Real-time Safety Alerting System for Construction Sites

by

Nima Abdollahpour

B.Sc., Sharif University of Technology, 2019

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Applied Science

in the
School of Mechatronic Systems Engineering
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2024

Declaration of Committee

Name:	**Nima Abdollahpour**
Degree:	**Master of Applied Science**
Thesis title:	**Real-time Safety Alerting System for Construction Sites**
Committee:	**Chair:** Jiacheng (Jason) Wang Associate Professor, Mechatronic Systems Engineering
	Mehrdad Moallem Supervisor Professor, Mechatronic Systems Engineering
	Mohammad Narimani Committee Member Senior Lecturer, Mechatronic Systems Engineering
	Kamal Gupta Examiner Professor, Engineering Science

Abstract

Construction sites represent complex, dynamic, environments where ensuring safety is crucial yet challenging. Traditional safety measures often fall short due to their reliance on manual monitoring and intervention, which are often time-consuming and prone to errors. To address these challenges, this study proposes a *Real-Time Safety Alerting System for Construction Sites*, by employing Bluetooth Low Energy (BLE) devices for indoor localization coupled with a customized Android application for real-time monitoring. Leveraging the FIND3 framework, significant customization was introduced to create a robust system capable of tracking workers and equipment within a construction site with the objective of alerting potential safety hazards.

The core components of the system include an Android application, server-side components, and an interactive front-end website. The Android app, installed on workers' phones, interfaces with BLE devices deployed across a construction site, facilitating precise indoor positioning using a fingerprinting algorithm. Server-side components, implemented using Go, Python, and Docker, provide administrative control, data management, and real-time monitoring. An interactive website displays the location of workers and equipment on a floor plan, alongside real-time safety alerts.

The system architecture further incorporates construction site zoning, dividing the floor plan into different zones with specific safety requirements. Machine learning algorithms sch as k-NN, Random Forest, and SVM are employed on the server-side to analyze location data for hazard detection and safety management. Real-time alarms and notifications are relayed to both workers and employers through the Android app and website, enhancing the overall safety management of the construction site.

Through rigorous testing and evaluation on an *emulated* construction site, this system demonstrates a promising approach to bolster construction site safety, providing a foundation for further enhancements and real-world deployment. This thesis delineates the design, implementation, and evaluation of this system on a proof-of-concept system, shedding light on the potential of integrating modern wireless communication, indoor localization, and machine learning technologies to improve construction site safety management.

Keywords: Construction Site Safety; Real-Time Monitoring; Bluetooth Low Energy (BLE); Indoor Localization; Fingerprinting Algorithm; Machine Learning for Hazard Detection

Dedication

To my beautiful wife, my brother, and my parents, who have always supported me.

Acknowledgements

I would like to express my deepest gratitude to a number of people whose support was invaluable during the course of this research.

First and foremost, I extend my heartfelt thanks to Dr. Mehrdad Moallem, who acted as my supervisor. His invaluable guidance, insightful feedback, and unwavering support throughout the duration of this project and my studies were instrumental in the successful completion of this research.

I am also deeply grateful to MITACS and SiteinSite for providing me the opportunity to delve into this fascinating area of research. Their funding and vision for the project have been a driving force behind its success.

Special thanks go to Dr. Mohammad Narimani for his assistance and valuable insights as a committee member.

I cannot forget the constant encouragement from my family. Despite being far away, they have been a continuous source of inspiration and learning for me.

To my friends, and especially to my wife, who have been pillars of strength and support during my studies, I owe a debt of gratitude. Your friendship and the relief it brought, especially during challenging times, were vital in keeping me motivated and focused. This thesis is not just a reflection of my work but a testament to the support and love I have received from each one of you. Thank you.

Table of Contents

Declaration of Committee **ii**

Abstract **iii**

Dedication **v**

Acknowledgements **vi**

Table of Contents **vii**

List of Figures **xi**

List of Acronyms **xiii**

1 Introduction **1**

1.1 Background and Motivation 1
1.2 Objectives and Scope 1
1.3 Contributions 2
1.4 Thesis Structure 3

2 Related Works **4**

2.1 Construction Site Safety 4
2.1.1 Importance of Safety Management 4
2.1.2 Common Hazards in Construction Sites 5
2.1.3 Safety Practices and Standards 6
2.2 Indoor Localization Techniques 7
2.2.1 Global Positioning System (GPS) and its Limitations 7
2.2.2 Radio Frequency-based Techniques 7
2.2.3 Fusion of Different Techniques 8
2.2.4 Comparison of Indoor Localization Techniques 8
2.3 Bluetooth Low Energy (BLE) Devices 10
2.3.1 Overview of BLE Technology 10
2.3.2 Advantages of BLE Devices in Localization 10

2.3.3 Limitations and Challenges . . . 10
2.3.4 Applications in Construction Site Safety . . . 11
2.4 Fingerprinting in Indoor Localization . . . 11
2.4.1 Overview of Fingerprinting Technique . . . 11
2.4.2 Fingerprinting Process . . . 12
2.4.3 Advantages and Limitations . . . 13
2.4.4 Fingerprinting in Construction Site Safety Applications . . . 14
2.5 Machine Learning and Clustering Algorithms in Indoor Localization . . . 15
2.5.1 Role of Machine Learning and Clustering in Indoor Localization . . . 15
2.5.2 Supervised Learning: From k-Nearest Neighbors (k-NN) to Neural Networks . . . 15
2.5.3 Unsupervised Learning: Clustering Techniques and Their Applications 15
2.5.4 Comparison: Supervised vs Unsupervised Learning in Localization . 16
2.5.5 Limitations and Challenges of ML Algorithms in Construction Sites 17

3 System Architecture and Construction Site Zoning 19
3.1 Overview . . . 19
3.2 BLE Device Integration . . . 20
3.2.1 Types of Beacons: Stations and PPEs . . . 22
3.3 Android Application . . . 23
3.3.1 Core Features of the Android App . . . 23
3.3.2 Challenges and Overcoming Them . . . 23
3.3.3 Customizations for Enhanced Compatibility . . . 25
3.3.4 Deep Dive into the Code . . . 26
3.4 Server-side Components . . . 27
3.4.1 Framework Selection . . . 27
3.4.2 Real-time Updates . . . 27
3.4.3 Website Functionality and Data Management . . . 28
3.4.4 Visual Aids . . . 28
3.5 Zone Division and Safety Requirements . . . 28
3.5.1 Zoning Concept . . . 28
3.5.2 Types of Zones . . . 29
3.5.3 Personal Protective Equipment (PPE) Beacons . . . 29
3.5.4 Visualization of Zones . . . 29
3.6 Real-time Monitoring and Compliance and Alarm System . . . 30
3.6.1 Monitoring in Real-Time . . . 30
3.6.2 Compliance and Alarm System . . . 30

4 Indoor Localization and Fingerprinting 33
4.1 Overview of Fingerprinting . . . 33

4.1.1 Offline Phase 33
4.1.2 Online Phase 33
4.2 Data Collection and Storage 34
4.2.1 Variables Impacting RSSI Readings 35
4.2.2 RSSI Data Collection from BLE Beacons 35
4.2.3 Data Storage Mechanism 36
4.3 Real-time Location Estimation 37
4.3.1 Process of Real-time Localization using Stored Data 37
4.3.2 Correlation between Historical and Current RSSI Data 38
4.3.3 Nuances of Making the Process Real-time 38
4.4 Building a Reliable Fingerprinting Database 39
4.4.1 Initial Setup 39
4.4.2 Periodic Recalibration 40
4.4.3 Database Reliability 40

5 Machine Learning Algorithms for Hazard Detection and Implementation 42
5.1 Overview of ML Structure for the Server 42
5.2 Naive Bayes Algorithm 43
5.2.1 Extending Traditional Naive Bayes 43
5.2.2 Implementation Details 44
5.3 k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Random Forest, and Other Algorithms 45
5.3.1 Overview of Python-Based Classification 45
5.3.2 Enhancements for Accuracy 47
5.4 Algorithm Comparison and Performance Evaluation 49
5.4.1 Consensus Prediction 49
5.4.2 Performance Metrics and Evaluations 50
5.4.3 Under the Hood: Code Mechanics 50

6 System Development and Performance Assessment 53
6.1 System Testing and Evaluation 53
6.1.1 Testing Methodology 53
6.1.2 Performance Metrics 54
6.2 Results and Analysis 56
6.2.1 Dynamic Test Analysis 56
6.2.2 Hardware Dependency of Response Time 57
6.2.3 Spatial Accuracy Analysis 58

7 Conclusion 60

Bibliography **62**

List of Figures

Figure 2.1 Comparison of various indoor positioning technologies in terms of accuracy, cost of implementation, and power consumption. Figure reproduced from [23]. . . . 9
Figure 2.2 Fluctuations in RSSI from a BLE beacon at a fixed distance. 11
Figure 2.3 Overview of the fingerprinting process. Figure reproduced from [23]. 13

Figure 3.1 High-level schematic representation of the indoor localization project, illustrating the integration of hardware and software components, and the bidirectional communication between the server and Android device. . . . 20
Figure 3.2 Bluetooth long range waterproof beacon K5P. . . . 21
Figure 3.3 Bluetooth long range waterproof beacon K5P and its transmission distances. . . . 23
Figure 3.4 Screenshot of the Learning Mode interface. . . . 24
Figure 3.5 Screenshot of the Tracking Mode interface. . . . 25
Figure 3.6 Screenshot of the Main Page. . . . 28
Figure 3.7 Floorplan with worker's real-time position. . . . 28
Figure 3.8 Sample floor-plan indicating various safety zones in a building. . . . 30
Figure 3.9 Example of a worker in a Safe Zone. . . . 31
Figure 3.10 Example of a worker in a Blue Zone without the required PPE, notification on the Website. . . . 32
Figure 3.11 Example of a worker in a Green Zone, notification on the Website. 32

Figure 4.1 Schematic representation of the fingerprinting process, illustrating the offline and online phases. . . . 34
Figure 4.2 Data representation for learning and tracking 37
Figure 4.3 Flowchart depicting database operations based on *db.go* functionalities. 41

Figure 5.1 Flowchart illustrating the process of sensor data processing and prediction. . . . 52

Figure 6.1 Photograph showcasing the test environment and beacons on the walls 54
Figure 6.2 Photograph showcasing the test environment and beacons on the walls 55

Figure 6.3 System's Confidence Over Time 57
Figure 6.4 Visualization of Spatial Accuracy 59

List of Acronyms

BLE Bluetooth Low Energy

BPNN Backpropagation Neural Network

DNN Deep Neural Networks

ELM Extreme Learning Machine

FIND Framework for Internal Navigation and Discovery

GPS Global Positioning System

HSE Health and Safety Executive

IPS Indoor Positioning Systems

k-NN k-Nearest Neighbors

OSHA Occupational Safety and Health Administration

PDR Pedestrian Dead Reckoning

PPE Personal Protective Equipment

RFID Radio Frequency Identification

RSS Received Signal Strength

RSSI Received Signal Strength Indicator

SVM Support Vector Machines

SVR Support Vector Regression

UWB Ultra-Wideband

VLC Visible Light Communication

WLAN Wireless Local Area Network

Chapter 1

Introduction

1.1 Background and Motivation

Construction sites are dynamic environments where safety is of paramount importance. Workers and employers must continuously monitor and manage potential hazards to prevent accidents and ensure a safe work environment. Traditional safety management approaches rely on human observation and manual intervention, which can be time-consuming and prone to errors. Moreover, construction sites often involve complex indoor environments, making it challenging to monitor worker and equipment locations accurately.

The rapid advancements in technology, specifically in indoor localization and wireless communication, have paved the way for new solutions that can address these challenges. Bluetooth Low Energy (BLE) devices and indoor localization techniques, such as fingerprinting, offer the potential for precise location tracking within construction sites. Additionally, machine learning algorithms can be utilized to analyze location data and proactively identify potential safety hazards.

1.2 Objectives and Scope

The primary objective of this thesis is to design and implement a real-time safety alerting system for construction sites, leveraging a network of BLE devices and a custom-built Android application. This system will track worker and equipment locations using fingerprinting for indoor localization and analyze the location data using machine learning algorithms. The project builds upon the existing FIND framework to provide an enhanced safety solution for construction site management.

The scope of the thesis includes:

1. Developing an Android application to integrate with BLE devices for precise indoor positioning.
2. Creating server-side components for administrative control using Go, Python, and Docker.

3. Implementing machine learning algorithms, such as k-Nearest Neighbors(k-NN), Random Forest, and Support Vector Machines(SVM), for hazard detection and safety management.

4. Dividing floor plans into different zones with specific safety conditions and ensuring compliance.

5. Designing a real-time alarm and notification system for both workers and employers.

1.3 Contributions

This thesis makes significant advancements in construction site safety through the innovative adaptation and application of the FIND framework, BLE technology, and advanced machine learning algorithms. The key contributions are:

- **Adaptation of FIND Framework for BLE Technology:** This adaptation has been pivotal in leveraging the strengths of BLE for precise indoor localization within construction sites. Key system features developed include:
  - *Alerting Systems:* Implementation of real-time safety alerting mechanisms on the Android application and the website.
  - *Zoning Mechanism:* Creation of different safety zones on construction site floor plans to enhance situational awareness and safety compliance.
  - *Integration of Various Beacon Types:* Utilizing a combination of stationary and Personal Protective Equipment(PPE) BLE devices for comprehensive site monitoring.
  - *Received Signal Strength Indicator(RSSI) Filtering:* Introduction of RSSI cutoff strategies to optimize data collection efficiency.

- **Handling Large Data Volumes in Multi-floor Buildings:** This aspect addresses the challenges posed by high-density BLE environments, particularly in multi-floor buildings. Optimizations include:
  - *Android App Enhancements:* Optimizing the app to scan and transmit data from nearby beacons efficiently, using a LinkedHashMap to manage recent BLE signals.
  - *Server-Side Data Management:* Employing a randomized search approach in machine learning processes for effective large-volume data handling.

- **Server-Side Components and Machine Learning Integration:** This crucial contribution involves the seamless integration of real-time monitoring and floorplan visualizations. Advanced machine learning algorithms such as k-NN, SVM, and Random Forest are utilized for accurate localization and hazard detection. Enhanced techniques for managing missing RSSI values have been implemented to improve the system's robustness and reliability.

Together, these contributions significantly enhance the Real-time Safety Alerting System for Construction Sites, showing great promise for improving safety management in dynamic and complex construction environments.

1.4 Thesis Structure

This thesis is organized into seven chapters, each detailing a distinct aspect of the study:

1. **Introduction**: This chapter presents the background and motivation for the study, the objectives, and the scope of the thesis, and provides an overview of the structure of the entire work.

2. **Literature Review**: This chapter reviews existing literature on construction site safety, indoor localization techniques, Bluetooth Low Energy (BLE) devices, fingerprinting in indoor localization, and machine learning algorithms in safety management.

3. **System Architecture and Construction Site Zoning**: This chapter presents an overview of the proposed system architecture, focusing on BLE device integration, Android application development, server-side components, zone division, safety requirements, and real-time monitoring and compliance systems.

4. **Indoor Localization and Fingerprinting**: This chapter elaborates on the fingerprinting technique used in the proposed system consisting of the data collection, processing, and location estimation methods.

5. **Machine Learning Algorithms for Hazard Detection**: This chapter provides an overview of the machine learning structure utilized on the server side. It gives detailed insights into the use of algorithms such as Naive Bayes, k-Nearest Neighbors (k-NN), Random Forest, and Support Vector Machines (SVM).

6. **System Development and Performance Assessment**: This chapter discusses the development of the Android application and server-side components. It further assesses the system's performance through rigorous testing and evaluation.

7. **Conclusion**: This final chapter summarizes the research, presents the findings, discusses the implications, and offers suggestions for future work.

The following chapters provide a comprehensive understanding of the real-time safety alerting system, its design and implementation, and evaluation of its effectiveness in enhancing construction site safety.

Chapter 2

Related Works

2.1 Construction Site Safety

Construction site safety is a critical concern due to the complex and dynamic nature of construction environments [5]. Ensuring the well-being of workers and preventing accidents is vital for the success of any construction project. Falling-at-height accidents are, without a doubt, one of the leading causes of fatal accidents at workplaces such as construction sites. In an investigation of identifying the root causes of accidents [5], it was revealed that accidents often occur due to the failure of workers to identify hazards, workers deciding to proceed despite the identification of hazardous conditions, or workers choosing to act unsafely regardless of work conditions.

Working at height, either intentionally or unintentionally, without proper precautionary measures, may lead to an accident or even death if the supervisor is not aware of the worker's unsafe condition. In such cases, the top site management cannot carry out precautionary measures in time under a fall protection plan. As noted in [5], sensor-based systems for improving construction safety have recently gained considerable attention.

A crucial part of modern safety management is the integration of data-driven accident prevention systems, which use machine learning to develop injury-risk mitigation frameworks and flag high-risk zones for proactive risk mitigation [5, 14]. In this section, we will discuss the importance of safety management in construction sites, the common hazards, and the existing safety practices and standards.

2.1.1 Importance of Safety Management

Safety management plays a crucial role in reducing workplace accidents, injuries, and fatalities in the construction industry. Effective safety management can lead to improved worker productivity, reduced project delays, lower insurance costs, and an enhanced reputation for the construction company [20]. Moreover, it is essential to comply with the legal and regulatory requirements set by the government and industry organizations to ensure a safe work environment. In recent years, the importance of integrating multi-method modeling approaches into safety risk management has been highlighted to improve project performance metrics such as quality, productivity, and cost [2]. Machine learning models are proving valuable in this context, proactively predicting risk levels and aiding decision-making, thereby dynamically enhancing safety performance [15].

2.1.2 Common Hazards in Construction Sites

Construction sites are replete with potential hazards, often leading to accidents and injuries. These environments pose significant risk factors. Thus, it is essential to identify and understand them to apply effective safety measures [20].

Here are some of the most common hazards found in construction sites:

1. Falls from height: One of the leading causes of injuries on construction sites is falls from elevated surfaces, such as scaffolds, ladders, or roofs. The lack of protective measures or faulty equipment often results in severe injuries or fatalities.

2. Struck-by accidents: These occur when workers are hit by moving vehicles, equipment, or falling objects. The high-velocity impact can lead to severe injuries and may even be fatal.

