WESTERN SYDNEY UNIVERSITY

Indoor Navigation for the Visually Impaired: Enhancements through Utilisation of the Internet of Things and Deep Learning

Payal Tusharkumar Mahida

A thesis submitted for the degree of

Doctor of Philosophy

School of Computer, Data and Mathematical Sciences
Western Sydney University

November 2021

Acknowledgements

My PhD journey has been full of excitement, frustration and satisfaction. I would like to take the opportunity to acknowledge the people who have inspired, guided and supported me to successfully complete my PhD study. First, I would like to thank Almighty God for his kindness, blessings and strength to overcome all the challenges I faced during the journey.

I would like to express my sincere gratitude to my supervisor, Dr Seyed Shahrestani, for his continued support, guidance and motivation throughout my PhD study. He helped me learn how to identify and solve research problems. His timely feedback and dedication helped me to complete my PhD with high quality. I would also like to extend my gratitude to my co-supervisor, Dr Hon Cheung, for his valuable feedback and comments. His extensive feedback helped me improve my presentation and writing skills. I will always be grateful for his kindness and patience.

I would like to extend my sincere gratitude to Western Sydney University (WSU) for identifying and recognising my hard work and granting me the WSU Post Graduate Research Award. I would like to thank WSU and Associate Professor Dongmo Zhang for supporting me with possible ways to grant allowances for presenting our work at conferences and providing other support in research, including English language support. Thanks to Ahmed, Farhad and Nabil for sharing valuable tips and materials on time. I would also like to thank Dr Milena for her feedback in teaching me academic writing skills and timely feedback. Special thanks to my friends and colleagues for their constant motivation and all their possible support to motivate me.

Last, but not least, I would like to thank my loving family. Their unconditional love, support and faith helped boost me most of the time. Special thanks to my mother, Jyoti, and father, Narendrasinh, for making me who I am today. I am also thankful for my parents-in-law and sisters and their family for their love and support. My sisters, Priyanka and Unnati, have always supported me and constantly admired my determination. My brother-in-law, Hiren, has constantly encouraged me in the learning journey and admired me for the clarity of my work. During the long and intense years of my studies, I lost people from my family who shared the dreams with me. I would like to honour their memory in this work, especially my brother-in-law, Rajdeep.

Thanks to my caring and loving daughter, Priyal, for constantly supporting and motivating me. Special thanks to her for helping me in sharing her ideas and views on my writing and helping to improve it. Thanks to my daughter, Hiya, for being my best companion, for motivating and understanding my situation most of the time, and for growing stronger and smarter by herself. Finally, and most importantly, thanks to my husband, Tushar, for his love, support and understanding. His moral support enabled me to successfully fulfil my dream. This work would not have been possible without his constant love and support. Thank you for being a motivating partner.

I am genuinely grateful to all the amazing people who believed in me and helped me in this wonderful journey. This work is dedicated to all of you.

Statement of Authentication

To the best of my knowledge, I declare that the work presented in this thesis is original. I hereby declare that I have not submitted this material, either in full or in part, for a degree at this or any other institution.

Signed: [redacted]__ On: 26/11/2021

Payal Tusharkumar Mahida

Contents

List of Tables

List of Figures

List of Abbreviations

2-D	Two-dimensional
A-GPS	Assisted Global Positioning System
AI	Artificial Intelligence
AOA	Angle of Arrival
BLE	Bluetooth Low Energy
BVIP	Blue Dot for Visually Impaired People
CART	Classification and Regression Tree
CSI	Channel State Information
DNN	Deep Neural Network
ELU	Exponential Linear Unit
GPRS	General Packet Radio Service
GPS	Global Positioning System
INS	Inertial-based Navigation Systems
IoE	Internet of Everything
IoT	Internet of Things
IP	Internet Protocol
IPIN2016	Indoor Positioning and Indoor Navigation 2016
IR	Infrared Radiation
IT	Information Technology
KNN	K-Nearest Neighbour
LED	Light-Emitting Diode
LOS	Line of Sight
MAE	Mean Absolute Error
MEMS	Micro-Electromechanical System
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NFC	Near Field Communication
OGM	Occupancy Grid Map
OJPS	Orthogonal Jump Point Search
PCA	Principal Component Analysis
POI	Point of Interest

PRM	Probabilistic Roadmap
QR	Quick Response
ReLU	Rectified Linear Unit
RF	Radio Frequency
RFID	Radio Frequency Identification
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RRT	Rapidly Exploring Random Tree
RSSI	Received Signal Strength Indicator
SELU	Scaled Exponential Linear Unit
SVM	Support Vector Machine
TDOA	Time Difference of Arrival
TOA	Time of Arrival
UHF	Ultra High Frequency
UWB	Ultra Wide Band
VI	Visually Impaired
VLC	Visible Light Communication
WHO	World Health Organization
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
XML	Extensible Markup Language

List of Authored Publications

The results and outcomes of the research work in this thesis are published in the following papers.

P. Mahida, S. Shahrestani, and H. Cheung, “Deep Learning-Based Positioning of Visually Impaired People in Indoor Environments,” Sensors, vol. 20, no. 21, p. 6238, Oct. 2020, doi: 10.3390/s20216238.

P. T. Mahida, S. Shahrestani, and H. Cheung, “An improved positioning method in a smart building for a visually impaired user,” in *International Conference on Internet of Things Research and Practice (iCIOTRP2019)*, Sydney, Australia, 2019, pp. 7–12 doi: 10.1109/iCIOTRP48773.2019.00010.

P. T. Mahida, S. Shahrestani, and H. Cheung, “Indoor positioning framework for visually impaired people using Internet of Things,” in *International Conference on Sensing Technology*, Sydney, NSW, Australia, 2019, pp. 198–203 doi: 10.1109/ICST46873.2019.9047704.

P. T. Mahida, S. Shahrestani, and H. Cheung, “DynaPATH: Dynamic Learning Based Indoor Navigation for VIP in IoT Based Environments,” in 2018 *International Conference on Machine Learning and Data Engineering (iCMLDE)*, Dec. 2018, Sydney, NSW, Australia, pp. 8–13, doi: 10.1109/iCMLDE.2018.00012.

P. T. Mahida, S. S. Shahrestani, and H. Cheung, “Comparision of pathfinding algorithms for visually impaired people in IoT based smart buildings,” in *International Telecommunication networks and applications conference (ITNAC 2018)*, Sydney, NSW, Australia, 2018, pp. 10–13, doi: 10.1109/ATNAC.2018.8615350.

P. T. Mahida, S. Shahrestani, and H. Cheung, “Localization techniques in indoor navigation system for visually impaired people,” in 2017 *17th International Symposium on Communications and Information Technologies*, ISCIT 2017, Cairns, Queensland, Australia, vol. 2018-January, pp. 1–6, doi: 10.1109/ISCIT.2017.8261229 doi: 10.1109/ISCIT.2017.8261229.

Abstract

Wayfinding and navigation are essential aspects of independent living that heavily rely on the sense of vision. Walking in a complex building requires knowing one's exact location to find a suitable path to the desired destination, avoiding obstacles and monitoring one's orientation and movement along the route. People who do not have access to sight-dependent information, such as that provided by signage, maps and environmental cues, can encounter challenges in achieving these tasks independently. They can rely on assistance from others or maintain their independence by using assistive technologies and the resources provided by smart environments.

Several solutions have adapted technological innovations to combat navigation in an indoor environment over the last few years. However, there remains a significant lack of a complete solution to aid the navigation requirements of visually impaired (VI) people. The use of a single technology cannot provide a solution to fulfil all the navigation difficulties faced. A hybrid solution using Internet of Things (IoT) devices and deep learning techniques to discern the patterns of an indoor environment may help VI people gain confidence to travel independently. This thesis aims to improve the independence and enhance the journey of VI people in an indoor setting with the proposed framework, using a smartphone.

The thesis proposes a novel framework, Indoor-Nav, to provide a VI-friendly path to avoid obstacles and predict the user's position. The components include Ortho-PATH, Blue Dot for VI People (BVIP), and a deep learning-based indoor positioning model. The work establishes a novel collision-free pathfinding algorithm, Orth-PATH, to generate a VI-friendly path via sensing a grid-based indoor space. Further, to ensure correct movement, with the use of beacons and a smartphone, BVIP monitors the movements and relative position of the moving user. In dark areas without external devices, the research tests the feasibility of using sensory information from a smartphone with a pre-trained regression-based deep learning model to predict the user's absolute position.

The work accomplishes a diverse range of simulations and experiments to confirm the performance and effectiveness of the proposed framework and its components. The results show that Indoor-Nav is the first type of pathfinding algorithm to provide a novel path to reflect the needs of VI people. The approach designs a path alongside walls, avoiding obstacles, and

this research benchmarks the approach with other popular pathfinding algorithms. Further, this research develops a smartphone-based application to test the trajectories of a moving user in an indoor environment. The accuracy of the approach is 99.99%, with accurate turns and orientations of the user. The study shows the use of inertial sensors of smartphone data to discern the patterns of the surrounding environment and position a user with an error of no more than 65 centimetres.

One of the main requirements of this field is to design a low-cost solution and employ technology for those who need it most. As such, self-positioning, safe and VI-friendly paths, with audible directions and warnings, are the main aspects of the proposed framework. The approaches and framework of the thesis contribute to analysing the capability of IoT implementations and deep learning techniques to enhance and improve the indoor journey of VI people. In future, the work will be extended to decrease the positioning errors further by using smartphones' sensors. In addition, future research will test pre-trained models deployed in a smartphone application and remove the dependency on internet connectivity.

Keywords: indoor, navigation, visually impaired, Internet of Things, deep learning, pathfinding, positioning, inertial sensor, smartphone

Chapter 1

Introduction

This chapter introduces the problems that visually impaired (VI) people face in navigating an indoor environment. Further, it discusses the limitations of existing indoor navigation systems, while the research questions highlight this study's intention to enhance the journey of VI people.

1.1. Introduction

A robust outdoor navigation solution is provided by a Global Positioning System (GPS); however, the use of GPS in an indoor environment is not always possible, as the satellite signals on which it relies cannot penetrate most walls [1]. Outdoor navigation has additional infrastructure aids that assist VI navigators, such as voice announcements on buses, talking crosswalks and Braille signs. However, according to Strategy Analytics, people spend 80 to 90% of their time indoors, including in malls, hospitals and other public buildings [2]. Even sighted people find it daunting to navigate complex buildings, such as hospitals, IKEA, and malls. According to the World Health Organization (WHO), 265 million people are estimated to be VI worldwide; among these, 39 million are blind, and 246 million have a low vision [3]. Therefore, VI people may require directional assistance or navigation aids when travelling in a complex indoor environment.

Despite visual maps, individuals face difficulty navigating a new location or room in a complex building. Enhancing one's natural ability to visualise the world and make independent decisions is a challenge for VI people. When travelling in a new environment or public building, VI people may require directional assistance or some form of navigation aid. The most remarkable change in a VI or blind person's life is losing independence with the loss of eyesight. A VI person faces challenges in accessing the world in three main areas: the physical world, symbolic world and social world [3]. The biggest challenge faced by VI people is interacting with the physical world. To interact with the physical world, navigation is a task that is mainly taken for granted by sighted people. External cues tell people to move, stop and change directions, yet navigation and mobility remain difficult for visually impaired people.

Knowing a building's landmarks while passing through to a destination can help people feel more confident about their route. Even when they are familiar with a particular environment, VI people can be unsure if they have passed a landmark on their way to their destination. When visiting a public building, VI people require consistent guidance and may need to know their location, giving rise to questions like *Where am I? Have I passed a landmark on my way to the destination? Am I following the right path?* These questions relate to different areas of the overall problem of indoor navigation. The first question is about knowing their position. The latter questions are related to path planning based on maps and knowing the history of positions—known as the trajectory of the path. Special technologies—such as raised line maps (i.e., tactile maps) and signage information in Braille—can assist a VI person in a complex environment [4]. However, not all VI individuals can read and understand tactile maps [5].

Moreover, tactile maps have limitations, including static information about changing surroundings, which poses difficulty for the VI person to position themselves [6]. A white cane is a luminous physical aid that allows a VI person to scan their surroundings for obstacles within a range; however, it fails to identify obstacles outside the range of 1.2 metres. In some cases, objects above knee level are not sensed. For example, a user may be unaware of a table's existence, as the cane may pass between the table legs, under the table-top.

The development of innovative modern technologies, such as the Internet of Things (IoT) and artificial intelligence, has created possibilities for providing interactive systems to assist VI people in navigating indoor environments independently. The lives of VI people may improve when IoT systems have a real-time understanding of conditions, events and movements in the physical world. The indoor navigation system to enable independent navigation for VI people must track trajectories with knowledge of the VI person's position. Implementation of IoT devices in an indoor environment will offer a visionary service for disabled communities. IoT devices include small items, everyday objects, vehicles, and the like embedded with electronic circuitry, providing network connectivity and capabilities to collect and exchange data [7]. The IoT allows these objects to be sensed and controlled remotely across existing or new infrastructure, creating opportunities for more direct integration with the physical world.

A complex indoor environment can be transformed into a smart environment, where each of its meaningful locations (rooms, doors, stairs and elevators) are connected and can communicate with each other. Providing intelligent indoor environments that can make autonomous and adaptive decisions will bring revolutionary change to the lives of VI people. The use of indoor environment sensor data based on IoT devices is a rising trend to be exploited as a sensing service to automate real-time decision making and automated reasoning. Data extracted from sensors in such an environment can provide change notifications that can be used to make real-time decisions [8] and offer context-aware semantic information, including locations.

For a navigation system, it is crucial to locate the user to enable interaction with the rest of the interconnected devices. Moreover, the user's location information can provide a wide range of other services, such as aiding tourists, finding emergency exits, tracking children in crowded places, and assisting police in rescue operations [9]. A novel approach with interactive and smart devices can effectively guide a VI person in a complex indoor environment. An IoT-based navigation system with audible instructions regarding landmarks and room-level accuracy to travel in a new building can allow VI people to move independently in an indoor environment. As such, a variety of newly developed technologies and systems have been generated and tested [10], yet the designs still suffer from limitations in accuracy, high hardware cost, and lack of additivity and security [11]. The indoor environment must be equipped with an embedded computing system that senses the surroundings using existing Wireless Fidelity (Wi-Fi)-based or new wireless infrastructure. These interconnected IoT devices exchange their data with a remote server, where the data can be analysed and processed. The contribution of this research is to provide a self-directed, accurate and audio-aided navigational system using IoT and deep learning to aid the indoor navigation of VI people.

1.2. Motivation

Directional signs or building maps serve as helpful navigational aids, especially for people entering a new indoor environment. Despite having navigational aids, a sighted person can lack confidence when visiting such a new environment. However, a VI person visiting such an indoor environment is usually not provided with equivalent navigational aids. Given the lack of such aids, it is challenging to seek destinations independently without asking for the help of a sighted person. For a person to follow a route, they must ordinarily have some concept or plan. A VI person can learn a route either by guidance from a sighted escort or by audio instruction. Further, once the route has been learnt, successful travel requires the individual to:

- know their position with respect to the location
- detect and avoid local and global obstacles
- know the moves to reach the destination
- know their directions and follow the suggested route
- detect the entries/exits of doors
- identify landmarks along the way.

Some of the current indoor navigation systems for VI people use Braille keyboards for user input, but not all VI people can read Braille. The accuracy of the system must be within centimetres, as the VI person may be unable to identify the two adjacent doors of a room in a mall or building.

GPS was developed with outdoor positioning applications in mind, with accuracy requirements of 10 to 30 metres [12]. GPS signals in an indoor environment are weak, as standalone GPS receivers cannot detect the satellites when indoors. Indoor positioning solutions using assisted GPS (A-GPS) are proposed to overcome this problem. However, A-GPS obtains assistance in the case of poor signals from cellular networks using mobile by improving the GPS receiver sensitivity by approximately 10 dB [13], which is insufficient to achieve indoor positioning accuracies of under two metres for VI people.

Collision-free pathfinding, positioning and knowing route landmarks can be significant challenges for VI people in an indoor environment. Despite significant progress in recent years, the navigation and pathfinding problem continues to attract research to find an optimal solution for VI people [14]. Research shows that selecting the shortest path to reach a goal location is greatly emphasised. However, the criteria for selecting a path for a low-vision pedestrian include factors apart from the shortest distance. The primary requirement of a pathfinding algorithm for VI users is a safe and line-shoring path with turn-by-turn audio instructions, supported by clues to highlight landmarks and nearby places. Users must be provided with a clear path of travel that avoids local and global obstacles, with minimum diagonal and irregular moves.

In addition, a VI person requires movement instructions, including steps and heading information, such as left and right motions. For example, Figure 1.1 displays an indoor navigation mobile application that suggests the pedestrian move 148 metres to the left. Such instructions can place VI people under pressure, as they have no notion of their surroundings or travelled distance [15].

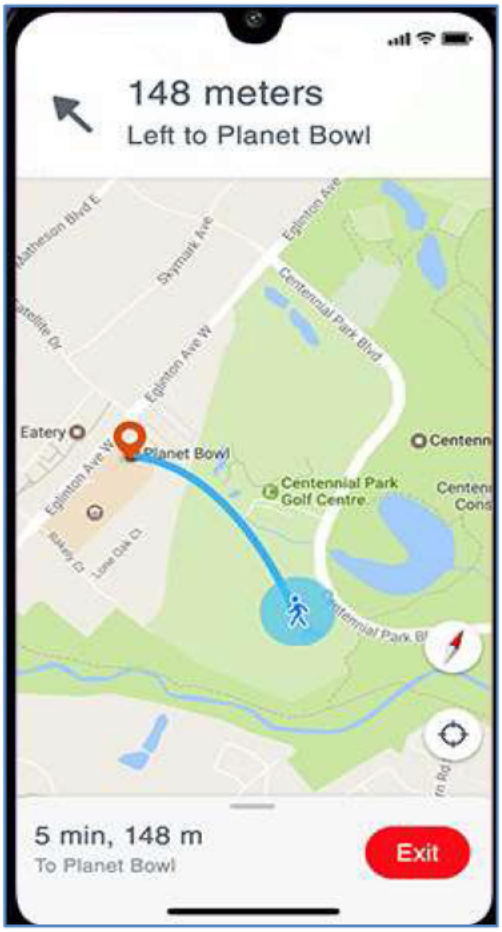


Figure 1.1: Indoor navigation application providing instructions [15]

For VI people, other challenges of indoor navigation systems include positioning, tracking and pathfinding. Indoor pathfinding relies on the shortest path between locations and must react to changes in the path during unexpected changes in the environment. The pathfinding algorithm for VI people must consider avoiding obstacles, including fixed furniture and walls. Moreover, considering technological advancements, the pathfinders must find a route that follows dynamic changes by avoiding non-working elevators, wet floors and similar obstacles. It is mandatory to achieve constant and correct feedback from the smart devices to display such dynamic features [7].

IoT device technology is about transforming physical objects into digital data products. The physical object is embedded with a sensing device to emit data like its location, state, and events. Using the IoT and other contemporary technology advancements can help navigation systems become predictive, collaborative, and intelligent to make decisions for VI people to navigate seamlessly in an indoor environment. The goal of this thesis is to use an IoT device and deep learning algorithms to help a VI person make smart moves independently and build self-confidence with accurate route descriptions, path navigational progress, such as passed landmarks, and accurate absolute positioning.

1.3. Research objectives and questions

The previous section demonstrated that the positioning and navigation of a VI person in an indoor environment are challenging and unresolved issues. The problems faced by VI people in navigating indoors require appropriate attention. Thus, this thesis aimed to address these problems via an innovative smart approach that enhances the indoor journey of VI people by providing an independent navigation system that uses advanced technological IoT devices and deep learning techniques.

1.3.1. Research objectives

The main research goal of the thesis was to use IoT devices and apply deep learning techniques to help VI people navigate an indoor environment independently. The research objectives are presented in Table 1.1.

Table 1.1: Research objectives

RO1: To review the state-of-the-art of indoor navigation for VI people, IoT protocol suite and deep learning techniques.
RO2: To develop an understanding and determine the effects of adapting an appropriate mapping technique for suitable IoT devices in the complex indoor infrastructure.
RO3: To explore indoor collision-free pathfinding algorithms, propose a dynamic and VI-friendly path generation mechanism, and measure its effectiveness.
RO4: To propose and develop an efficient tracking and positioning model to promote independent and centimetre-level accurate movements of VI people in an indoor environment.
RO5: To propose a framework for effective and efficient indoor navigation for VI people to meet their needs.

1.3.2. Research questions

It is essential to understand the difficulties faced by VI users when navigating an indoor environment. This understanding led to the following primary research questions.

> **RQ1:** What are the main factors hindering a VI person's indoor navigation? What are the critical issues to be resolved to create an effective and accurate indoor navigation system for a VI person?

Further, the research objectives led to the following sub-questions:

> **RQ2:** How can technological advancement, including IoT, help develop a smart navigation guide for VI people?
>
> **RQ3:** Which type of representation technique for indoor maps should be adopted to suit the requirements of VI people?
>
> **RQ4:** What are the criteria to be considered to evaluate the optimal pathfinding algorithm for a VI person? Which pathfinding algorithm is suitable for VI people to navigate in a dynamically changing environment?
>
> **RQ5:** How can the user's movements based on the suggested path be tracked? How can smart devices help a VI person self-localise without assistance?
>
> **RQ6:** How can indoor positioning accuracy be improved by using hybrid technologies for VI people?
>
> **RQ7:** What is the best deployment solution of the proposed architecture for indoor navigation?

To answer the identified research questions, a review of existing indoor navigation systems was performed. Based on the review results, traditional and existing methods were studied and simulated to a certain extent to conclude the limitation and usability of the systems. Further, scientific experiments and emulations were performed to address the problems faced by VI people in indoor environments related to pathfinding, tracking and positioning. The research goal in this thesis was to develop better surroundings for the community with low vision using IoT devices and deep learning techniques. The findings of the study will improve the independent indoor navigation of VI people. The results will potentially help develop a robust and useful mobile-based, low-cost application design with IoT-embedded surroundings. The study's contribution is original and in demand, considering the rapid explosion of IoT devices and the trend of deep learning-based applications in today's digital world.

1.4. Research methodology

The research methodology of this research is a combination of theoretical and experimental work and design. This section discusses the research methodology adapted after the review of the underlying needs of the VI people for indoor navigation. The research project sought to improve the living standard of VI people by using IoT devices and deep learning techniques. Although research has been undertaken on indoor navigation, few systems have been specifically designed to consider communities with low vision [16]. The literature shows that the indoor location-aware system structure [17] for pedestrians consists of four layers: the Wireless Local Area Network (WLAN) interface, localisation layer, tracking and fusion layer, and application layer. Figure 1.2 presents an architectural diagram of a location-aware system based on Wi-Fi. The indoor map digital information is fed as input to all the layers depicted in the figure. We initiated this research by studying existing pedestrian-based navigation systems.

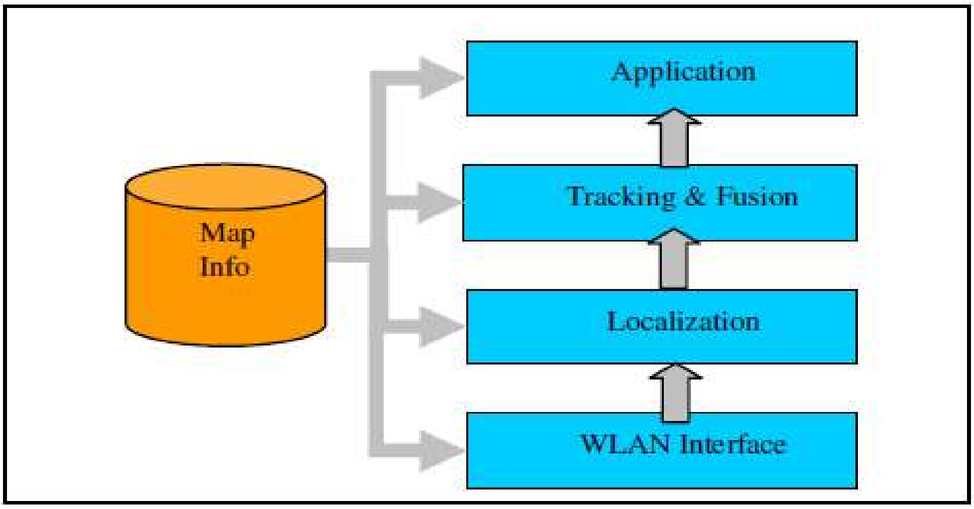


Figure 1.2: Architecture of Wi-Fi-based location-aware system for pedestrians [17]

There were numerous questions to be researched, so this study is divided into three primary stages to head in the best possible direction, as shown in Figure 1.3. Through this organisation of stages, the research is divided into smaller components to monitor progress at a granular level. The three stages of the research were research study; data collection, evaluation and testing; and framework design, implementation and performance. In the first stage, we performed a preliminary study to identify the issues faced by VI people while navigating indoors. We defined the parameters to consider in the literature review of the existing indoor positioning technology and its limitations.

In the research study, we initiated the study of popular positioning technology and algorithms to identify a single/hybrid technology to meet the requirements of VI people. Further, we explored the use of IoT and deep learning in indoor navigation to propose a suitable positioning technology and solution. The research questions and scope were framed to indicate the direction of the enquiry and focus related to the research. Finally, we reviewed literature papers and research on indoor navigation for VI people. Given that not much work has been undertaken for VI people, we explored the results and approaches of navigation systems for pedestrians indoors. The works related to the research study are presented in Chapters 1 and 2 of the thesis. The goal of the second phase is to propose a framework to help a VI person navigate an indoor environment accurately. The literature review revealed that the significant challenges in navigating indoors are generating an obstacle-free path, tracking and positioning VI people in an indoor environment. Therefore, this phase is divided into three significant stages: proposing a pathfinding algorithm that can avoid obstacles, a tracking system, and a positioning system for VI people.

Although numerous researches related to pathfinding has been undertaken for indoor navigation, very few studies have focused on addressing the needs of VI people. An appropriate indoor space representation structure acts as an initial stage for further decisions reflecting the needs of pathfinding algorithms for VI people. We investigated the function of existing discrete and sampling-based pathfinding algorithms with different indoor space scenarios based on the selected representation technique. With the thought, one of the algorithms can be adapted, and we initiated the process to relate its function to the needs of VI people. However, given that none of them considered the requirements of VI people, we proposed a novel algorithm, 'Ortho-PATH', to generate a path in an indoor environment for VI people. Further, we compared the results with the most popularly used algorithms for the indoor environment.

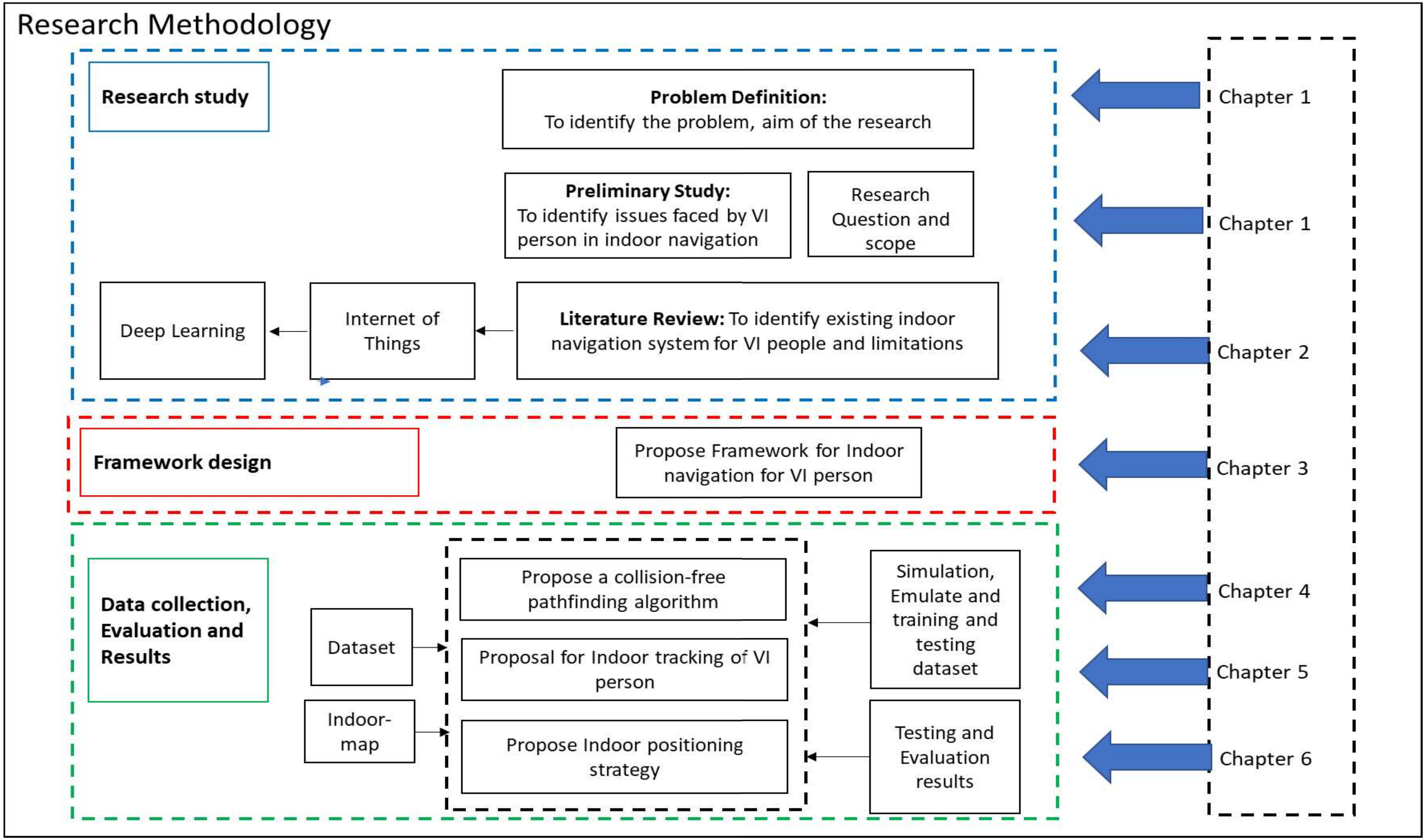


Figure 1.3: Research methodology

There is a need to track a moving user indoors and update the path as the user moves to recalculate the path if the user becomes lost. The accuracy of the tracking system must be within centimetres to help a VI person navigate individually—for example, to differentiate between two adjacent doors in a building. For a walking person, it is essential to know the distance travelled, turns taken, and heading of the user moving in an indoor space. With this basic understanding, we proposed a combination of the absolute and relative positioning of the user. In addition, we investigated whether convenient IoT devices, such as a smartphone or iBeacon embedded in the entry/exit could help a VI person navigate individually. Based on experiments and emulations, we developed a dynamic fusion algorithm based on inertial sensors—the accelerometer, gyroscope, and magnetometer of a user.

