi and triangulation methods. In *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pages 837–842. pages 60

Wu, T., Xia, H., Liu, S., and Qiao, Y. (2019). Probability-based indoor positioning algorithm using ibeacons. *Sensors*, 19(23):5226. pages 51

Wu, Y.-H., Chen, Y.-L., and Sheu, S.-T. (2016). Indoor location estimation using virtual fingerprint construction and zone-based remedy algorithm. In *2016 International Conference On Communication Problem-Solving (ICCP)*, pages 1–3. pages 43

Xia, B., Liu, T., Ding, T., and Wang, Z. (2021). An improved pso localization algorithm for uwb sensor networks. *Wirel. Pers. Commun.*, 117(3):2207–2223. pages 71

Xiong, W., Schindelhauer, C., and So, H. C. (2023). Globally optimized tdoa high-frequency source localization based on quasi-parabolic ionosphere modeling and collaborative gradient projection. *IEEE Transactions on Aerospace and Electronic Systems*, 59(1):580–590. pages 66, 67

Xiong, W. and So, H. C. (2023). Outlier-robust passive elliptic target localization. *IEEE Geoscience and Remote Sensing Letters*, 20:1–5. pages 67

Yang, B., Guo, L., Guo, R., Zhao, M., and Zhao, T. (2020). A novel trilateration algorithm for rssi-based indoor localization. *IEEE Sensors Journal*, 20(14):8164–8172. pages 40, 43, 70, 87

Youssef, M. and Agrawala, A. (2005). The Horus WLAN Location Determination System. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, page 205–218, New York, NY, USA. Association for Computing Machinery. pages 9, 13, 31

Zhang, J., Han, G., Sun, N., and Shu, L. (2017). Path-loss-based fingerprint localization approach for location-based services in indoor environments. *IEEE Access*, 5:13756–13769. pages 10, 39, 40, 41, 43

Zheng, J., Li, K., and Zhang, X. (2022). Wi-fi fingerprint-based indoor localization method via standard particle swarm optimization. *Sensors*, 22(13):5051. pages 1, 69