3. Poor Zoning Practices: An often overlooked but significant risk factor is the absence of proper zoning within construction sites. Without well-defined zones, workers may unknowingly enter areas where they might be exposed to specific hazards. For instance, some zones might be generally safe for all workers, while others might require specific safety equipment or training to enter. Certain zones might be restricted to specific personnel, and some might be explicitly marked as danger zones due to high-risk factors. The lack of clarity and enforcement of such zones often contributes to accidents and injuries.

4. Caught-in/between: Instances, where workers become trapped in or between machinery, equipment, or construction materials, can result in crushing injuries. Proper training and strict adherence to safety protocols can help prevent such incidents.

5. Electrocution: Exposure to live electrical wires or equipment may result in electrocution or electrical burns. Ensuring proper electrical safety practices significantly reduces this risk.

6. Trips and slips: Construction sites, uneven surfaces, debris, or wet floors can cause workers to trip or slip. Such accidents, although seemingly minor, can lead to severe injuries, particularly if the worker falls onto a sharp object or heavy machinery.

By recognizing these common hazards, construction sites can implement necessary safety practices and preventive measures to minimize the occurrence of accidents and enhance overall site safety.

2.1.3 Safety Practices and Standards

To encounter the risks inherent in construction sites, an array of safety practices and standards have been set in place. These measures are designed to promote a safe working environment, minimizing injuries and accidents. The use of Smart Personal Protective Equipment (PPE) combined with artificial intelligence techniques is one such practice, enhancing safety through the early prediction and notification of anomalies detected in the worker's environment [31]. In addition to these practices, the role of zoning practices cannot be underestimated. Effective zoning practices are key to improving safety in construction sites. The implementation of well-defined zones can help control access to certain high-risk areas, manage the flow of workers and machinery, and guide the use of necessary protective equipment [20]. A clear understanding of zones – safe zones, zones requiring equipment, worker-specific zones, and danger zones – is crucial for all personnel. This can be facilitated through regular safety training, clear signage, and the use of digital tools for real-time tracking and monitoring of workers' locations.

Moreover, various organizations and regulatory bodies have set safety standards for the construction industry. Notably, the Occupational Safety and Health Administration (OSHA) in the United States and the Health and Safety Executive (HSE) in the United Kingdom provide guidelines and enforce safety regulations for the construction industry. Their regulations are instrumental in promoting a culture of safety within the construction field.

In addition to these practices, data-driven methods such as machine learning models have been used to predict injury types and to develop safety controls in construction [3]. Also, the use of collaborative information integration frameworks for safety monitoring that collect, analyze, and disseminate safety information have shown to improve the collaborative working environment for safety inspection and monitoring in construction projects [43]. Advanced technologies such as machine learning and Indoor Positioning Systems (IPS) can play a significant role in enhancing zoning practices. For instance, machine learning algorithms can be used to predict and flag potential risk areas, thereby contributing to the dynamic delineation of zones based on risk levels [15]. IPS, on the other hand, can help monitor the real-time location of workers and ensure their adherence to designated zones, thereby minimizing accidents due to unauthorized access to high-risk areas.

In summary, construction site safety is a critical aspect of construction project management. Effective safety management practices help reduce accidents and injuries, improve worker productivity, and ensure compliance with legal and regulatory requirements. The next sections will explore the use of technology, specifically indoor localization techniques, and Bluetooth Low Energy (BLE) devices, to enhance safety management in construction sites.

2.2 Indoor Localization Techniques

Indoor localization techniques have been developed to counter the limitations of Global Positioning Systems (GPS) in indoor settings. GPS struggles to perform well indoors because of signal loss due to interference by building materials and other obstructions. Thus, indoor positioning techniques such as Bluetooth-based technologies, integrated Radio Frequency Identification(RFID) technologies, and intelligent fingerprint-based localization have been proposed to alleviate the above issues [7, 22, 49, 50].

2.2.1 Global Positioning System (GPS) and its Limitations

The global positioning system (GPS) technology has revolutionized outdoor localization and navigation, providing accurate positioning information anywhere on the globe. While highly effective for outdoor location services, signal loss due to building structures and electronic devices, as well as the requirement for a clear line of sight to at least four satellites, makes GPS less effective for indoor localization [22, 49]. The GPS signals are severely attenuated when they pass through walls, roofs, and other obstructions, leading to a significant loss in positioning accuracy. These limitations have led to the development of alternative indoor localization techniques.

2.2.2 Radio Frequency-based Techniques

These techniques exploit the propagation of radio waves to determine the position of objects or people within an indoor environment. Radio Frequency-based techniques such as WiFi, Bluetooth, and RFID have been widely utilized for indoor positioning due to their ability to penetrate walls and other obstructions [6, 11, 22, 32, 38, 39].

Wi-Fi Positioning

WiFi positioning is a common indoor localization technique that leverages the ubiquitous presence of WiFi access points in indoor environments. It typically operates based on the WiFi signals from multiple WiFi access points (APs) to determine the position of a device as well as using advanced machine learning techniques such as deep neural networks to enhance precision[28]. Techniques such as Received Signal Strength (RSS)-based positioning and fingerprinting have been developed for this purpose[11, 17, 28, 30, 50]. The main advantages of WiFi positioning are its widespread availability and relatively low cost. However, WiFi signals can be affected by multipath propagation and signal interference, which may reduce positioning accuracy.

Bluetooth-based Positioning

Bluetooth-based positioning is another popular indoor localization technique, often utilizing Bluetooth Low Energy (BLE) beacons where Received Signal Strength Indicator (RSSI) values of BLE signals are used to estimate location [33]. Similar to WiFi positioning, it typically relies on the RSSI from multiple BLE beacons with methods like trilateration algorithm for position calculation [33]. Bluetooth positioning utilizes RSSI from Bluetooth devices to estimate the distance between devices and thereby locate them within an environment. However, like Wi-Fi signals, Bluetooth signals can be affected by signal interference and environmental factors. Various studies have shown the efficacy of Bluetooth-based positioning in different settings such as office buildings, museums, and even for navigation applications[13, 33, 34, 38, 39].

RFID

While this thesis does not cover RFID, it's another type of Radio Frequency-based technique that is often used for indoor positioning and tracking with significant improvements in positioning accuracy by using integrated RFID technologies [22].

2.2.3 Fusion of Different Techniques

There is a growing trend in research to fuse different techniques for enhancing indoor positioning accuracy. This is evident in studies [22, 24, 25, 37].

For instance, [24] proposed an indoor positioning system using a Bluetooth receiver, accelerometer, magnetic field sensor, and barometer on a smartphone, demonstrating excellent localization performance. Similarly, [37] proposed a novel positioning method based on fusing trilateration and dead reckoning, employing Kalman filtering as a position fusion algorithm to improve positioning accuracy. The fusion of different techniques can lead to improved accuracy and reliability in indoor localization.

2.2.4 Comparison of Indoor Localization Techniques

Each indoor localization technique has its own advantages and limitations, and the selection of a technique depends on the specific requirements of the application, including the desired localization accuracy, system cost, and environmental constraints [27, 49]. It's crucial to have a comprehensive understanding of these techniques for their optimal use in different indoor scenarios.

A comparison of various indoor positioning technologies in terms of accuracy, cost of implementation, and power consumption are shown in Figure X (from [23]).

Global Positioning System (GPS), although highly effective in outdoor environments, is often unsuitable for indoor localization due to its large error range and difficulty penetrating physical structures such as walls. Its positioning accuracy can degrade significantly indoors, and in certain cases, this might lead to placing a user even in the wrong room.

WiFi-based positioning, on the other hand, is constrained by its environmental requirements. Since WiFi access points are power-hungry, they need to be connected to a power

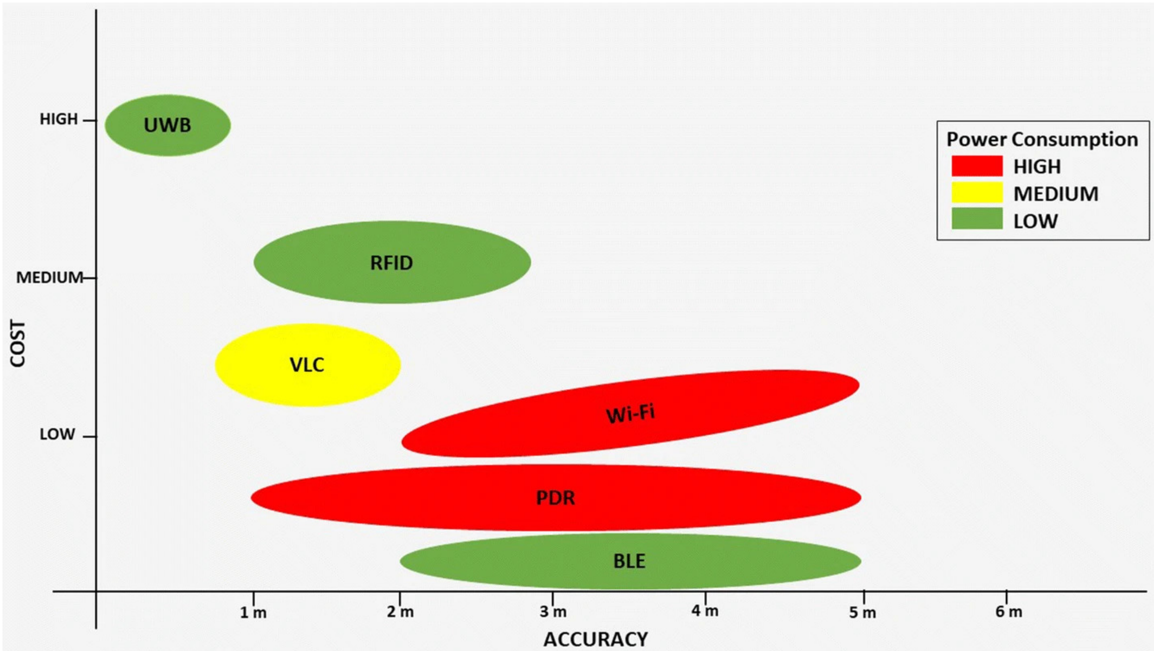

Figure 2.1: Comparison of various indoor positioning technologies in terms of accuracy, cost of implementation, and power consumption. Figure reproduced from [23].

source at all times. This requirement poses challenges for deployment in dynamic environments like construction sites, where access to consistent power may not be guaranteed. Furthermore, the cost of deploying and maintaining a WiFi infrastructure may be prohibitive for many applications.

Pedestrian Dead Reckoning (PDR), Ultra-Wideband (UWB), and Visible Light Communication (VLC) are other notable indoor localization techniques. However, PDR requires inertial sensors that may not be feasible in a construction setting due to the dynamic nature of the environment and frequent obstructions. UWB, while offering high accuracy, demands a more significant power supply and can be cost-prohibitive, making it less ideal for a construction site. VLC, requiring line-of-sight for effective operation, faces challenges in a construction environment where obstructions are common and lighting conditions are variable.

For the purposes of this project, Bluetooth-based positioning was chosen due to its balance between localization accuracy, cost, and ease of deployment. Unlike WiFi, Bluetooth low-energy (BLE) devices consume less power and are easy to install, making them an attractive choice for construction site environments. Moreover, they provide a satisfactory level of accuracy that is adequate for most indoor localization tasks.

Indoor environments, especially construction sites where conditions are constantly changing, present unique challenges to localization techniques. Non-line-of-sight scenarios can significantly affect the accuracy of most methods except for received signal strength (RSS) techniques. These scenarios can occur when the direct path between the transmitter and receiver is obstructed, causing the signal to be received via reflection, diffraction, or scattering. These multipath effects can lead to inaccurate position estimations, posing a significant challenge for indoor localization.

This is one of the primary reasons we have chosen to employ fingerprinting in our work. Fingerprinting-based techniques, which rely on RSS, have been found to be more robust to such issues. They leverage the unique signal characteristics of each location to create a "fingerprint" of the environment, which can then be used for accurate positioning, even in dynamic and challenging indoor environments like construction sites.

In the following sections, we will delve deeper into the application of Bluetooth technology, specifically Bluetooth Low Energy (BLE) devices, and the fingerprinting technique for indoor localization in construction sites.

2.3 Bluetooth Low Energy (BLE) Devices

BLE devices have become a popular choice for indoor localization due to their low power consumption and wide signal range [13, 25, 34, 37]. BLE technology, specifically, is becoming more integrated into indoor positioning due to its low energy consumption and capability of providing reliable location estimates [33].

2.3.1 Overview of BLE Technology

BLE technology, a power-efficient variant of classic Bluetooth technology, is designed for short-range communication between devices. It's widely used in indoor positioning systems, often leveraging RSSI measurements to estimate location. Mathematical filtering functions like median, mode, single-direction outlier removal, shifting, and feedback filtering are used to improve accuracy [13, 25, 34, 37]. It enables a wide variety of applications in localization, particularly due to its ability to be easily integrated into devices and systems [33].

2.3.2 Advantages of BLE Devices in Localization

BLE devices offer several advantages in localization, including low power consumption, ease of deployment, and the ability to provide high-accuracy positioning through mechanisms like RSSI-based positioning. Additionally, they can employ mathematical filtering functions for improved accuracy [13, 25, 33, 34, 37]. BLE technology allows for improved positioning accuracy in indoor environments, often providing a more reliable solution than GPS in these contexts [50].

2.3.3 Limitations and Challenges

Despite their advantages, BLE devices also have limitations such as signal instability due to environmental factors, and the need for careful placement of beacons to ensure accurate positioning. Additionally, BLE technology still faces challenges related to signal fluctuations and the impact of physical obstructions on signal strength, which can influence the accuracy of positioning systems [32]. Methods for dynamically estimating propagation models can be employed to improve accuracy in challenging environments [25, 32, 34, 37].

Figure 2.2 illustrates the fluctuations in RSSI values received from a beacon at a fixed distance, highlighting the instability of the BLE signals.

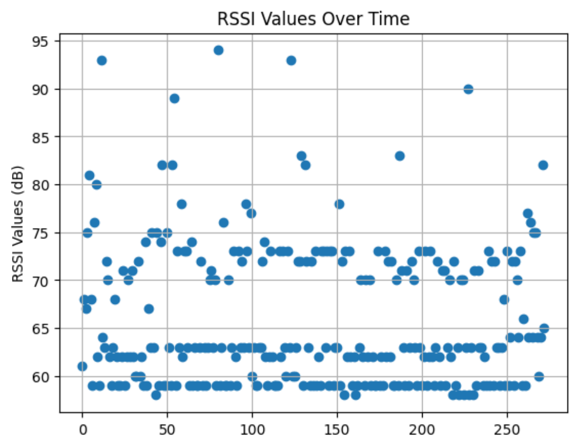

Figure 2.2: Fluctuations in RSSI from a BLE beacon at a fixed distance.

2.3.4 Applications in Construction Site Safety

BLE devices can be used to significantly contribute to construction site safety through indoor localization. They can track the location of workers, machinery, and potentially hazardous zones, helping to prevent accidents and ensure efficient operations [13, 25, 34, 37]. By integrating BLE technology with other systems, it is possible to develop comprehensive safety solutions that enhance awareness and responsiveness in the event of potential hazards or accidents [32].

2.4 Fingerprinting in Indoor Localization

Fingerprinting is a popular technique for indoor localization that uses the unique signal characteristics of different locations to create a map of the indoor environment [11, 12, 17]. Indoor positioning and object locating systems, in general, use a variety of technologies, among which fingerprinting based on received signal strength (RSS) has been shown to provide promising results [22, 50].

2.4.1 Overview of Fingerprinting Technique

Indoor localization through fingerprinting leverages the unique signature of the received signal strengths at different locations from multiple wireless transmitters, typically WiFi access points or Bluetooth beacons [6, 11, 17]. The two major phases involved are the offline and online phases, also known as the learning and positioning phases, respectively. During the offline phase, a radio map of the environment is created by measuring the signal strengths at different reference points. This map is then used during the online phase to estimate the position of a device based on the measured signal strengths from the transmitters [11].

2.4.2 Fingerprinting Process

Offline Phase: Data Collection and Radio Map Creation

The offline phase, also known as the learning phase, involves the data collection and processing stage [11, 17]. During this phase, the location of the device is known, and the signal strengths from various transmitters (e.g., WiFi access points and Bluetooth beacons) are recorded at multiple reference points within the environment [11]. A reference point in this context is a specific, predetermined location within the environment where signal strength measurements are taken. These reference points are strategically selected to cover the area of interest effectively and provide a comprehensive dataset for the radio map.

The received signal strength (RSS) from each transmitter is recorded into a vector $v_i = [v_{i1}, v_{i2}, ..., v_{in}]$ where i denotes the reference point and n is the number of transmitters.

A radio map is then created by associating each vector v_i with its corresponding location. Thus, the radio map, represented as a matrix M, is constructed as follows:

$$M = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}$$

where m is the number of reference points and each row in the matrix corresponds to a vector v_i. These reference points are crucial as they form the basis for the fingerprinting technique, wherein the unique signal characteristics at each point are used to create a distinct "fingerprint" that aids in the accurate positioning of devices during the online phase.

The more detailed the radio map, the higher the positioning accuracy that can be achieved in the online phase [17]. This phase can also benefit from the use of intelligent algorithms to optimize data collection and processing [45, 50].

Online Phase: Position Estimation

During the online phase, also known as the positioning phase, the device measures the signal strengths from the surrounding transmitters [11, 17]. The measured signal strengths are represented as a vector $v_o = [v_{o1}, v_{o2}, ..., v_{on}]$.

These measurements are then compared to the radio map created during the offline phase. The comparison is often performed using a similarity metric such as the Euclidean distance. For each vector v_i in the radio map, the distance $d(v_i, v_o)$ is calculated. The position of the device is estimated as the location associated with the vector that has the smallest distance to v_o, i.e., $\text{argmin}_{v_i \in M} d(v_i, v_o)$

This process can leverage various machine learning techniques such as clustering algorithms of neighboring reference points for more accurate positioning [45, 50]. The use of such techniques can further refine the position estimation, accounting for factors such as signal interference and changes in the environment.

Figure 2.3 provides an overview of the fingerprinting process, including the offline and online phases (from [23]).