To predict the position of the user in an indoor environment and track a user indoors, we studied existing methods and algorithms. Appropriate simulations have supported the results, emulations, and experiments to check the performance of the work at each stage. The following chapters have distributed the literature review with proposed methods with their simulations and experiments to enhance the navigation journey of VI people in an indoor environment.

1.5. Contributions

The key objective of this thesis is to propose an accurate and robust approach for indoor navigation and positioning of a VI person. Although indoor navigation research is actively tackling positioning problems from several domains, not much work has been undertaken for VI people. The section summarizes the publication outcomes as the thesis's contribution with novel approaches and methods resolving pathfinding, tracking and positioning problems faced by VI people in indoor navigation.

1. **Literature review:** The preliminary task of the research is to understand the positioning that enables wireless technologies and algorithms used in real-world scenarios. A comprehensive overview and study of key implementation technologies for localisation for VI people is undertaken. This work was published in:

    - P. T. Mahida, S. Shahrestani, and H. Cheung, "Localization techniques in indoor navigation system for visually impaired people," in *17th International Symposium on Communications and Information Technologies (ISCIT)*, Cairns, Australia Jul. 2017, pp. 1–6, doi: 10.1109/ISCIT.2017.8261229.

2. **Novel collision-free pathfinding algorithm:** Following the study of existing systems, we explored and gathered the requirements for a VI person. For a complete indoor navigation solution, an effective digital space representation method is a starting point. Therefore, we have explored different techniques for indoor environment representation to suit the requirements of VI people. Further, we validated traditional and popular pathfinding algorithms to determine their feasibility in meeting the needs of VI people. The performance evaluation and comparison of pathfinding algorithms' experimental results led us to identify that the design and algorithm needed to consider and focus on the critical requirements of a VI person. The contributions of the comparison were published in:

   - P. T. Mahida, S. S. Shahrestani, and H. Cheung, "Comparison of pathfinding algorithms for visually impaired people in IoT based smart buildings," in *International Telecommunication Networks and Applications Conference (ITNAC)*, Sydney, Australia 2018, pp. 10–3, doi: 10.1109/ATNAC.2018.8615350.

   To address the requirements of VI people in path generation, we developed an Ortho-PATH novel pathfinding algorithm that uses IoT devices embedded in the interiors of the indoor space. Additionally, we proposed a new framework, DynaPATH, to deploy grid-based indoor spaces and generate orthogonal, shore-line and VI-friendly paths. The contributed research article discussing the novel approach is:

   - P. T. Mahida, S. Shahrestani, and H. Cheung, "DynaPATH: Dynamic learning based indoor navigation for VIP in IoT based environments," in *International Conference on Machine Learning and Data Engineering (iCMLDE)*, Sydney, Australia Dec. 2018, pp. 8–13, doi: 10.1109/iCMLDE.2018.00012.

3. **Reliable tracking technique:** A VI person can follow an obstacle-free path provided by the DynaPATH framework. However, tracking the person to ensure they follow the correct path is equally important. We proposed a novel Blue Dot for VI People (BVIP) framework that combines absolute positioning using beacons and a relative micro-electromechanical system (MEMS)-based position learning technique. The relative positioning is based on the fusion of inertial sensors that track the user with dynamic threshold-based distance estimation, with 99.99% of heading and turn detection accuracy using a machine learning technique. We proposed and implemented an

Android-based tracking application that customises each user's walking style and step size. The novel approach was published in:

- P. T. Mahida, S. Shahrestani, and H. Cheung, "An improved positioning method in a smart building for visually impaired user," *International Conference on Internet of Things Research and Practice (iCIOTRP2019)*, Sydney, Australia 2019, pp. 7–12, doi: 10.1109/iCIOTRP48773.2019.00010.
- P. T. Mahida, S. Shahrestani, and H. Cheung, "Indoor positioning framework for visually impaired people using Internet of Things," in *International Conference on Sensing Technology*, Sydney, Australia 2019, pp. 198–203, doi: 10.1109/ICST46873.2019.9047704.

4. **Centimetre-level positioning technique:** Lack of sufficient external beacons in an area, such as a hallway, may generate difficulty in positioning a VI person. Therefore, we provided a robust independent inertial guidance tool to position a VI person in an indoor environment. To the best of our knowledge, this work is the first to propose and recommend regression-based neural training to estimate the position of a VI person based on the inertial sensors of a smartphone. We experimented with a deep neural network model to predict the position of a VI person as a complementary system to our navigation framework using external sensors. The absolute position error on a grid-based indoor space is approximately 65 centimetres, which is reasonable for VI people. The contributed research article that discussed the novel approach was:

   - P. Mahida, S. Shahrestani, and H. Cheung, "Deep learning-based positioning of visually impaired people in indoor environments," *Sensors*, vol. 20, no. 21, p. 6238, Oct. 2020, doi: 10.3390/s20216238.

1.6. Thesis structure

The structure of this thesis is presented in Figure 1.4 and organised as follows. **Chapter 2** presents the design and framework of Indoor-Nav for VI people to enhance and assist their indoor journey using IoT devices and a deep learning algorithm. The chapter presents the major components of the framework, while the further chapters discuss the implementation and experimental evaluation.

Chapter 3 discusses the limitations of the reviewed literature related to pathfinding, tracking and positioning in an indoor environment for VI people. Further, the chapter provides backgrounds on the basic concepts used in this thesis, including indoor navigation systems, wireless positioning technologies, IoT and deep learning.

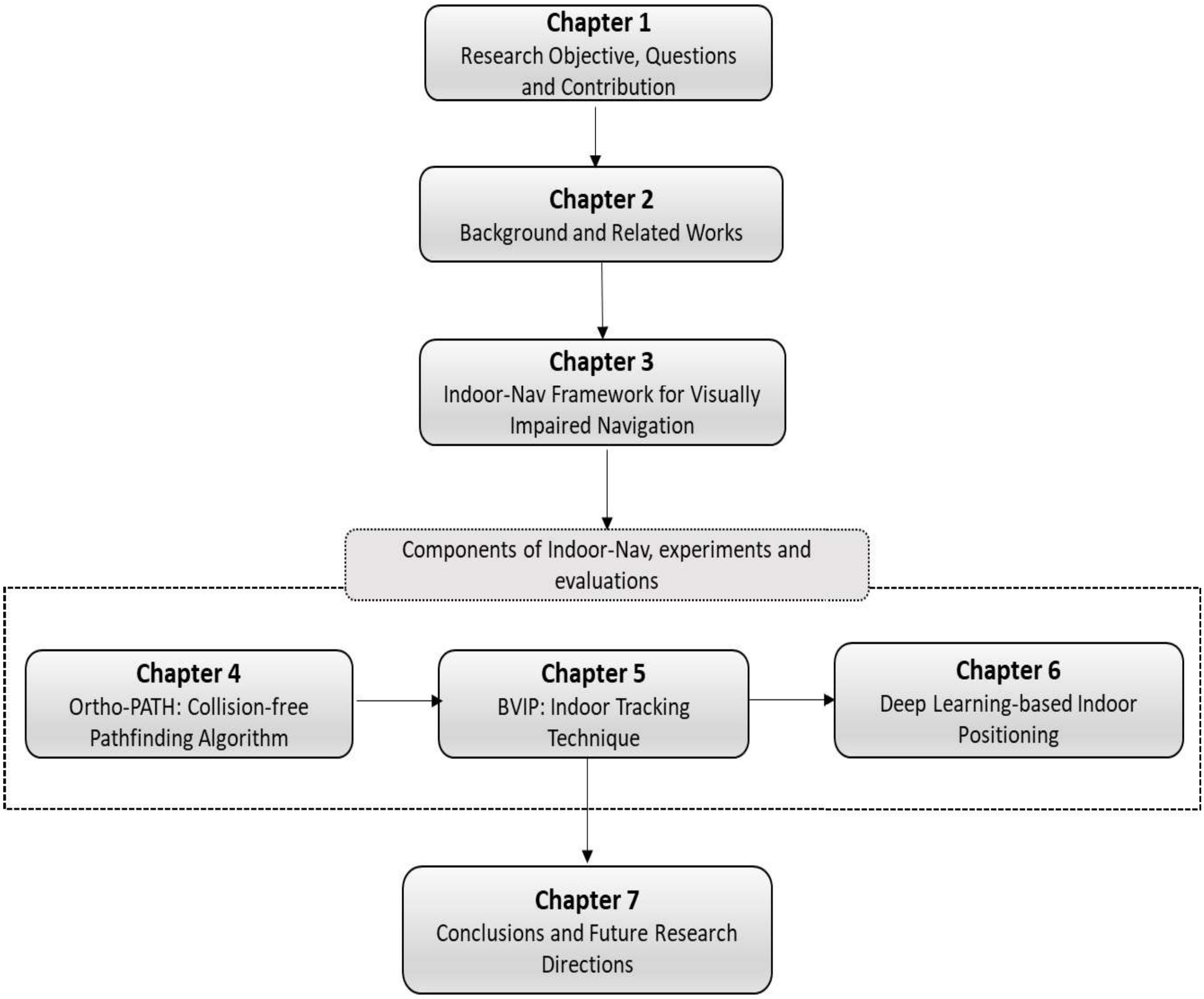


Figure 1.4: Structure of the thesis

Chapter 4 introduces various techniques to represent indoor space and traditional pathfinding algorithms. Further, it discusses the limitations of the traditional pathfinding algorithms for a VI agent moving in an indoor space. Finally, the chapter proposes a novel pathfinding algorithm, Ortho-PATH, which provides VIP-friendly paths that avoid global and local obstacles.

Chapter 5 presents an indoor tracking approach, BVIP, using beacons embedded in building locations and a smartphone held by the user. The approach proposes a reliable fusion algorithm that predicts the relative position of the VI user with threshold-based step detection, heading detection using an inertial sensor and a machine learning turn detection approach.

Chapter 6 explores a deep learning-based absolute positioning approach using a smartphone's inertial sensor dataset. The deep learning model is trained online and offline on a gathered public dataset produced by inertial sensors, including the accelerometer, gyroscope, and magnetometer of a smartphone to provide centimetre-level positioning.

Chapter 7 concludes this thesis, discusses the presented contributions in the context of the identified research questions, and offers a stance for ongoing directions for future research.

Chapter 2

Background and Related Works

This chapter introduces the significant challenges faced by VI people in navigating a complex indoor environment. The chapter illustrates the existing navigation systems provided to the community for navigating indoors, including their limitations. The chapter identifies the major components of an indoor navigation system from the literature review. It also introduces IoT and deep learning techniques applicable to establishing smart environments suitable for assisting with navigations by VI people. The chapter discusses the background of various wireless positioning technologies, with a comprehensive study of their advantages and disadvantages. Further, the chapter presents an in-depth literature review of the indoor navigation system components, including pathfinding algorithms, tracking, and positioning.

Some parts of the survey work reported in this chapter have previously appeared in:
P. T. Mahida, S. Shahrestani, and H. Cheung, "Localization techniques in indoor navigation system for visually impaired people," in *ISCIT 2017*, Cairns, Queensland, Australia, Jul. 2017, pp. 1–6.

2.1. Overview

An indoor navigation system provides path-guided information to help users reach their destination via determining the precise and accurate position of the user in an enclosed structure, such as a building, mall, hospital, or university campus. There is always a need for appropriate indoor positioning and pathfinding systems—especially a complete solution to allow VI people to navigate comfortably and independently. Positioning technology and algorithms, tracking a moving user, pathfinding and obstacle detection are the most challenging aspects of mobility in indoor navigation, especially for a VI person [1]. This chapter revisits the existing indoor navigation solutions for VI people to discuss their implemented strategies and limitations.

Section 2.2 introduces the general requirements of VI people while navigating indoors and details RSNAVI—an indoor navigation solution. The work highlights the architecture and significant components of the navigation system based on the literature review. Section 2.3 reviews and compares existing indoor navigation systems for VI people. Section 2.4 introduces the requirement for an appropriate map representation technique and discusses popular indoor collision-free pathfinding algorithms, while Section 2.5 discusses the characteristics, applicability, and reference model of IoT. Section 2.6 explores popular indoor positioning technologies used primarily in indoor navigation systems. Section 2.7 provides the background of the deep learning technique, including the structure and parameters for positioning an indoor user. Section 2.8 discusses our intensive research and review of components of indoor navigation systems, including positioning, tracking and pathfinding algorithms implemented for VI people, as well as their limitations. Section 2.9 discusses issues and gaps in existing indoor navigation and the need to provide a solution to meet the navigation requirements of VI people. Finally, Section 2.10 summarises the chapter.

2.2. Introduction

Existing outdoor navigation and mapping applications use GPS—a technology successful in navigation and finding locations worldwide within a range of five to 15 metres of accuracy [9]. However, a GPS-based outdoor navigation system does not help navigate indoors, given the lack of line of sight (LOS). After travelling a long distance, the power of GPS signals is relatively low and is compromised further by obstructions between its antenna and the transmitting satellite. Also, GPS signals are blocked and reflected by the walls of buildings, increasing the difficulty of locating a user indoors, given the insufficient signal strength inside buildings [13]. Multipath effect signals are also reflected and attenuated by noise interference from walls and furniture inside buildings [1], [13]. Complex indoor spaces usually have good signage or directional maps to find specific locations and indicate directions to move. These complex indoor environments include airports, exhibition halls, supermarkets, campuses, car parks and other environments [18]. Travelling independently in such a complex environment is a challenging task for a VI person. Tactile signs, sign boards, and Braille signs are solutions to virtually help VI users realise their position and direction to the destination in a complex indoor environment [19]. A VI person without advanced navigational aids performs obstacle detection and avoidance using traditional methods, such as a white cane or guide dog, in indoor and outdoor environments [20].

Moreover, providing merely a distance description without enriched information of indoor maps is insufficient and might negatively influence the mobility and autonomy of VI users [20]. It is necessary to provide indoor location-based navigational services to VI people in such a complex environment, considering the rapid growth and technological advancement of IoT devices and deep learning. Literature work in this field evaluates beacons, sensors, near field communication (NFC), radio frequency identification (RFID) and Wi-Fi as an interface providing location information in the indoor environment. These technologies interact with physical world entities, collect information, and evaluate and provide location information back to the system [21]. Several designs and implementations of connected indoor environments based on emerging technology have provided assistive technology to the special-needs community.

Figure 2.1 represents the architecture of the RSNAVI [22], an RFID-based context-aware indoor navigation system for VI users. There are three layers to the architecture of RSNAVI: sensing, network and application layers [22]. The sensing layer consists of various interconnected sensing devices that help identify obstacles in the path of VI people. The layer contains RFIDs, passive RFID tags and smartphones to monitor and track obstacles. The lower layer collects information about the surroundings and passes it to the middle layer. The middle network layer exchanges the information gathered from the sensing layer to Internet-located services. The application layer is a Java-based web service to retrieve and process the incoming information about the surroundings to the VI person. If a network connection between the control and monitoring gateway does not exist, the system uses SMS-based communication via a General Packet Radio Service (GPRS) modem.

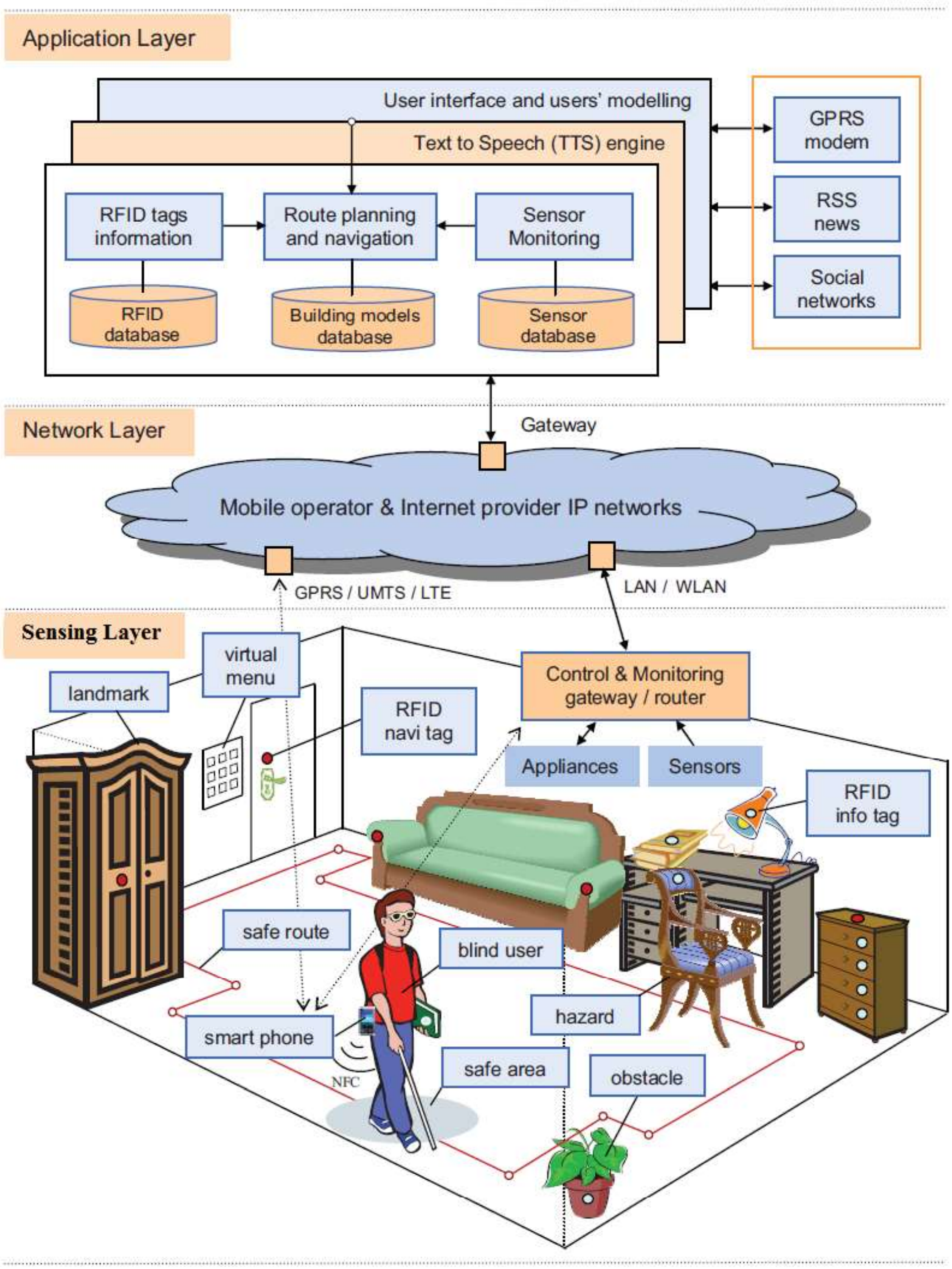


Figure 2.1: RSNAVI—architecture based on RFID context-aware indoor navigation for VI people [22]

Rosen suggested a position information model, including semantic and geometric information, to provide location-based on the dead-reckoning technique [22]. However, the system proposes using a multi-parametric optimal route considering the limitation of dead reckoning. The path is estimated in two phases—a rough and accurate estimate of various parameters, including static obstacles, emergencies, landmarks and turns.

Many navigation systems for VI people do not comply with the specific orientation and movement of VI people in an indoor environment. The accuracy required for VI people must be within centimetres to locate adjacent doors to find the correct room in a building. Researchers have invested significant effort to develop approaches or systems based on fast-developing technologies and sensors. Some of these include wireless communication transmissions, RFIDs and wearable computing devices [2], [23]. However, the research community has not converged to a single, universal system that provides high accuracy and quick response for indoor navigation systems. A navigation system comprises an indoor map, pathfinding and obstacle detection, position estimation and real-time indoor tracking, as shown in Figure 2.2. Generation and representation of indoor digital maps to local coordinate systems may help people find their way in an indoor environment. However, it is difficult for a VI person to interpret maps directly and accurately find their way to their destination.

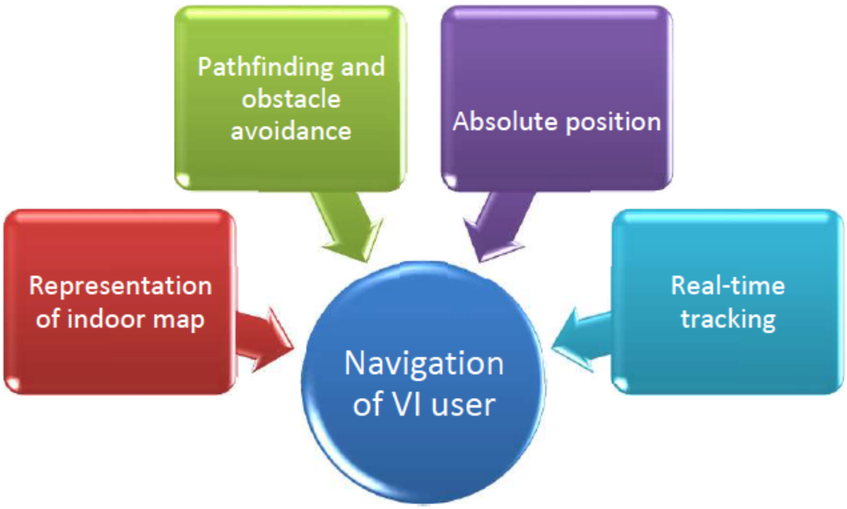


Figure 2.2: Framework and processes in indoor navigation for VI people

Representation of an indoor map is the first process for an indoor navigation framework. The indoor map provides information regarding locations, destinations, landmarks, routes and obstacles, such as fixed objects (including stairs, elevators and walls) and rearrangeable objects. The absolute positioning approach helps find the position of the user moving in the indoor environment. Alongside positioning, real-time tracking of a moving user is necessary, as the VI user may become lost or not follow the suggested path. Despite many challenges in existing navigation systems, IoT devices can offer VI users accurate, precise, audio-aided information to navigate independently. Section 2.3 discusses the state-of-the-art of existing indoor navigation systems with different implemented technologies and algorithms.

2.3. Indoor navigation systems for visually impaired people

Several approaches have sought to develop indoor navigation systems; however, not many deployments have been successful. As a result, the number of indoor location solutions reaching the market has risen dramatically. However, indoor navigation systems for VI people are beyond trials, and the experiment has not yet taken off [24]. Several projects have provided solutions to resolve indoor navigation, reflecting installation difficulties, lack of LOS, accuracy and interference. Most of these systems, such as BlindSquare [25] and Nearby Explorer [26], assist VI people to navigate outdoors using mobile devices and installed IoT assistive technology, including beacons, RFID, NFC and Quick Response (QR) codes [25],[26]. Some use short-range solutions, whereas others have long-range precision technology, increasing the system's cost [11]. An indoor navigation system for VI people has various criteria to consider. The primary concerns are accuracy, ease of use, real-time context awareness and obstacle-avoidance wayfinding mechanisms. Section 2.3.1 provides a literature review of existing indoor navigation systems for VI people, including their features and limitations.

2.3.1. Literature review of indoor navigation systems

The first indoor navigation system used the infrared radiation (IR) approach [27]. The primary use of IR signals is to detect or track objects. IR exists just below the red edge of the visible spectrum, which makes this technology less intrusive than those based on visible light [27]. The IR-based system has a more extended range yet suffers from interference [28] from florescent lights and sunlight [1]. Some authors have proposed navigation systems based on Zigbee and Bluetooth [24], [28], [29], [30] to achieve high accuracy, but the radio signals complicate the propagation and deteriorate the results. Moreover, Bluetooth devices require 10 to 30 seconds per scan. Hence, these systems' latency rate and power consumption are unsuitable for real-time positioning applications [28].

Nakajima [31] proposed using Visible Light Communication (VLC) and geomagnetic sensors to position and localise users in indoor environments. A Panasonic Cloud environment is used to infer position information with a geomagnetic correction algorithm. However, the authors noted fluctuation in geomagnetic values because of noise and disturbance in the building. In addition, the position of the mobile device and floor of the building deteriorated the reliability of the solution. SUGAR [11] is an ultra wide band (UWB)-based navigation system that provides high accuracy, with a 95% confidence interval. However, the implementation of wayfinding involves a grid-based A* algorithm with high computation complexity, given the square cell division of the indoor region.

Guerrero suggested a micro-navigation system using an infrared camera to detect the user's position and movement [32]. The system implements static obstacle detection using an Extensible Markup Language (XML)-based tree structure of the room containing the corresponding data. However, the experiment and simulation of the system demonstrated that the system degrades with multiple users travelling in the environment. PERCEPT is an indoor navigation system based on Kiosk and passive RFID tags used for positioning [30]. It is a macro-navigation system to assist VI people to travel from one room to another located on different floors, based on the shortest path algorithm. However, the reported system does not detect any surrounding obstacles in the environment. Further, the author recommended improving the system by providing direction information based on steps and user preferences.

The Path-Guided navigation system presents wall-mounted grid-based IR tags for positioning [28]. The author also developed a waist wearable device for VI users that transmits information to their smartphones. A vision-based IR technology is used in the system to make it cost-effective and accurate. RFID tags have received growing interest and demand because of the simplicity and low-cost type of small IoT devices. RFID is used in various navigation solutions to improve the daily needs and lives of VI people [19], [33], [34]. Saleh proposed an indoor navigation aid based on active RFID tags and QR codes for VI and sighted users, respectively [19]. The system determines the user's position using eight attenuation power levels and signal strength using a smartphone. Emidio proposed an Ultra-High Frequency (UHF) RFID technology for long-range detection and tags for positioning the user [35]. The author evaluated the user's movement in a scattered environment considering the dynamic nature of indoor spaces. Dristi sought to develop an assistive navigation system to assist VI people with high accuracy [36], installing ultrasound pilots in the environment to provide accurate positioning. However, this system involves a high monetary cost to deploy, and the initial setup restricts its applicability to wide-scale implementation, given the cost. Further, lockage and disturbance in radio frequency (RF) signals degrade the accuracy of the system in certain dead spots.

With the growing demand for smartphones, Chen proposed a navigation system based on a handheld mobile device and the existing Wi-Fi infrastructure of the building [37]. The researcher implemented a clustering algorithm and Kalman filter for the refinement of position and movement information. The solution uses Wi-Fi technology with a fingerprinting database.

However, it suffers from low performance and high computation requirements. Gallagher implemented a sensor fusion based system that runs locally on a mid-range smartphone [38]. It relies on a Kalman filter that fuses all the sensors available on the smartphone, including a Wi-Fi chipset, accelerometer and magnetic sensors. Wi-Fi-based positioning systems have also been proposed for navigation systems using the Received Signal Strength Indicator (RSSI) [22] [19]. However, because of multipath distortions and power consumption limitations, RSSI-based navigation systems do not estimate positions with high accuracy. Therefore, Wi-Fi-based RSSI signals with fingerprinting techniques are predominantly used for position determination. However, the fingerprinting-based navigation technique is highly dependent on training procedures and maintenance efforts [37]. Advanced technologies such as UWB and ultrasound used in the navigation systems provide centimetre-level accuracy [11], [36]. However, these systems use Ubisense sensors and ultrasound tags, which increases the overall cost estimation of the system.

NavCog is a smartphone-based turn-by-turn navigation system for VI users, employing a network of Bluetooth low energy (BLE) beacons with an approach of K-nearest neighbour (KNN) algorithm [39]. The author claimed that NavCog achieves more precise localisation information than GPS- and Wi-Fi-based methods. Participants were less concerned with precision yet demanded the application make them aware of missed turns and provided a rerouted path to the destination. LowViz [40], AudiblEye and NavVis are some of the latest indoor navigation mobile applications for VI people. LowViz is an indoor navigation app launched in 2015 that uses various technologies, including inertial sensors, Wi-Fi and BLE beacons, to guarantee high accuracy [40]. LowViz uses a combination of positioning algorithms, including sensor fusion and post-processing methodology, for data refinement to provide high accuracy. However, context-aware real-time pathfinding is not yet included in the system.

Table 2.1 summarises and compares the existing indoor navigation systems based on localisation technology and techniques, accuracy, space, wayfinding algorithm, obstacle avoidance and cost. It provides an insight into the systems' positioning technology, distance estimation metrics, location estimation, accuracy, space, pathfinding algorithm, obstacle avoidance and cost. The table shows the limitations of each technology in meeting the accuracy and dynamicity requirements of VI people.

Table 2.1: Summary and comparison of indoor navigation systems for VI people

Ref.	Wireless technologies	Distance estimation metrics	Location estimation	Accuracy	Space	Pathfinding algorithm	Obstacle avoidance	Cost
[31]	VLC + Wi-Fi + Bluetooth	AOA	Custom algorithm using geomagnetic sensor	1–2 m	3-D	–	No	Low
[11]	UWB + Wi-Fi	TDOA and TOA	Custom algorithm based on Compass	20 cm	2-D	A*	Static	High
[32]	Infrared camera, LEDs + Wi-Fi + Bluetooth		Triangulation	15 cm	2-D	Shortest path	Static	High
[30]	Bluetooth + passive RFID + Wi-Fi	-	-	–	3-D	Shortest path	No	Mediu m
[22]	Wi-Fi + BLE + passive RFID tags Inertial sensors	Dead-reckoning technique	Dead reckoning		4-D	Multi-parametric optimisation	Static	Low
[28]	Bluetooth + infrared tags	-	Step detection (accelerometer)	±1.6 m	3-D	Dijkstra's algorithm	Static	Low
[19]	Wi-Fi + active RFID + QR codes	RSS, power level	Closest tag algorithm	High	3-D	Dijkstra's algorithm	Static	Mediu m
[41]	UHF RFID tags + Wi-Fi	RSS	Proximity	High	2-D	Dijkstra's algorithm	Dynamic	High
[42]	Smartguide attached to cane + RFID + Wi-Fi	GPS + compass	Dead reckoning	High	3-D	Shortest path	–	High
[37]	Wi-Fi	RSS	Fingerprinting, map adaptive Kalman filtering	High	2-D	Shortest path	No	Low
[28]	Ultrasound beacon + Wi-Fi + wearable computer		Triangulation, step and degree detection	22 cm	2-D	–	Dynamic	High

Note: AOA = angle of arrival, LED = light-emitting diode, TDOA = time difference of arrival, TOA = time of arrival.

2.3.2. Building blocks of indoor navigation systems

The literature review related to indoor navigation in Section 2.3.1 explored the direction and helped identify the four major components of an indoor navigation system. The study of various navigation systems and the process of investigating the building blocks of a robust indoor navigation system for a VI person rely on a combination of dynamic mobility and orientation skills [43], [44]. Figure 2.3 presents the building blocks of an indoor navigation system for VI people. Indoor localisation technology builds the backbone network for the indoor space to interact with the connected indoor world. It helps estimate the location of the user and other interior objects in the indoor environment. However, automatic positioning is not all about navigation systems. Other factors are required, such as indoor data design, considering real-time changes in the environment. Unlike outdoor space data models, indoors must consider geometric, topological, semantic and temporal data, reflecting structure, connectivity and real-time data in the environment, updated regularly in Cloud services [45]. Based on these environmental and obstacle data, the pathfinding algorithm suggests a real-time turn-by-turn optimal route from A to B to the user. Speech recognition and audio output services are convenient ways for VI users to interact with the system [20].

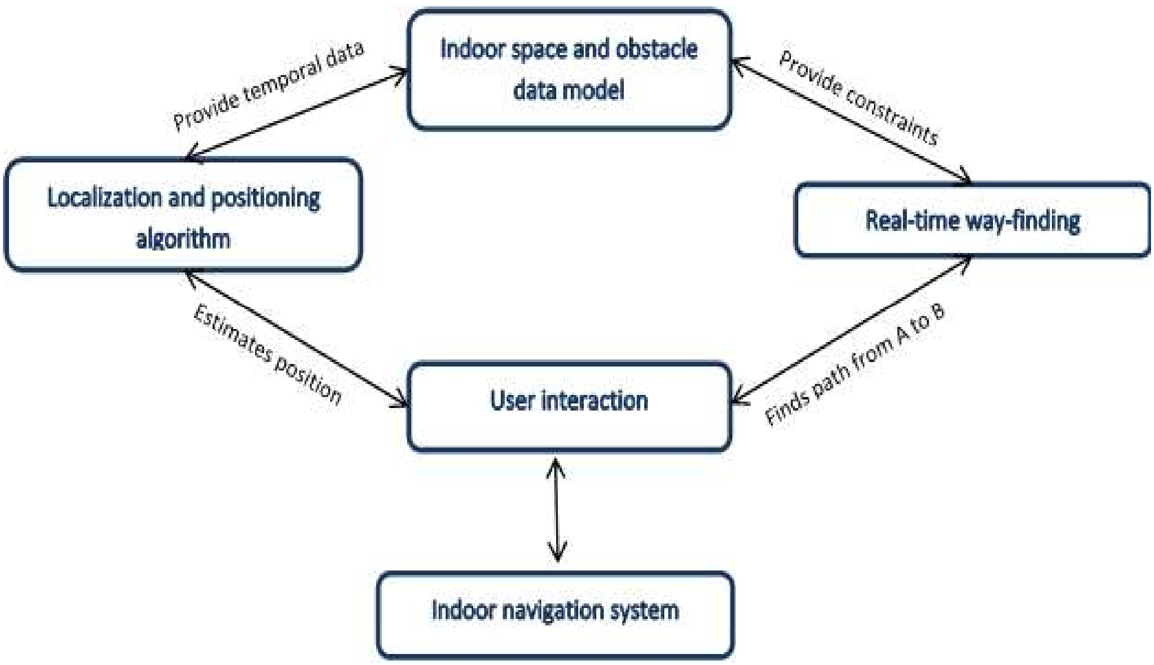


Figure 2.3: Building blocks of indoor navigation framework

From related studies, it is concluded that localisation technology, a positioning algorithm, an indoor space and obstacle data model, real-time wayfinding and user interaction are the most significant components of an indoor navigation system. The gaps relevant to each component of the navigation system are further discussed in the following sections.

2.4. Representation of indoor space and pathfinding algorithms

Collision-free pathfinding is an essential component of an indoor navigation system that allows an agent to find and plan a route from the start location to the goal location, avoiding obstacles along the way. One possible obstacle is a 'U'-shaped obstacle, as shown in Figure 2.4. An efficient pathfinding process should scan a larger area and find the shorter path (blue) rather than following the longer path (red). As shown in the figure, the agent can follow a shorter and more efficient path by identifying an obstacle from a far distance than the traditional one. Both the routes (red and blue) avoid obstacles, but the blue route makes an intelligent move to detect an obstacle from a far distance, making this route optimal and efficient.

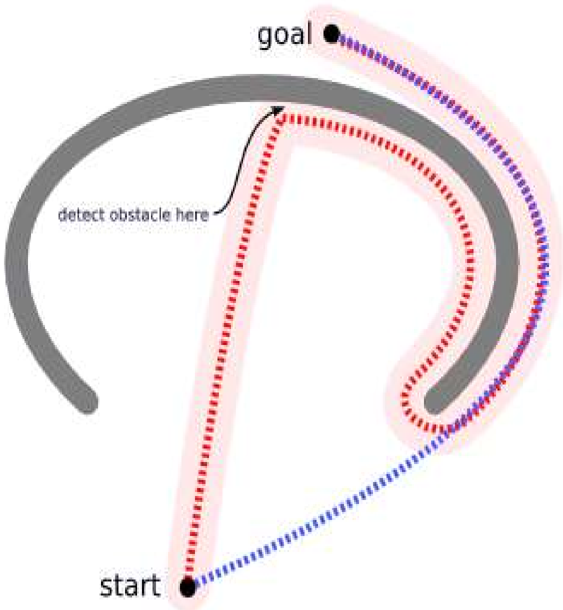


Figure 2.4: Local obstacle avoidance in pathfinding algorithm

Alongside local obstacles, a pathfinding algorithm must also avoid global obstacles, such as blockage of a route because of a wet floor or unavailability of vertical connectors, such as lifts, elevators and stairs. Adapting VI users' preferences and considering the environment in selecting a path are prominent features of a pathfinding algorithm. For example, as shown in Figure 2.5, a shorter path via a non-working lift can be avoided by employing a pathfinding algorithm over a suboptimal path. Such runtime dynamic adaptation in a complex environment helps solve VI people's limitations and enables them to travel independently.

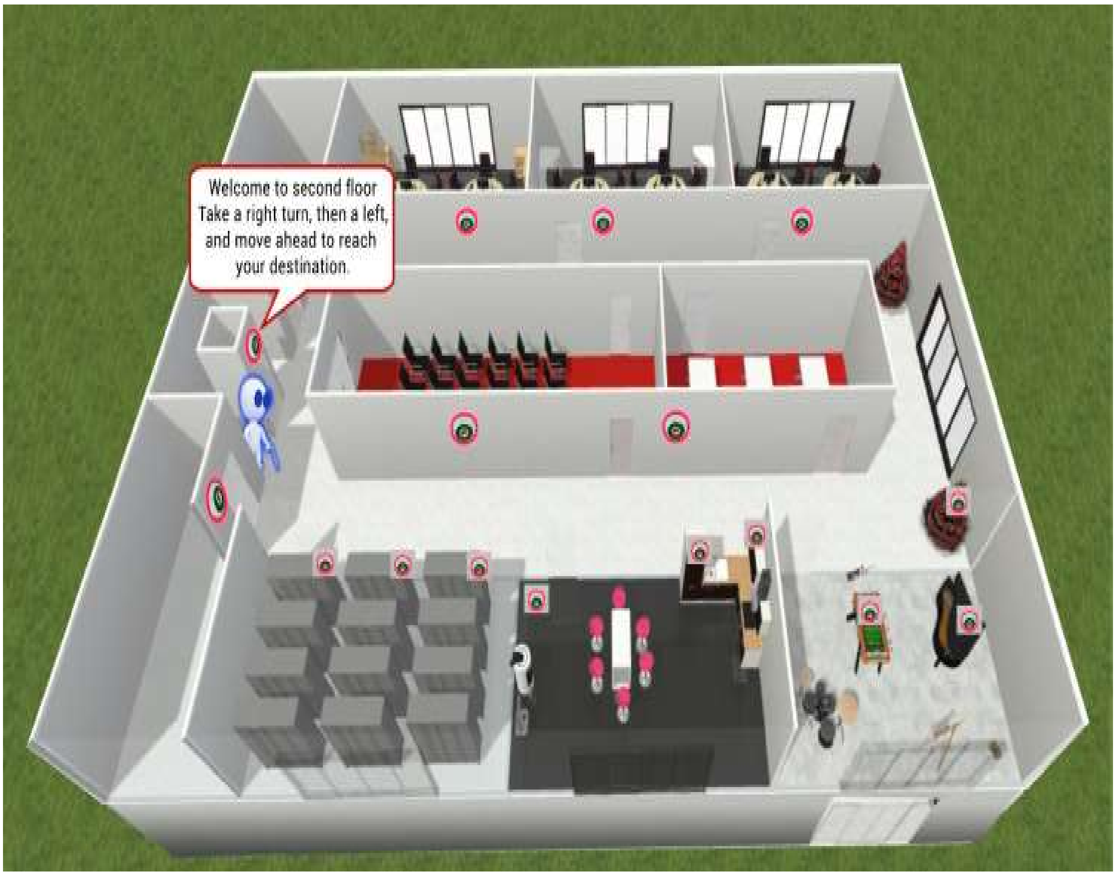


Figure 2.5: Global obstacle avoidance in the pathfinding algorithm

The pathfinding process consists of three main components: representation of an indoor environment, a heuristic function, and a pathfinding algorithm using the heuristic function to guide the indoor search [46]. The decision to represent indoor maps and obstacles mapping on the indoor floor plan is an input to a pathfinding algorithm [47], [48], [49].

2.4.1. Map representation techniques

The indoor navigation process models the indoor space onto a pathfinding graph. The graph captures locations—including rooms, foyers and corridors—as nodes, connecting links as distance, states and transitions between them [50]. The map representation techniques of an indoor environment are classified into grid graph and skeleton graph representations that can be further divided into visibility (roadmap) graph and tree graph representations, as shown in Figure 2.6. This thesis studied various graph representation techniques to determine the most suitable fit for the work.

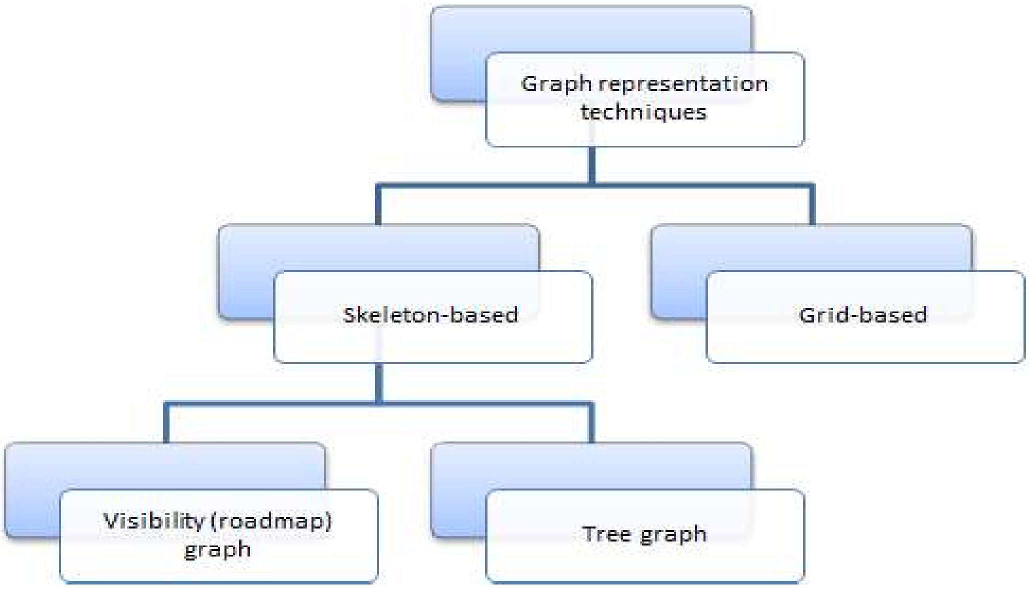


Figure 2.6: Graph representation techniques

The grid-based representation technique decomposes the indoor space into uniform cells or grids to generate grid-based graphs. The skeleton-based graph representation denotes the indoor environment's spatial structures, including rooms, stairs and corridors, as nodes with their connectivity as edges [51].

2.4.1.1. Grid-based representations

The grid-based map representation is a popular way to discretise a map into a search graph [51]. Figure 2.7 (a) shows the original plan of an example indoor environment with three obstacles. A square grid-based representation in its pictorial form is depicted in Figure 2.7(b). A grid point is marked as either traversable/walkable (0) or blocked (1), based on the topology of the map [46]. Grids with such binary representation are known as binary occupancy grids [52]. The blocked grid point shows the location of the obstacle and accessibility of a VI person through space. An indoor map represented as a grid can be assigned a number or probability value to each grid point. Different numbers code the traversable and non-traversable locations. Such occupancy grids are known as probability occupancy grids [53].

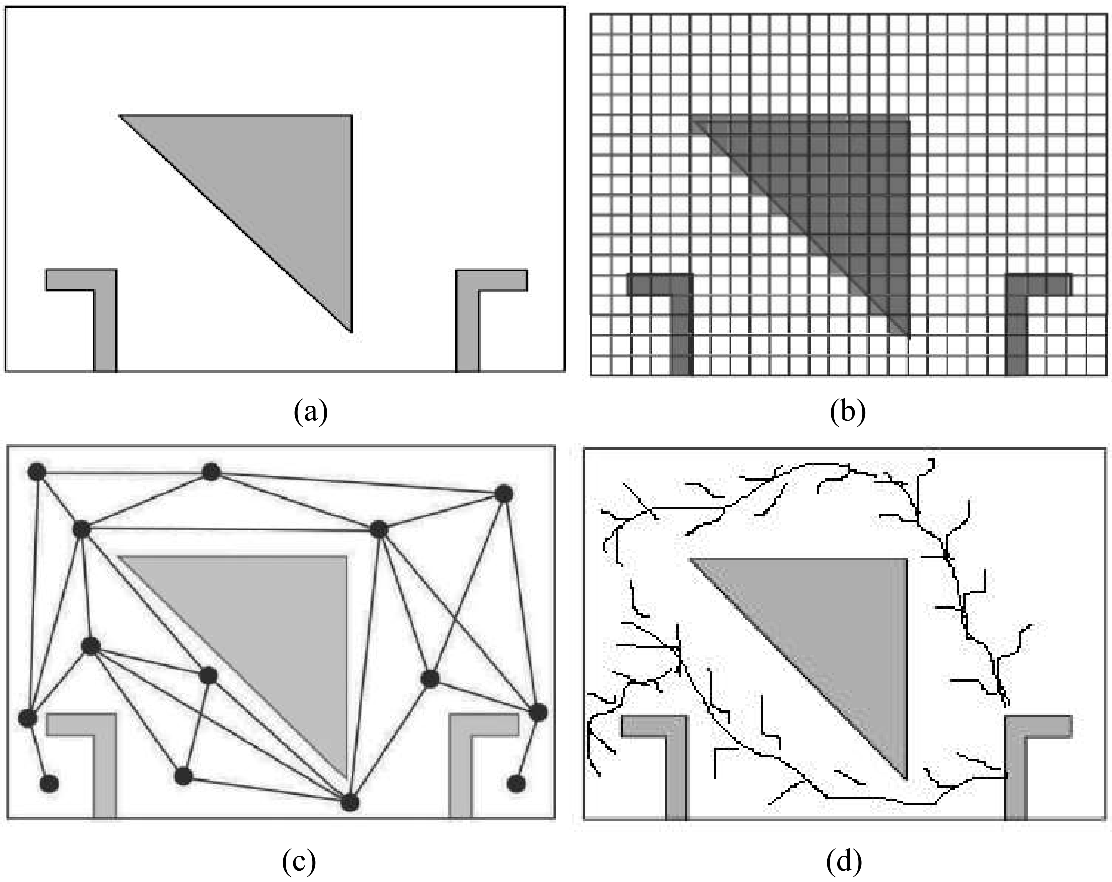


Figure 2.7: Graph representation: (a) original map, (b) grid-based graph, (c) skeleton-based visibility graph and (d) tree graph

2.4.1.2. Skeleton-based representations

The skeleton-based representation technique selects special features of the space configuration [54] and reduces the indoor space into a countable set of graph nodes. Skeleton-based approaches are further categorised as two types of irregular graphs: visibility (roadmap) graphs and tree graphs [54]. Figure 2.7 (c) and (d) show examples of the two categories—visibility and tree structure graphs of skeleton-based graphs with obstacles. In general, skeleton-based graphs generate new nodes around the corners of obstacles to avoid any blockages in the generated path.

2.4.1.3. Visibility graph representation

A visibility graph is a graph representation technique in which the nodes are visible to each other, as shown in Figure 2.7 (c) [53]. Two nodes are visible to each other if a straight-line segment joining them does not intersect any obstacle. It generates a graph with vertices representing reachable objects and edges representing the movements from one position to another. Unlike grid-based approaches, a visibility graph may not find the shortest path, as the nodes represent a coarser representation of finer grid points [51]. However, a visibility graph requires less memory to store the map representation of an indoor space [55] and reduce computation time [56].

2.4.1.4. Tree graph representation

A visibility graph must generate thousands of connections and nodes to find a path. Unlike visibility graphs, a tree representation generates randomised tree data structures requiring fewer nodes [57]. A tree representation—the rapidly exploring random tree (RRT)—is discussed in this thesis. Tree representation includes some of the same desirable properties as in roadmap representation, but this approach leads to better performance and good behaviour consistency [57]. Table 2.2 summarises the properties and performance of the path representation technique, as aforementioned. The memory-intensive property refers to memory usage for storing the representation of the map. The dynamic environment suggests the suitability of the representation technique in a space with moving objects. Optimality represents a vital feature to find the shortest path. Representation guarantees to find an optimal solution. Efficient path-smoothing defines the quality of the process required to be performed on the generated path to remove sharp turns or discontinuities in the environment to provide a realistic path. Hence, choosing an appropriate representation technique based on the application is the first factor that will affect the performance and robustness of a pathfinding algorithm.

Table 2.2: Summary of graph representation techniques

Properties	Grid representation	Roadmap representation	Tree representation
Memory-intensive	Yes	No	No
Dynamic environment	Yes	No	Yes
Optimal representation	Yes	No	No
Representation complete	Yes	No	Yes
Efficient path-smoothing	Yes	No	Yes

The graph search algorithm processes rapidly, with fewer numbers of nodes in the map representation. However, the more nodes and the closer the nodes are to each other, the better the quality of the path. Grid-based graphs preserve most of the spatial information about the dynamic environment collected by the sensors attached to objects in the environment and can provide information about the objects' locations and states. Obstacle detection and avoidance can be achieved efficiently with sensory information of the environment. Therefore, we selected grid-based representation for our simulation and experiments in the pathfinding algorithms discussed in Chapter 4.

2.4.2. Indoor pathfinding algorithms

Figure 2.8 presents the categories of pathfinding algorithms that this research discussed, implemented, and used in experiments. The algorithms are categorised as discrete pathfinding, classified as Dijkstra's and A* algorithms and sampling-based pathfinding algorithms categorised as Probabilistic Roadmap (PRM) and Rapidly exploring Random Trees (RRT). The aforementioned popular pathfinding algorithms were explored and simulated in MATLAB to check their feasibility for VI people in an indoor environment assumed to be embedded with IoT devices in a grid-based indoor map to avoid obstacles.

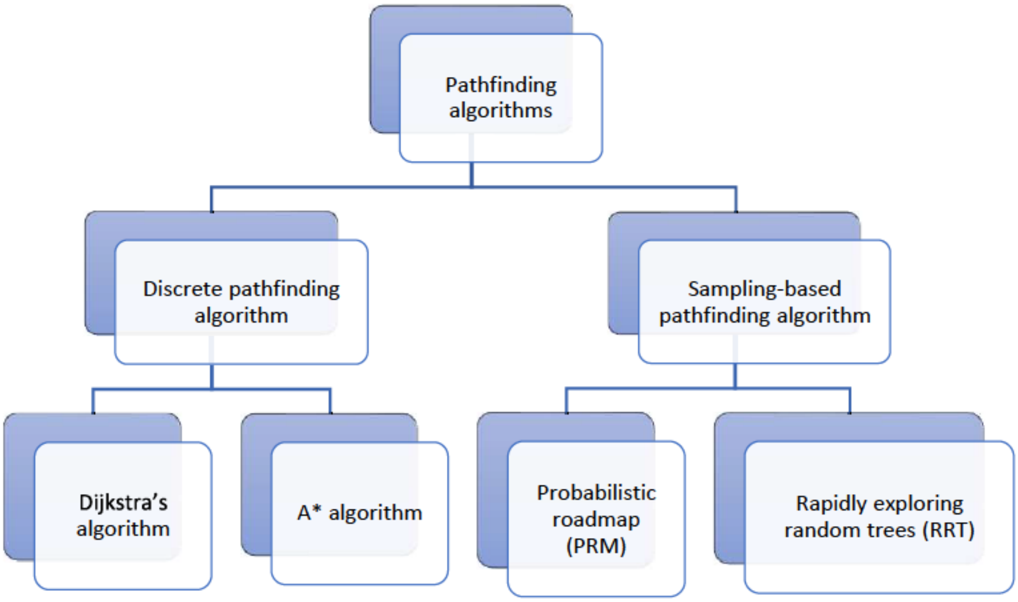


Figure 2.8: Classification of pathfinding algorithms

Aside from seeking the shortest route, VI people have specific requirements, including minimum turns, following a path close to walls, avoiding obstacles and following an optimal route that avoids significant risks. Much effort has been invested into designing a computationally efficient obstacle-free path algorithm for robots [58]; however, in the robotic environment, the robot's shape and angular movement are major constraints in planning a valid path. In our case, agents moving in the environment are VI humans, who have no such movement constraints, except the limitation of visualising the environment. Routes generated by a pathfinding algorithm must be computed in real-time, and the chosen path must be sufficiently realistic and human-like for a VI user [46]. Chapter 4 discusses the implementation and simulation of a grid-based map embedded with IoT devices to provide collision-free paths in an indoor environment for VI people to navigate.

2.5. Applicability of the Internet of Things

The recent advancements and convergence of MEMS technology, wireless communications and digital electronics have resulted in the development of miniature devices to sense, compute and communicate wirelessly over short distances. These miniature devices' interconnections can help in various applications, including environmental monitoring, infrastructure monitoring, traffic monitoring and retail, as shown in Figure 2.9. The figure presents a schematic interconnection of IoT objects, where the application domains were chosen based on the scale of the effect of the data generated.

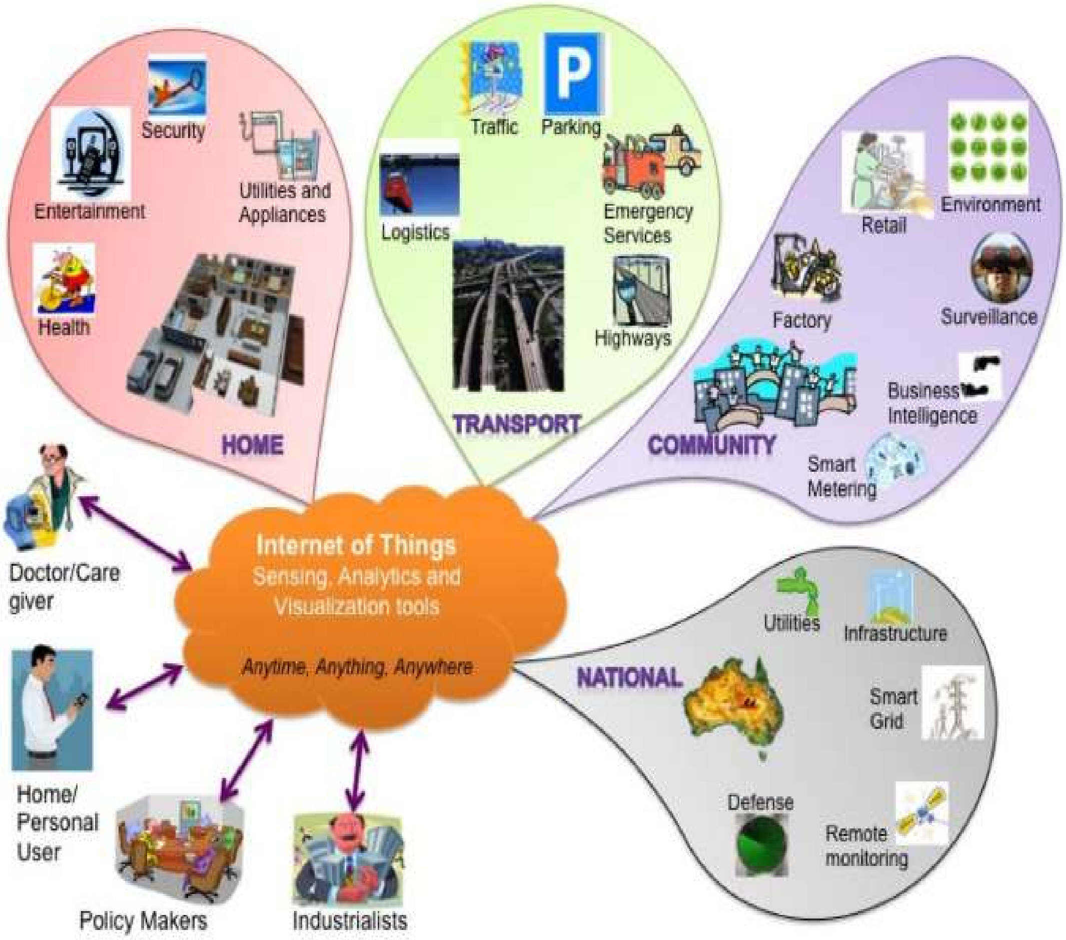


Figure 2.9: IoT schematic showing applications and end-users [59]

2.5.1. Internet of Things devices and their characteristics

The term 'Internet of Things' (IoT) defines interrelated, interconnected objects that can sense, collect and transfer data over a wireless network without human intervention [7]. The 'thing' in IoT refers to a piece of connected equipment, furniture, solar panel or connected automobile embedded with a sensing device that alerts about possible changes in events, including fuel, temperature, and fire. The IoT is an extension of Internet connectivity into physical devices and everyday objects. The physical devices are embedded with electronic devices, Internet connectivity and hardware, including sensors, actuators and RFIDs. These devices communicate and transfer data over the Internet and can be controlled and monitored remotely.

Cisco refers to the IoT as the 'Internet of Everything' (IoE), with four key components: processes and standards, things, the Internet and data—as shown in Figure 2.10. The processes and standards allow things to connect over the Internet to exchange data using industry standards that guarantee interoperability and enable automated processes. The two main requirements of IoT are sensing and addressing [60]. Sensing is essential to identify and collect critical parameters for analysis while addressing provides Internet Protocol (IP) connectivity to identify things.

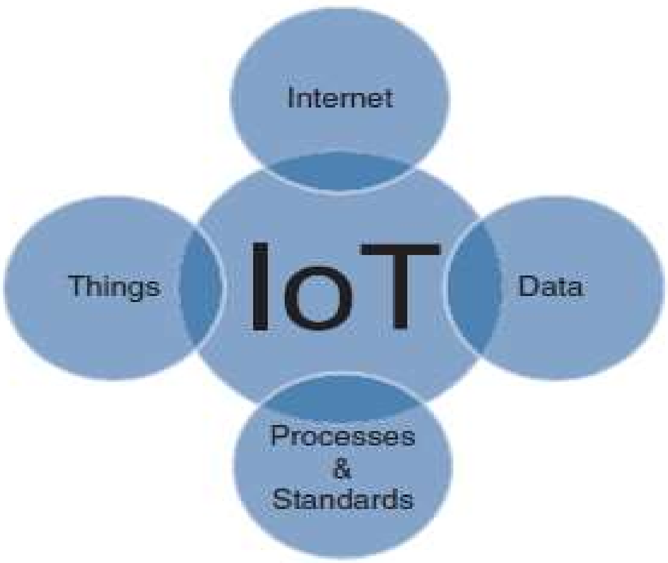


Figure 2.10: IoT—complete definition [60]

Smart objects are physical objects that contain embedded technology to sense and interact with their environments by enabling communications, including sensors and actuators. These smart objects are fundamental building blocks of IoT networks.

2.5.1.1. Characteristics of Internet of Things

The fundamental characteristics of smart objects in IoT are as follows:

- Sensing: Each object connected to an IoT network should be capable of sensing the environment and generating operational data.
- Connectivity: Smart objects participating in global information and IoT infrastructure must be interconnected to make intelligent decisions from the collected IoT data.
- Centralised: The IoT is everywhere, so different disconnected and disaggregated data streams need to be connected and brought to a central location to perform analysis.
- Intelligence: The system collects raw sensor data and converts them into a contextualised meaning. The collected raw information helps make decisions after being converted to intelligent data.
- Energy: The IoT cannot rely solely on batteries; thus, energy harvesting, power efficiency, charging and infrastructure are necessary parts of the IoT design.
- Security: Securing the endpoint devices and moving data in the IoT networks is essential to provide a complete model solution.
- Coordination: A well-designed user interface will help information technology (IT) and non-IT professionals to coordinate and conclude about the occurrences and events around them with IoT data.

This subsection discusses the primary IoT devices used in the thesis, with the general architecture of the IoT devices.

Sensors: A sensor measures some physical quantity and converts the measurement reading into a digital representation. Sensors are not limited to human-like sensory data but can measure a wide spectrum of rich and diverse measurement data with good precision. Different sensors include invasive, non-invasive, active, passive, contact, absolute, and relative [61]. The categories of sensors are shown in Figure 2.11, based on the physical phenomenon.

Sensor Types	Description	Examples
Position	A position sensor measures the position of an object; the position measurement can be either in absolute terms (absolute position sensor) or in relative terms (displacement sensor). Position sensors can be linear, angular, or multi-axis.	Potentiometer, inclinometer, proximity sensor
Occupancy and motion	Occupancy sensors detect the presence of people and animals in a surveillance area, while motion sensors detect movement of people and objects. The difference between the two is that occupancy sensors generate a signal even when a person is stationary, whereas motion sensors do not.	Electric eye, radar
Velocity and acceleration	Velocity (speed of motion) sensors may be linear or angular, indicating how fast an object moves along a straight line or how fast it rotates. Acceleration sensors measure changes in velocity.	Accelerometer, gyroscope
Force	Force sensors detect whether a physical force is applied and whether the magnitude of force is beyond a threshold.	Force gauge, viscometer, tactile sensor (touch sensor)
Pressure	Pressure sensors are related to force sensors, measuring force applied by liquids or gases. Pressure is measured in terms of force per unit area.	Barometer, Bourdon gauge, piezometer
Flow	Flow sensors detect the rate of fluid flow. They measure the volume (mass flow) or rate (flow velocity) of fluid that has passed through a system in a given period of time.	Anemometer, mass flow sensor, water meter

Figure 2.11: Categories of sensors [62]

Factors such as smaller size, form factors and decreasing cost enhance the economic and technical feasibility of having an increased density of sensors. Recently, smartphones have become commonplace IoT devices with dozens of sensors inside. Figure 2.12 shows the different types of sensors in a smartphone, including a camera, pedometer, accelerometer and gyroscope. These sensors can be embedded/installed in the infrastructure of a building to collect surrounding local data. Chapter 3 discusses the use of RFIDs to identify the dimension and location of the interior and furniture in the grid-based space in pathfinding algorithms. Chapter 4 proposes a solution using smartphones and beacons to monitor the user's movement in an indoor environment. Finally, chapter 5 discusses using inertial sensors of a smartphone to estimate the user's absolute position using deep learning techniques.