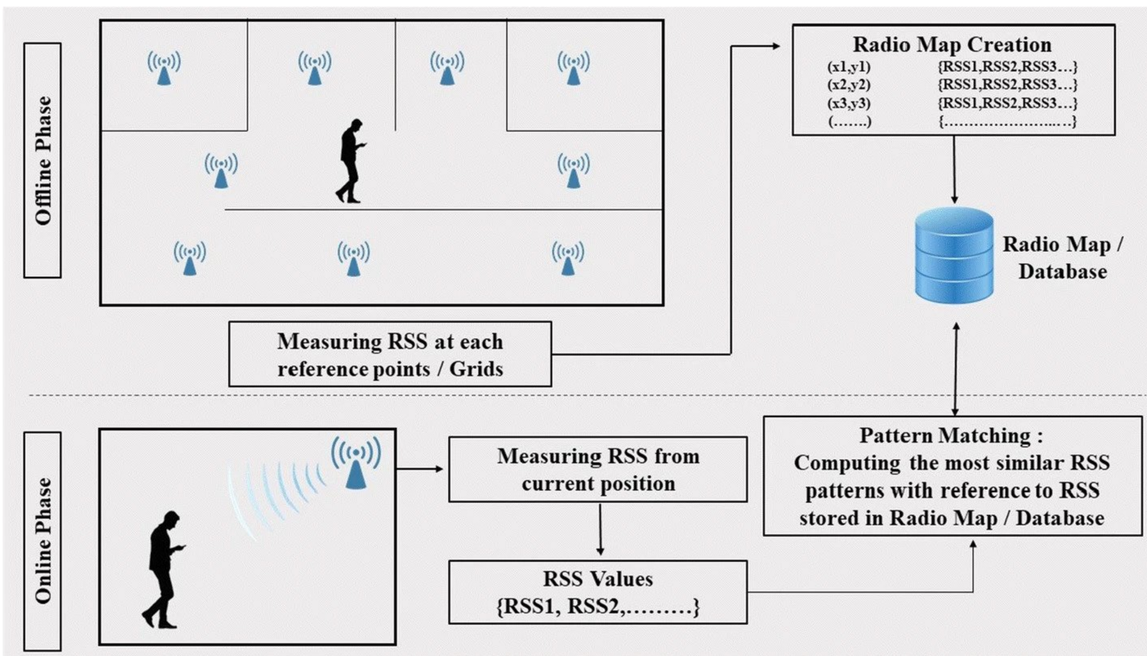

Figure 2.3: Overview of the fingerprinting process. Figure reproduced from [23].

2.4.3 Advantages and Limitations

The primary advantage of fingerprinting techniques is their ability to provide relatively high accuracy in complex indoor environments where methods such as GPS fail to deliver [17]. The accuracy achievable through fingerprinting largely depends on several factors, including the density of reference points, the stability of the signal environment, and the algorithm used for location estimation. Typically, fingerprinting methods can achieve an accuracy within a few meters, and in some cases, with a very dense network of reference points and a stable signal environment, the accuracy can be further refined to within a meter [17, 50].

However, it is essential to note that the primary limitation of fingerprinting methods is the need for extensive and labor-intensive data collection during the offline phase, which can be challenging for large or dynamic environments [17, 50]. Additionally, changes in the environment or in the transmitter characteristics can result in outdated radio maps, necessitating frequent updates [17]. Therefore, while fingerprinting techniques can offer high accuracy, this is contingent on the initial and ongoing efforts to maintain an up-to-date and comprehensive dataset.

2.4.4 Fingerprinting in Construction Site Safety Applications

Fingerprinting techniques, employing received signal strength (RSS) from wireless devices, hold significant promise for enhancing safety in construction sites by enabling precise indoor localization. This application can facilitate tracking workers and equipment, identifying hazardous areas, and promptly alerting workers to potential safety risks [12, 17]. However, the dynamic and ever-changing nature of construction sites necessitates specific adaptations of fingerprinting methods to ensure their effective application in such environments [12].

- **Dynamic Updating of Fingerprint Database:** Given the constantly evolving layout of construction sites, it's crucial to regularly update the fingerprint database. This adaptation ensures that the localization system accurately reflects the current site conditions, thereby maintaining the reliability of location-based safety alerts.

- **Enhanced Signal Processing:** Construction sites are replete with factors that can interfere with signal transmission, such as heavy machinery and varying building materials. Advanced signal processing techniques are therefore essential to mitigate the effects of noise and multipath propagation on the accuracy of RSS readings.

- **Increased Reference Point Density:** The complex layout of construction sites, coupled with their frequent modifications, necessitates a higher density of reference points for accurate indoor localization. More BLE beacons or similar transmitters should be strategically placed throughout the site to ensure comprehensive coverage and maintain high localization accuracy.

- **Machine Learning Adaptations:** Adapting machine learning algorithms to better handle the variability and unpredictability of construction environments is crucial. Algorithms that are robust to environmental changes or that can adapt over time will significantly enhance the system's reliability in accurately identifying and tracking locations within the site.

In summary, while fingerprinting offers a promising solution for indoor localization in construction sites, adapting it to the unique challenges of these environments is key to its successful implementation. These adaptations focus on ensuring the system's accuracy, reliability, and responsiveness to the dynamic conditions of construction sites.

2.5 Machine Learning and Clustering Algorithms in Indoor Localization

2.5.1 Role of Machine Learning and Clustering in Indoor Localization

Machine learning (ML) has emerged as an essential tool for enhancing safety management in construction sites, by providing solutions for risk prediction, incident identification, and real-time monitoring [12]. In the context of indoor localization, ML algorithms can refine position estimates, improve accuracy, and predict potential safety incidents by identifying patterns in the positioning data [16, 48]. Specifically, in the fingerprinting process, the system can learn from the collected Received Signal Strength (RSS) data for accurate location estimation [47, 50]. This technique is being adopted in various systems, such as mobile phones for floor localization, to determine the current floor level using WiFi access points [21]. Clustering algorithms such as k-means and spectral clustering can identify underlying structures in the data, improving the accuracy and efficiency of the indoor positioning system [4, 35, 45].

2.5.2 Supervised Learning: From k-Nearest Neighbors (k-NN) to Neural Networks

Supervised learning algorithms, such as k-NN and neural networks, are frequently used in the fingerprinting process of indoor localization. The k-NN algorithm, known for its simplicity and robustness, estimates the position of a device based on the k closest reference points in the radio map [12, 44]. The efficacy of this method, however, is dependent on factors such as the chosen value of k, the distance metric used, and the density and distribution of reference points [51]. On the other hand, neural networks and Extreme Learning Machine (ELM) techniques offer the potential for higher accuracy, given adequate data and computational resources [1, 18, 19, 28, 46]. Other supervised learning algorithms, such as Decision Trees, Support Vector Machines, and Support Vector Regression(SVR), can also be employed to enhance the accuracy and reliability of indoor localization systems and to detect anomalies or potential safety threats based on the collected data [42, 50].

2.5.3 Unsupervised Learning: Clustering Techniques and Their Applications

Unsupervised learning plays a crucial role in indoor localization, with clustering techniques like spectral clustering offering significant enhancements [35]. These techniques can be utilized during the offline phase to cluster reference points based on their RSS readings, thereby potentially enhancing the efficiency of the subsequent online phase and improving the overall performance of the localization system [4, 29, 45]. Advanced techniques such as the Fusing A Group Of FingerprinTs (FAGOT) method and Generalized Principal Component Analysis (GPCA) can exploit the intrinsic complementarity among different types of fingerprints for improved accuracy [16, 41]. Triangle and Centroid localization algorithms based on distance compensation show promise in practical localization scenarios [40]. Notably, the unsupervised clustering technique applied to RF fingerprint data can improve localization accuracy by taking into account architectural structures such as floors, doors, and aisles, as demonstrated in multi-floor indoor environments [8]. Hierarchical clustering methods also provide a promising avenue for location estimation, as they integrate RSS data from both WiFi and GSM networks and partition the RSS space for optimal transmitter selection [26]. The potential to integrate these unsupervised learning methods with filtering techniques like the Kalman filter can yield further improvements in localization accuracy [4].

2.5.4 Comparison: Supervised vs Unsupervised Learning in Localization

Both supervised and unsupervised learning approaches offer unique advantages in indoor localization, and their application depends on the specifics of the task. Supervised learning proves effective when labeled data is available, as is often the case during the offline phase of fingerprinting [50]. In contrast, unsupervised learning can be particularly beneficial when the objective is to identify structures or patterns in the data, rather than predicting a specific outcome [29, 45].

In terms of supervised learning algorithms, k-nearest Neighbors (k-NN) and neural networks stand out in the domain of indoor localization, more specifically in the fingerprinting process [12]. An improved version of k-NN, the Kernel Difference-weighted k-Nearest Neighbor (KDF-KNN), optimizes the weight distribution of k neighbors through a constrained optimization problem, enhancing the algorithm's accuracy for pattern classification [51]. Another improved dynamic prediction fingerprint localization algorithm has been proposed based on KNN, which greatly improves location accuracy [44].

On the other hand, neural networks offer the potential for higher accuracy, conditional on sufficient data and computational power. For instance, a spectral clustering and weighted backpropagation neural network (SWBN) method was proposed for 3D indoor localization, significantly reducing localization median error and training time compared to the Backpropagation Neural Network (BPNN) and k-NN methods [35]. Furthermore, Deep Neural Networks (DNN) demonstrate remarkable proficiency in learning discriminative features from noisy wireless signal measurements [28]. In multi-floor environments, the extreme learning machine (ELM) localization technique exhibits high-precision localization performance by leveraging multiple individual ELMs for floors and geographically formed data clusters for each floor, proving its ability to outperform existing schemes [46].

In the realm of unsupervised learning, clustering techniques show promise for indoor localization. Algorithms such as K-means are frequently employed to group the reference points based on their RSS readings, which can enhance the efficiency of the subsequent online phase. A K-means-based method for floor estimation via fingerprint clustering of WiFi and various other positioning sensor outputs significantly improves the complexity and speed of floor detection while achieving accuracy close to traditional fingerprinting methods

[36]. In an effort to improve the accuracy of the UWB sensor-based indoor positioning system, the K-means algorithm with an additional average silhouette method has been used. This approach helps to define the optimal number of clusters based on the silhouette coefficient, resulting in a more accurate grouping of UWB data. The integration of the Kalman filter further refines the system by reducing noise and interference in the signal. Consequently, the system significantly reduces average localization error when combining the K-means algorithm with the Kalman filter [4]. Furthermore, a WiFi localization approach has been proposed that fuses a group of fingerprints for higher positioning accuracy in indoor environments, showing better performance than other systems [16, 45]. However, conventional clustering methods usually require the direct or indirect predefinition of the clustering pattern and the number of clusters, which might lead to unsatisfactory estimation accuracy if improperly selected. To mitigate this issue, the Improved Clustering Algorithm of Neighboring Reference Points Based on KNN, for instance, incorporates the k-means clustering algorithm to analyze the geometric proximity between reference points and the test point in the online phase, resulting in significantly improved positioning accuracy [45]. Alternatively, the Cluster Filtered KNN (CFK) scheme employs clustering techniques to categorize neighbors into different clusters and selects one as a delegate, leading to better performance compared to the standard k-NN algorithm [29].

Supervised and unsupervised learning algorithms each carry their strengths, and their application depends on the specific task at hand. Supervised learning shows efficacy when labeled data is available, as in the case of the offline phase of fingerprinting. Unsupervised learning proves advantageous when the goal is to identify structures or patterns in the data rather than predicting a specific outcome. Also, consideration of architectural aspects and controlled computational complexity significantly improves the accuracy of Wi-Fi multi-floor indoor positioning, as demonstrated by a combined strategy employing unsupervised clustering and majority voting committees of backpropagation artificial neural networks [8].

2.5.5 Limitations and Challenges of ML Algorithms in Construction Sites

Despite the potential of ML and clustering in construction safety management, there are several limitations and challenges to consider. For instance, these algorithms often require significant amounts of labeled data for training, which can be challenging to acquire in dynamic construction environments [50]. Moreover, the predictive performance of these algorithms can be influenced by numerous factors, including the quality of the data, the selection of features, and the configuration of the algorithms [10, 51].

Probabilistic techniques show good performance in Wireless Local Area Network(WLAN) location estimation, but they require a large number of training samples for calibration, which can lead to high offline manual effort [9]. Techniques that dynamically estimate the propagation models that best fit the propagation environments using real-time RSS measurements can improve the accuracy of positioning [32]. Some algorithms, such as the weighted fusion algorithm, can also improve the accuracy of WiFi indoor positioning, but they still face challenges related to WiFi signal fluctuation and differences between probability values [30].

Beacons using Bluetooth low-energy (BLE) technology offer advantages such as low power consumption, miniaturization, wide signal range, and low cost, but their indoor positioning accuracy can be affected by noise, motion, and fading, all of which are characteristics of a Bluetooth signal and depend on the installation location [25]. Techniques like using an extended Kalman filter to process input data including noise can enhance the accuracy of beacon-based indoor positioning technology [25]. Fusion of techniques like trilateration and dead reckoning can achieve high positioning accuracy, with experimental results showing an accuracy of less than one meter, but the efficacy varies depending on the situation [37].

Evaluating the performance of different algorithms is crucial to select the most suitable one for a specific indoor localization task. This comparison can be carried out using a range of metrics, including accuracy, precision, recall, and F1-score for classification tasks, silhouette coefficient, Davies-Bouldin index, or within-cluster sum of squares (WCSS) for clustering tasks. Cross-validation techniques can also be employed to provide a robust assessment of an algorithm's performance.

Notably, the optimal algorithm for a specific task may change as new data is collected or the application requirements evolve. Therefore, maintaining a flexible approach, routinely reassessing the performance of the chosen algorithms, and considering the integration of multiple algorithms into a single system could yield better results. It's imperative that these factors be continually monitored and adjusted in line with the changing dynamics of the construction site [51].

Chapter 3

System Architecture and Construction Site Zoning

3.1 Overview

This thesis presents an innovative system designed to enhance safety management in construction environments. At its core, the system integrates Bluetooth Low Energy (BLE) technology, a sophisticated Android application, and advanced server-side components. The system is built upon the Framework for Internal Navigation and Discovery (FIND), which has been adapted and customized to meet the specific requirements of construction site safety. The primary aim is to leverage modern technological advancements to create a more secure and efficient construction workplace.

By harnessing the capabilities of BLE devices for precise indoor localization, the system provides real-time updates on worker positions and movements. This is crucial in dynamic environments like construction sites, where the safety of personnel is constantly at risk due to the nature of the work and the presence of various hazards.

The Android application plays a pivotal role in the system, acting as an interface between the workers and the technology. It is designed for ease of use, ensuring that workers can effortlessly interact with the system without any significant disruption to their workflow. Through this application, workers receive timely alerts and notifications about potential hazards, contributing to proactive safety measures.

The server-side component, intricately designed using a pre-existing framework adapted to the project's needs, serves multiple functions. It processes the raw tracking data using machine learning algorithms to pinpoint a worker's location, stores historical data for retrospective analyses, and most crucially, cross-references the worker's location with pre-defined zones in the construction site. These zones, classified based on safety parameters, dictate specific safety protocols. The system instantly detects any protocol breaches, like a worker entering a hazardous zone without the necessary Personal Protective Equipment (PPE).

Upon such detection, the system immediately alerts the worker via the Android app and flags the event for site supervisors or safety officers through the web interface.

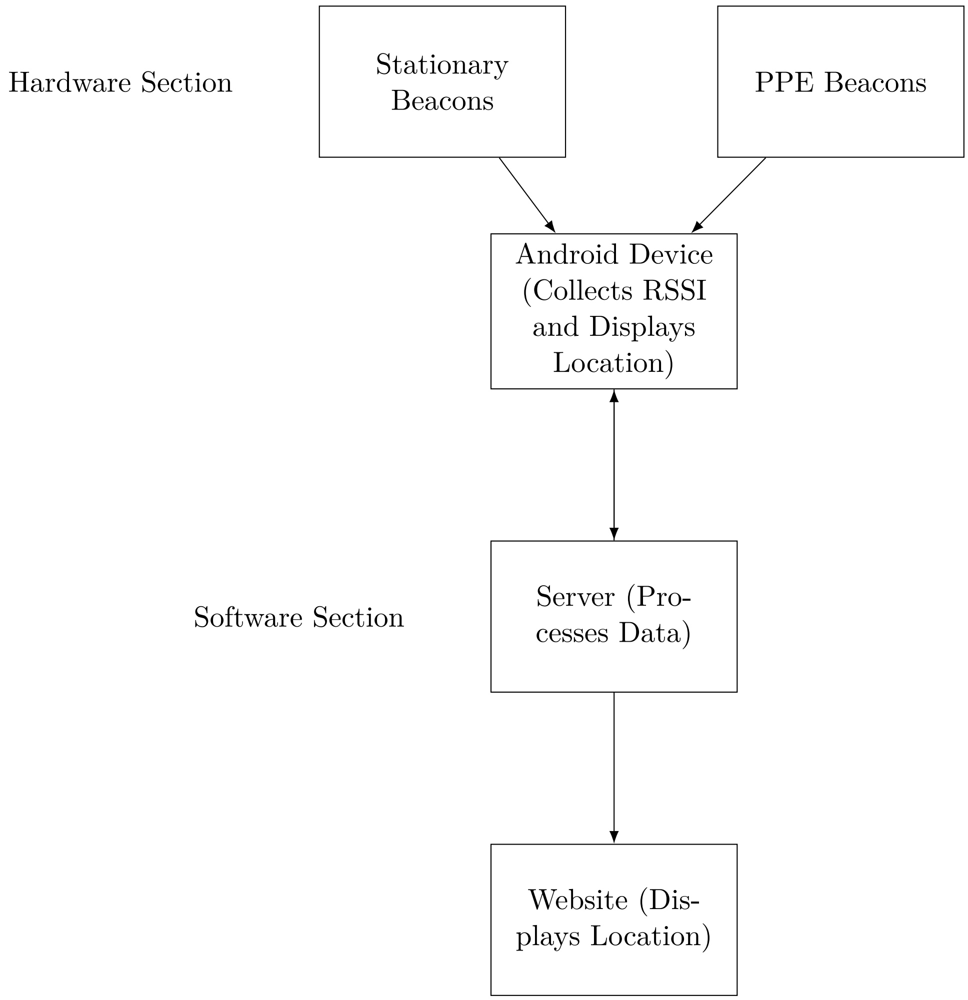

Figure 3.1: High-level schematic representation of the indoor localization project, illustrating the integration of hardware and software components, and the bidirectional communication between the server and Android device.