Figure 2.12: Sensors in a smartphone [61]

2.5.2. Reference model of Internet of Things

The general reference model for a navigation system based on IoT is shown in Figure 2.13. There are five layers in the reference model of the IoT, including the sensing layer, network layer, Cloud layer, real-time analyser and interaction layer.

2.5.2.1. Layer 1: infrastructure layer

The infrastructure layer includes the physical devices, sensors, tags and actuators required to sense and measure the local surroundings of the environment. These are IoT devices acting as endpoint devices to send and receive information. They may be embedded in the ground or other infrastructure to provide access to digital data. To track and monitor these physical objects, addressing and identification must occur at the lowest layer of the IoT reference model. Each object is given an identity by the sensing layer, including sensors, actuators and tags.

2.5.2.2. Layer 2: network layer

The communication technologies depend on the locations and types of IoT devices. However, wearables typically communicate via a short-range technology, such as Bluetooth, with a nearby collecting device, such as a smartphone. The device may further forward the collected data to the infrastructure. Sensors installed in urban fixtures also use a variety of communication technologies. The communication and processing in independent and standalone sensors typically use existing networks, such as Wi-Fi and Zigbee.

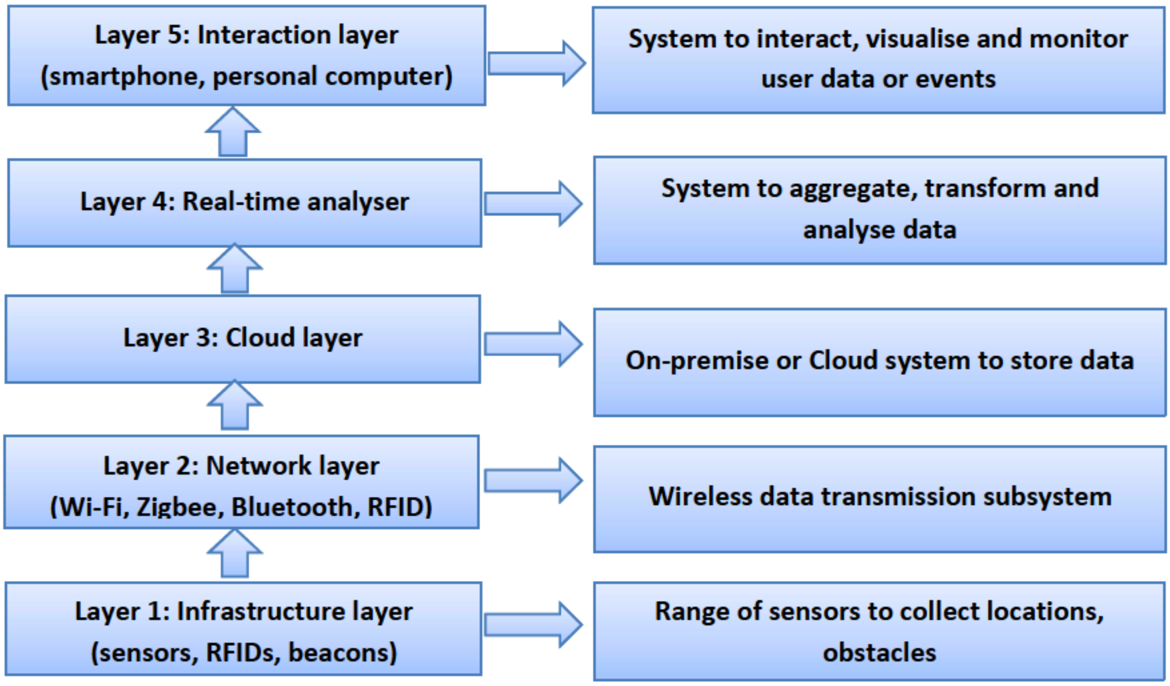


Figure 2.13: A reference model for the general system for IoT

2.5.2.3. Layer 3: Cloud layer

The data are in constant motion in an IoT application. Some applications may require data not to be processed at the network layer and be stored in memory or disk for later use. An additional layer is recommended to store data in a Cloud or on-premise database server. Further, the raw data can be converted to event-based data in the above layer.

2.5.2.4. Layer 4: real-time analyser

A real-time analyser is a layer that creates views of the collected data, as demanded in the application. It combines data from multiple sources and filters and further selects featured data to incur intelligent information to client applications.

2.5.2.5. Layer 5: interface layer

This layer is equivalent to the application layer of the transmission control/Internet protocol suite, where the outcomes of the processed data are communicated via the interface layer. A smartphone application is generally used to interact and convey data to the agent of the system.

The layer also ensures the security of the devices in the network. Sensing and computing services have become increasingly mobile and capable of recognising and adapting to dynamic environments. Technological evolution has advanced to develop tools and resources to meet the needs of disabled people. Various assistive technologies have been studied to help people with visual impairments during navigation of an unfamiliar environment [39-52]. The IoT can transform ways of accessing dynamically changing environments. However, it still faces difficulties in some grey regions, such as architectural issues and indoor navigation. Chapters 4 and 5 focus on the utilization of IoT devices in the proposed framework for indoor navigation by VI people.

2.6. Indoor wireless positioning technologies

Over the last decade, several indoor positioning and navigation techniques have developed approaches to position a user indoors. Figure 2.14 displays the traditional approach and location model of an unknown target node $P(x,y)$, with known locations of the transmitter nodes, Tx_i (x_i,y_i), in an x-y plane. R_i denotes the range of Wi-Fi signals or signals based on any other underlying technology. The signal may include noise because of interference from each transmitter node. D_i is the actual distance between the i^{th} base node and target node, $P(x,y)$.

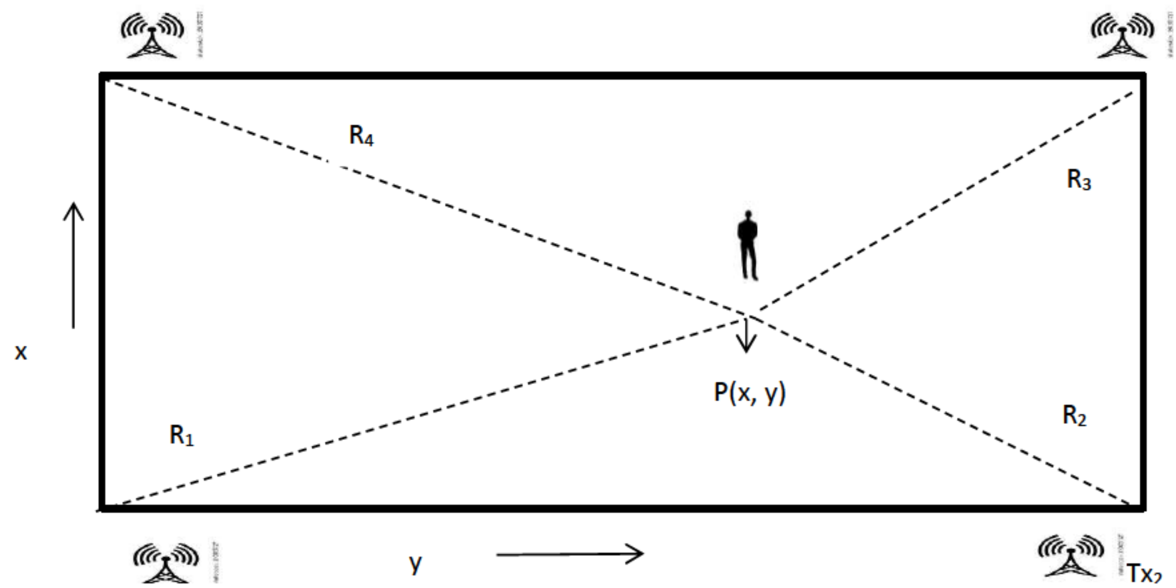


Figure 2.14: Location model of unknown node *P(x,y)*

In other techniques, including Time Difference of Arrival (TDOA), the base nodes are required to know their positions in a coplanar scenario. Unlike Time of Arrival (TOA) , TDOA requires knowing the time difference, including arrival measurements, to localise a target node. In Angle of Arrival (AOA) estimation, base nodes determine the angle of the arriving signal. To allow base stations to estimate AOA, they should be equipped with antenna arrays, and each antenna array should be equipped with RF in both front and end components. However, this incurs higher cost, complexity and power consumption [3]. Figure 2.16 shows the most prominent state-of-the-art indoor positioning technologies. The major categories are IR, radio frequencies and ultrasound. The radio frequencies are further categorised as narrow and wideband, including Wi-Fi, RFIDs, BLE-based beacons and Zigbee in narrow-band frequencies.

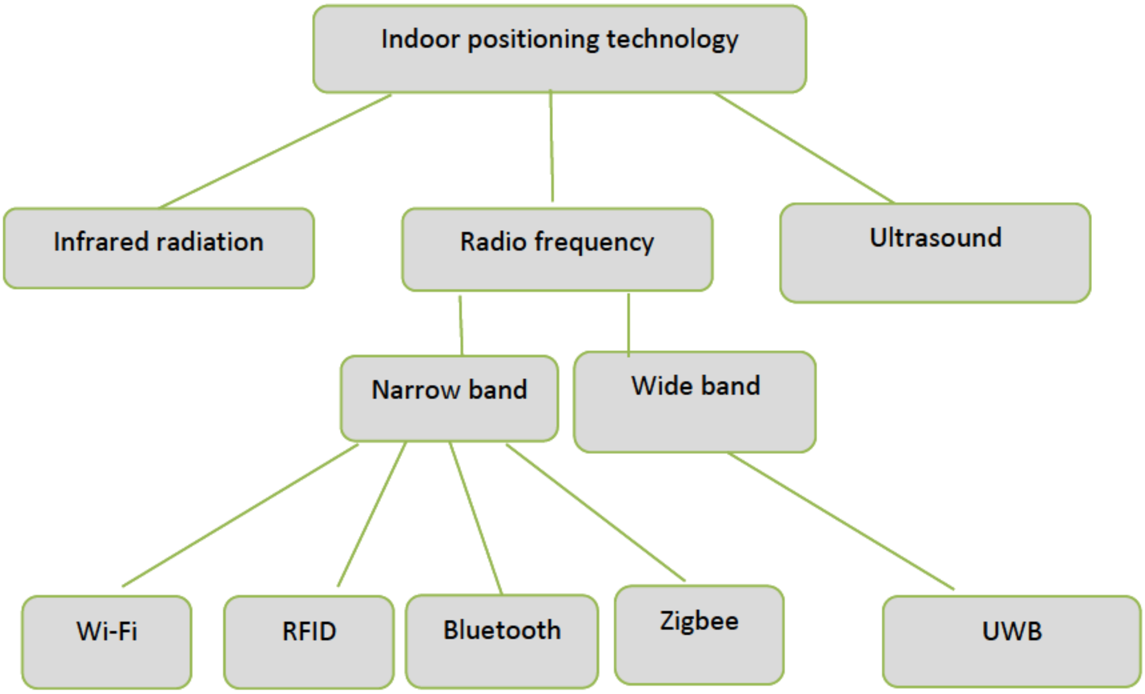


Figure 2.16: List of technologies used in indoor navigation systems

The following subsections discuss popular wireless technologies suitable for indoor navigation, with their advantages and disadvantages.

2.6.1. Infrared radiation

IR-based positioning systems are one of the most common positioning technologies available for television, mobile telephones, and personal digital assistants. An IR signal uses an invisible spectrum of light and has a longer wavelength than visible lights, and transmits in two different ways: Direct IR and Diffuse IR [4]. The advantage of using an IR-based system is that it is small, lightweight and easy to carry. However, it has some limitations for location determination, such as interference from florescent light and sunlight, maintenance costs and security issues [5]. Further, IR-based systems have a limitation of limited coverage. Proximity and AOA positioning methods are frequently used with IR-based systems [4].

2.6.2. Wi-Fi

Wi-Fi is the common name for the IEEE 802.11 standard. Wireless connectivity is more prevalent than ever in everyday life. Each wireless router broadcasts a signal that is received by devices in the area [64]. Wi-Fi-based systems have very low costs for implementation, as they are part of indoor infrastructure nowadays. Using Wi-Fi-based indoor positioning and navigation systems depends on knowing the list of wireless routers available in an area in which the system operates. RSSI-based positioning is the most popular WLAN positioning method to extract signals in Wi-Fi networks and run on off-the-shelf WLAN hardware [65]. TOA, TDOA and AOA positioning mechanisms are not common in WLAN because of the angular measurements and time delay complexity [27]. RSSI-based positioning in WLAN positioning systems achieves accuracy varying from 3 to 30 metres.

2.6.3. Radio frequency identification

RFID-based systems consist of RFID tags and readers [66]. RFID tags can be attached to infrastructure and to fixed and moving furniture in a building. The RFID reader detects the identification number stored in the RFID tag and then transmits the location information to a base station [67]. The tags consist of a microchip that can typically store up to 2 kilobytes of data and a radio antenna. Tags emit radio signals that readers receive and vice versa. Both tags and readers use predefined radio frequencies and protocols to send and receive data between them [27]. There are two types of tags: active and passive. An RFID reader consists of different components to connect to a server, including an antenna, transceiver, power supply, processor and interface. Given the dynamic nature of the data, it is impossible to write all the information at the installation time. Therefore, it is appropriate to have a tag database to address this issue [66]. Although different positioning methods can be used with RFID, proximity is the most used and senses the presence of RFID tags rather than the exact position [27]. RFID tags add speed, accuracy and efficiency and help identify, track and monitor items in a system.

2.6.4. Bluetooth

Bluetooth is a wireless communication method used by two devices over short distances. Bluetooth is the IEEE 802.15 standard with a gross low and short bit rate range of up to 10 metres. The maximum distance for Bluetooth communication is up to 100 metres for a class 1 Bluetooth set [28]. The devices can send a maximum of 3 Mb/s. Thus, implementation and deployment can be costly. Bluetooth technology uses proximity and RSS-based methods to estimate unknown distances [27]. Bluetooth beacons are hardware transmitters based BLE devices that broadcast their identifiers via Bluetooth to nearby portable devices. The BLE beacons can’t determine the physical location with pinpoint accuracy. The BLE beacons use Bluetooth smart low energy radiation to send out the signals to the device, it is essential that the mobile device switch their Bluetooth on for interaction with the BLE beacon in the range. The maximum range (at a transmission power of +4 dBm) of different beacon hardware types is as follows [68]:

- indoor beacon: 80 metres
- outdoor beacon: 80 metres
- pocket beacon: 50 metres
- keychain beacon: 25 metres
- long-range beacon: 300 metres.

2.6.5. Zigbee

Zigbee is a standard that provides network, security and application support services operating on the top of IEEE 802.15.4 [27]. Zigbee is a wireless technology standard that can be regarded as a short-distance and low-rate Wireless Personal Area Network. The signal coverage of Zigbee in an indoor environment is 20 to 30 metres—relatively less than in outdoor environments. This technology achieves positioning by coordination and communication with neighbouring nodes. Usually, RSSI values are used to estimate the distance between Zigbee nodes [65].

2.6.6. Ultra wide band

UWB signals used for positioning have received considerable attention because of their prominent feature of providing centimetre-level positioning accuracy [13]. This technology is referred to as a base-band, impulse and carrier-free technology [27]. UWB uses a low power density, wide bandwidth, which increases the reliability of the technology. The low frequency of UWB pulses enables the signal to effectively pass through obstacles, such as walls or objects [27]. The high bandwidth offers high data throughput for communication. However, UWB hardware is expensive, making it costly for wide-scale use. TOA and TDOA positioning methods have higher accuracy than other algorithms, but they require clock synchronisation [27].

2.6.7. Ultrasonic

Ultrasonic is a mechanical, oscillating sound pressure wave with a frequency higher than the upper limit of the human hearing range [27]. Ultrasonic devices can be used to detect objects and measure distances. Sound tracking is used to achieve centimetre-level accuracy. As a result of the strong decay profile of acoustic waves, the sound system is limited to a 10-metre operating range if not scaled with additional nodes [6]. This technology is associated with RF signals to fulfil the synchronisation requirement, which increases the overall cost of the system [69]. The relative distance between the devices can be estimated using TOA measurements of ultrasound pulses travelling from emitters to receivers. The emitter's coordinates can be estimated by multilateration from three or more ranges to some fixed receivers deployed at known locations [65].

Table 2.3 summarises all the above-discussed positioning enabled wireless technologies based on parameters such as range, accuracy, the requirement of LOS, interference and positioning methods.

Table 2.3: Summary of positioning enabled wireless technology

Positioning technology	Range	Accuracy	LOS	Interference	Positioning methods
Infrared	Short	1–2 m	Yes	Yes	AOA, proximity
Wi-Fi	Medium	1–5 m	No	No	Proximity, TOA, TDOA, RSSI, fingerprinting
Zigbee	Medium (20–30 m)	5–10 m	No	No	RSSI, fingerprinting
Bluetooth	Short	2–5 m	Yes	Yes	RSSI, fingerprinting
RFID	Short (passive) Medium (active)	1–2 m	No	No	RSSI, fingerprinting, TOA, proximity
UWB	Long	Cm	No LOS	No	TDOA, AOA
Ultrasonic	Long	3 cm–1 m	No LOS	No	TOA, Multilateration

No single positioning technology can provide accuracy and precision that meets the requirements of VI people. Therefore, seamless cooperation among hybrid solutions with different technologies may help provide accurate location information for a VI person to navigate a complex environment.

2.7. Background of deep learning system

The core concept of any artificial intelligence (AI) system is to perceive the local surroundings and take action based on this perception. For humans, vision is only one aspect of perception, as people perceive the world through their sight, as shown in Figure 2.19. For example, a room with an identifier room number (Administration – 231) can be seen by a sighted person who interprets and locates the room processing in the brain.

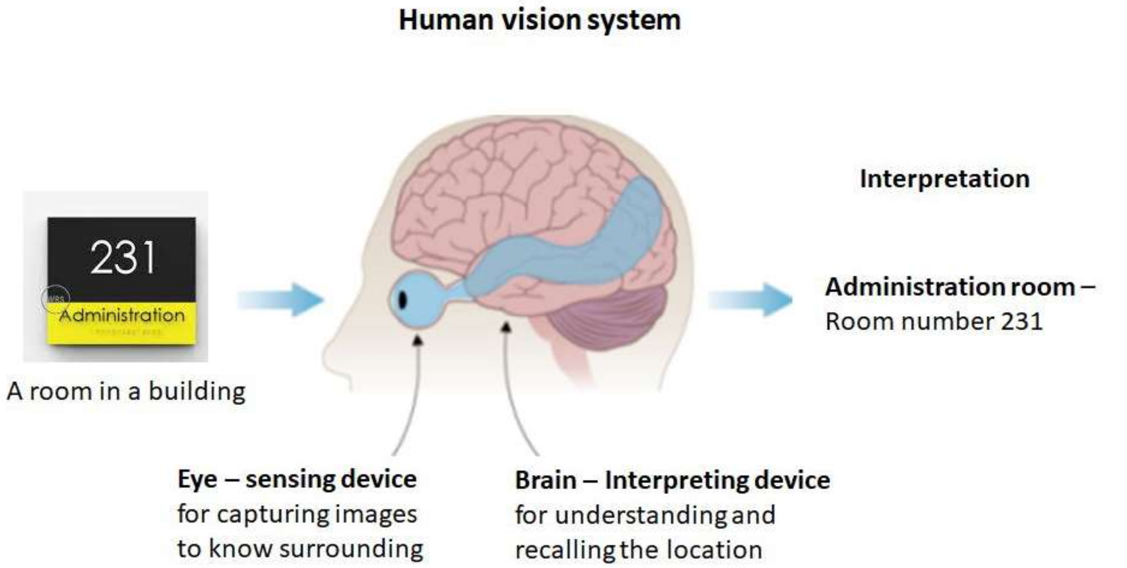


Figure 2.19: Human vision system

Figure 2.20 presents an overview of the positioning system using an AI-based deep learning system with two major components—the sensing and interpreting device. A smartphone with various sensors can act as a sensing device to mimic human eyes, while the robust deep learning algorithm mimics brain function to interpret and classify the location.

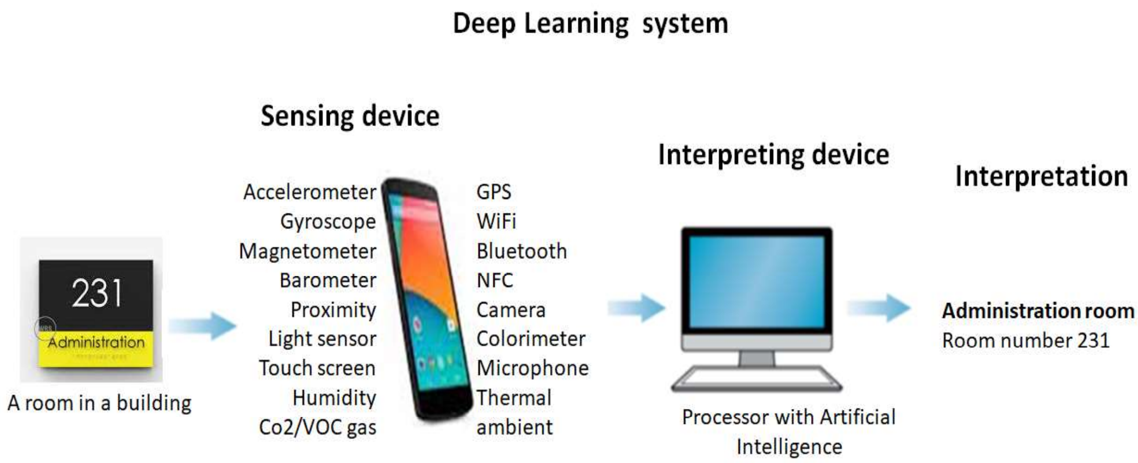


Figure 2.20: Overview of a positioning system using a deep learning technique

Deep learning is a specific subfield of machine learning—a new take on learning representations from data that emphasises learning successive layers of increasingly meaningful representations [70]. Contemporary deep neural networks (DNNs) often involve tens or even hundreds of successive layers of representations, and all have learnt automatically from exposure to training data. DNNs map the input to target mapping via a deep sequence of simple data transformations (layers), and this data transformation is learnt by exposure to training data. As shown in Figure 2.21, the learning process finds a set of random values as weights of all the layers in a network, such that the network correctly maps inputs to the associated target values. The loss function of the network uses the prediction value *Y'* and the actual target value *Y* to compute the distance score. The optimiser uses the loss score value as a feedback signal to adjust the value of the weights. A network with minimal loss is one for which the outputs are close to the target values.

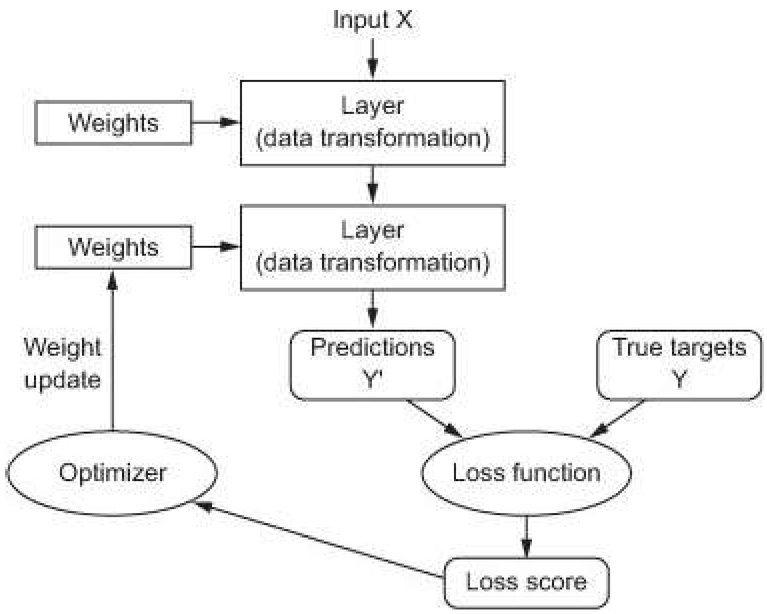


Figure 2.21: Feedback system in DNN [70]

Optimisers are algorithms or methods used to change the attributes of the neural network, such as weights and learning rate, to reduce losses. Optimisers are used to solve optimisation problems by minimising the function. Various optimisers can be used in different applications, including Adam, Adamax, RMSprop and Adaptive Gradient Algorithm (Adagrad).

Deep learning has attained a level of public attention over the history of AI to solve many industry issues. This thesis used a multilayer perceptron (MLP) to position the VI user in an indoor environment with the help of inertial sensors of the smartphone. The MLP is characterised as fully connected layers, where each perceptron relates to every other perceptron. The MLP model is a class of feedforward artificial networks that define a mapping function, as in Equation (2.1) [71]:

$$y = \psi(\sum_{i=1}^{n} \omega_i x_i + b) = \psi(w^T x + b) \quad (2.1)$$

where y is the target, w denotes the vector of weights, x is the vector of inputs, b is the bias, and Ψ is a non-linear activation function. The activation function converts the sum of input signals into an output signal. The activation function determines how the sum of the input signals activates and fires the target value. This study implemented Rectified Linear Unit (ReLU), Softplus, Exponential Linear Unit (ELU) and Scaled Exponential Linear Unit (SELU) to determine the best-suited deep learning model. The work proposed using a regression-based training algorithm to generate the MLP weights, mapping the inertial sensor data of a smartphone. In this case, the inputs of the MLP correspond to the three-axis inertial sensor measurements of a smartphone, and the output layer delivers the coordinates of a point in two-dimensional (2-D) local space, x and y. Figure 2.22 presents the MLP-based DNN with three hidden layers consisting of 128, 64 and 128 neurons.

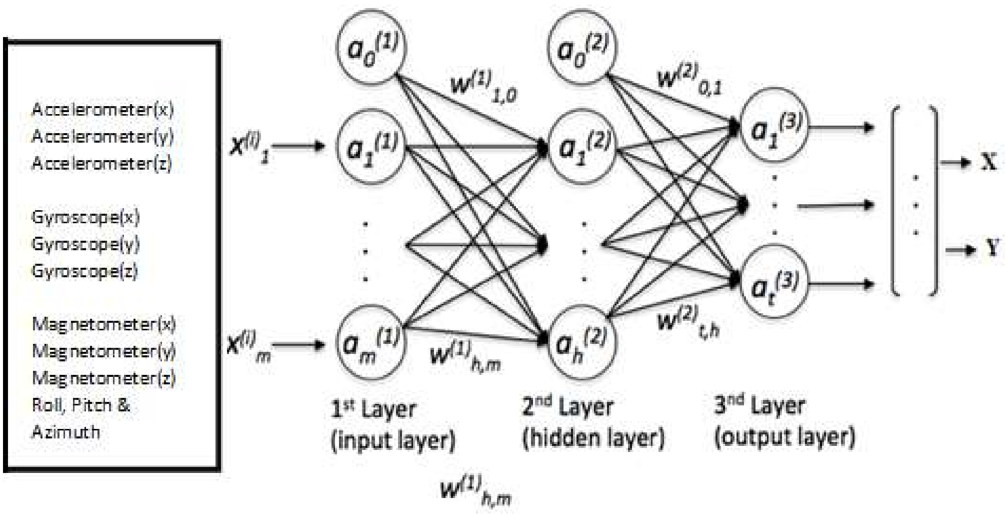


Figure 2.22: MLP network structure

An additional batch normalisation layer is introduced to perform optimisation on the input layers to mitigate the effect of unstable gradients with the given neural network. The batch normalisation layer works by performing a series of operations on the incoming input data [71]. The batch normalisation layer is adopted between the hidden layers for equal distribution among the input of hidden layers and faster convergence. The hidden layer weights were updated by a reduction in the loss function L, as expressed in Equation (2.2) using the back-propagation algorithm:

$$L = \frac{1}{m}\sum_{i=1}^{m}(yi - f(xi))^2 \tag{2.2}$$

where *m* represents the number of samples of input features, y_i represents the actual coordinates of the i^{th} sample, and $f(x_i)$ is a function to predict the position from the i^{th} sample of input features. The work is implemented in Google's TensorFlow—a popular open-source deep-learning library that uses Keras as a high-level application programming interface for its library. To train an MLP classifier using TensorFlow, the following parameters must be set: number of layers in the network, epoch (number of iterations), a block size of the learning, seed size and maximum iteration number. TensorFlow offers various tools for production deployment and maintenance, debugging and visualisation, and running models on embedded devices and browsers. Keras is ideal for the rapid implementation of deep learning models. Chapter 5 discusses the implementation and results of a DNN to position a user in an indoor environment.

The literature review of an indoor navigation system discussed in Section 2.3 did not focus on all building blocks. Therefore, the following Section 2.8 discusses and highlights the state-of-the-art of pathfinding algorithms, tracking and positioning approaches used in existing indoor navigation systems about meeting the needs of VI people.

2.8. Related works

Pathfinding algorithms are a strong pillar in the context of robot movement or game development [46], as indoor navigation systems guide users to move independently from one location to another, and the pathfinding approach computes the best path between the goal and destination locations. The best path for most users can be described as a combination of criteria, such as the fastest or shortest routes. However, a VI user may select a route following a specific landmark or a simple path with few turns. Section 2.8.1 discusses the state-of-the-art indoor pathfinding algorithms used in an indoor navigation system. The pathfinding algorithms generate a path, but the VI person faces difficulty ensuring the directions he takes and moving independently.

A monitoring and tracking technology can help a VI person follow a suggested path to reach the destination. Section 2.8.2 presents work related to indoor positioning and tracking in an indoor navigation system. Existing indoor positioning technologies for VI people can be categorised as vision-based, non-vision based and IoT device based [14]. However, their popularity differs in accuracy due to noise and interference, signal availability, energy consumption, installation cost, and high maintenance in the indoor environment [72]. Considering these challenges, the performance of indoor location-based services depends on the appropriate choice of technology and approach.

2.8.1. Related works: indoor pathfinding algorithms

A significant amount of work has been done to develop a navigation system for VI people using existing traditional pathfinding algorithms, such as Dijkstra's and A* [37], [73], [74]. Most navigation systems' objective is to provide solutions that focus on positioning a user and providing an optimal path within the environment [75]—usually the shortest path with the minimum number of turns. The author emphasises using Dijkstra's algorithm because of its optimal capability to find the shortest route [75]. The author highlights using a personalised approach of combining user preference and a smoothening path metric to reduce the jagged path generated by the basic Dijkstra's algorithm [75]. However, the computation time required to traverse the environment using Dijkstra's algorithm is relatively high.

A variation of Dijkstra's algorithm, OPTIPATH, is a greedy search variant of Dijkstra's algorithm to find an optimal path, considering the minimum number of turns and rerouting the path based on obstacles in the environment [58]. It includes a new criterion in the basic Dijkstra's algorithm to find a route with fewer angular turns between two nodes. A multilayer dictionary storage structure for Dijkstra's algorithm is proposed to find the optimal path from the start to the goal location, with the minimum turns in terms of degree [76]. The door-to-door approach is a two-level hierarchy pathfinding algorithm based on Dijkstra's algorithm [77]. The two levels are coarse and fine, where the coarse level helps determine a path between rooms, and the fine level finds a path between doors in the environment. Using a graph-based model, it returns a coarse path that ignores internal obstacles in the environment. However, the quality of the path suggested by the algorithm is unsafe for VI users, as it causes sudden changes in direction and diagonal movements.

A microscopic pathfinding algorithm is proposed to avoid static and moving obstacles for VI people [78]. The algorithm considers five optimisation parameters—length, obstacles, landmarks, directions and intersections. The path suggested ensures a VI person follows a path close to walls, even in a narrow passage connecting two walls. SUGAR is a microscopic pathfinding algorithm and uses the A* algorithm on a cell-based region in an indoor environment [11]. However, the computation complexity of the algorithm increases because of the division of the region into cells. Goyal proposed a pathfinding algorithm based on image processing that avoids local obstacles captured by the camera [79].

The literature suggests that the pathfinding algorithm primarily focuses on determining the shortest optimal path. However, a VI person may not mind travelling an additional few metres to attain a safe route that avoids obstacles and blockages. Much less effort has been invested in improving the quality of the path by considering the safety of users. Frequent changes in direction and sudden diagonal movements must be avoided to provide a safe route. Thus, aside from a short route, there is a primary need to provide a safe route to VI people that avoids local and global obstacles.

2.8.2. Related works: indoor tracking and positioning approaches

Conventional studies on pedestrian inertial-based systems focus on step detection using accelerometer sensors. Most of the movement tracking algorithms are based on peak detection, thresholds and spectral analysis [80]. However, many other approaches have been proposed and evaluated based on complexity, computational overhead and real-time usage [81]. Peak detection uses the periodic characteristics of the repetitive motion of a moving user. A peak detection algorithm extracts the local peak of a step from the normalised measure of an accelerometer sensor [82]. A handheld device may predict fake signals of human motion in some solutions, which leads to false step counts [83]. Peak detection algorithms are low in complexity, yet are limited to specific environments, step modes and device poses [83].

Spectral analysis-based step detection approaches use the periodic characteristics of the normalised acceleration values in the time domain by employing transformation. These techniques were not favourable for the current study's application, as the method requires large computational loads [80]. Literature review shows a threshold-based approach using gyroscope measurements for detecting the stance of a user's steps [84]. The stance posture while walking can be detected with gyroscope measurements that fall in the threshold range. These approaches propose using a single measurements, unlike few other approach using multiple values from both the accelerometer and gyroscope measurements [85]. The approach checks the multiple threshold values, and a valid step is detected if all thresholds are in the specified range.

However, threshold-based detection methods suffer from poor performance in different walking styles of users.

For a VI person, the positioning error needs to be within a few centimetres to locate a user in the correct room within a building. The system should also be able to estimate and update the location of the moving user rapidly. The literature review of such work focuses on predicting accuracy, positioning error and using technologies with minimal resources. Vision-based positioning technologies require the receivers and the moving object or person to be in the LOS to estimate the position measurements [86]. This category involves a vision-based camera and infrared ultrasonic system [87]. Guerrero suggested using a micro-navigation system with an infrared camera, Wiimote—an augmented white cane to detect the user's position and movement [32]. However, the system requires massive resources and a high computation operation to evaluate the user's position.

Non-vision-based positioning technology includes narrow and wideband wireless RF and magnetic field-based technologies [86]. Indoor positioning has been attempted using Wi-Fi, infrared, RFID, ultrasound, Bluetooth or a combination of technologies [88], [89]. A radial-based network, including infrared and RFID, has acceptable localisation error. However, it suffers from high cost, requiring additional hardware and offensive calibration processes [87]. Ultrasound waves are used to estimate and track the position of a user in ultrasound-based systems. However, blockage of the LOS might result in incorrect measurements [90]. SUGAR [11] uses multiple UWB tags that achieve a suitable localisation error of up to 38 cm for VI people. However, installing the UWB system is expensive, and the positioning is purely based on a UWB tag. Nakajima proposed using VLC and geomagnetic sensors to position and localise the user in an indoor environment [31]. The system provides localisation errors of up to one to two metres, which is insufficient for VI people.

Several attempts have been made to develop indoor navigation systems; however, not many have been successfully deployed. NavCog is a smartphone-based turn-by-turn navigation system for blind users using a network of BLE beacons with an approach of KNN algorithm [39]. The system achieves precise localisation; however, the solution must reroute the path when the user misses the turns. LowViz is the latest mobile application to assist VI people in indoor navigation [40]. The system uses a wide range of technologies, including sensors, Wi-Fi and BLE beacons, to guarantee low localisation error. However, context-aware real-time pathfinding is yet to be included in the system, and the app may fail when the signals from external devices fail. Thus, while various newly developed technologies have been generated and tested, the designs still suffer from limitations in localisation error, hardware cost, availability and lack of additivity.

Recently, there has been a considerable new interest in indoor localisation techniques, driven by the proliferation of smartphones and other mobile devices. Traditional approaches, such as Wi-Fi-based fingerprinting or distance-based methods, have low prediction accuracy because of shallow learning [91]. To manage the shallow learning problem, DNNs are implemented for self-extraction of appropriate low- and high-level features of the given raw data [92], [93], [94]. DNN approaches have shown good performance against signal fluctuations, noise effects and time-consuming manual tuning [95]. The deep networks dynamically learn from the environment by mapping noisy and complex input data to the corresponding output [71]. However, to the best of the author's knowledge, insufficient work has been done to provide deep learning-based positioning for VI people. Given the limitations of positioning systems for VI people, this thesis indirectly reviewed general positioning techniques using Wi-Fi, inertial sensors of smartphones and channel state information (CSI).

A novel indoor classification approach is proposed with Wi-Fi fingerprints to predict the correct floor and locations using a DNN [91]. The positioning system based on deep learning uses heterogeneous network data, including Wi-Fi and cellular networks with recurrent neural network (RNN) algorithms, with a high average error of 9.19 metres [96]. The positioning error is approximately nine metres—unsuitable for a low-vision person. Another study applied an RNN-based indoor positioning solution [97] to RSS data to exploit the sequential correlation of RSS data. The work achieved an average localisation error of 0.75 metres, with 80% of the errors below one metre. The positioning system based on the integration of two techniques, namely linear discriminate analysis and RSS based MLP provides 99.15% prediction accuracy and 0.98-metre positioning error [95]. RSS-based approaches have high variability at a fixed position each time. In addition, RSS-based localisation systems have coarse information because of multipath channels from different antennas. RSS-based approaches usually have one to three metres of localisation error, which is difficult to improve further [98].

A localisation technique based on CSI fingerprints collected using a single access point is proposed that uses a Principal Component Analysis (PCA) feature extraction technique [99]. The technique provides different positioning errors in different rooms varying from between 0.6 and 1.08 metres [99]. The work compared two positioning methods, including MLP and convolutional networks implemented on RSS and CSI data [93]. The observation shows that RSSI data could achieve an average 0.92-metre localisation error, with the highest error of nine metres. However, the results with CSI data achieved a 0.92-metre positioning error, with maximal localisation of 1.92 metres. Besides Wi-Fi network information, the magnetic field signals captured from the magnetometer are similar to the earth's non-constant magnetic field [100], [101]. In addition, each building has a unique magnetic field with some local anomalies. Thus, the static magnetic field can be used in indoor localisation and navigation systems [34], [36]. A recurrent DNN approach applied on magnetic signals in an indoor environment achieves a localisation error of 1.062 metres, compared with an average error of 3.14 metres with BLE fingerprinting results [102].

Despite extensive research to improve the algorithms and technologies discussed in earlier sections, some significant issues remain unresolved related to the accuracy, infrastructure selection and computational complexity. Many indoor solutions focus on using high computing devices. Most positioning systems have focused on solving the underlying issue as a classification problem using Wi-Fi signals by providing room-specific information. This thesis aimed to mitigate the infrastructure dependency and proposed positioning a VI person using a commonly carried and convenient device—a smartphone.

2.9. Issues and gaps in existing indoor navigation systems

Most navigation solutions are geared to the needs of people with good eyesight. The systems designed for low-vision people have limitations in their functionality or features. The following are the limitations of positioning and navigation systems in indoor environments:

- **Attenuation of signals and no LOS:** A high attenuation level exists in radio signals because of various building materials, such as plywood, glass, iron, roofing tiles and bricks [7]. Lack of knowledge of a propagation model and no direct LOS might fail to provide an exact measure of the wireless device in an indoor environment.
- **Avoiding local and global obstacles in the path:** Indoor navigation is challenging for VI people, as they must know about and avoid approaching obstacles. Therefore, indoor navigation systems for VI people must consider avoiding local and global obstacles, as discussed in Section 2.4.
- **VI-friendly paths:** People with low vision prefer to walk along a straight route, avoiding obstacles, yet most pathfinding and generation approaches focus on the

shortest route. There is a demand for pathfinding algorithms that provide the appropriate information with straight-line pathways and paths along walls. Designated path generation tools help people with a reduced peripheral field find their destination.

- **Indoor position accuracy:** Indoor positioning and navigation developments use various algorithms and technologies, including IR, ultrasound, magnetic and vision-based [7]. All approaches have technical shortfalls in coverage range and accuracy. For VI users to navigate indoors, centimetre-level accuracy is required to facilitate a small framework of indoor spaces to differentiate events such as two adjacent entrances of stores and detecting different floor levels. Thus, there is a high demand to improve the relative accuracy of navigation systems indoors.
- **Mapping and deployment setup:** Choosing an efficient mapping and representation technique is essential for developing an effective navigation system for VI users [8]. In addition, the deployment setup should take care of IoT devices with their configurations and periodic updates and storage in the cloud.

2.10. Summary

This chapter has discussed the need for an indoor navigation system and highlighted the issues and gaps in existing systems for VI people. This chapter has also discussed the limitations of existing indoor navigation systems developed for VI people. Further, the chapter introduced IoT devices' characteristics and applicability in indoor navigation systems. The chapter derived four major components of indoor navigation systems from an intensive literature review of existing systems and understanding the gaps for VI people. The components include map representation, collision-free pathfinding algorithms, tracking and positioning approaches, and user interfaces for a VI person in indoor environments.

Moreover, the chapter provided the background of indoor wireless positioning technologies, including their coverage and accuracy. It introduced various wireless positioning technologies, discussing the expected range, accuracy, requirement for LOS, interference and positioning methods. The chapter discussed many positioning approaches that have used wireless signals to estimate the user's absolute position in an indoor environment. The work in the thesis adapted the deep learning technique to predict the user's position using the measurement of inertial signals. Further, this chapter introduced the concept of deep learning and its applicability in an indoor navigation system. The chapter studied the literature solutions for positioning and tracking an indoor user and discussed their gaps in providing a complete solution for independent navigation. Apart from positioning and tracking, a map representation of the indoor environment and providing a collision-free path for VI people are mandatory. The chapter discussed the state-of-the-art of map representation techniques and popular pathfinding algorithms. Based on the highlighted research in this chapter, the next chapter proposes a novel framework to solve the challenges discussed and faced by VI people. The following chapters propose and evaluate approaches to remove the existing gaps and provide a robust framework for VI people to navigate indoors.

Chapter 3

Indoor-Nav: Novel Framework for Visually Impaired Navigation

This chapter introduces Indoor-Nav—a novel framework to provide an end-to-end solution for VI people to travel independently in an indoor environment, using IoT and deep learning techniques. The chapter discusses and highlights the layers, interaction between devices and components of the proposed framework.

3.1. Overview

The field of indoor navigation for VI people has been studied extensively; however, developing technology to provide a reliable and accurate solution to meet the needs of VI people remains a significant challenge. This chapter presents a general overview of a practical solution that is reliable, cost-effective and accurate in guiding VI people to navigate independently in an unknown environment. Section 3.2 introduces the significant problems and challenges that the thesis aimed to resolve using the proposed framework. Section 3.3 presents the layers and components of the framework, with deployment details. Section 3.4 summarises the chapter.

3.2. Introduction

Vision loss significantly affects the lives of VI people in most activities that are essential for movement and independence—especially navigation. Research shows that VI people spend 80 to 90% of their time inside a building or indoor environment. Thus, given that vision is an essential human sense, VI people face unique challenges in navigating an unknown indoor environment, such as finding their desired goal location and determining correct paths in an unknown indoor environment. Orientation and mobility training programs help VI people learn safe, efficient and effective navigation with appropriate travel skills. However, there remain issues that make VI people dependent on an individual's guidance while travelling in an unknown building.

Many assistive devices for VI people have been studied and reviewed in the literature. However, most remain in an active development stage or do not provide a complete solution to resolve all the issues faced by VI people. In addition, some do not meet user needs and expectations, as they are too large for a human to hold, uncomfortable to wear, or complicated and costly, as discussed in chapter 2. Existing research from the literature review indicates that VI people require detailed information about the environment in the range of 5 to 20 metres to navigate confidently. Further, they require information about obstacles, specific landmarks, self-positioning and safe paths to reach the desired location.

Table 3.1 summarises the significant problems that VI people experience when navigating an unfamiliar building. The table presents problems, challenges and solutions with technologies based on the literature review.

Table 3.1: Problems and challenges of VI people, with applicable solutions

	Problems and challenges	Solution
1	Information on commercial maps is limited and insufficient for a VI person to understand	Spatial maps with details of interiors
2	Difficulty detecting and avoiding fixed and global obstacles	RFIDs, proximity beacon, camera
3	Commercial tools provide shorter paths with multiple turns and are not VI-friendly	Reliable pathfinding algorithm
4	No commercial tools have detailed information about landmarks and safe routes, considering the needs of a VI person	Proximity sensors/beacons
5	Difficulty in knowing orientation and relative position when in motion	Fusion algorithm for tracking user
6	No appropriate tool to accurately self-position a VI person	Accurate and reliable positioning technique (cm)
7	Difficulty interacting and navigating in a crowded place with lack of accessibility information	A handheld assistive device with audio interaction

Challenge#1: People with vision loss usually keep the setup of their home or work environments constant to help them locate items and avoid barriers while moving. However, in a public building, the only way to know the environment is through using maps. Usually, maps are provided as a kiosk or a visual board that guides sighted people to navigate a complex building. However, these commercial maps do not provide detailed context-aware information readable for a VI person [4]. Therefore, as mentioned in Table 3.1 (#1), dynamic changes in complex buildings make navigation tasks difficult for VI people. Thus, maps must include spatial information of fixed, moving and temporary obstacle position changes in the indoor space for VI people. A possible solution to the limited information on commercial maps is to provide spatial maps with details of interiors and changes in the environment.

Challenge#2: To detect and avoid obstacles in the indoor environment is crucial for VI people, as discussed in Table 3.1 (#2). The problems are discussed much in chapter 4 more in detail, including illustrations of global and local obstacles. Each object in the indoor space needs to be embedded with RFIDs to recognise the interiors and track their location and help VI people avoid the obstacles when travelling indoors. The framework proposes using Bluetooth beacons installed at each door's entry/exit, escalators, and lifts as landmark identification. These IoT devices help VI people to differentiate between two adjacent doors and navigate independently.

Challenge#3: Further, a noticeable difficulty for VI people discussed in Table 3.1(item #3), as observed in the literature review and studies, is that most commercial tools provide a shorter route than a safe path for pedestrians to move. The paths provided by most navigation systems consider the shortest route rather than the safest route along walls and with minimum turns. Therefore, an appropriate and reliable pathfinding algorithm is proposed to meet the safety requirement of VI people. This work focuses on providing a VI-friendly path considering the requirements and safety of VI people. The work requires a unique indoor building map, i.e. occupancy grid maps (OGMs). Therefore, there is a need for a reliable and VI-friendly pathfinding algorithm that helps VI people with obstacle-free and safe paths with minimal turns to travel safely on the way, as discussed further in Chapter 4.

Challenge#4: Identification of entry/exit or doors is required for a VI person to navigate indoors independently. Therefore, the work proposes implementing the beacon/proximity sensors on specific landmarks, including doors, entry/exit, escalator, and lift. That helps the VI people differentiate landmarks, defined as an indoor POI and thereby provide safe routes as mentioned in Table (item #4).

Challenge#5,6: In a complex building, self-positioning and orientation recognition are significant challenges for a VI person, as discussed in table 3.1 (item #5, #6). The work proposes a fusion tracking algorithm in Chapter 5 that helps a VI person with rerouted routes knowing their orientation and position when they get lost. In addition, to overcome problems involved with lack of accessibility, a complimentary, independent Android application is proposed in Chapter 6 to help a VI person self-learn his/her position holding a smartphone using deep learning techniques.

Challenge 7: Vision loss causes difficulty accessing the physical world (item #7). Thus, an effective means of communication and interface for a VI person would include speech and audio interactions using handheld devices. Smartphone-based solutions are most likely to be accepted because of their comfort and lightweight by VI people. Moreover, the latest smartphone comprises various sensors, including an accelerometer, gyroscope, magnetometer and more. The inbuilt voice recognition and audio interactions fulfil the need of VI people. Therefore, the proposed framework suggests a reliable and accurate positioning fusion approach using beacon and smartphone sensor signals to position and track an indoor user.

3.3. Proposed framework: Indoor-Nav

This section introduces the Indoor-Nav framework that this thesis proposes to provide users with the capabilities to manage their indoor journey without direct human assistance. The framework provides a communication platform enabling IoT devices to support and sense the indoor space over the Internet. In addition, it offers users safe paths, tracking and self-positioning facilities. The design of the framework considers the fact that IoT devices, including smartphones, have limited power consumption. Figure 3.1 presents the overall technology layers and communication between layers for the framework and its components to perform.

The framework includes four layers: infrastructure, processing, communication and interaction layers. The lowest layer acts as the sensing layer to collect information about the indoor space to generate a spatial map with information on obstacles and landmarks, using beacon and RFID devices. The updated spatial map or grid-based indoor map is converted to OGMs and provided to the above processing layer, consisting of storage and processing devices, such as Raspberry Pi, a mini processor and a Cloud-based server. This layer calculates safe paths, estimates the user's position and helps track the user. The communication layer includes wireless technologies for the device and applications to send/receive data to and fro at a periodic interval. Finally, the interaction layer comprises an Android smartphone-based application that interacts with the user to instruct and guide VI people while navigating indoors.

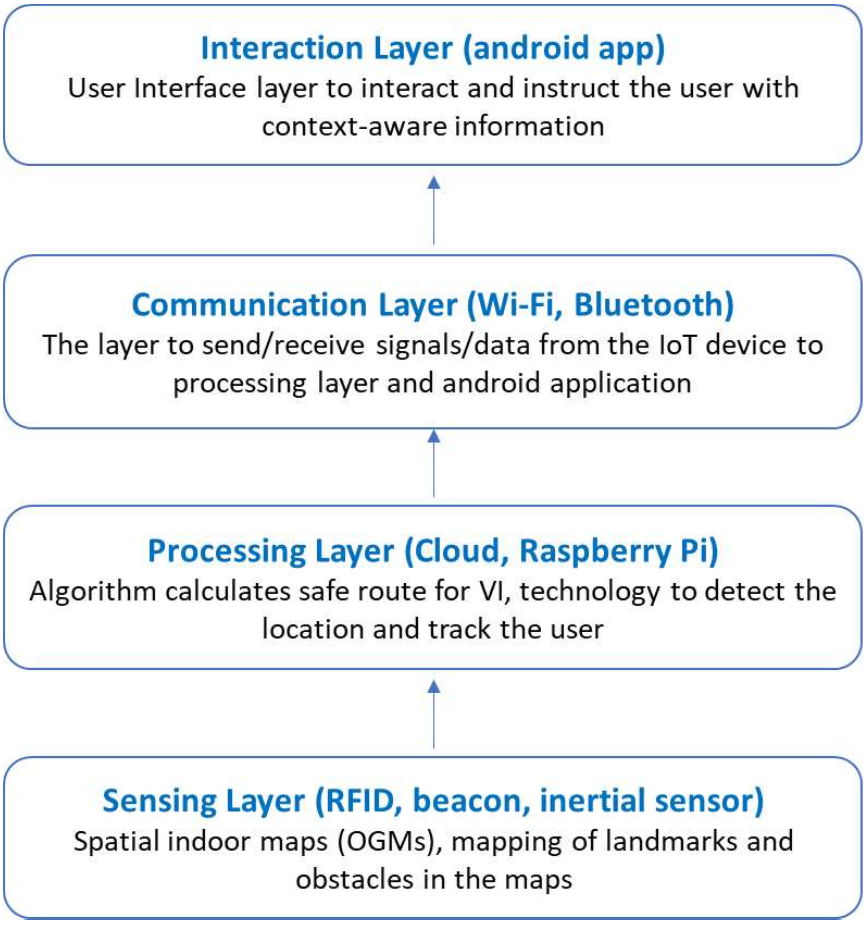


Figure 3.1: Overall layered structure of Indoor-Nav framework

Each of the layers have their own significance. The lowermost sensing layer helps to sense the environment using IoT devices and builts the spatial map including information related to the landmarks and obstacles in the indoor space. Communication layer sends the sensed information to the VI person via mobile device. An android app is the interaction layer that assist the VI person with with voice to know and accordingly act to move in the given indoor space.

Figure 3.2 presents the distributed architecture of devices in the framework. The architecture consists of IoT devices, including RFIDs embedded in each obstacle, including fixed and rearrangeable furniture, and iBeacons installed on corridors, entry/exit doors and escalators. These sensors help identify the proximity information of the barriers or landmarks in the indoor space. The signals of inertial sensors, including the accelerometer, gyroscope and magnetometer of a smartphone, are collected to sense the trajectories of the moving user [103], [104]. Inertial sensors help track indoor users in real-time and locate the user's relative position. A gyroscope measures the rate of change of the sensor's orientation—that is, the sensor's angular velocity. An accelerometer measures the change in gravitational acceleration in an embedded device. A magnetometer measures the local magnetic field, consisting of both the earth's magnetic field and the magnetic field due to magnetic material surrounding the sensor.

The raw data of the sensors is collected via a Raspberry Pi RFID reader. Further, they are processed in the Cloud using the deep learning technique to predict the absolute positioning of the user. The VI user (i.e., the agent) interacts with the system using a smartphone-based Android app and receives audio instructions to navigate indoors.

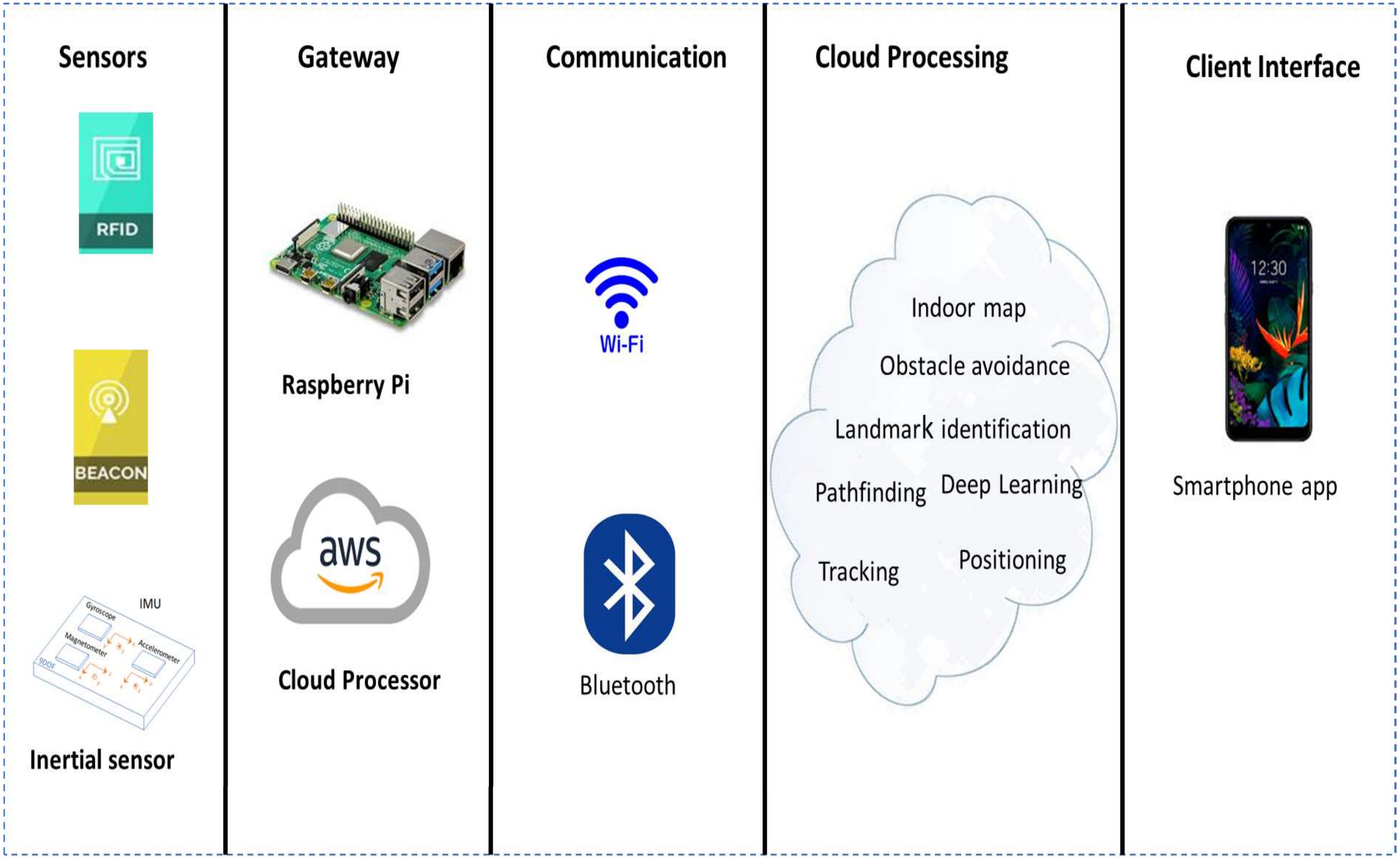


Figure 3.2: IoT devices connected with Cloud in Indoor-Nav

With recent advancements in wireless protocols, sensors and other IoT devices can be embedded in building infrastructure. In addition, device connectivity is improved with Cloud services, such as Amazon Web Services and Google. The work considers the deploymnent of the DNN algorithms on cloud based platforms. Therefore, the connectivity to Internet and cloud becomes bottleneck for the framework to be functional. Smartphones and wearables, including smartwatches, are common equipment used for human interfacing and are considered end devices. The proposed framework may not be functional in absence of any of the components. In Future, we may propose the solution limiting the infrastructure and dependencies. The framework focuses on connecting VI people and events and help people self-locate and re-route if they become lost in a complex building.

3.3.1. Architecture of Indoor-Nav

The Indoor-Nav architecture has three major components based on the problems identified by the thesis. The components of the framework include initial map setup, navigation guide (path estimation) and navigation guide (tracking and positioning), as shown in Figure 3.3. This section provides a brief introduction to the components of the architecture. In addition, the components, along with experiments and simulations, are discussed in more detail in later chapters. The initial map setup detailed presented in the following subsection 3.3.1.1 that takes the OGMs as input, and updates them from the RFID information embedded on each object in the indoor space [105], [106]. Further, subsection 3.3.1.2 discusses the path estimation component that takes the updated OGMs as input to generate a VI-friendly and safe path. The third component of the architecture, an Android application that helps a VI person track and self-position using fusion algorithm and deep learning techniques, is detailed in subsection 3.3.1.3.

3.3.1.1. Initial indoor map setup

Indoor maps, with their unique properties, such as fixed barriers (including walls, corridors and interiors), can be converted into OGMs using robotic sensors with laser and range sensors [107], [108], [109]. The conversion of sensor data to OGMs is not part of this thesis—this thesis assumes that OGMs are provided to the framework. However, this section briefly introduces the concepts of OGMs and this study's related assumptions. OGMs present the indoor space occupancy representation primarily for robot navigation and discretise the indoor environment into grids or fixed-size cells [109]. The structure of the distributed space using OGMs gives the discretisation of the environment into grids. Each i^{th} grid cell or pixel m_i in the space is either occupied or free. For example, figure 3.4 represents the OGM of a 4×4 grid space.