This architecture's brilliance lies in its scalability, real-time monitoring capability, and adaptability. The subsequent sections delve deeper into each component's specifics, offering insights into their design, operation, and the rationale behind the choices made during the system's development.

3.2 BLE Device Integration

Bluetooth Low Energy (BLE) beacons are tiny devices that transmit small amounts of data over short distances using Bluetooth technology. Their primary function is to broadcast identifying information to smart devices within their range.

In the realm of this project, BLE beacons play a pivotal role. Construction sites are dynamic, with workers moving around constantly amid potentially dangerous machinery and conditions. Accurate tracking of each worker's location is paramount to ensure their safety. BLE beacons provide a cost-effective, energy-efficient, and accurate solution to address this need.

Considering the project's goals, the setup was designed meticulously. The environment in focus was a room of dimensions 5 m x 2 m, in which 10 BLE beacons were strategically placed. Several factors influenced this decision:

- **Coverage**: For precise tracking, it's essential that there are no blind spots in the room. Ten beacons ensure comprehensive coverage of the entire area.

- **Obstructions**: Construction sites often have a myriad of obstructions, from machinery to unfinished structures. More beacons mean the system can locate a worker's position more accurately even if some beacons' signals are obstructed.

- **Signal Redundancy**: In the unfortunate event a beacon fails or its signal gets interrupted, having multiple beacons ensures that the system remains functional and continues to track workers efficiently.

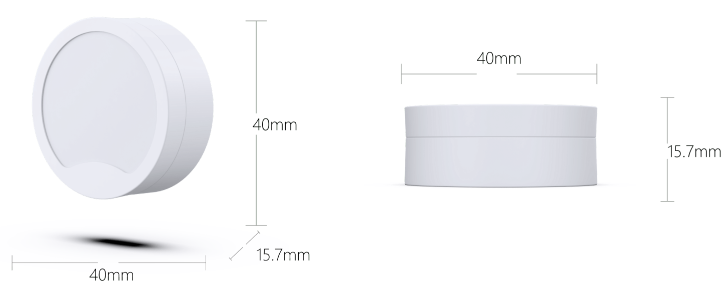

Figure 3.2: Bluetooth long range waterproof beacon K5P.

The placement and distance of the beacons were also of prime importance. In our setup, beacons were affixed to the walls. While the distances between them varied, they were typically spaced 1.5 m apart. The rationale behind this spacing includes:

- **Signal Strength**: Spacing them too far apart might weaken the signal strength, reducing tracking accuracy.

- **Overcoming Obstructions**: Given that construction sites are rife with obstructions, closer spacing ensures that even if one beacon's signal is blocked, others can compensate for it.

- **Power Consumption**: While BLE beacons are energy efficient, having them too closely spaced could result in unnecessary energy consumption. The 1.5 m spacing represents a balanced approach.

Diving deeper into the specifics of the beacons used:

- They are **IP67 Waterproof**. Given the unpredictable nature of construction sites where exposure to rain or other elements is frequent, this feature ensures that the beacons remain functional in all conditions.
- Their compact size (**dia. 40mm*15mm**) means they can be conveniently placed without being obtrusive.
- Powered by an **easy-to-replace CR2477 battery**, they boast a long life of up to **4 years**.
- They are **fully compatible with BLE 5.0**, which is known for its Long Range feature.
- With a **300 meters long range** and a maximum **8dBm TX power**, they are ideal for construction sites.

3.2.1 Types of Beacons: Stations and PPEs

Within the architecture of the Real-time Safety Alerting System, beacons have been differentiated into two primary types, each serving its distinct function:

1. **Station Beacons (Access Points)**: These beacons are strategically fixed at various locations within the construction site. They act as stationary reference points, broadcasting their identifying signals continuously. The placement and density of these station beacons ensure that the entire construction site is adequately covered, allowing for accurate location triangulation irrespective of where a worker might be within the premises.
2. **Personal Protective Equipment (PPE) Beacons**: Unlike the stationary nature of the station beacons, PPE beacons are mobile and are attached to various Personal Protective Equipment pieces like helmets, vests, or boots. These beacons transmit their unique identifiers, enabling the system to recognize not just where a worker is but also if they are equipped with the requisite safety gear for that zone. This dual beacon system—combining stationary with mobile—ensures a robust and dynamic safety compliance check.

The coalescence of these two types of beacons provides the system with a holistic view of the construction environment. Station beacons provide the geographical context, while

PPE beacons give insights into the workers' safety gear adherence. Together, they form the backbone of the system's real-time monitoring and safety compliance checks.

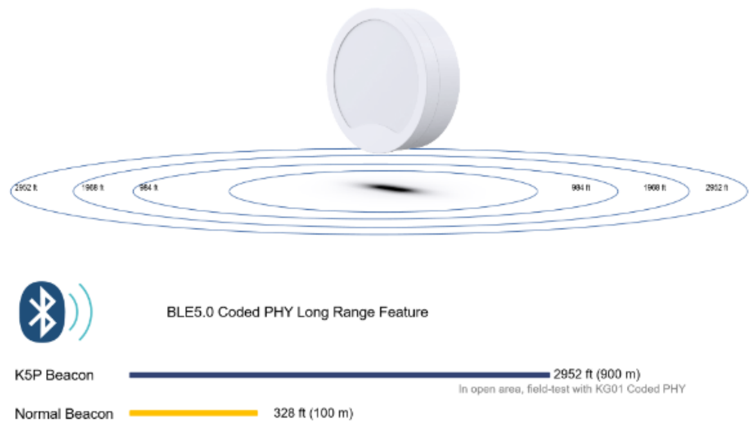

Figure 3.3: Bluetooth long range waterproof beacon K5P and its transmission distances.

3.3 Android Application

A key aspect of the Real-time Safety Alerting System for Construction Sites is the mobile application. Built upon the foundation of the FIND framework, this Android app is a focal point for interfacing with BLE beacons, enabling precise location tracking of workers on the construction site.

3.3.1 Core Features of the Android App

The Android app predominantly operates in two modes:

- **Learning Mode**: Here, the app goes through the offline phase of the fingerprinting process. The user inputs the name of the location, essentially marking it as a reference point. The application then proceeds to collect RSSI values from the available beacons. It is recommended for the user to pause for about 1-2 minutes in each location to ensure an adequate range of RSSI values is captured.

- **Tracking Mode**: This corresponds to the online phase of fingerprinting. In this mode, the app continuously tracks the worker's location based on the RSSI values of nearby beacons.

3.3.2 Challenges and Overcoming Them

One of the initial challenges faced while working on the Android app was the non-functionality of the existing code from the FIND3 framework. The framework's codebase utilized scan-

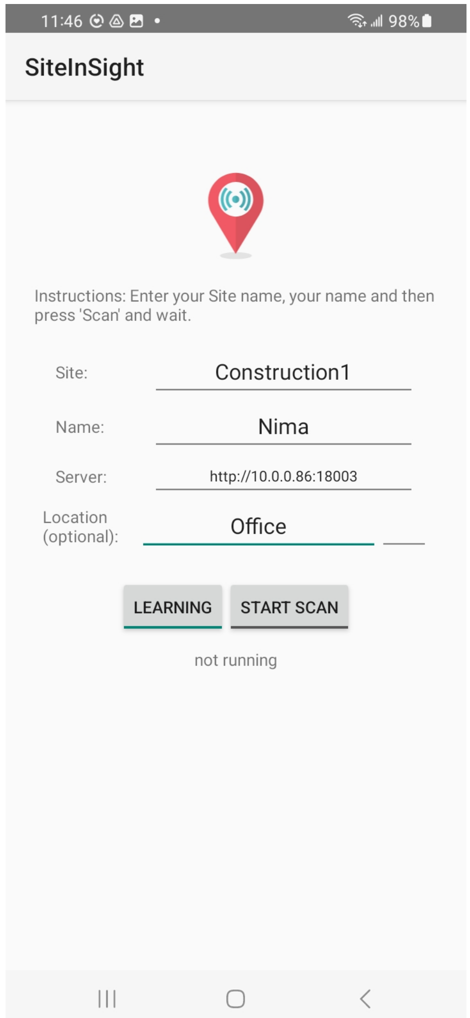

Figure 3.4: Screenshot of the Learning Mode interface.

ning methodologies for WiFi, which recent Android updates have rendered obsolete. Due to the restrictions imposed by these updates, the WiFi scanning code could not be executed, posing a significant roadblock in the app's development.

To address this issue, the Android code was redeveloped, focusing exclusively on scanning Bluetooth signals. This rework not only ensured that the app was compliant with the latest Android guidelines but also allowed for a more specialized and efficient approach tailored for BLE beacons. While this process was undoubtedly time-consuming, it resulted in a more robust and reliable application that was better aligned with the project's requirements and the evolving landscape of Android development.

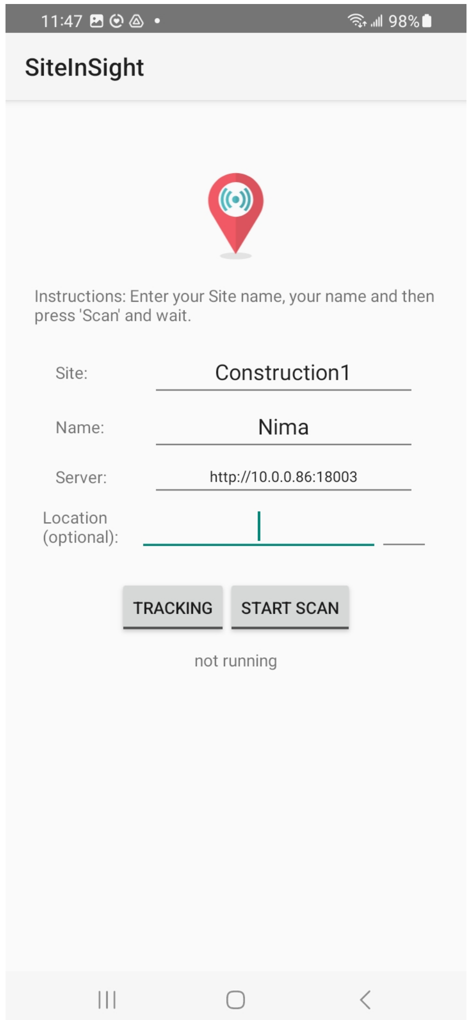

Figure 3.5: Screenshot of the Tracking Mode interface.

3.3.3 Customizations for Enhanced Compatibility

While the base application from the FIND3 framework provided a robust starting point, customizations were essential to tailor the app to the unique requirements of construction sites:

- **RSSI Cutoff Value**: Given the specific architecture and layout of construction sites, not all beacon signals are relevant for all locations. By setting a cutoff value of -70 db, the app ensures that only the RSSI values of beacons in the immediate vicinity (presumably in the same room) are considered. This approach mitigates potential inaccuracies that might arise from distant beacon signals.

- **Safety Notifications**: A critical change in the app is its ability to push notifications in real time when a worker enters a designated danger zone. Such an alert mechanism can be crucial for preventing accidents and ensuring compliance with safety protocols.

- **Data Transmission Logic**: A significant modification was made concerning how frequently the app sends data to the server.
  - *Learning Mode*: In the learning mode, the app is configured to send data to the server each time a new RSSI value is detected. This frequent data transmission is essential for the learning phase, as it allows the system to collect a rich dataset of RSSI values from various locations, thereby enhancing the accuracy of the fingerprinting database.
  - *Tracking Mode*: The tracking mode, however, adopts a different approach for data transmission. The app accumulates data and transmits it to the server only after every four new RSSI signals are received. This strategy is implemented for two primary reasons. Firstly, it allows for a significant change in the contents of the `LinkedHashMap` that stores the latest BLE signals, ensuring that the data sent reflects a meaningful variation in the user's environment. Secondly, it reduces the server's load by preventing it from being inundated with frequent, minor updates, thereby optimizing overall system performance. Additionally, this methodology helps in balancing the need for timely data transmission without introducing excessive delays, ensuring that the system remains responsive and accurate in tracking the user's movements within the construction site.

3.3.4 Deep Dive into the Code

Central to the Android application's functionality is its ability to conduct fingerprinting based on Bluetooth signals. This process involves differentiating these signals through the unique names of the BLE beacons emitting them.

Within the app, there is a maintained LinkedHashMap of bluetoothResults which keeps track of RSSI values. Whenever a new beacon's RSSI value is detected, it checks if the beacon is already present in the map. If so, the existing value is averaged with the new one, providing a smoothed estimate. If the beacon is new, and if the map already has ten beacons, the oldest beacon value is removed to make space for the new one.

The variable `counter_n` keeps a count of new RSSI values. In Learning Mode, data is sent to the server for every new value. In Tracking Mode, data is dispatched for every fourth value.

The logic behind sending data every time in Learning Mode, even with just a single new RSSI value, ensures faster and more dynamic updates. This is in line with the essence of the Learning Mode – to quickly understand and adapt to the environment. On the other hand, in the Tracking Mode, the decision to transmit data once every four values ensures a balance between real-time tracking and not overburdening the server with too many frequent updates. Additionally, the app's ability to average RSSI values from an existing beacon ensures smoother location tracking, particularly useful in mitigating the effects of transient signal fluctuations.

This section provides a comprehensive insight into the Android application's structure, operation, and custom modifications. The blend of the base FIND3 framework with tailored adjustments ensures a robust and precise tracking system tailored for the nuances of construction sites.

3.4 Server-side Components

3.4.1 Framework Selection

Choosing the right framework for a project of this scale and importance was pivotal. The underlying rationale for selecting an existing framework for the website stemmed from the various advantages it offered. Primarily, the chosen framework provided a base website template capable of handling site names, managing diverse construction sites, workers, and even the nomenclature of reference points.

The built-in structure of how the front end communicates with the back end, particularly the process of invoking APIs for Python code execution, proved to be beneficial. It allowed modifications in the machine-learning Python code and adaptations in the website template to better suit the requirements of a zoning-based construction site. We have replaced the map of the FIND framework with the ability to show the floorplan. Once a worker's location is determined server-side, the website seamlessly selects the appropriate floor plan corresponding to the worker's current floor level. Broadly, this framework offered a starting point that was closely aligned with our desired end product, saving considerable development time while providing a reliable foundation.

3.4.2 Real-time Updates

A pivotal enhancement in the augmented FIND framework was the incorporation of floorplans, transforming them into dynamic maps that visually represent workers' locations within the construction site. This innovative feature enables a seamless overlay of real-time positioning data onto the site's floorplan, offering a comprehensive and intuitive spatial perspective. The system operates efficiently in the background: each time the server confirms a worker's location, it immediately updates the corresponding position on the floorplan. To maintain up-to-the-minute accuracy, the front-end interface periodically retrieves these updated images, specifically every 30 seconds, thereby enabling supervisors and safety officers to track the movement of workers with near real-time precision. Such a feature significantly enhances situational awareness and reinforces safety protocols on the construction site.

3.4.3 Website Functionality and Data Management

Beyond real-time tracking, the website has been fortified with other functionalities essential for comprehensive site management. A crucial feature is data logging. The implemented Go code persistently logs data, ensuring a complete record of all locations a worker has visited. This historical data provides invaluable insights, especially in post-incident analyses or for routine safety audits.

3.4.4 Visual Aids

To offer a more tangible sense of the user interface and the ease of use it brings, here are some visual aids:

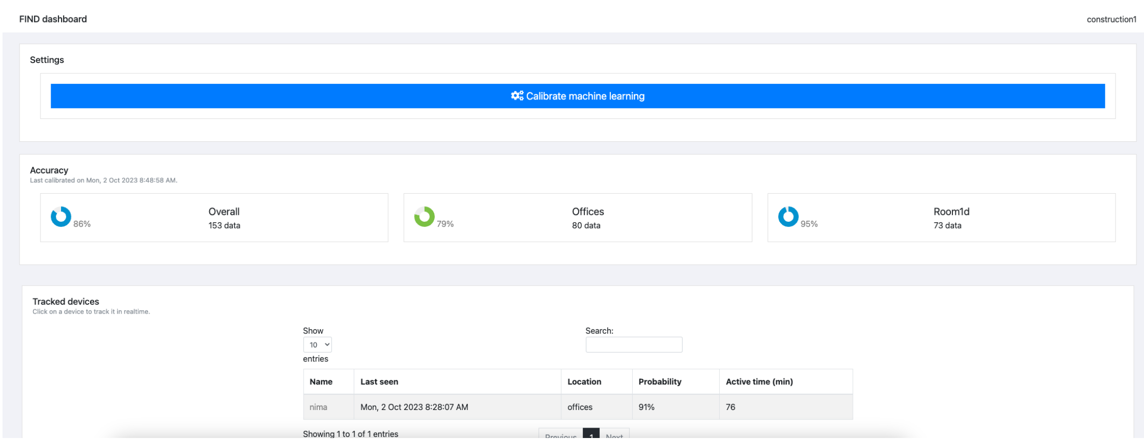

Figure 3.6: Screenshot of the Main Page.

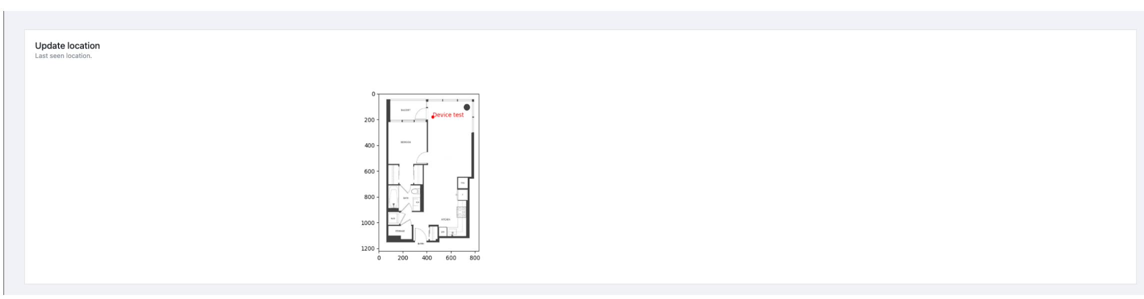

Figure 3.7: Floorplan with worker's real-time position.

3.5 Zone Division and Safety Requirements

3.5.1 Zoning Concept

In the construction industry, zoning serves as a paramount strategy to delineate different areas of a site based on the risks associated and safety precautions required. Through proper zoning, potential hazards are anticipated, and the necessary measures are put in place to ensure worker safety. Essentially, zones act as spatial guidelines for workers, outlining where they can move freely, where additional caution is required, and which areas mandate the use of personal protective equipment (PPE).