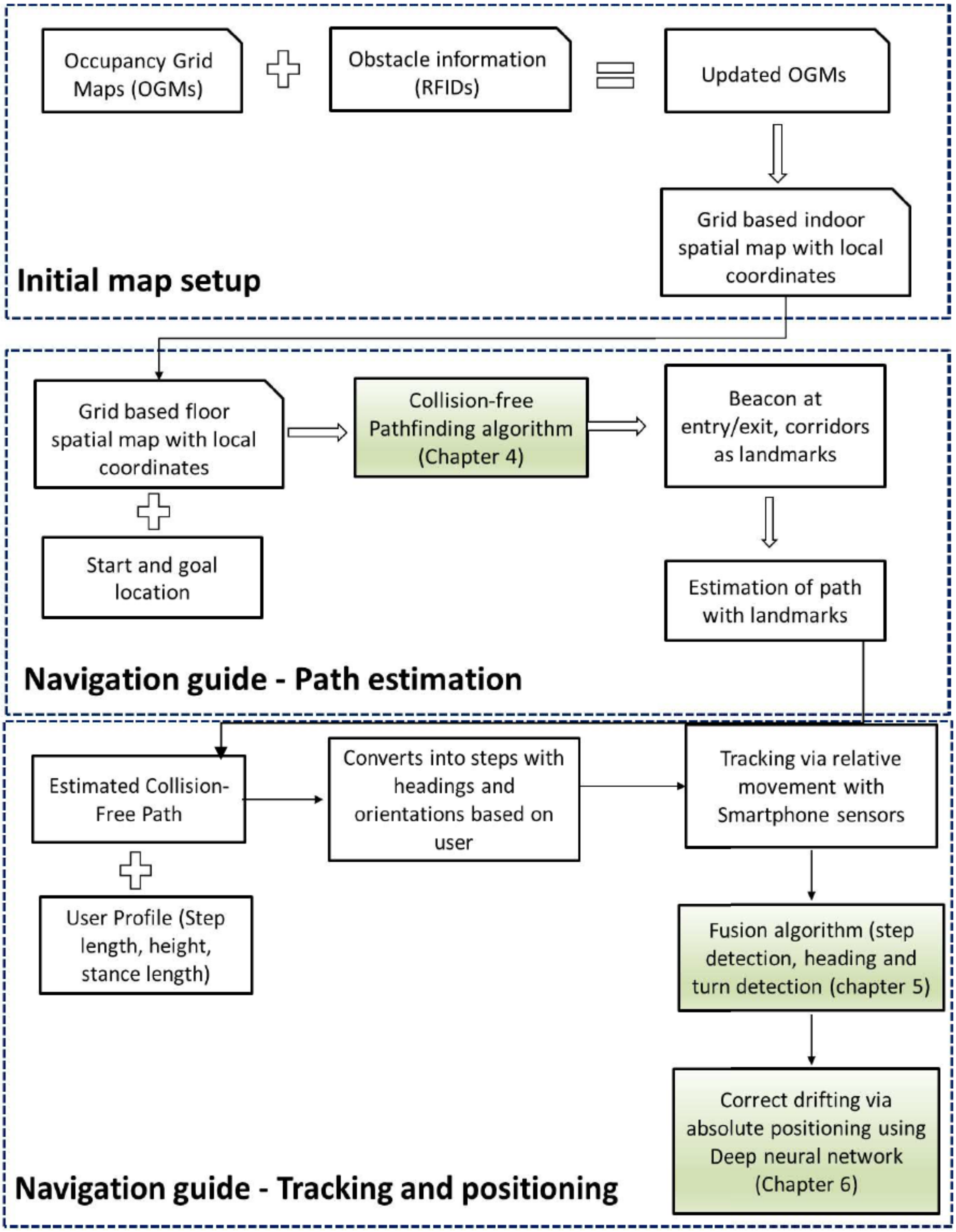


Figure 3.3: Components of Indoor-Nav framework

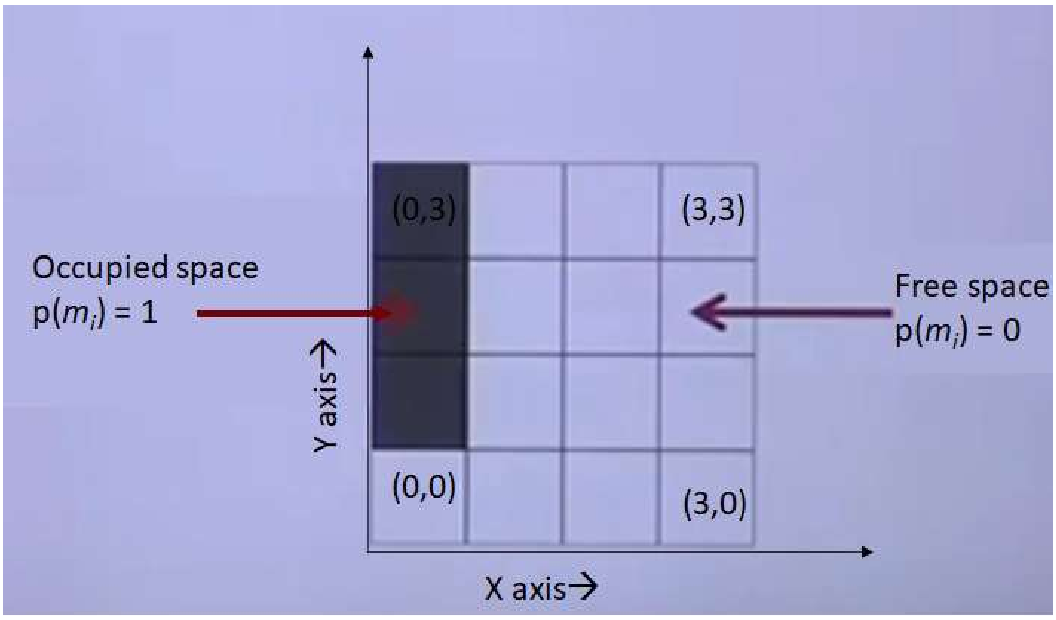


Figure 3.4: Representation of OGM

The black pixel specifies an occupied space, and the white pixel shows the free space. This work used a binary occupancy grid to represent the occupied workspace (obstacle) with a true value (1) and free workspace with a zero (0). Every cell in the representation is a binary random variable that models the occupancy with a 0 or a 1:

- If cell is occupied: $p(m_i) = 1$
- If cell is not occupied: $p(m_i) = 0$.

The indoor maps used for the experiments discussed in Chapters 4, 5 and 6 employ OGMs as grid-based maps.

3.3.1.2. Navigation guide: path estimation

Figure 3.5 shows the dynamic learning component, DynaPATH, for generating a collision-free path for VI people. As shown in the figure, the proposed approach expects each unique area of a building—such as rooms, stairs, lifts and corridors—to be equipped with a sensing layer of IoT devices. They are augmented with IoT devices that provide global information about the environment via periodic sensing. For instance, a lift or elevator not working can be recorded by the sensing layer, and thus the pathway using it can be avoided in navigation for a VI person. The numbers in Figure 3.5 denote the sequence of steps of a VI-friendly path.

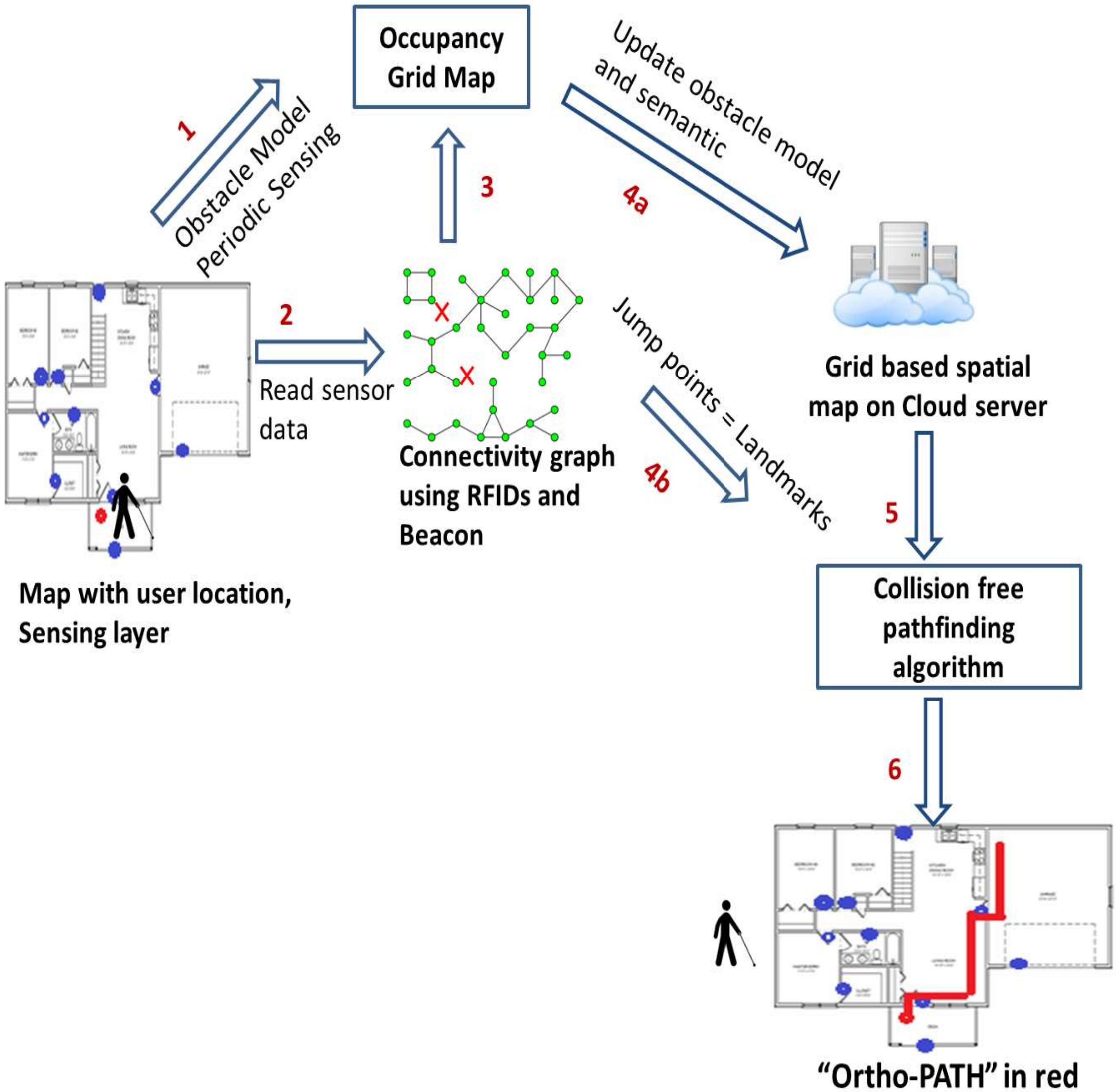


Figure 3.5: DynaPATH—path estimation for VI people in Indoor-Nav

The indoor map with basic structural information, including locations of walls, stairs, elevators, is converted in a local OGM as discussed in subsection 3.3.1.1 in Step 1. The step generates the basic OGM covering the fixed structure of the building. However, the indoor space has other moving obstacles such as furniture, interiors expected to be embedded with the IoT devices. The component DynaPATH senses the indoor environment from sensor data received from the embedded IoT layer on top of an indoor environment in step 2 to generate the connectivity graph using the IoT devices, including RFID and beacons. Later, in step 3, the component updates the OGM with the obstacle model by reading the sensing IoT-based layer with the dimensions of the obstacle and updates the basic OGM. Our approach creates an intelligent connected graph based on information using a sensing layer and grid map. The obstacle model senses the environment periodically and updates the Cloud server in step 4a. The semantic and topological information about these nodes is updated to the Cloud server periodically.

Step 5 provides two inputs, namely 1) the location of landmarks and 2) the grid-based spatial map stored in the cloud Server to the pathfinding algorithm to evaluate the collision-free path for the VI people. The nodes generated from the connected graph from step 4b acts as jump points or walkable points to the pathfinding algorithm, Ortho-PATH, in the given environment. Steps 2 and 3 updates the obstacle model if the IoT sensing layer finds any change. The updated grid-based spatial map and specific locations as jump nodes serve as inputs to the chosen pathfinding algorithm, Ortho-PATH, producing safe and VI-friendly paths for VI people. The solution aims to optimise computation time to evaluate a safe path in the indoor environment. The DynaPATH demonstrated using Figure 3.5 generates safe route for VI people following the dynamicity of the given indoor environment. The suggested path generated in step 6 avoids global and local obstacles, with straight moves along the walls (line-shore), making the path VI-friendly.

3.3.1.3. Navigation guide: tracking and positioning

This component proposes an Android-based application for indoor positioning of a VI person, based on inertial sensors of a smartphone and the DNN technique. Our system design is based on Lambda architecture and can be used for the real-time analysis of a VI person moving in a smart building using the MLP algorithm.

Figure 3.6 defines the high-level reference architecture used to estimate the location of the subject. The system applies a deep learning-based algorithm to detect the position of the VI user in the building. The system collects data from the iBeacon and the inertial sensor of the smartphone and smartwatch of the subject. The Raspberry Pi collects information about the movement of the user. The data collected by the Raspberry Pi is sent in the form of Apache AVRO messages to a big data platform created using Apache Kafka. AVRO is an open-source data serialisation system that helps with data exchange between systems, programming languages and processing frameworks. Kafka is a streaming platform that handles real-time data. The data come in the form of AVRO messages and are inserted into the system through the Kafka REST server. The Kafka REST server uses an AVRO schema that persists in the

Schema Registry server to obtain information from the messages. The data are then processed in an implementation of a Lambda architecture.

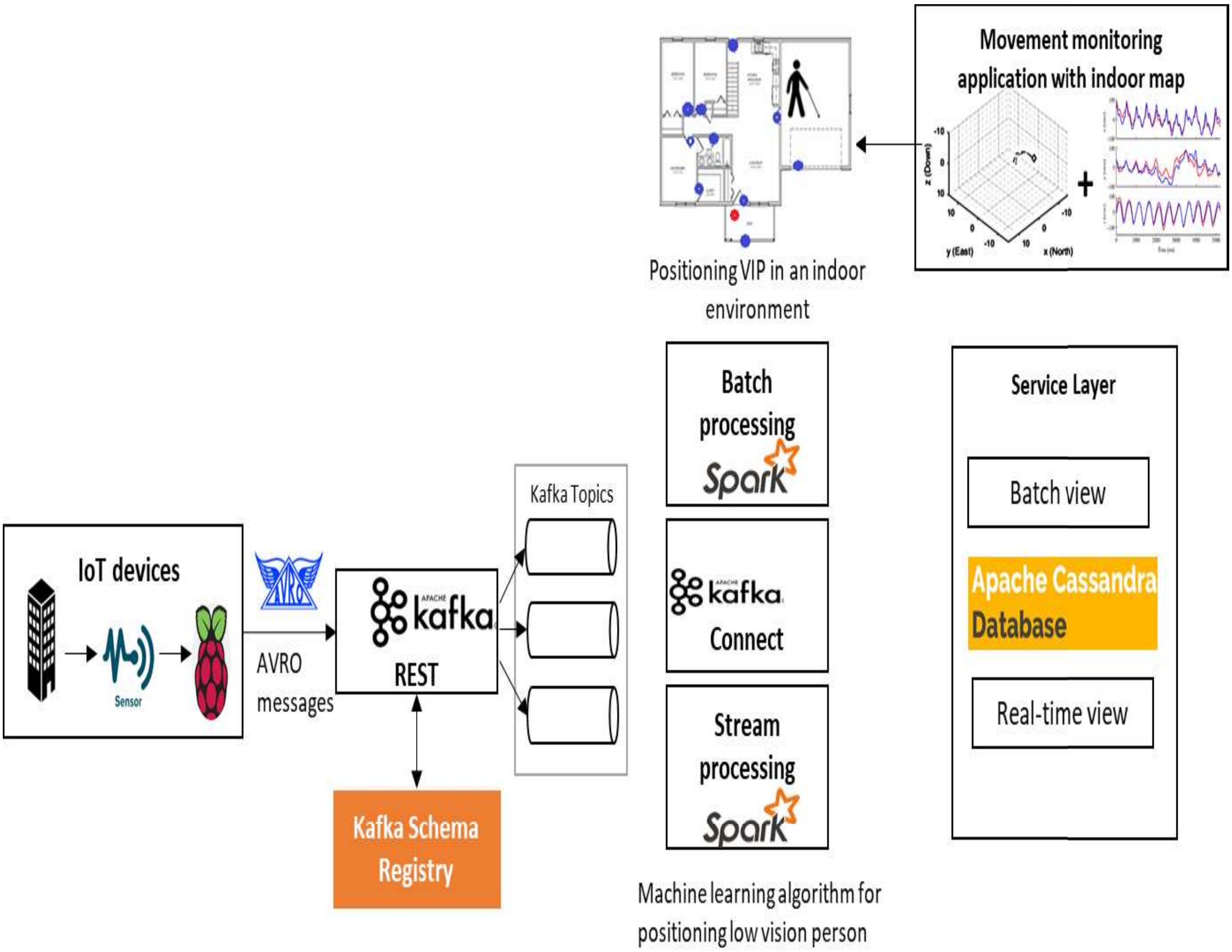


Figure 3.6: Deployment of indoor positioning and tracking system in the Cloud

Kafka Connect is a server that takes messages from the Kafka topics and inserts them in a Cassandra database. In batch processing, all data stored in the database are used as training data for the regression DNN algorithm to predict the location estimation of VI people. The DNN algorithm used for location estimation is represented as the Lambda function. For stream processing, it is unfeasible to retrain the deep learning algorithm each time new streaming data enter the system, as this process would take too long compared with the time in which the prediction should be made. Considering this time restriction, depending on the size of the streaming data, the deep learning algorithms should be retrained at a frequency of several hours.

Figure 3.7 displays the general interactions between the Android-based application residing in a smartphone and the Cloud-based pre-trained model established by a deep learning model. The

DNN is tested and deployed on the AWS cloud but could be easily deployed on any other platforms. The framework proposes a navigation guide tool for positioning a VI user in an indoor environment, using a regression-based deep learning model. The smartphone application reads the inertial sensor signals, including the accelerometer, gyroscope and magnetometer. Some more features are extracted from the raw data by the component and fed as input to the DNN model for training to estimate the user's location. The DNN network model trains the inertial sensor measurements of the smartphone to estimate the corresponding position of the moving VI user in the building. IPIN2016, a public dataset, is used to train and test the proposed model. The Android-based smartphone app informs the absolute local position of the user to the agent through an audio interface. The implementation and experiments of the DNN, with the results, are discussed in Chapter 6.

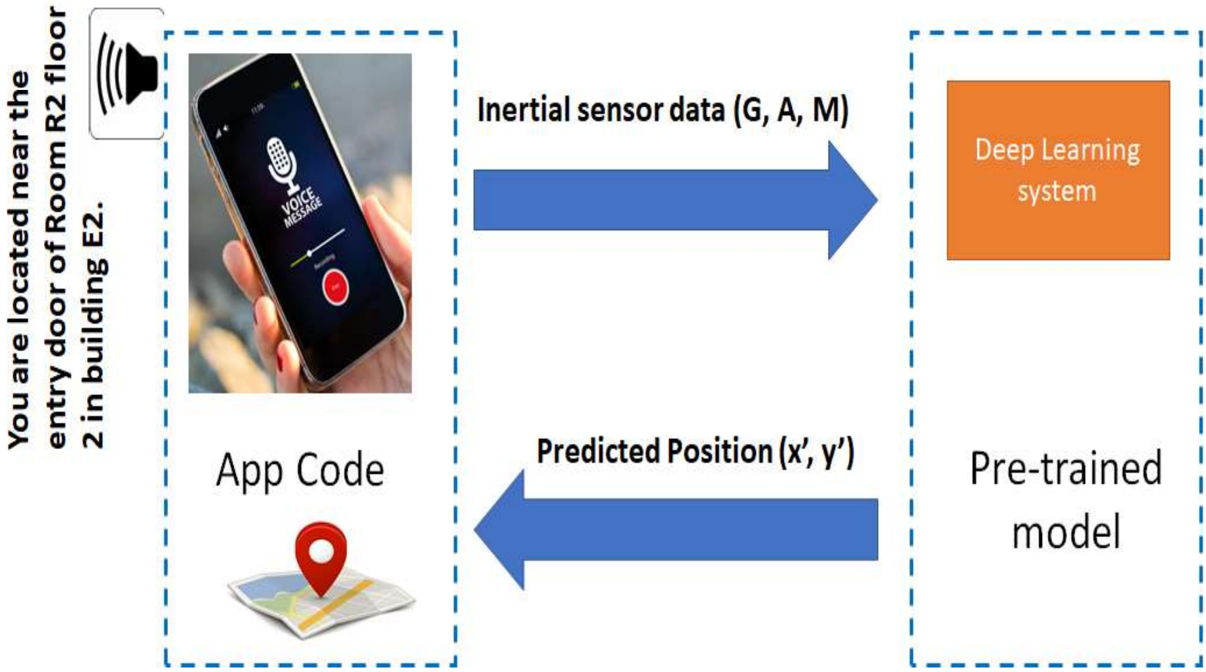


Figure 3.7: Interaction between the smartphone app and pre-trained model

3.4. Summary

The novel integration of IoT with deep learning using an innovative approach has created scope for accurate and reliable smart indoor navigation for VI people. This chapter has summarised the proposed Indoor-Nav framework for assisting VI people in navigating an indoor environment, including its layers, devices and operation. The components of the framework are implemented independently, enabling on-demand smart navigation for VI people. A range of experiments and simulations will be presented in Chapters 4, 5 and 6 to evaluate the proposed approaches related to pathfinding, tracking and positioning a VI user indoors. The study aims to validate and demonstrate the efficiency of the proposed framework, Indoor-Nav, in guiding VI people to navigate indoors independently. Chapter 4 demonstrates the effectiveness, suitability and limitations of the traditional pathfinding algorithm in estimating paths for VI people. It proposes a reliable algorithm, Ortho-PATH, to estimate a VI-friendly and safe path for VI people. Chapter 5 presents a fusion tracking algorithm based on an inertial sensor and beacon to validate whether the VI person follows the suggested path accurately. Finally, chapter 6 describes the capability of smartphone inertial sensors to localise an indoor user using deep learning techniques and a smartphone to position a VI user.

Chapter 4

Ortho-PATH: Collision-free Pathfinding Algorithm

This chapter presents the collision-free pathfinding algorithms used in indoor spaces embedded with IoT devices to allow VI users to navigate the existing infrastructure of a building. The work includes experiments and simulations of the most popular pathfinding algorithms with an indoor floorplan. The algorithms fail to provide a VI-friendly path that reacts to environmental changes. Thus, the work proposes an innovative pathfinding algorithm, Ortho-PATH, to overcome the limitations of traditional pathfinding algorithms, with high optimality in providing timely responses and improved safety for VI people.

Some parts of the work reported in this chapter have previously appeared in:
P. T. Mahida, S. Shahrestani, and H. Cheung, "DynaPATH: Dynamic learning based indoor navigation for VIP in IoT based environments," in *iCMLDE*, Sydney, NSW, Australia, Dec. 2018, pp. 8–13.
P. T. Mahida, S. S. Shahrestani, and H. Cheung, "Comparison of pathfinding algorithms for visually impaired people in IoT based smart buildings," in *ITNAC 2018*, Sydney, NSW, Australia 2018, pp. 10–13.

4.1. Overview

This chapter discusses the collision-free pathfinding algorithms used for indoor navigation services using IoT devices. With the growing development of IoT, an indoor navigation service provides an immediate visionary sense for senior and disabled communities. An indoor environment with rooms, doors and corridors, can be converted into a smart environment with the help of IoT devices. When communicating with each other, the IoT devices exchange data that can help the movement of VI people in indoor environments. However, one of the significant challenges in such an interconnected world of IoT devices is providing a reliable and safe path for VI people [110]. Thus, this chapter introduces a collision-free pathfinding algorithm for VI people. The primary goal of this chapter is to provide an inclusive understanding of traditional and popularly used pathfinding algorithms. The chapter highlights the drawbacks of traditional pathfinding algorithms for VI people. It identifies the constraints of using them in an indoor navigation system for VI people, and proposes a novel pathfinding algorithm, Ortho-PATH, to address the constraints.

Section 4.2 discusses pathfinding algorithms in general, including the difficulties faced in avoiding global and local obstacles. Section 4.3 discusses various categories of traditional and popular pathfinding algorithms used in indoor environments to suit VI people. Section 4.3.1 introduces the concept of the heuristic function used in most pathfinding algorithms, while Sections 4.3.2 and 4.3.3 discuss traditional and popular pathfinding algorithm categories—discrete and sampling-based pathfinding algorithms. Sections 4.3.2 and 4.3.3 discuss the working and pseudocodes of the different pathfinding algorithms: Dijkstra's, A*, PRM and RRT algorithms. Section 4.4 describes the innovative algorithm, Ortho-PATH, which incorporates features that closely address the needs of VI people, with the theoretical background of its application. Section 4.5 discusses the feasibility and applicability of the Ortho-PATH pathfinding algorithm over traditional path-planning algorithms, with simulation results and evaluations done in MATLAB. Finally, Section 4.6 summarises the chapter.

4.2. Introduction

Finding and planning a path is a fundamental component of applications in GPS, indoor navigation, video games, robotics, logistics and crowd simulation [46], [111]. Sight is a key sense used to find a path, and loss of the visual sense can lead to major problems for VI people in finding their way. Researchers have invested effort into helping pedestrians find the optimal and shortest path in public buildings, such as airports, museums and malls [77]. However, very few have considered the legibility of pathfinding aids for people with visual impairment [78], [112], [113]. Pedestrians have many factors influencing how easily they find their goal location in their indoor environment journey. The ability, experience and knowledge of a person about the indoor environment help pedestrians determine their way effectively. Indoor navigation is a process of taking a precomputed path from a pathfinding algorithm and determining a user's movement along the found path, considering the user's limitations [114].

An indoor pathfinding process encompasses querying a specific path from the user's current location to the target location in an indoor environmental space. Public buildings, such as hospitals, are often busy and have various interiors to confuse and obstruct a VI person navigating the environment. The interiors and furniture of the building act as obstacles for a VI person. While walking towards the goal location, avoiding obstacles is crucial [115]. Therefore, a revised definition of a pathfinding algorithm considering users' low vision involves constructing a valid collision-free path between start location A to goal location B [116]. An adaptive and safe (collision-free) path helps the VI person navigate indoor environments using advancements in IoT-based obstacle-sensing technology. There are two main types of obstacles in a public building: static and moving. Obstacles whose positions do not move are static obstacles, including walls, doors and fixed furniture. Moving obstacles include those whose positions and states change with time. Based on the frequency with which objects are moved, they can be further categorised as: (1) static landmarks (e.g., avoiding stairs or the route through a particular landmark) (2) constantly moving (e.g., people and changes in the interior), (3) temporal change (based on time—e.g., a door closed during a specific hour or a wet floor).

Despite significant progress in recent years, the pathfinding problem attracts research to produce optimised pathfinding architecture for humans with limited heavy devices to carry, unlike robots [76], [117]. Researchers have invested significant effort into providing a path that selects the shortest route to the goal location. However, the criteria for selecting a path by a pedestrian with visual limitations does not only include distance, but also includes:

- avoiding local and global obstacles
- the minimum number of turns
- the minimum computation time
- shore-lining paths (along the walls).

Any optimal path should meet these path-selection criteria to provide a realistic, human-like, adaptable and dynamic indoor environment.

4.3. Collision-free pathfinding algorithms

Pathfinding algorithms for the indoor environment must determine an obstacle-free path for VI people. Representation of the indoor environment is an important aspect to be considered before proposing a pathfinding algorithm. As discussed in section 2.4, the indoor map representation data are stored as dense (grid-based) or sparse (skeleton-based) graphs. In other words, the data can be represented as a weighted graph to search the goal node from the start node with a specific heuristic. In provided simulations, the grid-based dense technique has been used, as discussed in Section 2.4. The connectivity of indoor spaces can be represented with a graph that captures the relationship between a collection of nodes V and edges E. A network of rooms in a building is represented as nodes in the graph. The accessible routes connecting rooms with horizontal or vertical connectivity, such as stairs and elevators, act as edges. Each edge E is assigned a weight, as the distance between the nodes or an estimation needed to travel along the edge. A graph $G = (V, E)$ contains n ($n > 2$) nodes, named V_1, V_2 … V_n, and edges E_1, E_2, E_3 … E_n. The neighbourhood of a node $n \in V$ is denoted by $N(n)$ and defined by Equation (4.1):

$$N(n) = \{m \in V: (n, m) \in E \tag{4.1}$$

A graph search algorithm starts at node V_{start} and attempts to find a path to a goal node V_{goal} by exploring the neighbouring nodes V via the connecting edges E. A node is ‘unexpanded’ if the algorithm has not reached it. A node is ‘alive’ or ‘open’ if it has been reached, yet has at least one neighbour not yet reached. A node is ‘closed’ or ‘dead’ when the search algorithm has reached it, and so have all its neighbours.

Among various challenges in implementing indoor navigation for VI people, this research sought to answer a few inevitable questions. By performing simulations on traditional pathfinding algorithms, this study aimed to answer the following primary questions:

- How optimal are the current traditional pathfinding algorithms for VI users?
- Which pathfinding algorithm will provide optimal paths in an indoor navigation system for VI people, considering real-time changes in the environment of a building?

Figure 4.1 presents the design to fetch obstacle information and feed it as input into the pathfinding algorithm, as discussed in Sections 4.3.1 and 4.3.2.

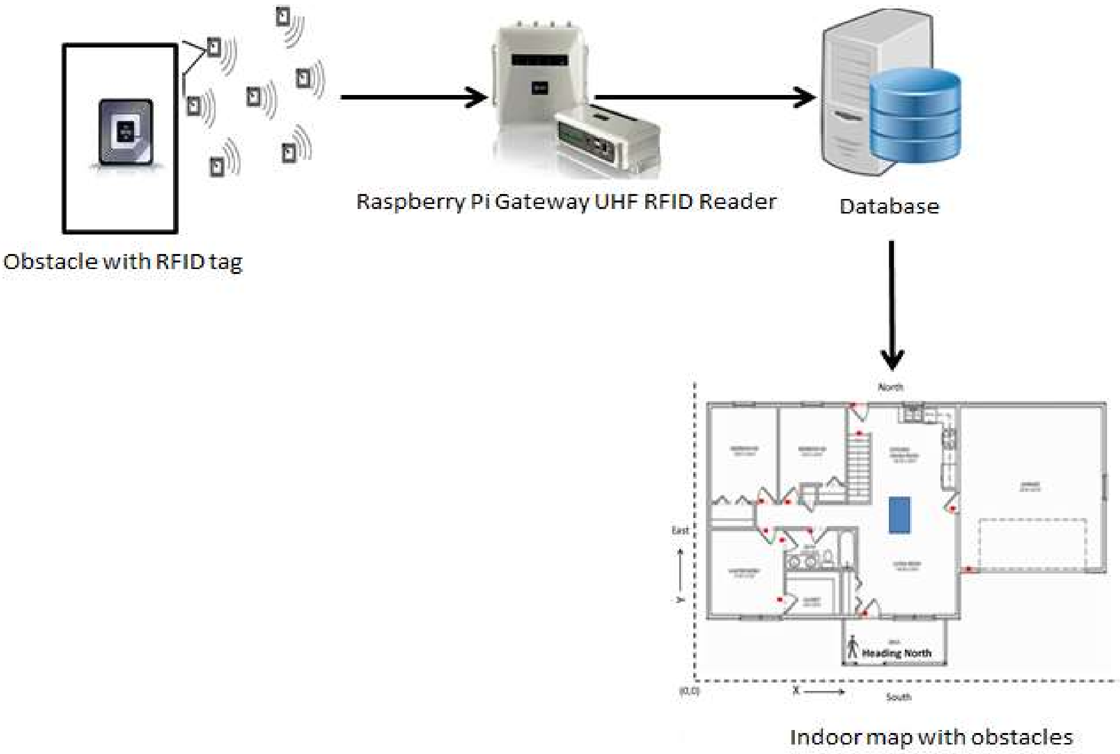


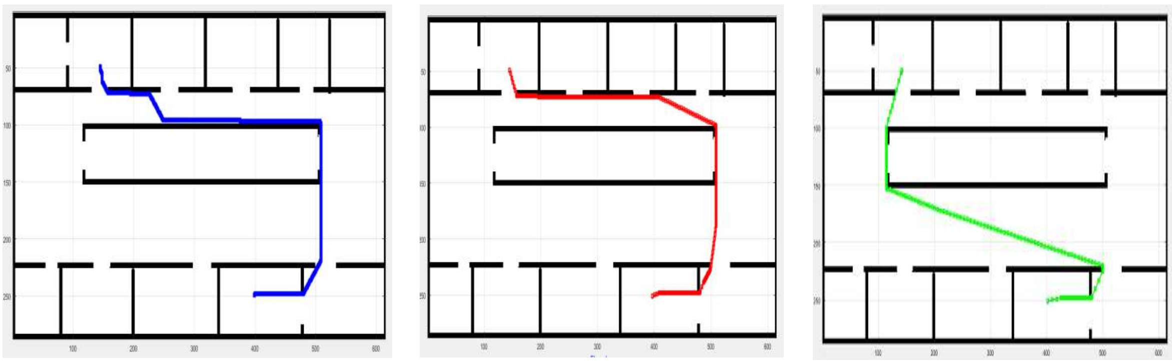
Figure 4.1: Fetching obstacle information using IoT device

Each interior of the building, including fixed and moving furniture, is embedded with an RFID tag. All the obstacles that might hinder a VI person need an RFID tag. Each tag in the area is equipped with detailed dimensions of the object. An RFID tag emits RFID signals sent continuously and read with a Raspberry Pi Gateway UHF RFID reader. Static and moving obstacles can be traced using an RFID reader and then fed into the database system. With obstacle information from the database system, the building's indoor floor plan is fed into the pathfinding algorithm to generate a VI-friendly path. Further, the pathfinding algorithm searches for an optimal path in the indoor space, using a grid-based graph representation technique, as discussed in Chapter 2, and the heuristic function, discussed in the following section.

4.3.1. Heuristic functions

A 'heuristic' is defined as searching for an optimal path from the start node to the goal node [118]. Heuristics are essentially used to guess whether a node being evaluated will lead to the goal node. Heuristics help increase pathfinding algorithms' efficiency, as they are inclined to limit the number of searched nodes. A heuristic function, *h(n)*, increases the chance of finding an optimal path using heuristic information [53]. The function *h(n)* takes the goal node *n* as input and returns a non-negative real number as an estimate of the cost of the path to reach the goal node *n* from the start node [92]. The function *h(n)* is an underestimate if *h(n)* is less than or equal to the actual cost of the lowest-cost path to the goal node *n*. If the heuristic is admissible and does not overestimate the cost of the path from the start node to the goal node, then the path to node *n* is guaranteed to be optimal [119]. This function helps guess the most promising neighbour of the current node while searching and estimating the search direction with a greater chance of leading to the goal node.

In a grid-based map representation, each straight, horizontal or vertical move from a traversable node to one of its neighbours may or may not have a uniform cost. The map representation includes non-traversable nodes, such as walls and other obstacles where a path is not allowed through the nodes. A discrete pathfinding algorithm (Dijkstra's) was implemented in MATLAB to determine the value of the appropriate neighbouring node m. Figure 4.2 (a), (b) and (c) show the paths generated for varying numbers of neighbouring nodes, with m = 1, 4 and 8.



(a) (b) (c)

Figure 4.2: Quality of path generated from m neighbours: (a) m = 1, (b) m = 2, 4 and (c) m = 8, 16

As the number of neighbouring nodes to be explored increases, the generated path's quality also increases, with fewer directional changes. However, as the number of neighbouring nodes increases, the computation time also increases in finding an optimal path. Therefore, the generated route from m = 8 can be considered the base heuristic with less turns that generates an optimal path with better quality considering the needs of the VI people, as shown in Figure 4.2.

4.3.1.1. Heuristic functions

This subsection discusses the distance measurement techniques between the two nodes, including Euclidean distance, Chebyshev octile distance, Manhattan distance and diagonal distance algorithms [78], [120]. In these experiments, there are eight neighbouring nodes.

Manhattan distance:

The Manhattan distance is a standard distance heuristic for a grid-based technique to estimate the distance between two nodes in an *n*-dimensional space, known as L1 distance [117]. The distance between two locations using Manhattan distance is the absolute difference between these two points in a grid on a horizontal or vertical path. Instead of calculating a straight path through two points, Manhattan distance calculates the sum of the absolute difference of its horizontal and vertical components. Equations (4.2) and (4.3) calculate the Manhattan distance between two nodes, A and B:

$$d_x = |x_2 - x_1| \text{ } and \text{ } d_y = |y_2 - y_1| \tag{4.2}$$

$$Manhattan\ (A, B) = d_x + d_y \tag{4.3}$$

where d_x represents the distance between two x-coordinates (x_1, x_2) of nodes A and B, and d_y represents the distance between two y-coordinates (y_1, y_2) of nodes A and B. The function *Manhattan (A, B)* denotes the Manhattan distance between A and B.

Euclidean distance:

Euclidean distance between two points in Euclidean space is a straight-line distance between two points, where an agent may move in any direction. It is the cartesian coordiantes of the points using the Pythagorean theorem. The Euclidean distance between the two nodes A and B is depicted in Equation (4.4):

$$Euclidean\ (A, B) = \sqrt{{d_x}^2 + {d_y}^2} \tag{4.4}$$

where d_x represents the distance between two x-coordinates (x_1, x_2) of nodes A and B, and d_y represents the distance between two y-coordinates (y_1, y_2) of nodes A and B, as shown in Equation (4.2). The *Euclidean (A, B)* is the Euclidean distance between nodes A and B.

Figure 4.3 represents the distance between the start and goal locations using Euclidean and Manhattan distance in blue and red. Considering the needs of VI people, the Manhattan distance is preferred, as it avoids an angular move. VI people tend to avoid making diagonal moves between two locations and prefer walking along walls.

Diagonal move:

The diagonal move is an eight-way movement heuristic across diagonals when the distance between two adjacent points (diagonal and non-diagonal) is the same. Equation (4.5) presents the calculation of diagonal move:

$$Diagonal\ (A, B) = D * \left(d_x + d_y\right) + (D_2 - 2 * D) * \min(d_x, d_y) \tag{4.5}$$

where *Diagonal (A, B)* is the function to calculate diagonal distance between two nodes A and B. The distance d_x represents the cost between two x-coordinates (x_1, x_2) of nodes A and B, and d_y represents the cost between two y-coordinates (y_1, y_2) of nodes A and B. For $D = 1$ and $D_2 = 1$, the heuristic distance is known as Chebyshev distance [121]. When D = 1 and $D_2 = sqrt(2)$, the heuristic distance is known as octile distance.

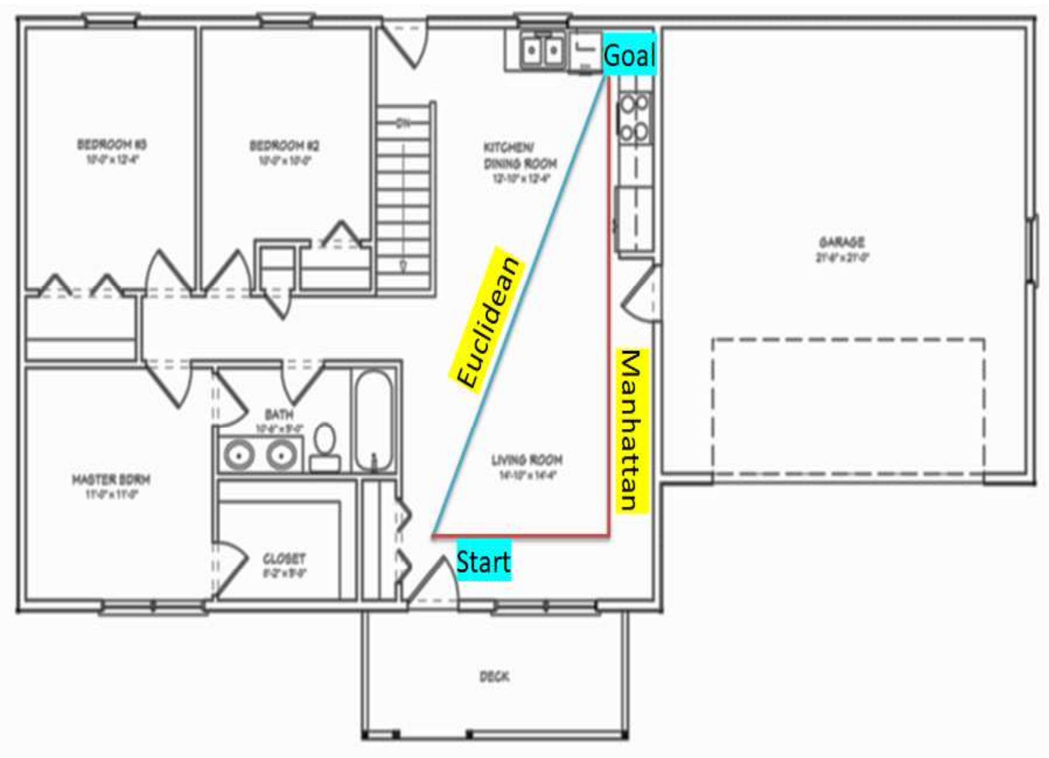


Figure 4.3: Distance heuristic based on Euclidean and Manhattan techniques

Further in the following section, traditional pathfinding algorithms are discussed, with their pseudocodes and simulations.

4.3.2. Discrete pathfinding algorithm

Pathfinding is a complicated problem in a continuous domain [116]. However, discrete pathfinding algorithms resolve pathfinding issues by converting the continuous configuration space into discrete space. The cell decomposition method placed over the configuration space marks discrete positions of obstacles and possible path movements. This method generates a finite and countable number of states *S*. There has been significant research on discrete search mechanisms, given their importance in robotics, games and navigation [122]. Breadth-first, depth-first, best-first, Dijkstra's and A* algorithms are popular and most used in pedestrian navigation as discrete search mechanisms [123]. This section discusses the most popular algorithms, Dijkstra's and A*, as discussed in Section 2.4, and presents the simulation results using the algorithms in Section 4.5.

4.3.2.1. Dijkstra's algorithm

Dijkstra's algorithm is a classic shortest path algorithm between two points, given its optimisation capability [75]. Hence, it can be referred to as a breadth-first search algorithm for finding the shortest paths from a single source node to all other nodes. When Dijkstra's algorithm searches in the forward direction, the cost is frequently called cost-to-come because it represents the minimum cost required to reach the goal node [124]. This work deals with a backward version of Dijkstra's algorithm, in which the cost is called cost-to-go because it represents the cost of going from the start to the goal node. Algorithm 4.1 displays the pseudocode of Dijkstra's algorithm.

For graph G with vertices V and edge E, Algorithm 4.1 represents Dijkstra's algorithm to find a path from the V_{start} to V_{goal} nodes. The algorithm calculates the cost c_j for all nodes and inserts the vertices in an empty list, *open-list*. Based on the available shorter route, the *update* (v_i, c_j, *open-list*) function inserts a new vertex v_i with the calculated minimum cost c_i into the list *open-list* for the paths not already in the list. The nodes with minimum cost heuristics are further expanded in the algorithm until the goal node is reached. Also, open nodes in a heap are reordered (and the search tree adjusted) whenever a cheaper path to the goal is found through a recently expanded neighbour. Many navigation systems in indoor environments for VI users propose the use of Dijkstra's algorithm [75]–[77], [125], [126], [127].

Algorithm 4.1: Dijkstra's

Input: G, V, E, V_{start}, V_{goal}

for all v_{start} ∈ V_{goal} **do**

insert(v_{goal}, 0, open-list)

while the open-list is not empty **do**

v_j = get-shortest(open-list)

if(v_j ∈ V_{start} **then**

return SUCCESS

for all v_j such that $e_{i,j}$ ∈ E **do**

if v_i is unexpanded or $c_i > d_{i,j} + c_j$ **then**

$c_i = d_{i,j} + c_j$

update (v_i, c_i, open-list)

set back pointer from v_i to v_j

return FAILURE

4.3.2.2. A algorithm*

The A* algorithm is an informed-search algorithm that solves pathfinding using a heuristic function to estimate the cost between the start node and goal node. The A* algorithm augments the original Dijkstra's shortest path algorithm by adding a heuristic. It is extensively used for pathfinding in computer games [46]. The algorithm does not waste time exploring all nodes, but only expands directions that look promising to find the shortest path to the goal node. The heuristic in the algorithm helps determine how close the search is to the goal node. The A* algorithm selects the path that minimises the cost function $f(n)$, as in Equation (4.6):

$$f(n) = g(n) + h(n) \tag{4.6}$$

where n is the goal node, $g(n)$ is the cost function of the path from the start to goal node n, and function $h(n)$ is a heuristic to estimate the cost of the cheapest path to the goal node.

Algorithm 4.2 displays the pseudocode for the A* algorithm that uses a heuristic-based approach to estimate the cost to traverse between two nodes. The function denoted with $h_{start,I}$ in the algorithm is the estimated heuristic function that calculates the cost between the start node V_{start} and goal node V_{goal}. The A* algorithm proposes a path between the start node and the goal node based on the selected heuristic. Hence, A* is complete in a finite graph, yet may not be resolution complete in a countably infinite graph, depending on its heuristic [128]. The heuristic function $h(n)$ gives the A* algorithm an estimate of the minimum cost from node n to its goal node. The A* algorithm turns into Dijkstra's algorithm if $h(n)$ is zero, as only $g(n)$

is present in the cost function. The A* algorithm guarantees to find the shortest route if $h(n) < g(n)$. However, the lower the $h(n)$ value, the more nodes in A* expand, making it slower. If $h(n) = g(n)$, the A* algorithm tries to find the best path and does not expand any additional nodes. However, in situations where the A* algorithm does not guarantee the shortest path, it can run faster than the other two cases.

If $h_{i,j} \leq d_{i,j}$, then the heuristic is said to be admissible or optimistic. When the A* algorithm uses an admissible heuristic, the path from the start node to the goal node is guaranteed to be optimal concerning the cost. As long as the heuristic does not overestimate the true cost, the algorithm will never miss an opportunity to find a less expensive path through a node on the open list. As with other best-first search algorithms, the A* algorithm modifies the search tree to reflect better paths through nodes on the open list as they are discovered.

Algorithm 4.2: A* algorithm

Input : G, V, E, V_{start}, V_{goal}
for all $v_i \in V_{goal}$ **do**
    insert(v_i, h_{start}, i, open-list)
while the open-list is not empty **do**
    v_j = get-feasible(open-list)
    if($v_j \in V_{start}$ **then**
        return SUCCESS
    for all c such that $e_{i,j} \in E$ **do**
        if v_i is unexpanded or $d_{i,goal} > d_{i,j} + d_{j,goal}$ **then**
        $d_{i,goal} = d_{i,j} + d_{j,goal}$
        $c_i = h_{start,i} + d_{i,goal}$
        update (v_i, c_i, open-list)
        set back pointer from v_i to v_j
return FAILURE

The performance of the heuristic functions discussed in Section 4.3.1 (Euclidean, diagonal and Manhattan distances) for a typical indoor environment is shown in Figure 4.4 (a), (b) and (c). The floor plan used for the simulation is kept uniform for other simulations in the chapter, as discussed in Section 4.6. The floor plan includes 25 rooms, passages, lifts and stairs. Figure 4.4 (a), (b) and (c) show paths generated by the algorithms based on the Euclidean, diagonal and Manhattan distances, respectively.

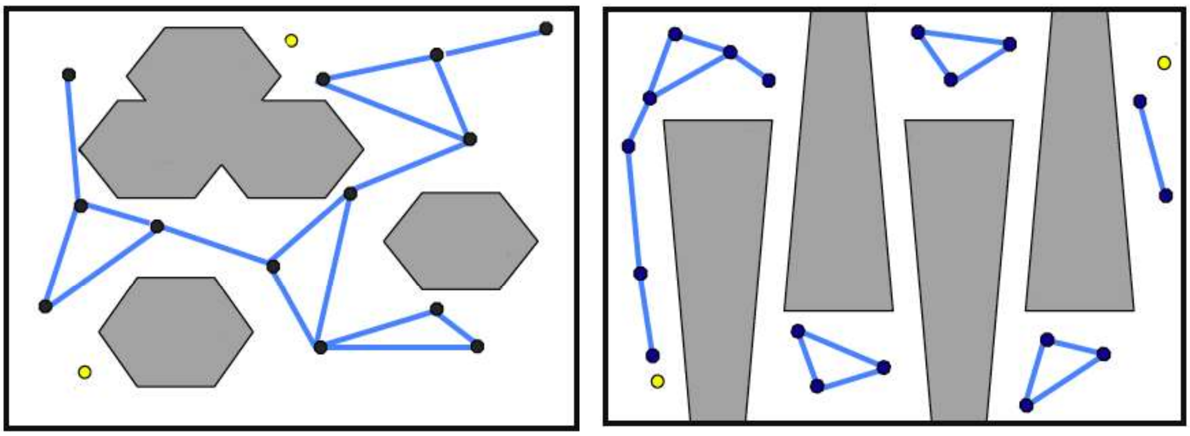
Figure 4.4: Quality of path generated by (a) Euclidean, (b) diagonal and (c) Manhattan heuristic functions

4.3.3. Sampling-based pathfinding algorithm

Sampling-based pathfinding algorithms have proven valuable in various application domains, such as robotics, aerospace, manufacturing, medicine and computer animation [129]. In a given configuration space with obstacles, a general problem in navigation for VI users is to follow a collision-free path [51]. Sampling-based methods provide approximate paths, avoiding the excessive computation cost in representing free configuration space in discrete pathfinding algorithms [56]. A sampling-based pathfinding algorithm explores a given map's configuration space by sampling the configuration space, and builds a graph representing its connectivity [74]. This section discusses the two popular sampling-based algorithms: PRM pioneered by Kavraki and Latombe and RRT by Lavalle and Kuffner. The work assumes that the obstacle information from RFID tags is provided to the pathfinding algorithm, as discussed earlier.

4.3.3.1. Probabilistic roadmap algorithm

The PRM algorithm first constructs a roadmap graph representing a set of collision-free paths, and then queries the shortest path connecting a start node to the goal node using the roadmap graph [130]. In sampling-based algorithms, the number of samples determines the computation complexity to find a feasible path [131]. The PRM algorithm operates by randomly sampling the configuration space coordinates, then further mapping them into an obstacle space. In a PRM algorithm, the randomly generated sample nodes are referred to as 'waypoints'. The waypoints are selected randomly, without any bias or prior knowledge about the map. The indoor space with sample nodes of the map is used to build a connectivity graph. The algorithm attempts to generate a connectivity graph in the configuration space by connecting to the nearest neighbours with some distance *d*. Figure 4.5 (a) represents the roadmap generation for a given sample map with static obstacles. Figure 4.5 (b) illustrates the inefficiency of PRMs in generating paths with narrow passages. An algorithm is complete if it terminates in a finite time, returns a valid solution if one exists, and is otherwise a failure [130]. Therefore, the PRM is a probabilistically complete and suboptimal algorithm.



(a) (b)

Figure 4.5: PRM (a) generation and (b) problems

The PRM algorithm is used in multi-query applications consisting of two phases—PRM generation or pre-processing phase and query phase, as depicted in Algorithm 4.3 (a) and (b) [130]. Roadmap generation begins with an empty graph G as depicted in algorithm 4.3 (a). At each iteration, a random vertex V_{rand} is created and added as a vertex to graph G if it is found to not collide with any obstacles in the map.

Algorithm 4.3 (a): PRM generation

Input: G, N

1: **for** i=1 to N
2: V_{rand} = *GenerateRandomSample()*
3: *if*($V_{rand} \in v_{free}$)
4: *G.add*(V_{rand})
5: ***for** each* V_{near} in G.nearestneighbours (V_{rand}, d)
6: **if**(NoObstacle(V_{near}, V_{rand}))
7: G.addEdge(V_{near}, V_{rand})

Algorithm 4.3 (b): PRM query

Input: G, V_{start}, V_{goal}

1: *PathPlan*(G, V_{start}, V_{goal}, discrete_algorithm())

V_{free} is the set of configurations for which the search does not collide with any obstacles. Connections are attempted between V_{rand} and a neighbouring node V_{near} located at a distance d if not colliding. Thus, the new edge is added to the graph between the V_{rand} randomly-generated node and V_{near} neighbouring node in the map. The PRM roadmap generation phase creates a generic roadmap without considering any start or goal nodes. The approach's advantage is that it can be reused to query multiple nodes with different start and goal nodes. Multi-query planners are used when the multiple goal nodes need to be searched in the same environment [132]. A graph is created, stored and (if required) improved over time in multi-query planners. In the query phase, as shown in Algorithm 4.3 (b), a discrete algorithm, such as Dijkstra's algorithm, is applied to search for the shortest path between V_{start} and V_{goal} nodes. This section discusses simplified and basic versions of sampling-based pathfinding algorithms. Some extensions of the PRM algorithm (e.g., near obstacles, deformed objects and closed chain systems) are provided to improve the computational speed and optimality [131].

4.3.3.2. Rapidly exploring random tree algorithm

In some applications, computing a roadmap a priori may be practically unfeasible or wasteful. Therefore, incremental sampling-based pathfinding algorithms, such as RRT, have emerged [57]. Like PRM, this approach works by generating random nodes in the map and connecting them to form a special type of graph, called a 'tree'. However, the difference lies in the way the construction is undertaken to generate this tree. On each iteration, the algorithm generates a random node V_{rand} in the map and checks if it is free, V_{free}, as shown in Figure 4.6. Further, the algorithm searches for the closest neighbouring node, V_{near}, in the existing tree and attempts to connect the V_{rand} and V_{near} nodes via a straight line that corresponds to moving straight in the physical space. The algorithm only selects the neighbouring node, V_{near}, in the range of threshold distance *delta*. Figure 4.6 depicts the selection of the new node for generating the connected graph. The node in red, V_{rand}, represents the random node generated by the algorithm, whereas the node in black, V_{near}, represents the closest node in the existing tree. Given that V_{rand} is not within the defined threshold value, V_{new} is the new node generated that is a distance delta away from V_{near} along the line towards V_{rand}.

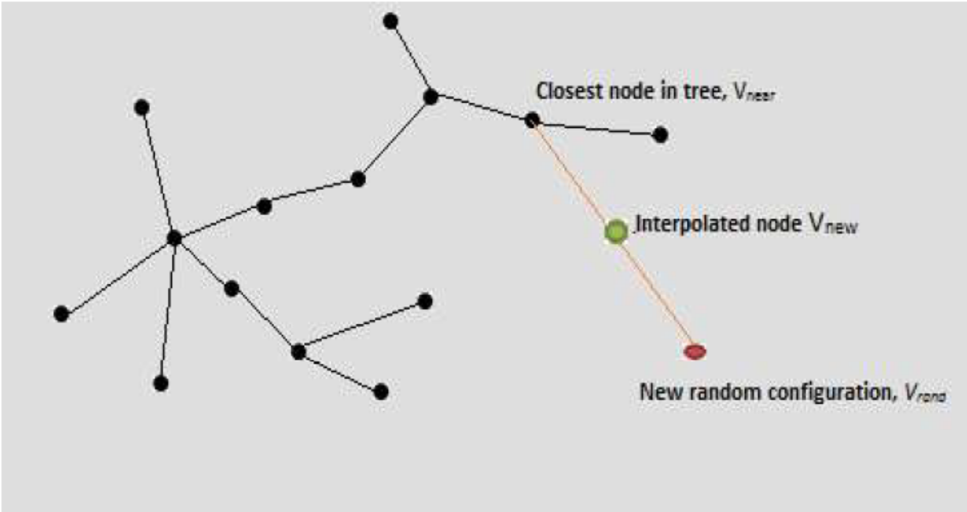


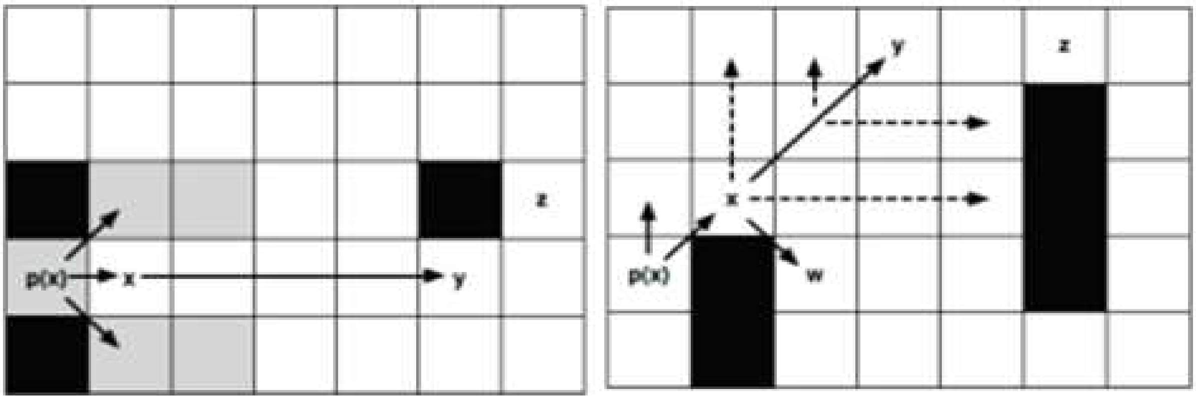
Figure 4.6: Selection of nodes in RRT algorithm

Algorithm 4.4 represents the pseudocode to build the path using RRT algorithm. However, the random node is abandoned if the algorithm cannot find the closest satisfying node. The algorithm will select a new random node V_{rand} to continue the search. The iteration is stopped as soon as the tree finds the goal node. If a successful connection between V_{rand} and V_{near} can be found, then V_{rand} is added to the tree graph, with V_{new} as the parent node [130].

```
Algorithm 4.4: Generate_RRT
Input:  Vstart, T, N, delta
T.init(Vstart)
for i=1  to N do
        Generate a random configuration, Vrand
        If Vrand is in Vfree using the CollisionCheck()
                Find Vnear, neighbour node in the tree to Vrand
                If(dist(Vrand, Vnear) > delta  (too far)
                    Find Vnew such that dist(Vnew, Vnear) <= delta
                    Vrand = Vnew
        If(LocalPlanner(Vrand, Vnear) (if reachable)
                T.addNode(node Vrand, parent Vnew)
```

4.4. Ortho-PATH: proposed pathfinding algorithm

Ortho-PATH enhances orthogonal jump point search (OJPS)—an efficient pathfinding approach implemented on rectangular grids [133]. This work proposes Ortho-PATH, a novel pathfinding algorithm to fulfil VI people's basic requirements: safety and line-shore paths. It is an optimal search algorithm for speeding up the search by selectively expanding only specific nodes, known as jump points, on a grid map [110]. This algorithm defines jump nodes as the nodes that can be reached in straight lines, suitable for walking in a straight line. As a result, the average search time is reduced across all benchmarks in the algorithm [133]. The jump point technique expands very few and specific nodes from a grid map, rather than all nodes in the path. Figure 4.7 (a) presents the basic idea of straight or diagonal jump point nodes, with node *x* and parent node *p(x)*. The non-dominated neighbour of *x* lies immediately to the right, unlike the A* algorithm, which generates a neighbour node and adds it to the open list. The algorithm moves to the right without adding a new node to the open list, and moves in the direction until it encounters a node *y*. The node expansion gradually speeds up by identifying the jump point successors in the case of both straight and diagonal moves, as shown in Figure 4.7 (a) and (b).



(a) (b)

Figure 4.7: Examples of straight (a) and diagonal (b) jump points [133]

Rather than searching the nodes at runtime on the grid-based map, the jump point nodes are pre-calculated in the Ortho-PATH algorithm. The algorithm's objective is to provide a VI-friendly path with a reduction in computation time, and remove symmetry by recursively jumping over the nodes that can be reached optimally by a path that does not visit the current node. The Ortho-PATH does not evaluate jump nodes, as the algorithm pre-calculates [133]. Algorithm 4.5 defines the process of identifying jump point successors. The current node is represented as *x*, the start node as V_{start} and the goal node as V_{goal}, which act as inputs to the algorithm. Individual jump point nodes are identified as successors, as depicted in Algorithm 4.6 (a). Instead of adding each adjacent neighbour to the open list, the algorithm tries to find a more distant node by pruning the neighbouring nodes.

Algorithm 4.5: Ortho-PATH

Input: *x: current node, V_{start}: start, V_{goal}: goal*

Successor(x) ⇐ 0

Neighbours(x) ⇐ avoid_all_grids(x, neighbours(x))

for all n ∈ neighbour(x) do

    n ⇐ jump_node(x, V_{start}, V_{goal})

    if(direction(x, n) is straight)

        add n to successor(x)

return successor(x)

Algorithm 4.6 (a) Ortho-PATH—Identify successors

Input: x, G, V, E ,*x: current node, V_{start}: start, V_{goal}: goal*

1: successors(x) ← ∅

2: neighbours(x) ← prune_node (x, neighbours(x))

3: **for** all n ∈ neighbours(x) do

4:   n ← jump(x, direction(x, n), s, g)

5:   add n to successors(x)

6:   **if**(direction(x, n) is straight)

7:         add n to successor(x)

8:

9: return successors(x)

Algorithm 4.6 (b): Ortho-PATH planning

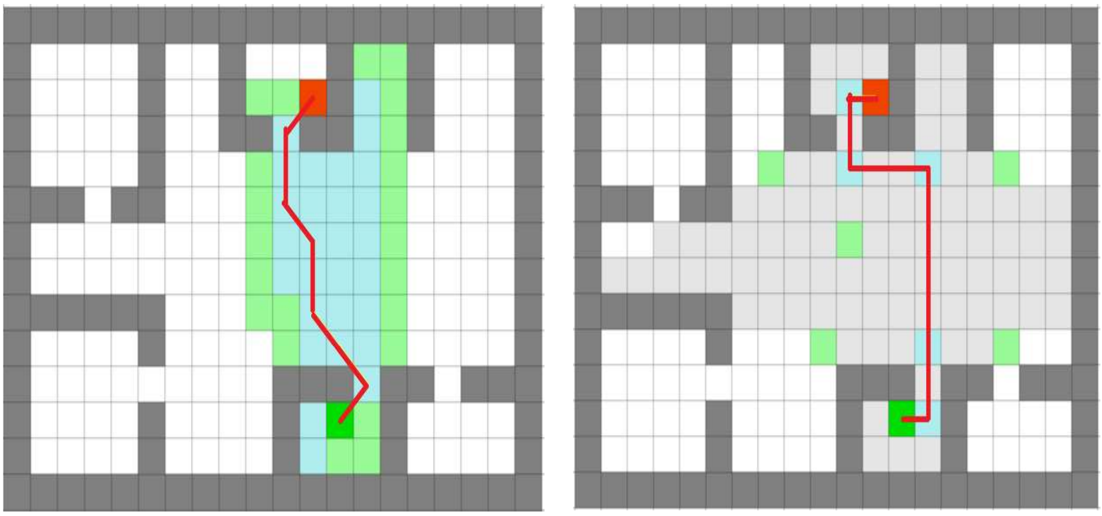
1: Plan(G, Vstart, Vgoal , successors)

Suppose the edge *(x, n)* forms a straight move travelling right from x to node n. Node n is considered the jump point. Such nodes are added to the list of successors in the algorithm. This process of finding successors continues until the set of neighbours is exhausted. The algorithm attempts to establish the list of jump points from the start node by stepping in direction d by avoiding obstacles. It follows the rules such that the jump nodes are orthogonal to each other. Algorithm 4.6 (b) plans a path from the start node to the goal node through the identified jump nodes.