3.5.2 Types of Zones

To enhance clarity and streamline worker movements, various zones have been identified within construction sites. These zones are color-coded for easy identification:

1. **Common working zone (Green Zone)**: This zone, accessible to all workers, is characterized by areas with permanent guard rails. Workers can operate here without the need for specialized equipment, provided they maintain a distance of less than 3 metres (3m in Canada and 2m in Hong Kong) from the adjacent ground level.

2. **Controlled working at height zone (Purple Zone)**: Before entering this zone, workers must don appropriate PPE. The zone is marked by anchor points fixed at specific distances. Examples of such PPE include lifelines, fall arrestors, and safety harnesses. Typical areas under this zone are roof or floor edges where guard rails are absent or incomplete.

3. **Danger working at height zone (Red Zone)**: This zone signifies areas with the highest risk. It includes regions elevated more than 3 meters from the ground, where workers might resort to using ladders or any unauthorized working platform. It's crucial to note that platforms that haven't been inspected—at least every 21 calendar days as per Alberta regulations—or approved fall under this zone.

3.5.3 Personal Protective Equipment (PPE) Beacons

To further enhance safety when entering specific zones like the Purple Zone, employees can be equipped with PPE beacons. By sticking these beacons onto their PPE, a system can be devised to alert or deny access to workers if they aren't equipped with the requisite safety gear. This digital mechanism provides an added layer of security, ensuring only those with appropriate gear can access high-risk zones.

3.5.4 Visualization of Zones

For a clearer understanding of the zoning within a construction site, refer to the illustration below, which demarcates different zones in a typical building setup.

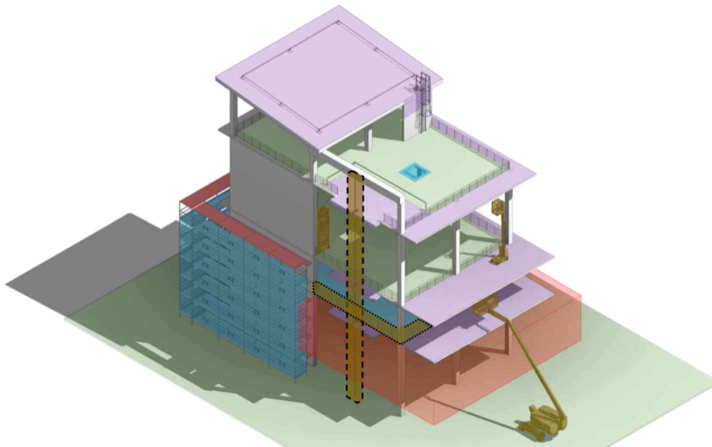

Figure 3.8: Sample floor-plan indicating various safety zones in a building.

3.6 Real-time Monitoring and Compliance and Alarm System

3.6.1 Monitoring in Real-Time

Central to the safety alerting system's efficacy is its capability to track and assess workers' locations in real time. This monitoring hinges on the reference points input during the learning mode of the application. When users demarcate a reference point, they also specify its color, which corresponds to the safety zones previously discussed.

For instance, when the system undertakes location analysis, it not only identifies the workers' exact position but also discerns the safety requirements associated with that zone based on the color of the location. Consequently, this allows the system to determine if the worker is adhering to the stipulated safety protocols or not. As an example, if a worker is detected within the Safe working at height zone (Blue Zone), the system will cross-reference to check if the individual is equipped with the required PPE. If not, the system will trigger an alert.

3.6.2 Compliance and Alarm System

In the event of a breach in safety compliance or any other potential hazard, the system is designed to immediately alert the involved parties.

- **Worker Alert**: The worker's phone plays a crucial role in providing instantaneous feedback. If a safety protocol is breached or if the worker enters a high-risk area without the necessary precautions, their phone will vibrate and display a visual alarm. This immediate feedback ensures that the worker can promptly rectify their position or equip the necessary safety gear.

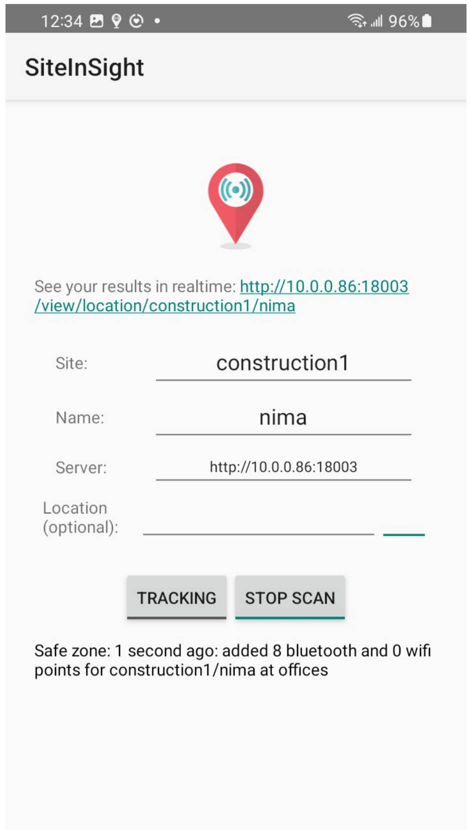

Figure 3.9: Example of a worker in a Safe Zone.

- **Employer Notification**: Parallelly, the system ensures that the employer or the site supervisor is also notified of the breach. This is achieved through visual alerts displayed on the website dashboard. This dual notification mechanism—informing both the worker and the supervisor—ensures rapid response and corrective action.

The essence of this monitoring and compliance system is to foster a culture of proactive safety. By integrating real-time monitoring with instantaneous alerts, the system goes beyond passive tracking, actively ensuring that every worker on the construction site operates within the ambit of safety guidelines.

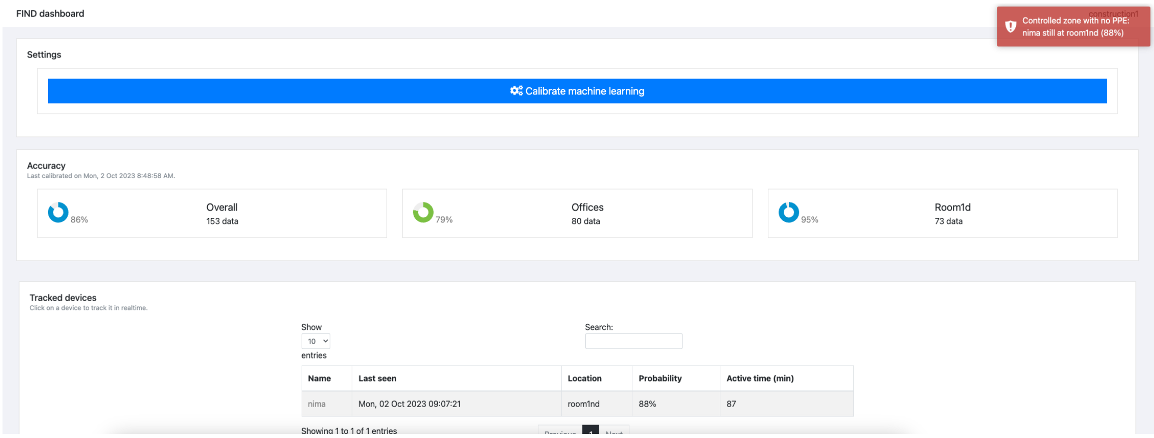

Figure 3.10: Example of a worker in a Blue Zone without the required PPE, notification on the Website.

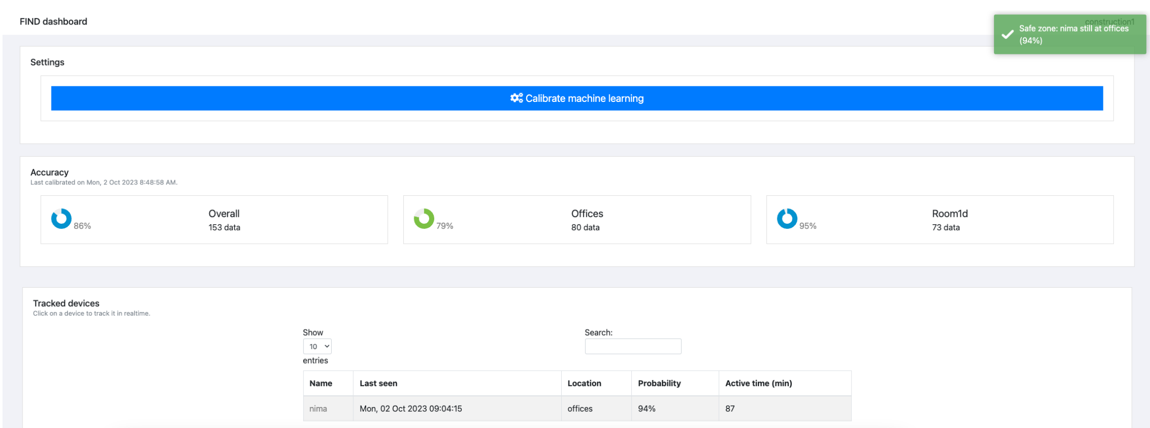

Figure 3.11: Example of a worker in a Green Zone, notification on the Website.

Chapter 4

Indoor Localization and Fingerprinting

4.1 Overview of Fingerprinting

Indoor localization has emerged as a pivotal technology, especially in environments where GPS signals are weak or non-existent. Fingerprinting is one of the prevailing techniques for indoor localization, offering a blend of accuracy and feasibility which makes it a prime choice in various applications like healthcare, retail, and industrial settings.

Fingerprinting-based localization primarily revolves around two phases: the offline phase and the online phase.

4.1.1 Offline Phase

The offline phase, often referred to as the training or calibration phase, is where a radio map of the environment is constructed. In this phase:

- Reference points (RPs) are defined across the area of interest.
- At each RP, the Received Signal Strength Indicator (RSSI) values from available signal sources like Wi-Fi access points or Bluetooth Low Energy (BLE) beacons are collected.
- The collected data, comprising of RSSI values and the corresponding coordinates of the RPs, is stored in a database. This database, often referred to as the radio map, encapsulates the fingerprint of the environment.

4.1.2 Online Phase

The online phase, also known as the tracking or localization phase, is where the real-time location of a device is estimated. In this phase:

- The device measures the RSSI values from the surrounding signal sources.

- These measured values are then compared with the radio map created during the offline phase.
- Various algorithms and techniques are employed to match the current RSSI readings with the stored fingerprints to estimate the most probable location of the device.

The essence of fingerprinting lies in its simplicity and the ability to work in complex indoor environments without requiring any additional hardware. The primary challenge, however, is maintaining the accuracy and reliability of the system amidst the dynamic nature of indoor environments. Factors such as interference, multipath fading, and hardware inconsistencies can significantly impact the system's performance. Overcoming these challenges necessitates a well-structured data collection process, a robust database, and efficient matching algorithms, the details of which are elucidated in the ensuing sections of this chapter.

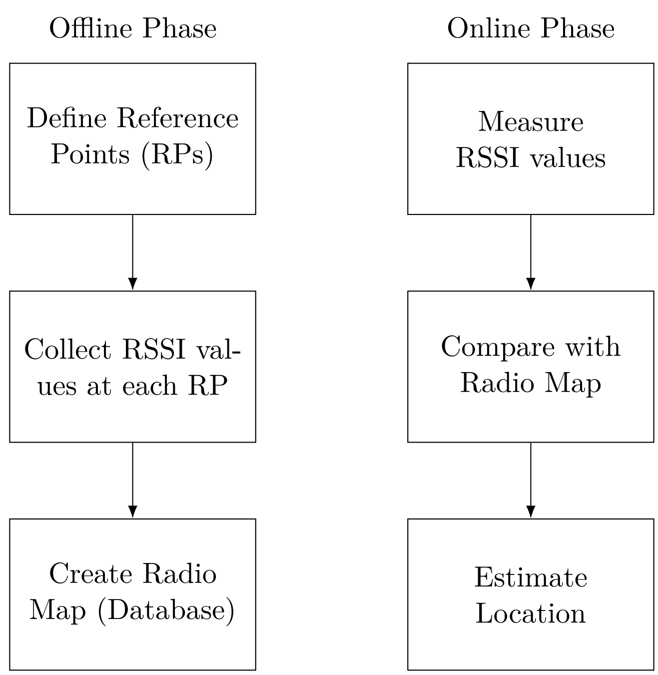

Figure 4.1: Schematic representation of the fingerprinting process, illustrating the offline and online phases.

4.2 Data Collection and Storage

In the pursuit of creating a reliable system for locating workers in real time, the cornerstone is the methodological collection and storage of data. This section delves into the intricate process of how data, specifically RSSI values from BLE beacons, are gathered, the variables affecting these readings, and the subsequent storage mechanism employed.

4.2.1 Variables Impacting RSSI Readings

Radio Signal Strength Indicator (RSSI) readings are pivotal in understanding the proximity of devices, yet these readings are not absolute. Various environmental factors contribute to the fluctuations in RSSI values, particularly when the signal strength is weak. To address this, a cutoff value is implemented to ensure the Android device captures data only from nearby beacons, reducing the noise and fluctuations in the readings.

- **Distance:** The RSSI value diminishes as the distance between the beacon and receiver increases, illustrating an inverse relationship between signal strength and distance.
- **Obstructions and Multipath Propagation:** Physical obstructions such as walls, human bodies, or other electronic devices, as well as signals reflecting off surfaces, can cause interference, leading to fluctuations in RSSI values. To mitigate these effects, a Sliding Window Technique (a method where a fixed-size subset of the most recent data points is continuously analyzed for noise reduction), as discussed in Chapter 5, is employed to stabilize the RSSI readings amidst environmental interference.
- **Beacon Transmission Power:** The power level at which beacons transmit their signals directly influences RSSI readings. A higher power results in stronger signals and vice versa.
- **Hardware Inconsistencies:** Discrepancies in device manufacturing can produce different RSSI readings for beacons at the same distance. However, in the context of fingerprinting used for this system, such inconsistencies are less impactful compared to methods like triangulation. The fingerprinting approach focuses on recognizing patterns of RSSI values corresponding to specific locations, making it inherently resilient to such hardware inconsistencies.

By acknowledging and addressing these variables, the system ensures a more reliable and context-aware interpretation of RSSI values. The methodologies applied, including the use of a cutoff RSSI value, Sliding Window Technique, and the fingerprinting approach, collectively enhance the accuracy of determining the proximity between the beacon and receiving device.

4.2.2 RSSI Data Collection from BLE Beacons

RSSI values are collected from BLE beacons using an Android application. The application scans for nearby beacons and retrieves their advertisement packets, which contain various pieces of information, including the RSSI. The code snippet below illustrates the callback method triggered when beacons are discovered:

```
// Callback function triggered upon discovering nearby beacons
```

```
private KBeaconsMgr.KBeaconMgrDelegate beaconMgrExample = new KBeaconsMgr.KBeaconMgrDel
    public void onBeaconDiscovered(KBeacon[] beacons) {
        for (KBeacon beacon: beacons) {
            // Process beacon data, including RSSI values
            // Filtering and logging RSSI values above a threshold
            if (beacon.getRssi() > -70) {
                // Process beacon data
            }
        }
    }
};
```

This methodology ensures a systematic approach to data collection, where only beacons meeting certain criteria (e.g., a minimum RSSI value) are processed to ensure data reliability.

4.2.3 Data Storage Mechanism

Upon retrieval, the RSSI data undergoes a structured storage process, ensuring its availability for subsequent real-time location calculations and historical analysis. The underlying mechanism encompasses the following stages:

- **Data Structuring:** Each captured reading is part of a 'SensorData' object, encompassing details such as timestamp, device information, and the actual sensor data, as illustrated in the 'sensorData.go' script. This structuring is critical for maintaining data integrity and easing future retrievals or queries.

- **Database Schema:** The system employs a robust database, structured as per the 'db.go' script, facilitating efficient data storage and retrieval. Tables such as 'sensors', 'devices', and 'locations' are crucial, each serving a unique purpose in the data management process.

- **Data Validation and Insertion:** Prior to storage, each 'SensorData' instance undergoes a validation process, ensuring that no essential fields are missing and that each entry is unique. Post-validation, data is inserted into the 'sensors' table, with any new sensor type leading to the dynamic creation of a new column, thereby maintaining a flexible database schema.

- **Efficiency in Storage:** Utilization of a 'stringsizer' mechanism optimizes the storage process. This functionality condenses the sensor data into a string form, minimizing the storage footprint and streamlining the insertion process.

In summary, this system's efficacy in tracking a worker's real-time location hinges on the meticulous collection and storage of RSSI values, acknowledging the potential for environmental interference. The Android-based collection, coupled with a structured and dynamic storage mechanism, lays a solid foundation for the subsequent processes of worker localization and movement analysis.

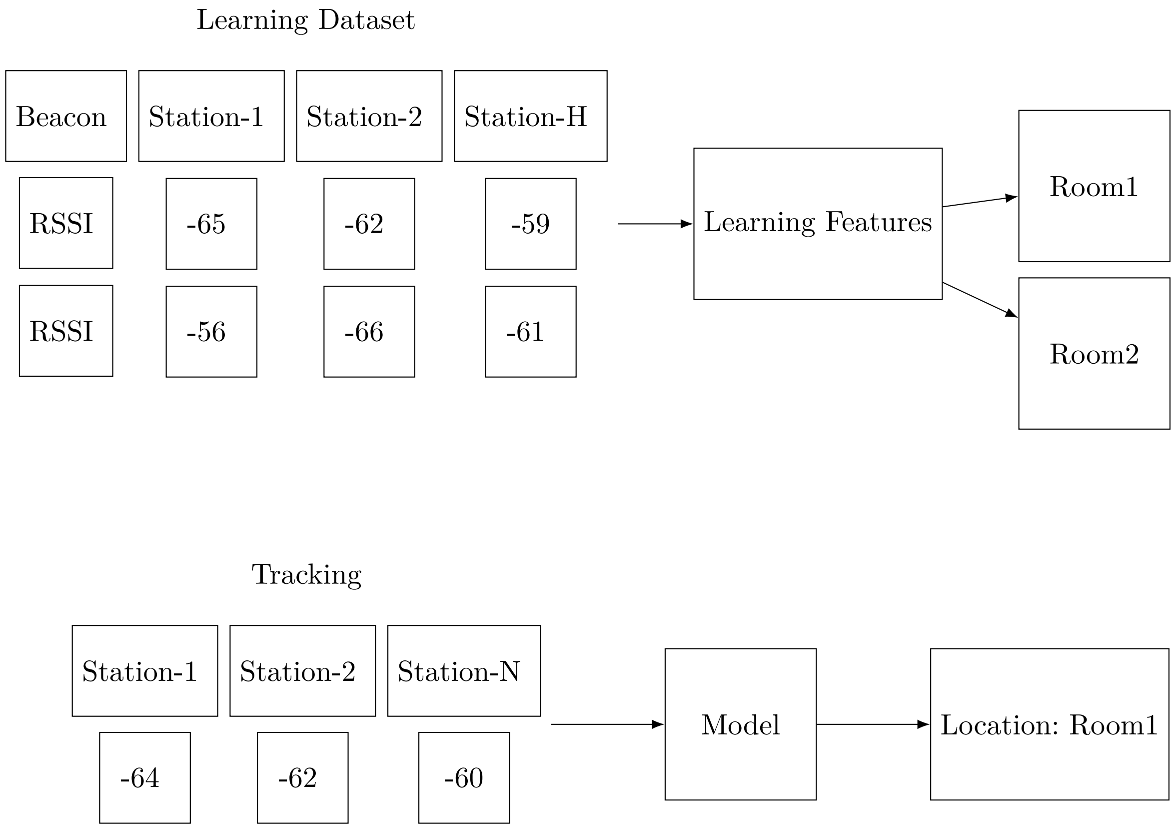

Figure 4.2: Data representation for learning and tracking

4.3 Real-time Location Estimation

Real-time location estimation is crucial in construction site safety alerting systems. Ensuring that the workers are promptly located with a high degree of accuracy can provide immediate insights, thus improving the overall safety of the site. With BLE devices acting as stations and an Android application installed on workers' phones, the following describes the process and nuances of real-time location estimation based on fingerprinting.

4.3.1 Process of Real-time Localization using Stored Data

The stored data comprises RSSI values from different BLE stations corresponding to various locations inside the construction site. The real-time location estimation process is as follows:

1. A worker's device sends current RSSI values obtained from BLE stations in its proximity.

2. The system in the backend, specifically the AnalyzeSensorData function, processes this data.

3. The system then compares this real-time data with the stored fingerprinting database using machine learning techniques.

4. Based on the comparison, the system predicts the most likely location of the worker's device. These predictions are structured as location-probability pairs, indicating the likelihood of the device being in those locations.

5. If a probable location is determined, the system updates the database with this new prediction using the AddPrediction function. If the location cannot be confidently determined, the system labels it as unknown.

4.3.2 Correlation between Historical and Current RSSI Data

For the purpose of location estimation, the correlation between historical RSSI data and the current data is of paramount importance. The more significant the correlation, the higher the confidence in the location prediction. Here's a deeper look:

- Historical RSSI data serves as a reference map. It represents the unique RSSI values at different locations.

- The current RSSI data from a worker's device represents a sample, which needs to be matched against the reference map.

- The closer the match between the current data and a stored pattern, the higher the probability that the device is in that specific location.

- Techniques like the Naive Bayes, as seen in the nb1.Classify function, plays a crucial role in making this correlation. They help in statistically determining the likelihood of the device being at a particular point. Other Machine Learning algorithms that we use are explained in Chapter 5.

To make this process more efficient and reliable, we limit the vector size for RSSI values to 10. This means that the system focuses on the 10 most recent (and often closest) RSSI values observed by the Android device. Limiting the vector size not only speeds up the computational process but also ensures that the data used for localization is the most relevant, thus potentially increasing the accuracy of location predictions.

4.3.3 Nuances of Making the Process Real-time

Making the location estimation process real-time introduces certain challenges and considerations:

- **Speed considerations:** The process needs to be swift. Any delay can render the location data irrelevant, especially if workers are constantly moving. Efficient algorithms, database operations, and optimized data structures are essential to ensure quick processing. Leveraging optimized algorithms and limiting the RSSI vector size to 10 ensures that the system can process data quickly without significant latency.

- **Data volume:** In a bustling construction site, multiple devices might be sending data concurrently. Handling this large influx of data without causing system bottlenecks is imperative.

- **Data accuracy:** RSSI values can fluctuate due to various reasons, including interference from other devices, physical obstructions, or device orientation. Ensuring that such fluctuations do not severely affect the location prediction accuracy is crucial.

- **Server Load:** Continual requests and data processing can put a strain on the server. Efficiently written code, as seen in the provided files, and possibly load balancing, can help in managing this load, ensuring smooth and uninterrupted operations.

- **Data Reliability:** Not all RSSI values may be reliable. Factors like interference, device malfunction, or even environmental factors can affect RSSI readings. The system must be robust enough to handle such anomalies, possibly by discarding outliers or considering them with less weight during the analysis.

In conclusion, real-time location estimation in a dynamic environment like a construction site is a complex process. It requires an amalgamation of efficient algorithms, robust database operations, and reliable data. The system described here, which incorporates fingerprinting techniques with BLE devices, offers a promising solution to address the inherent challenges and ensure worker safety on construction sites.

4.4 Building a Reliable Fingerprinting Database

Building a reliable fingerprinting database is crucial for accurate real-time location estimation. This section details the initial setup, periodic recalibration, and highlights the significance of database reliability for real-time location estimation.

4.4.1 Initial Setup

The initial setup is critical in laying the foundation for the fingerprinting database and involves defining reference points and establishing baselines.

- **Defining Reference Points:** Reference points were strategically selected to be 1.2 meters apart. This distance strikes a balance between location accuracy and exactness, proving satisfactory for the project's needs.

- **Initial Data Collection:** The data collection process is a meticulous activity that requires gathering RSSI values from BLE beacons using an Android application installed on workers' phones. At each defined reference point, data should be collected for a duration of approximately 1 to 2 minutes. This duration allows for the collection of a substantial amount of data, typically resulting in 120 to 150 RSSI vectors per reference point, which enriches the fingerprinting database and enhances the accuracy of location estimations.

- **Establishing Baselines:** The collected data at each reference point establishes a baseline of RSSI values, which are pivotal for the fingerprinting process. These baselines serve as the reference data for real-time location estimations.

Through careful selection of reference points and meticulous data collection, the system lays a solid foundation for the subsequent processes of worker localization and movement analysis in real-time.

4.4.2 Periodic Recalibration

Periodic recalibration or updates to the database are imperative to account for environmental changes that could affect the RSSI values. The system provides an option for recalibration that the admin can use to re-initiate the learning process.

- **Manual Recalibration:** The admin can trigger a manual recalibration to update the fingerprinting database.

- **Automatic Recalibration:** The system will recalibrate whenever new data comes in, whether it's in the form of a new reference point or new RSSI values for a previously added reference point.

4.4.3 Database Reliability

The reliability of the fingerprinting database is paramount for the success of the real-time location estimation phase. The database stores the collected data in a structured manner, facilitating efficient retrieval and analysis for real-time location estimation.

The *db.go* file contains various functions crucial for managing the fingerprinting database. The `MakeTables` function, for instance, creates two essential tables: a `keystore` table and a `sensors` table. These tables are pivotal for storing and managing the fingerprinting data.

Here's a breakdown of some key functionalities provided by the *db.go* file:

- **Data Insertion and Retrieval:** Functions like `AddSensor`, `GetSensorFromTime`, and `GetAllForClassification` are crucial for inserting new sensor data and retrieving existing data for various purposes including classification.

- **Prediction Management:** Functions `AddPrediction` and `GetPrediction` play vital roles in managing predictions related to the location of devices.
- **Data Validation:** The `AddSensor` function can also be seen as a way to validate data before insertion, ensuring the integrity of the fingerprinting database.
- **Database Debugging and Management:** Functions like `Dump`, `Delete`, and `Debug` are essential for database debugging and management.

In essence, the *db.go* file encapsulates a comprehensive set of functionalities required for managing a reliable fingerprinting database, which in turn, is crucial for accurate real-time location estimation.

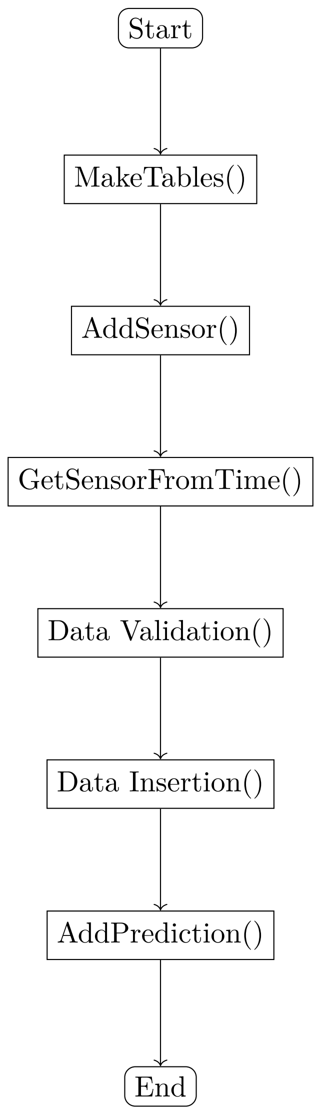

Figure 4.3: Flowchart depicting database operations based on *db.go* functionalities.

Chapter 5

Machine Learning Algorithms for Hazard Detection and Implementation

5.1 Overview of ML Structure for the Server

The modern era of construction safety is increasingly reliant on advanced technological solutions. For this project, we have adopted the FIND (Framework for Internal Navigation and Discovery) framework, adapting it with necessary customizations to meet our specific requirements. The server component, integral to our system, is expertly designed to handle the continuous flow of data from various Bluetooth Low-Energy (BLE) devices with a focus on concurrency and efficient data processing. This is crucial for the real-time demands of our application. In addition to these adaptations, we have also incorporated advanced machine-learning techniques to ensure accurate location predictions. The integration of the FIND framework with our tailored modifications and diverse technological tools exemplifies our commitment to creating a cohesive and effective system tailored to the unique challenges of construction site safety.

Upon receiving the sensor data, the server forwards a request to the Python classifier. This classifier, trained on pertinent datasets, furnishes the predicted location, which is subsequently relayed by the Go server to the client.

The machine learning architecture on the server is structured to facilitate sophisticated analysis and processing of the sensor data. Primarily, this involves the utilization of an external AI for machine learning-based fingerprinting, and a custom "Extended Naive Bayes1" algorithm. Each serves as a crucial pillar in deducing the final location.

Key Components of the Framework:

- **Concurrency for Performance**: Exploiting Go's intrinsic concurrency model, notably goroutines and channels, both the AI model and the Naive Bayes method are executed concurrently. This parallel execution mechanism ensures swift analysis, a requisite for real-time systems.

- **AI Endpoint**: The AI model, encapsulated within an endpoint, is invoked through a POST request. Herein, the server sends over the sensor data and requisite model information, post which the endpoint returns its predictions, thereby facilitating seamless communication between Go and Python.
- **Extended Naive Bayes1 Method**: Besides the AI Endpoint, the server also employs its Naive Bayes derivative for predictions. By leveraging the underlying principles of the Naive Bayes algorithm, but with tailored optimizations, it provides an alternative prediction pathway, ensuring redundancy and reliability.

5.2 Naive Bayes Algorithm

Naive Bayes is rooted in Bayes' theorem, given by:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Where:

- $P(A|B)$ is the posterior probability.
- $P(B|A)$ is the likelihood.
- $P(A)$ is the prior probability.
- $P(B)$ is the marginal likelihood.

The effectiveness of the real-time safety alerting system is critically dependent on its capability to consistently track workers within the construction environment. A key aspect of ensuring this reliability is redundancy. Therefore, alongside the primary Python-based machine learning classifiers, the system's Go server includes a Naive Bayes algorithm as a backup. This setup ensures that accurate predictions are maintained even if unexpected challenges arise with the primary classification method.

5.2.1 Extending Traditional Naive Bayes

Naive Bayes stands out for its computational efficiency and simplicity. It's based on the application of Bayes' theorem with the "naive" assumption of independence between every pair of features. Despite this assumption, in many real-world scenarios, especially in text classification and sentiment analysis, the algorithm has demonstrated remarkable accuracy.

The system incorporates an "Extended Naive Bayes" approach, tailored to meet the specific nuances and demands of the construction site environment. This extension involves several optimizations to enhance the predictive capabilities of the algorithm within this domain. Key aspects of this extension include:

- **Dynamic Data Handling:** The algorithm is designed to manage dynamic data efficiently. It processes incoming sensor data from various BLE beacons and categorizes them based on location and sensor type.

- **Enhanced Probability Estimation:** The core of this extension involves sophisticated probability estimation. The algorithm uses a Gaussian filter to apply a smoothing effect on the RSSI values. The Gaussian filter is mathematically represented as:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

  where $G(x)$ is the Gaussian function, μ is the mean (usually set to 0 for a standard Gaussian), σ is the standard deviation, and x represents the RSSI value. This smoothing reduces the impact of signal anomalies.

- **Probability Normalization:** After calculating probabilities for each location, these are normalized to ensure consistency across predictions. The normalization formula is given by:

$$P_{normalized}(location) = \frac{P(location)}{\sum_{i=1}^{n} P(i)}$$

  where $P_{normalized}(location)$ is the normalized probability for a given location, $P(location)$ is the raw probability of the location, and the denominator represents the sum of raw probabilities for all locations. This normalization ensures that the probabilities across all locations sum up to 1, maintaining a consistent probability distribution.

- **Efficient Data Storage and Retrieval:** A key feature of the Extended Naive Bayes algorithm is its efficient data management. It stores the learned data in a structured format that allows for quick retrieval and analysis, ensuring that the system's predictions are always based on the most current and relevant data.

This bespoke extension of the traditional Naive Bayes algorithm provides a robust solution for the complex and ever-changing environment of a construction site, where accuracy, speed, and reliability are of utmost importance.

5.2.2 Implementation Details

- **Concurrent Execution**: To ensure real-time predictions without lag, the Naive Bayes classification (nb1.Classify) operates concurrently as a separate goroutine. This concurrent execution model ensures non-blocking analysis, catering to the real-time demands of the system.

- **Structured Prediction Formation**: Once the classification is completed, results are structured into a standardized format (algPrediction). This format entails predicted locations and their associated probabilities, allowing for easy integration with other system components and ensuring clarity in decision-making processes.

- **Adaptive Learning with Gaussian Filtering**: One of the standout features of the extended implementation is the application of Gaussian filtering on the data. This acts as a smoothing mechanism, which can be particularly crucial in handling the noisy and fluctuating RSSI values typical in dynamic environments like construction sites. This not only refines the prediction but also aids in reducing the influence of outliers.

In essence, while the system's Python-based classifiers provide powerful predictive capabilities, the Go server's Extended Naive Bayes algorithm acts as a crucial safety net, ensuring consistent and reliable localization predictions, even in the face of unforeseen adversities.

5.3 k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Random Forest, and Other Algorithms

This section explores the nuances of various machine-learning algorithms crucial for classifying sensor data for indoor localization, focusing on their implementation specifics and parameter tuning.

5.3.1 Overview of Python-Based Classification

The primary class, *AI*, is central to both training and classification, using a range of machine learning algorithms, including:

- k-Nearest Neighbors (k-NN)
- Linear Support Vector Machines (SVM)
- Radial Basis Function SVM (RBF SVM)
- Decision Tree
- Random Forest
- Neural Network
- AdaBoost
- Gaussian Naive Bayes
- Quadratic Discriminant Analysis (QDA)

- Gradient Boosting

Each algorithm delivers its prediction, collectively enhancing location analysis accuracy, accessible via the *classify* method in the *AI* class.

Machine Learning Algorithm Implementations and Hyperparameter Tuning

Detailed below are the specific implementations and hyperparameter tuning processes for key algorithms:

k-Nearest Neighbors (k-NN)

The k-NN algorithm determines the neighbors of a given point using various distance metrics. A frequently employed metric is the Euclidean distance, which is defined as:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

Utilizing the Euclidean distance metric, the k-NN model's key parameters include the number of neighbors (tested with values 3, 5, 7, 9), weights (uniform or distance), and the distance metric (Euclidean or Manhattan).

Support Vector Machines (SVM)

SVM operates by identifying an optimal hyperplane that best segregates the dataset into classes. The decision function for this hyperplane can be represented by:

$$y(x) = w^T \phi(x) + b$$

The SVM models, both linear and RBF, were tuned with parameters such as cost (C) ranging from 0.001 to 10, kernel types (linear, RBF), and gamma values for the RBF kernel (0.1 to 100).

Random Forest

Random Forests, as an ensemble of decision trees, often resort to criteria like the Gini impurity to determine the most effective data split. The Gini impurity is given by:

$$Gini(p) = 1 - \sum_{i=1}^{J} p_i^2$$

Where p_i represents the probability of selecting an item from class i, and J indicates the total number of classes.

The Random Forest algorithm's parameters included the number of trees (10 to 200), maximum depth (None to 20), and minimum samples split (2, 5, 10).

Neural Network

The Neural Network implementation involved a multi-layer perceptron with varying configurations of hidden layers (e.g., (50,50,50), (50,100,50), (100,)), activation functions (tanh, relu), solvers (sgd, adam), alpha values (0.0001, 0.001, 0.05), and learning rate policies (constant, adaptive).

Hyperparameter Tuning Hyperparameter tuning was conducted using RandomizedSearchCV coupled with StratifiedKFold for cross-validation. This approach, being more computationally efficient than grid search, randomly samples a given number of parameter settings from the specified hyperparameter space. For each machine learning model, the best parameter set was determined based on the cross-validation results, ensuring an optimal balance between model complexity and generalization ability.

Feature Importance and Model Evaluation For tree-based models (Decision Tree, Random Forest, Gradient Boosting), feature importances were logged, providing insights into the most influential factors in location prediction. Confusion matrices were generated to evaluate the classification performance, and cross-validation results were analyzed to assess model robustness and effectiveness.

Additional Notes

- The Gaussian Naive Bayes algorithm was used without hyperparameter tuning due to its lack of hyperparameters.
- The Gradient Boosting algorithm involved additional parameters like subsample rates and maximum tree depths to control the model's complexity and prevent overfitting.

5.3.2 Enhancements for Accuracy

To ensure the reliability and robustness of our indoor localization, the following improvements and techniques were integrated into our methodology:

1. **Handling Missing RSSI Values**: Often, RSSI (Received Signal Strength Indicator) values might be missing or inconsistent in the dataset. Using a sliding window approach, it dynamically fills in any missing RSSI values. This not only enhances the consistency of our dataset but is also pivotal in enhancing prediction accuracy.
2. **Hyperparameter Optimization**: A critical aspect of any machine learning model's performance is the selection of appropriate hyperparameters. Our expanded hyperparameters dictionary, catering to multiple classifiers including Gradient Boosting, underscores our dedication to fine-tuning every algorithm.

3. **Efficient Hyperparameter Search with RandomizedSearchCV**: The traditional grid search approach for hyperparameter tuning can be arduous. We capitalized on *RandomizedSearchCV*, coupled with *StratifiedKFold* during cross-validation. This ensured efficient hyperparameter sampling and maintained an even distribution of classes during evaluations.

4. **Extensive Logging for Debugging and Transparency**: Throughout the process, comprehensive logging was integrated. This facilitated the monitoring of crucial system aspects, from data preprocessing to model evaluation, ensuring reproducibility, ease of debugging, and transparent model decision insights.

5. **Performance Metrics for Model Evaluation**: For a nuanced understanding of our model's effectiveness, we incorporated performance metrics such as confusion matrices. This offered insights into the true positive rates, potential misclassifications, and areas for model refinement.

The Pervasive Challenge of Bluetooth Noises and Missing Values

Bluetooth, with its benefits for short-range communications, presents challenges in dynamic environments like construction sites. Construction materials, machinery, and movement lead to issues like signal attenuations, reflections, multi-path fading, and occasional signal dropouts. This results in the RSSI values, essential for indoor localization, being noisy or occasionally missing.

Sliding Window Technique for Filling Missing Data

In addressing the challenges of fluctuating and missing RSSI values in a construction site environment, the project employed a modified sliding window technique. This technique is particularly tailored for handling missing data in RSSI readings. The approach is as follows:

- **Identifying Missing Values**: Initially, RSSI readings with missing values are identified and temporarily replaced with NaN (Not a Number) for processing.

- **Window-Based Imputation**: A forward and backward rolling window is applied to each RSSI series. This window computes the mean of the surrounding values, effectively filling in the missing data points. The size of the window is defined as a parameter, providing flexibility in handling different data scenarios.

- **Repeated Application**: The rolling mean process is repeated to ensure maximal coverage of missing values. If any NaN values persist after the first pass, the process is applied again to further refine the data.

- **Final Replacement**: Any remaining NaN values after the rolling mean applications are replaced back to 0.0, ensuring a complete dataset for subsequent analysis.

This technique, while differing from a traditional moving average, offers a robust solution for mitigating the impact of signal loss or interference, common in dynamic construction site environments. It enhances the reliability and integrity of the RSSI data, which is critical for the accuracy of the indoor localization process.

Project Benefits from the Modified Sliding Window Technique

Employing this modified sliding window approach yielded several key advantages for the project:

- **Enhanced Data Integrity**: The approach ensures a continuous and complete stream of RSSI data, crucial for the accuracy of real-time localization and any retrospective analysis or audits.
- **Increased Robustness**: The system's resilience to the unique challenges of construction sites, such as environmental interference and signal fluctuation, is significantly bolstered.
- **Improved Localization Accuracy**: By ensuring a more consistent and complete dataset, the system's indoor positioning accuracy is inherently enhanced.

Overall, this tailored sliding window technique has proven to be an invaluable asset in maintaining the precision and reliability of the Bluetooth-based indoor localization system, despite the inherent challenges of the operational environment.

5.4 Algorithm Comparison and Performance Evaluation

A hallmark of an effective real-time alerting system is its ability to consistently provide accurate predictions, regardless of the environmental variables or inherent challenges of the domain. To this end, this system amalgamates predictions from both the AI model and the Extended Naive Bayes1 method, and through a meticulously engineered consensus prediction mechanism, optimizes the final location predictions.

5.4.1 Consensus Prediction

The heart of this system's prediction lies in the `determineBestGuess` function. It amalgamates the predictions from different algorithms, weighing each based on a unique metric of "efficacy." The consensus prediction is mathematically represented using a weighted average:

$$\hat{y} = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \tag{5.1}$$

Here, $\hat{y}$ represents the consensus prediction, w_i are the weights derived from each algorithm's historical performance metrics, and y_i are the individual predictions from each algorithm.

The weights w_i are calculated as the product of the prediction probability provided by each algorithm (y_i) and an 'informedness' score. This 'informedness' score is a statistical measure quantifying the algorithm's accuracy and reliability in various contexts. It reflects the historical accuracy of the algorithm in making location predictions, thereby encapsulating its performance in similar scenarios. Therefore, the weights represent a combination of each algorithm's current prediction confidence and its historical reliability.

This composite measure of raw prediction probability and historical 'informedness' ensures that the system prioritizes predictions that are both confident and historically well-informed. Once these scores are calculated, they are normalized to produce a consensus prediction that is both accurate and robust, leveraging the strengths and compensating for the weaknesses of individual algorithms. This methodology enhances the overall accuracy and trustworthiness of the consensus prediction, providing a balanced and data-driven approach to determining the weights.

5.4.2 Performance Metrics and Evaluations

A myriad of machine learning algorithms are in play, each with its unique strengths, assumptions, and underlying mechanics. To gauge their individual and collective performance, various evaluation techniques and metrics are used:

- **Feature Importance**: Particularly pertinent for tree-based algorithms such as Decision Trees, Random Forests, and Gradient Boosting, feature importance metrics are logged. This aids in understanding the significance of each sensor, revealing which ones are crucial in predicting locations accurately.

- **Confusion Matrix**: An indispensable tool in classification tasks, the confusion matrix offers a granular breakdown of predictions. It uncovers the true positives, false positives, true negatives, and false negatives for each algorithm, painting a vivid picture of where each model shines or falls short.

- **Cross-Validation**: Employing Stratified K-fold cross-validation ensures each fold of the data retains the original class distribution, providing an unbiased evaluation. This method critically appraises each algorithm's performance, ensuring the reported metrics aren't artifacts of any particular data split.

5.4.3 Under the Hood: Code Mechanics

This system integrates the computational strengths of Python and Go to optimize location prediction accuracy. On the Python side, the focus is on machine learning and data analysis. A randomized grid search combined with cross-validation is employed to fine-tune the hyperparameters of each classifier. This approach systematically explores a range of hyperparameter values, ensuring that each algorithm operates at its peak efficiency. The randomness in the grid search introduces variability, preventing the model from settling into local optima and promoting a more comprehensive exploration of the hyperparameter space.

Cross-validation, a key component of this process, involves dividing the dataset into multiple subsets. The algorithm is trained on some of these subsets and validated on the others. This method helps in assessing the algorithm's performance and generalizability, ensuring its robustness across different data samples.

On the Go server side, the core of consensus prediction is realized through the `determineBestGuess` function. This function employs a sophisticated method to calculate an efficacy score for each location prediction made by an algorithm. The efficacy of each algorithm is quantified through a meticulous statistical process, which involves creating a confusion matrix for each algorithm and location pair. This matrix tabulates instances of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). From this matrix, several efficacy metrics are derived:

- Precision: Precision $= \frac{TP}{TP+FP}$
- Recall: Recall $= \frac{TP}{TP+FN}$
- F1 Score: F1 $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
- Informedness: Informedness $=$ Recall $+$ Specificity $- 1$, where Specificity $= \frac{TN}{TN+FP}$

These metrics offer a comprehensive view of each algorithm's performance. Precision measures the accuracy of positive predictions, Recall assesses the algorithm's ability to detect positive instances, and the F1 Score provides a balance between Precision and Recall. Informedness offers an overall measure of the algorithm's performance, considering both its sensitivity and specificity.

The efficacy scores for each location are then aggregated across all algorithms in the `determineBestGuess` function. This aggregation is pivotal as it reflects not just the confidence of each algorithm in its current prediction but also its historical accuracy and reliability. The final consensus prediction is formed by normalizing these aggregated scores, ensuring that the location prediction is not only the most probable but also the most reliable, based on historical data.

By combining the robust processing capabilities of Go with the advanced data analysis and machine learning prowess of Python, this system demonstrates an exemplary blend of software engineering and data science. This synergy enables the system to accurately localize workers within construction sites, significantly enhancing safety standards. The dual-language approach harnesses the strengths of both paradigms — the efficient concurrency model of Go and the sophisticated data processing of Python — ensuring a high-performance and accurate prediction system.

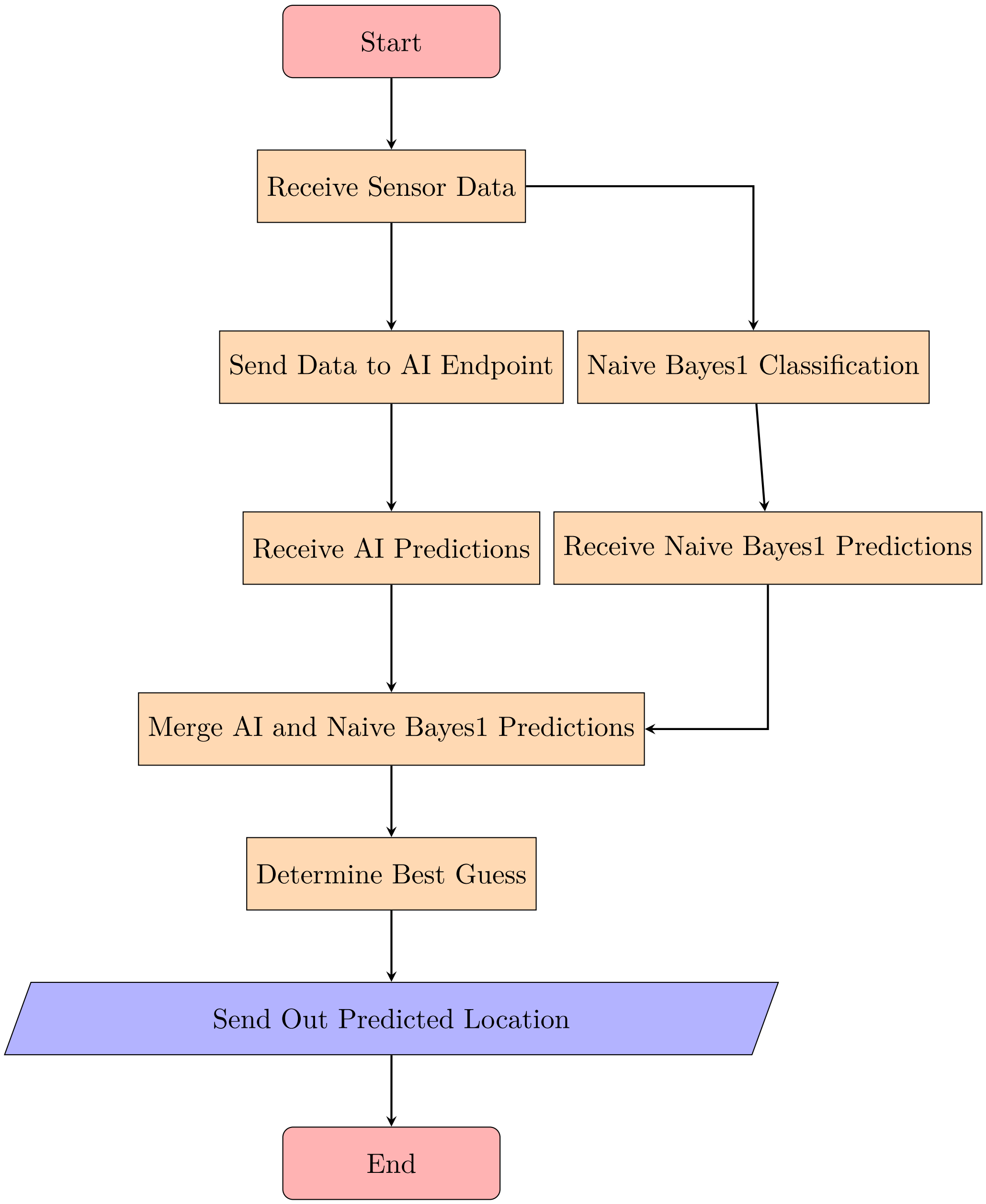

Figure 5.1: Flowchart illustrating the process of sensor data processing and prediction.

Chapter 6

System Development and Performance Assessment

6.1 System Testing and Evaluation

6.1.1 Testing Methodology

The Real-time Safety Alerting System was subjected to a comprehensive testing regime to ascertain its functionality, reliability, and performance. Customized components including the Android application, frontend website, and server were meticulously examined. Testing was conducted within an office space simulating a construction site environment, furnished with various items such as benches, drawers, and other objects mimicking potential obstructions and interference on a real construction site.

The testing environment was a room measuring 5×2 meters, wherein 10 beacons were installed as reference points, spaced 1.2 meters apart from each other. This setup was intended to closely resemble the spatial arrangements and potential challenges encountered on a construction site.

The testing phase was bifurcated into a learning phase and an evaluation phase. During the learning phase, the system was allowed to gather data for approximately 2 minutes at each reference point, generating less than 150 RSSI vectors per reference point. Subsequent to the data collection, two primary evaluation tests were conducted:

- **Static Test:** The static test involved standing at each reference point to evaluate the accuracy of location predictions. The aim was to ascertain the correctness of the system's predictions while in a stationary position.

- **Dynamic Test:** The dynamic test entailed walking between reference points to gauge the system's accuracy during movement. This test also provided insights into the lag experienced in updating the user's location as they transitioned between reference points.

Two photographs showcasing the test environment are attached to provide a visual perspective of the simulation setting (See 6.1 and 6.2).

Figure 6.1: Photograph showcasing the test environment and beacons on the walls

6.1.2 Performance Metrics

Performance evaluation of the system was primarily based on accuracy metrics, crucial for assessing the system's efficacy in real-time location tracking within the simulated construction environment. The system employs a fingerprinting approach, where accuracy is determined by the system's ability to match the estimated location with the correct reference point. In this context, a prediction is considered accurate if it corresponds to the actual reference point where the user is located.

Figure 6.2: Photograph showcasing the test environment and beacons on the walls

The output from the machine learning algorithm, such as

["location":"Room1","probability":0.89,"location":"Room2","probability":0.1,"location":"Room3"]

provided a probability score associated with each location prediction.

A meticulous manual verification was conducted post-testing to ascertain the accuracy of the location predictions. Utilizing timestamps, the exact location of the user at specific times was compared against the system's predictions stored in the SQL database. This manual cross-verification, combined with the fingerprinting approach, enabled a thorough assessment of the system's accuracy and performance in various simulated construction site scenarios.

6.2 Results and Analysis

6.2.1 Dynamic Test Analysis

The dynamic test, focused on the system's ability to accurately track a moving worker, provides valuable insights into its practicality and responsiveness. This analysis evaluates the results based on the system's overall accuracy, delay in location updates, and confidence over time.

Overall Accuracy In the dynamic test, the system achieved an overall accuracy of approximately 97.3%, a testament to the fingerprinting-based system's reliability in tracking workers within the simulated construction site. This accuracy rate underscores the system's proficiency in correctly identifying worker locations in relation to specific reference points, a crucial factor in ensuring safety. The comprehensive evaluation involved a total of 670 location estimations, demonstrating the system's robust performance and effectiveness at scale. These results affirm the system's capability to operate efficiently in real-world scenarios, showcasing its potential as a reliable tool for enhancing safety standards in dynamic construction environments.

Response Time: An essential aspect of any real-time tracking system is its response time, which measures the speed at which the system updates its location predictions during transitions. The average delay observed during the dynamic tests was approximately 1.26 seconds, representing the time taken for the system to correctly update the location after a transition has occurred.

System's Confidence Analysis As demonstrated in Figure 6.3, the system's confidence over time fluctuates based on the user's movement and transition between reference points. It's evident that there's a momentary drop in system confidence during these transitions. This decrease in confidence during transitions is likely due to the system re-calibrating its predictions based on the changing RSSI vectors. Over time, the confidence generally stabilizes, especially when the user remains stationary, reinforcing the system's adaptability and reliability.

Overall Confidence Considering the entirety of the dynamic test, the system's overall confidence in its predictions averaged out to approximately 79.5%. While this figure may not appear exceptionally high at first glance, it is important to contextualize this level of confidence within the specific challenges and complexities of real-time location tracking in a dynamic site environment. In such environments, factors like varying signal interference, obstacles, and the diverse range of movements by workers present significant challenges to achieving pinpoint accuracy. The confidence level achieved by the system is indicative of its ability to provide reliable location tracking most of the time, which is a considerable

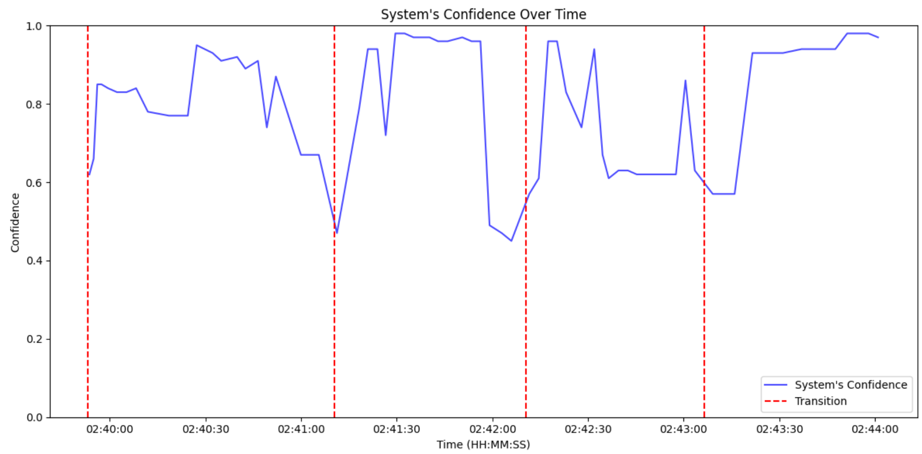

Figure 6.3: System's Confidence Over Time

achievement given the inherent unpredictability and complexity of the operational environment.

In conclusion, while there is room for improvement, the 79.5% confidence level, in the context of a construction site environment, represents a meaningful advancement in safety technology. It underscores the system's utility and reliability in providing real-time location tracking, thus bolstering its potential for enhancing safety protocols in actual construction sites. The dynamic test results affirm the system's capability to function effectively within a simulated construction site environment, demonstrating high accuracy and confidence rates with minimal delay. This makes the system a promising solution for safety alerting on actual construction sites.

6.2.2 Hardware Dependency of Response Time

An important aspect to consider in the evaluation of our system's response time is the dependency on the hardware used during testing. For this study, the Samsung A13 smartphone was employed for the data collection and real-time monitoring tasks. It is pertinent to note that the Samsung A13 is not a high-end device, and its processing capabilities are modest compared to more advanced models available in the market.

- **Samsung A13 Specifications:** The Samsung A13 is powered by an Exynos 850 chipset, featuring an octa-core processor with a clock speed of up to 2.0 GHz. The device offers adequate performance for routine tasks but may not exhibit the high processing power of flagship models. This could potentially impact the response time of real-time applications such as our safety alerting system.

- **Impact on Response Time:** During our dynamic test, the system demonstrated an average response time of approximately 1263.25 milliseconds. It is crucial to acknowledge that this response time is intrinsically linked to the processing capabilities of the Samsung A13. Devices with higher processing power are likely to yield faster response times, thereby enhancing the real-time tracking efficiency of the system.

This observation underlines the importance of considering the hardware specifications when deploying the system in a real-world scenario. The response time and overall system performance can vary significantly based on the smartphone model used by the workers. Future studies and implementations should take into account this hardware dependency, potentially testing the system across a range of devices to gauge performance variations.

6.2.3 Spatial Accuracy Analysis

The evaluation of the system's spatial accuracy revealed important insights into its performance. The maximum observed error in the distance was found to be 1.2 meters. This measurement is particularly significant in the context of the system's design, where reference points were deliberately spaced 1.2 meters apart.

It is essential to understand that the system's predictive model was trained using data gathered at these reference points. Therefore, the maximum error aligning with the spacing interval indicates that when the system does err in its predictions, it most commonly confuses a location with its immediate neighboring point.

However, it is important to note that the system's accuracy is not inherently limited to the reference point distance. The accuracy could potentially be finer than 1.2 meters if the environmental conditions, signal stability, and algorithmic efficiency are optimal. In other words, the 1.2 meters spacing provides a framework for the system's learning and prediction, but the actual accuracy in live scenarios can vary slightly depending on various factors such as signal interference, beacon placement accuracy, and algorithmic performance.

In summary, while the system's training on reference data set within 1.2 meters intervals plays a significant role in its accuracy, it does not strictly limit the system to this distance in terms of error. The maximum error of 1.2 meters simply reflects the most common scenario where the system confuses a location with its nearest neighbor within the grid formed by the reference points.

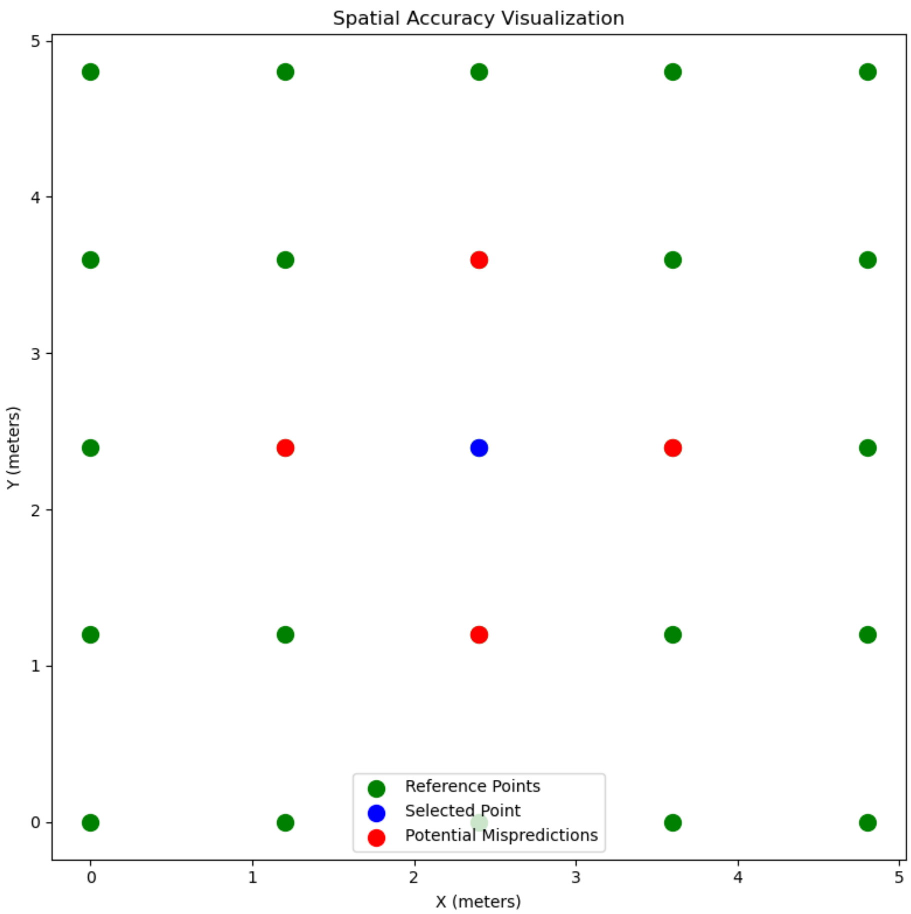

Figure 6.4: Visualization of Spatial Accuracy

Chapter 7

Conclusion

The development and implementation of the Real-Time Safety Alerting System for Construction Sites, as detailed in this thesis, represent a substantial advancement in enhancing safety measures within the inherently hazardous environment of construction sites. This system, a fusion of BLE technology, sophisticated machine learning algorithms, and indoor localization techniques, embodies a groundbreaking approach in safety management.

Central to this endeavor was the innovative adaptation and augmentation of the FIND framework, demonstrating its versatility and adaptability for construction site safety requirements. The system's architecture, comprising an intuitive Android application, advanced server-side components, and an interactive front-end website, formed the backbone of this real-time monitoring and alerting ecosystem.

Significant contributions of this thesis include:

- The customized adaptation of the FIND framework to incorporate BLE technology, enabling precise indoor localization.
- Development of essential system features, such as real-time alerting mechanisms, RSSI filtering, and the creation of safety zones, all tailored for the construction site context.
- Handling large data volumes, particularly in multi-floor buildings, through Android app optimizations and server-side data management techniques.
- Integration of advanced machine learning algorithms like k-NN, SVM, and Random Forest, which enhanced the accuracy and reliability of the system.

The system demonstrated high accuracy (97.3%) and confidence (79.5%) in dynamic test scenarios, affirming its potential as a reliable tool for safety in construction environments. The maximum spatial error observed was 1.2 meters, aligned with the spacing of the reference points, indicating the system's ability to accurately predict locations within the proximity of these points. The response time, averaging around 1.26 seconds, highlighted the influence of hardware capabilities on the system's performance, as seen in tests conducted using the Samsung A13 smartphone.

Looking forward, there are numerous avenues for further enhancement and expansion of the system. These include:

- Enhancing the response time for tracking faster for scenarios where faster movement or emergency situations are involved.
- Improving machine learning algorithms for more refined hazard detection.
- Integrating the system with other industrial systems for broader applicability.
- Expanding the system's use to other areas beyond construction sites.
- Conducting comprehensive field deployments and longitudinal studies to gather empirical data on the system's impact on safety culture.

In conclusion, this thesis presents a novel and effective solution to the longstanding challenges of safety management in construction sites. It not only contributes a significant technological innovation to the construction industry but also underscores the potential of integrating interdisciplinary technologies to address real-world challenges. The Real-Time Safety Alerting System for Construction Sites stands as a testament to the benefits of merging modern technology with practical applications, heralding a new era in construction site safety management.

Bibliography

[1] Abebe Belay Adege, Hsin-Piao Lin, Getaneh Berie Tarekegn, and Shiann-Shiun Jeng. Applying deep neural network (dnn) for robust indoor localization in multi-building environment. *Applied Sciences*, 8(7):1062, 2018.

[2] Maryam Alkaissy, Mehrdad Arashpour, Baabak Ashuri, Yu Bai, and Reza Hosseini. Safety management in construction: 20 years of risk modeling. *Safety science*, 129:104805, 2020.

[3] Maryam Alkaissy, Mehrdad Arashpour, Emadaldin Mohammadi Golafshani, M Reza Hosseini, Sadegh Khanmohammadi, Yu Bai, and Haibo Feng. Enhancing construction safety: Machine learning-based classification of injury types. *Safety science*, 162:106102, 2023.

[4] Taner Arsan and Mohammed Muwafaq Noori Hameez. A clustering-based approach for improving the accuracy of uwb sensor-based indoor positioning system. *Mobile Information Systems*, 2019:1–13, 2019.

[5] Amin Assadzadeh, Mehrdad Arashpour, Ali Rashidi, Alireza Bab-Hadiashar, and Sajad Fayezi. A review of data-driven accident prevention systems: Integrating real-time safety management in the civil infrastructure context. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, volume 36, pages 289–296. IAARC Publications, 2019.

[6] Paramvir Bahl and Venkata N Padmanabhan. 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. Ieee, 2000.

[7] Luis Brás, Nuno Borges Carvalho, Pedro Pinho, Lukasz Kulas, and Krzysztof Nyka. A review of antennas for indoor positioning systems. *International Journal of Antennas and Propagation*, 2012, 2012.

[8] Rafael Saraiva Campos, Lisandro Lovisolo, and Marcello Luiz R de Campos. Wi-fi multi-floor indoor positioning considering architectural aspects and controlled computational complexity. *Expert systems with applications*, 41(14):6211–6223, 2014.

[9] Xiaoyong Chai and Qiang Yang. Reducing the calibration effort for probabilistic indoor location estimation. *IEEE Transactions on Mobile Computing*, 6(6):649–662, 2007.

[10] Wan-Young Chung et al. Enhanced rssi-based real-time user location tracking system for indoor and outdoor environments. In *2007 International Conference on Convergence Information Technology (ICCIT 2007)*, pages 1213–1218. IEEE, 2007.

[11] Chen Feng, Wain Sy Anthea Au, Shahrokh Valaee, and Zhenhui Tan. Received-signal-strength-based indoor positioning using compressive sensing. *IEEE Transactions on mobile computing*, 11(12):1983–1993, 2011.

[12] Xingbin Ge and Zhiyi Qu. Optimization wifi indoor positioning knn algorithm location-based fingerprint. In *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 135–137. IEEE, 2016.

[13] Romeo Giuliano, Gian Carlo Cardarilli, Carlo Cesarini, Luca Di Nunzio, Francesca Fallucchi, Rocco Fazzolari, Franco Mazzenga, Marco Re, and Alessandro Vizzarri. Indoor localization system based on bluetooth low energy for museum applications. *Electronics*, 9(6):1055, 2020.

[14] Ahmed Gondia, Mohamed Ezzeldin, and Wael El-Dakhakhni. Machine learning–based decision support framework for construction injury severity prediction and risk mitigation. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 8(3):04022024, 2022.

[15] Ahmed Gondia, Ahmed Moussa, Mohamed Ezzeldin, and Wael El-Dakhakhni. Machine learning-based construction site dynamic risk models. *Technological Forecasting and Social Change*, 189:122347, 2023.

[16] Xiansheng Guo, Lin Li, Nirwan Ansari, and Bin Liao. Accurate wifi localization by fusing a group of fingerprints via a global fusion profile. *IEEE Transactions on Vehicular Technology*, 67(8):7314–7325, 2018.

[17] Suining He and S-H Gary Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490, 2015.

[18] Gao Huang, Guang-Bin Huang, Shiji Song, and Keyou You. Trends in extreme learning machines: A review. *Neural Networks*, 61:32–48, 2015.

[19] Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2):513–529, 2011.

[20] Umair Khalid, Amrit Sagoo, and Medjdoub Benachir. Safety management system (sms) framework development–mitigating the critical safety factors affecting health and safety performance in construction projects. *Safety science*, 143:105402, 2021.

[21] Kornkanok Khaoampai, Kulit Na Nakorn, and Kultida Rojviboonchai. Low complexity floor localization algorithm for mobile phone. In *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–6. IEEE, 2014.

[22] Hakan Koyuncu and Shuang Hua Yang. A survey of indoor positioning and object locating systems. *IJCSNS International Journal of Computer Science and Network Security*, 10(5):121–128, 2010.

[23] Jayakanth Kunhoth, AbdelGhani Karkar, Somaya Al-Maadeed, and Abdulla Al-Ali. Indoor positioning and wayfinding systems: a survey. *Human-centric Computing and Information Sciences*, 10(1):1–41, 2020.

[24] Keonsoo Lee, Yunyoung Nam, and Se Dong Min. An indoor localization solution using bluetooth rssi and multiple sensors on a smartphone. *Multimedia Tools and Applications*, 77:12635–12654, 2018.

[25] Seoung-Hyeon Lee, Il-Kwan Lim, and Jae-Kwang Lee. Method for improving indoor positioning accuracy using extended kalman filter. *Mobile Information Systems*, 2016, 2016.

[26] Kejiong Li, John Bigham, Eliane L Bodanese, and Laurissa Tokarchuk. Location estimation in large indoor multi-floor buildings using hybrid networks. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2137–2142. IEEE, 2013.

[27] Tsung-Nan Lin and Po-Chiang Lin. Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. In *2005 international conference on wireless networks, communications and mobile computing*, volume 2, pages 1569–1574. IEEE, 2005.

[28] Shing-Jiuan Liu, Ronald Y Chang, and Feng-Tsun Chien. Analysis and visualization of deep neural networks in device-free wi-fi indoor localization. *IEEE Access*, 7:69379–69392, 2019.

[29] Jun Ma, Xuansong Li, Xianping Tao, and Jian Lu. Cluster filtered knn: A wlan-based indoor positioning scheme. In *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–8. IEEE, 2008.

[30] Rui Ma, Qiang Guo, Changzhen Hu, and Jingfeng Xue. An improved wifi indoor positioning algorithm by weighted fusion. *Sensors*, 15(9):21824–21843, 2015.

[31] Sergio Márquez-Sánchez, Israel Campero-Jurado, Jorge Herrera-Santos, Sara Rodríguez, and Juan M Corchado. Intelligent platform based on smart ppe for safety in workplaces. *Sensors*, 21(14):4652, 2021.

[32] Santiago Mazuelas, Alfonso Bahillo, Ruben M Lorenzo, Patricia Fernandez, Francisco A Lago, Eduardo Garcia, Juan Blas, and Evaristo J Abril. Robust indoor positioning provided by real-time rssi values in unmodified wlan networks. *IEEE Journal of selected topics in signal processing*, 3(5):821–831, 2009.

[33] Aigerim Mussina and Sanzhar Aubakirov. Rssi based bluetooth low energy indoor positioning. In *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–4. IEEE, 2018.

[34] Joonghong Park, Jaehoon Kim, Sungwon Kang, et al. Ble-based accurate indoor location tracking for home and office. *Computer Science & Information Technology (CS & IT)*, pages 173–181, 2015.

[35] Linlin Peng, Junyu Liu, Min Sheng, Yan Zhang, Danni Hou, Yang Zheng, and Jiandong Li. 3d indoor localization based on spectral clustering and weighted backpropagation neural networks. In *2017 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–6. IEEE, 2017.

[36] Alireza Razavi, Mikko Valkama, and Elena-Simona Lohan. K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization. In *2015 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7. IEEE, 2015.

[37] Jenny Röbesaat, Peilin Zhang, Mohamed Abdelaal, and Oliver Theel. An improved ble indoor localization with kalman-based fusion: An experimental study. *Sensors*, 17(5):951, 2017.

[38] Adam Satan. Bluetooth-based indoor navigation mobile system. In *2018 19th international carpathian control conference (ICCC)*, pages 332–337. IEEE, 2018.

[39] Adam Satan and Zsolt Toth. Development of bluetooth based indoor positioning application. In *2018 IEEE international conference on future IoT technologies (Future IoT)*, pages 1–6. IEEE, 2018.

[40] Ye Shang, Zhigang Liu, Jinkuan Wang, and Xianda Xiao. Triangle and centroid localization algorithm based on distance compensation. 2012.

[41] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005.

[42] Zhi-li Wu, Chun-hung Li, Joseph Kee-Yin Ng, and Karl RPH Leung. Location estimation via support vector regression. *IEEE Transactions on mobile computing*, 6(3):311–321, 2007.

[43] Qingwen Xu, Heap-Yih Chong, and Pin-Chao Liao. Collaborative information integration for construction safety monitoring. *Automation in Construction*, 102:120–134, 2019.

[44] Lu Xuanmin, Qiu Yang, Yuan Wenle, and Yang Fan. An improved dynamic prediction fingerprint localization algorithm based on knn. In *2016 Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pages 289–292. IEEE, 2016.

[45] Weixing Xue, Xianghong Hua, Qingquan Li, Weining Qiu, and Xuesheng Peng. Improved clustering algorithm of neighboring reference points based on knn for indoor localization. In *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, pages 1–4. IEEE, 2018.

[46] Jun Yan, Guowen Qi, Bin Kang, Xiaohuan Wu, and Huaping Liu. Extreme learning machine for accurate indoor localization using rssi fingerprints in multifloor environments. *IEEE Internet of Things Journal*, 8(19):14623–14637, 2021.

[47] Jun Yan, Kegen Yu, Ruizhi Chen, and Liang Chen. An improved compressive sensing and received signal strength-based target localization algorithm with unknown target population for wireless local area networks. *Sensors*, 17(6):1246, 2017.

[48] Jun Yan, Lin Zhao, Jian Tang, Yuwei Chen, Ruizhi Chen, and Liang Chen. Hybrid kernel based machine learning using received signal strength measurements for indoor localization. *IEEE Transactions on Vehicular Technology*, 67(3):2824–2829, 2017.

[49] Da Zhang, Feng Xia, Zhuo Yang, Lin Yao, and Wenhong Zhao. Localization technologies for indoor human tracking. In *2010 5th international conference on future information technology*, pages 1–6. IEEE, 2010.

[50] Xiaoqiang Zhu, Wenyu Qu, Tie Qiu, Laiping Zhao, Mohammed Atiquzzaman, and Dapeng Oliver Wu. Indoor intelligent fingerprint-based localization: Principles, approaches and challenges. *IEEE Communications Surveys & Tutorials*, 22(4):2634–2657, 2020.

[51] Wangmeng Zuo, David Zhang, and Kuanquan Wang. On kernel difference-weighted k-nearest neighbor classification. *Pattern Analysis and Applications*, 11:247–257, 2008.