This work compares the operation of the A* algorithm with the Ortho-PATH pathfinding algorithm with three cases: (1) determining the number of nodes to be explored, (2) developing a line-shore and safe route and (3) post-processing the path to provide formal instructions to users. Case 1 discusses validating the number of nodes explored to find an optimal path from the start node to the goal node. Case 2 discusses the safest path for VI users. Finally, case 3 compares the post-processing time taken by A* and Ortho-PATH to provide appropriate instructions based on the evaluation of the path.

4.4.1. Case 1: relationship of node exploration with computation time

Figure 4.8 (a) and (b) show the number of nodes explored to generate the route between the start and goal node by A* and Ortho-PATH algorithms, respectively. There is a close relationship between the number of nodes explored and computation time taken by the pathfinding algorithms of A* and Ortho-PATH. The red and green boxes represent the start and goal nodes, respectively. The blue nodes represent the neighbouring nodes explored to reach the goal node. Based on the results shown in the figures, A* explore 28 nodes, compared with only 5 jump nodes with the Ortho-PATH algorithm.

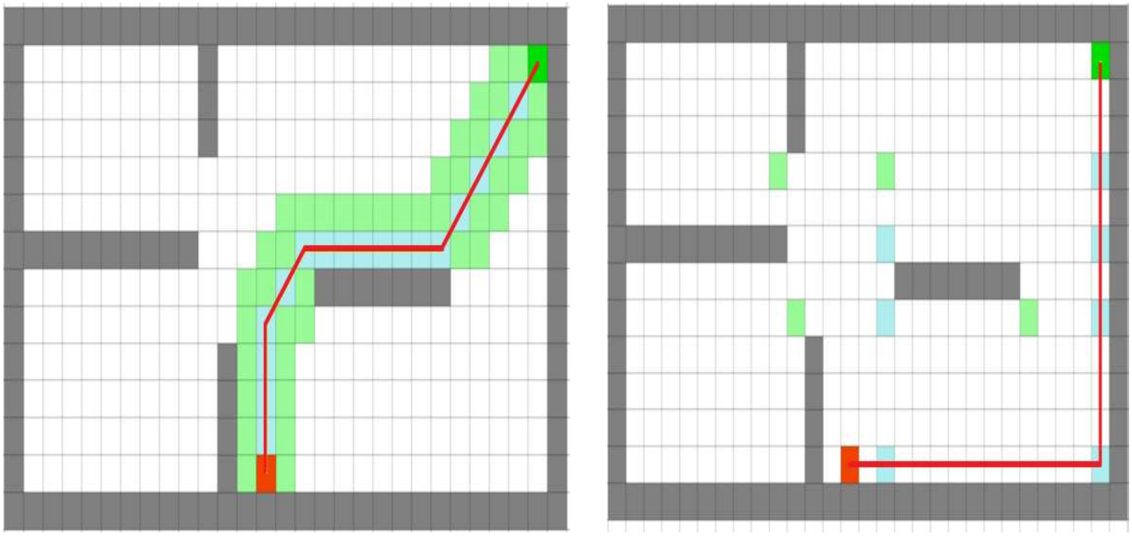


(a) (b)

Figure 4.8: Algorithm with *n* explored neighbouring nodes by (a) A* and (b) Ortho-PATH

4.4.2. Case 2: line-shore and safe route versus shorter route

Figure 4.9 (a) shows the path generated by the A* algorithm, and Figure 4.9 (b) presents the path generated by the Ortho-PATH algorithm. The path generated by A* is shorter with directed moves. However, Ortho-PATH generates a longer route with a line-shore path. The Ortho-PATH is not an optimal algorithm, as it develops a longer route, but it is acceptable and safe for VI people. It avoids obstacles and provides a line-shore path for VI people to feel more comfortable. Thus, this case scenario depicts the choice of a path with a slightly longer distance that is safe and acceptable to VI people.

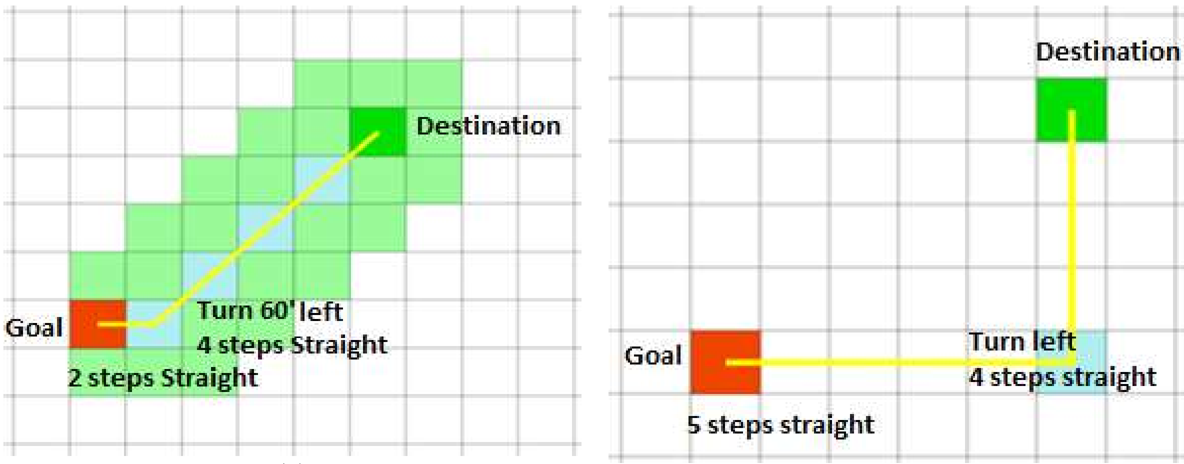


(a) (b)

Figure 4.9: Path by (a) A* and (b) Ortho-PATH

4.4.3. Case 3: path quality

Figure 4.10 (a) and 4.10 (b) show paths generated by the A* and Ortho-PATH algorithms, respectively. Once the path is generated, the user requires audio instructions to follow the path. This case discusses each algorithm's audio instruction sample based on the path generated. The path quality and steps instruction in the A* algorithm involve angular turns and directions, as shown in Figure 4.10 (a). The path suggests moving straight for two steps, then turning 60 degrees to the left and walking four steps to reach the destination. In contrast, the path generated by Ortho-PATH has simple instructions, such as turning left and right, and the number of steps.



(a) (b)

Figure 4.10: Path quality in (a) A* and (b) Ortho-PATH

As discussed before, the cases and their justifications show that the Ortho-PATH algorithm is suitable for VI users' indoor navigation. This algorithm usually provides the shortest route. The following section discusses the simulation results of all algorithms to check the feasibility of the algorithms in an indoor environment.

4.5. Evaluation and simulation results

The section discusses the simulation platform and results with performance metrics used to verify the analytical modelling and the effectiveness of the pathfinding algorithms discussed in Sections 4.3 and 4.4. These sections discuss the simulation platform and results of the pathfinding algorithms, including A*, Dijkstra's, RRT, PRM and Ortho-PATH, for VI people. The dimensions of the map and obstacles in the environment used in the simulations are assumed to be known to the platform, based on the design discussed in Section 4.3.

The simulation platform was tested on two floorplans with different designs. One of the floor plans included 25 rooms, passages, lifts and stairs. The layout of the floor plan was divided into 180 grid points, horizontally and vertically. The other floorplan consisted of 10 rooms, one conference room and a corridor of 600 grid points, horizontally and vertically. An obstacle-avoidance mechanism was implemented in each of the algorithms to avoid static obstacles presented on the map. Therefore, if an obstacle is in the path of the traversable area, the path is prevented by the algorithms. Figure 4.11 presents the main screen of the simulation platform developed in MATLAB to simulate different pathfinding algorithms. With the start and goal locations, the indoor environment's floor plans are provided as input to the system. Based on the selection of obstacles and their location, the path is computed by the selected algorithms. There are several criteria to consider to evaluate the performance and feasibility of such algorithms in an indoor environment. In this study, different scenarios were explored, varying the positions of the obstacles and the start and goal nodes to generate different paths.

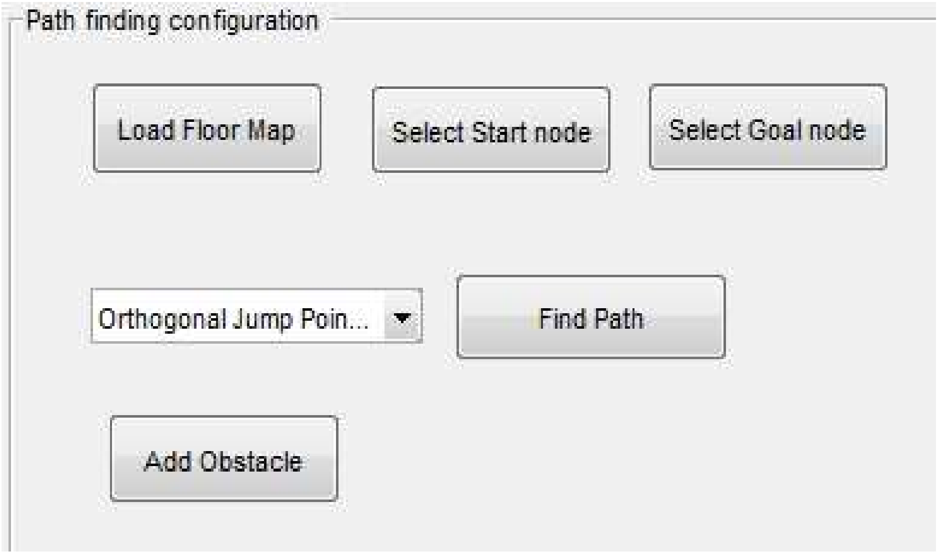


Figure 4.11: Main screen of MATLAB simulation platform

4.5.1. Simulation parameters and metrics

This section presents the details of the simulation parameters and metrics. The algorithms implemented in the simulation parameters included the A*, Dijkstra's, PRM, RRT and Ortho-PATH algorithms in MATLAB version R2016a. The simulations were performed with an inbuilt map simplified into a 2-D grid layout. Table 4.1 describes the simulation parameters.

Table 4.1: Simulation parameters

Grid layout	**180 × 180 (32,400 nodes)** **600 × 300 (180,000 nodes)**
Path finding algorithm	Discrete pathfinding (Dijkstra, A*) Sampling-based (PRM and RRT, nodes varying 100 to 500, Ortho-PATH)
Heuristic function	Neighbouring nodes = 8 Manhattan distance
Performance metrics	Quality of path, execution time, path length, traversed nodes and risk factor

Table 4.1 lists two different grid layouts used for the simulation—one with 180 × 180 grids with 32,400 nodes, and another larger layout of 600 × 600 grids with 180,000 nodes. Manhattan distance with eight neighbouring nodes was used as the heuristic function. The pathfinding algorithms were compared regarding the quality of the generated path, execution time, path length, nodes traversed and risk factor. The research compared the pathfinding algorithms based on features such as high-quality and VI-like paths. A high-quality feature included the length of the path, execution time and nodes traversed as primary categories. Path length and execution

time acted as cost functions to the pathfinding algorithm. Execution time was the total time taken to generate a high-quality path by a pathfinding algorithm. The number of nodes to be traversed depicted the nodes visited along the path. A VI-like path considered the safety of a VI person and was a less risky path with a minimum number of sudden turns. A line-shore path is a path that passes along the walls avoiding the intersections or crossings on the way. The risk and danger percentage factor of a generated path depict the ratio of turns passing through an intersection over the total number of turns.

4.5.2. Simulation results

This section discusses the simulation results of all implemented pathfinding algorithms: A*, Dijkstra's, PRM, RRT and Ortho-PATH. Figure 4.12 presents the path generated by Dijkstra's in blue and A* in red from the source (room location 117C) to destination (room location 103) in the given map (180 × 180). Figure 4.13 shows the path generated by these discrete algorithms with obstacles. Dijkstra's optimality feature outperformed to provide the shortest path in the indoor environment. However, it gave a route passing through an intersection area and had more angular variations to traverse. Low-vision people tend to avoid walking in central or intersection areas. Thus, VI people would choose the A* algorithm, with a safe and feasible path, rather than Dijkstra's algorithm, with an optimal and shorter route.

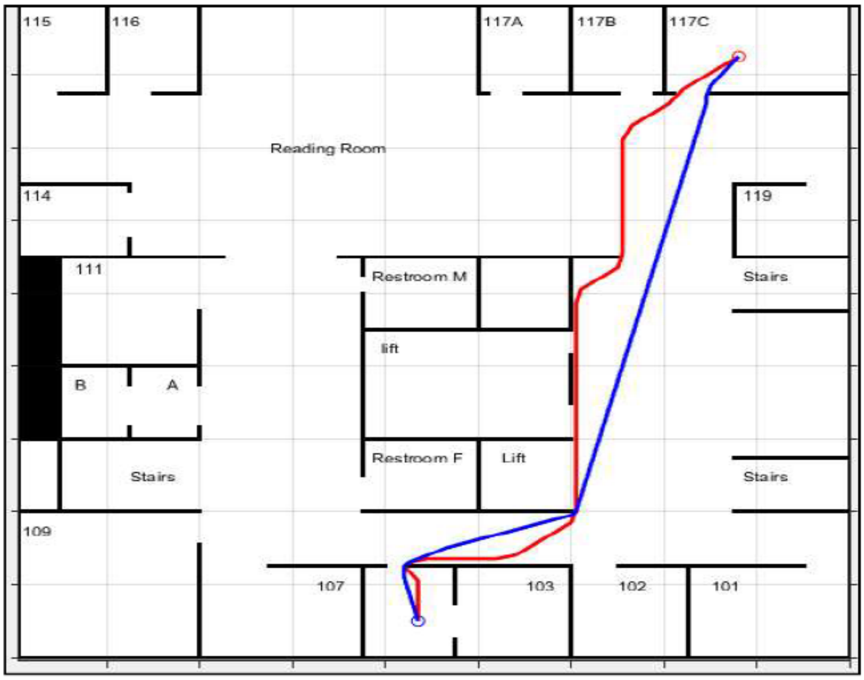


Figure 4.12: Paths generated by Dijkstra's (in blue) and A* (in red) (without obstacles)

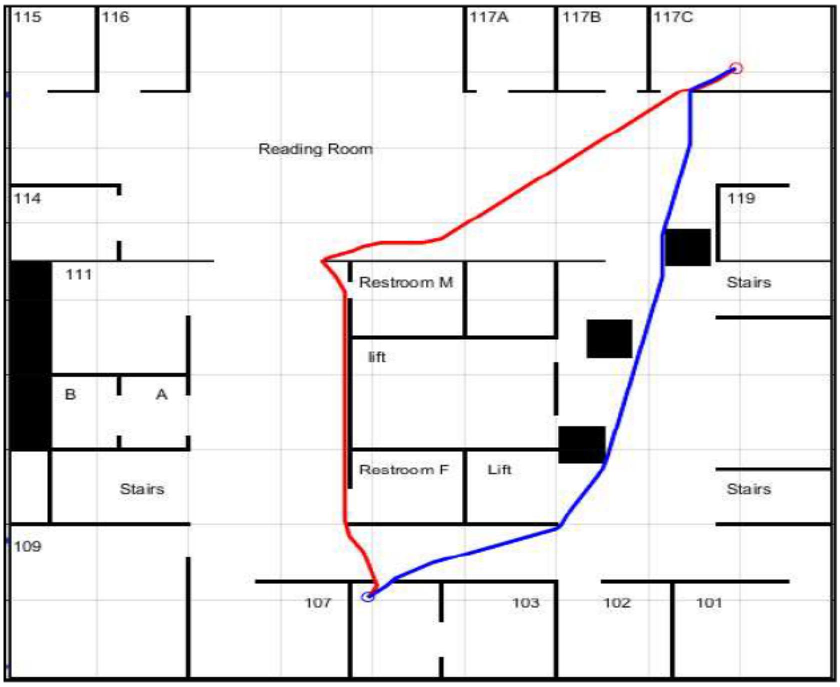


Figure 4.13: Paths generated by Dijkstra's (in blue) and A* (in red) (with obstacles)

Figure 4.14 (a), (b) and (c) show the path generated by PRM with varying numbers of nodes—100, 300 and 500, respectively. The simulation results showed the average computation time and path length, given the variation in the number of nodes from 100 to 500. The quality of the path was compromised by the randomness property of sampling-based algorithms with a smaller number of nodes. The additional number of nodes produces robust and feasible solutions for avoiding obstacles. Of course, the additional number of nodes increases the complexity. However, this increased number of nodes yields better options for improving the quality of paths and providing VI-friendly paths.

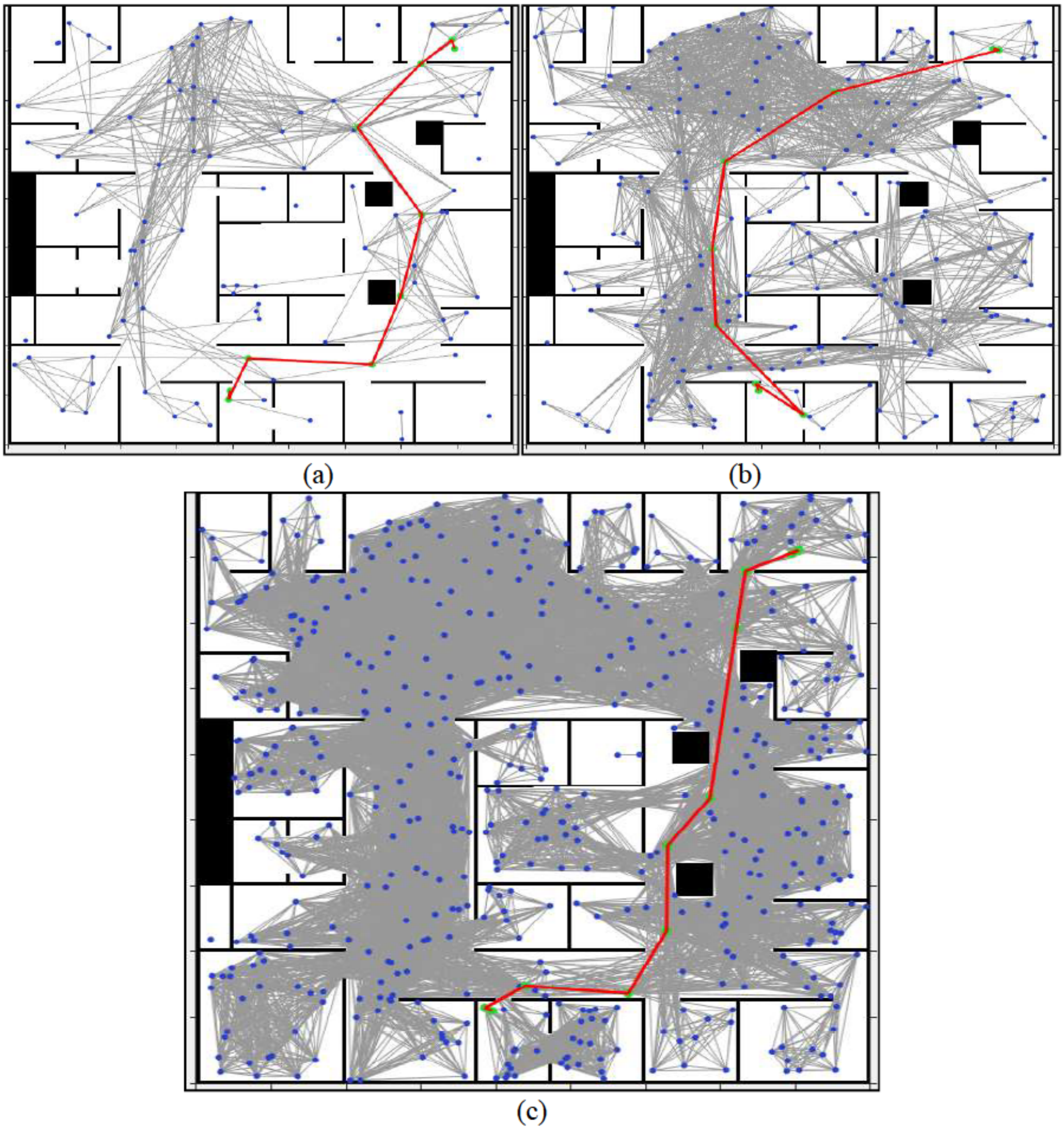


Figure 4.14: Path generated by PRM (in red) with (a) 100 nodes, (b) 300 nodes and (c) 500 nodes (blue dots)

Figure 4.15 presents the quality of a path generated by executing RRT—a sampling-based algorithm—on the same floor plan.

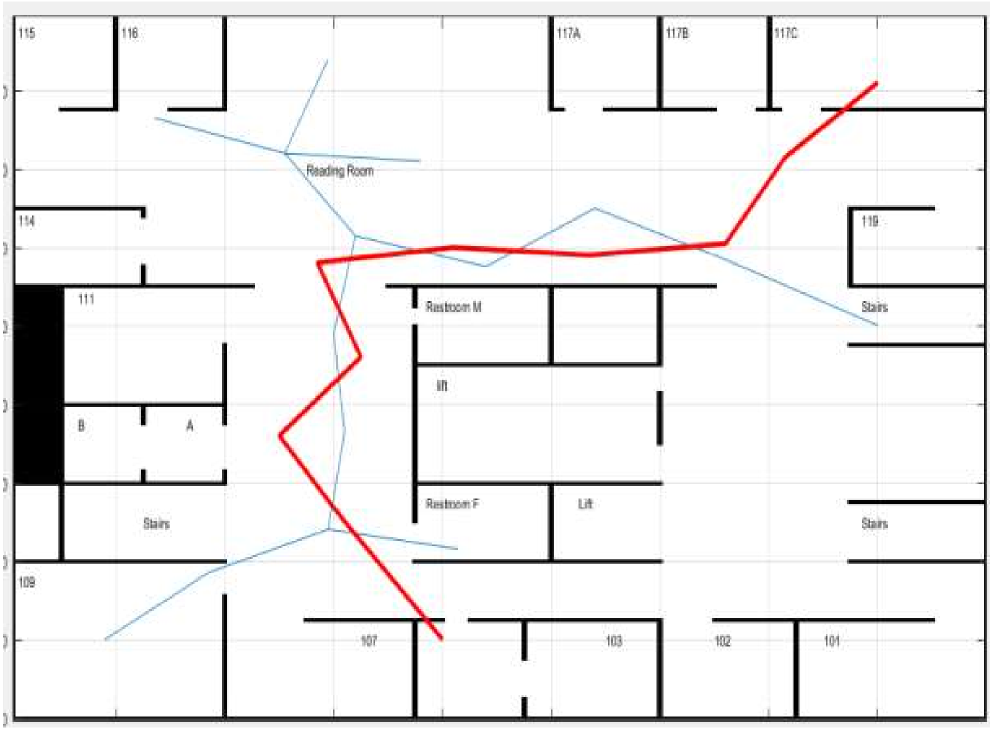


Figure 4.15: Path generated by RRT

The results from Figure 4.15 demonstrate that the path generated with sampling-based methods had many zigzags. Therefore, these methods require post-processing to smooth the path. However, once the initial map was ready, it was accessible to re-query and found a path for another pair of start and goal nodes. The path generated with the algorithms requires processing to guide the agent to follow the suggested route. The post-processing of the path maintains the algorithms to produce optimal and quality paths—that is, retaining smoothness as required by the VI people and suggesting heading and steps mapped along the suggested path. Figure 4.16 presents the orthogonal path generated by the Ortho-PATH pathfinding algorithm.

Figure 4.16: Path generated by Ortho-PATH with jump points

Figure 4.17 presents a combined picture of the paths generated by the A*, RRT, Dijkstra's and Ortho-PATH pathfinding algorithms with the same source and destination. The traditional pathfinding algorithms were tested for a different scenario to make sudden angular turns and heading changes. With such sudden moves, the algorithms must undergo the burden of either pre-processing or post-processing approaches.

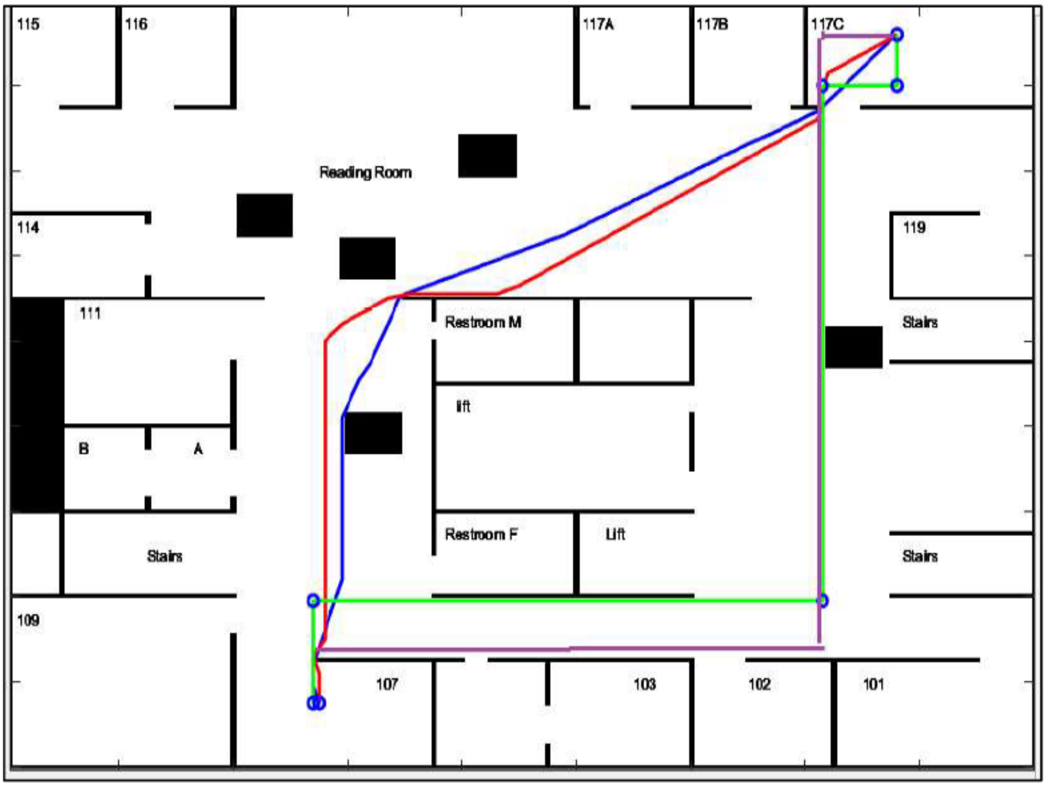


Figure 4.17: Paths generated by A*, RRT, Dijkstra's and Ortho-PATH

Unlike the A* algorithm, Dijkstra's algorithm provided a path passing through an intersection area and with more angular variations to traverse. As aforementioned, low-vision people tend to avoid walking in central or intersection areas. Therefore, a safe and optimal path would be the choice of a VI person. The simulation results showed that randomised pathfinding algorithms, such as RRT and PRM, address path-quality problems. Routes generated by these algorithms may have unwanted motions and be practically unfeasible for a VI person. Ortho-PATH outperformed the other algorithms and generated a more human-like or VI-like path, with only orthogonal moves. It is one of the rare pathfinding algorithms that provides the quality path as the highest priority.

Further, this study evaluated the primary test metrics to obtain a high-quality path, based on the execution time and path length of each algorithm. Path length and execution time are considered cost functions in any algorithm. The algorithms were tested for both environments—with and without obstacles. Figure 4.18 presents the path length of all five pathfinding algorithms. The figure shows that Dijkstra's optimality feature outperformed to provide the shortest path in both scenarios. The paths generated by A* and Ortho-PATH were longer by 5% and 27%, respectively, compared with Dijkstra's algorithm, with obstacles. A* and Dijkstra's provided relatively shorter routes compared with Ortho-PATH.

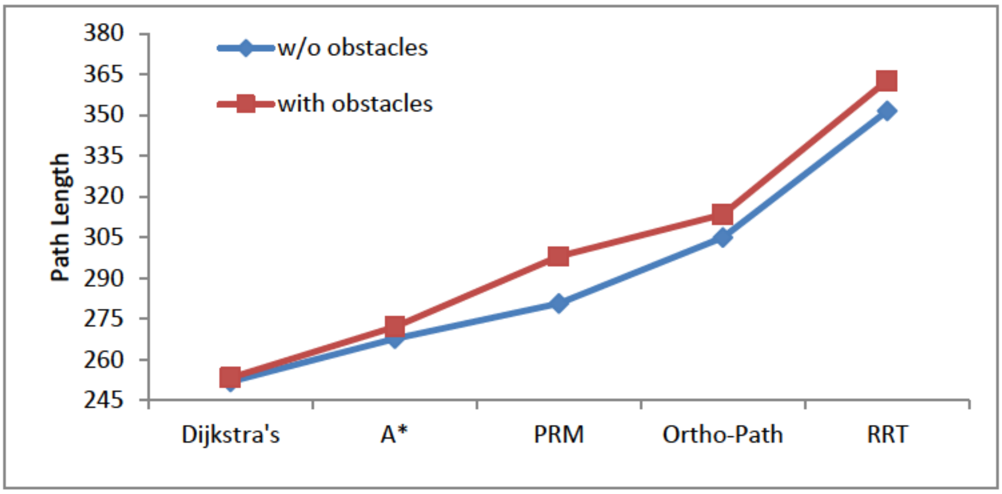


Figure 4.18: Length of paths generated by A*, RRT, Dijkstra's and Ortho-PATH

Discrete pathfinding algorithms can compute an optimal path, yet are limited to low-dimensional spaces that can be discretised without leading to a combinatorial explosion. The proposed simulation results indicate that, as the grid size increased, the execution time for discrete algorithms increased significantly. RRT, given its randomness property, was unsuitable in its naïve form, without pre-processing or post-processing. Ortho-PATH and PRM provided suboptimal results that were almost equivalent. Ortho-PATH computed orthogonal moves, and hence had longer routes. However, the path is favourable to VI people, as there were no diagonal moves.

Figure 4.19 indicates that A* outperformed by computing the path in minimum execution time—approximately five times faster than RRT. For high-dimensional maps, grid-based discrete algorithms, such as Dijkstra's, are impractical solutions because they search the entire space and are computationally expensive. Dijkstra's and PRM pathfinding algorithm maintained an average level of time, yet were slower than Ortho-PATH. The execution time of PRM, RRT and Ortho-PATH includes querying and generating random nodes and evaluating the path's length. A* and Ortho-PATH outperformed in both scenarios without obstacles and with 30% blockage with obstacles. The Ortho-PATH algorithm performs better than the three algorithms, including Dijkstra, PRM and RRT, maintaining a reasonable difference from the A* algorithm.

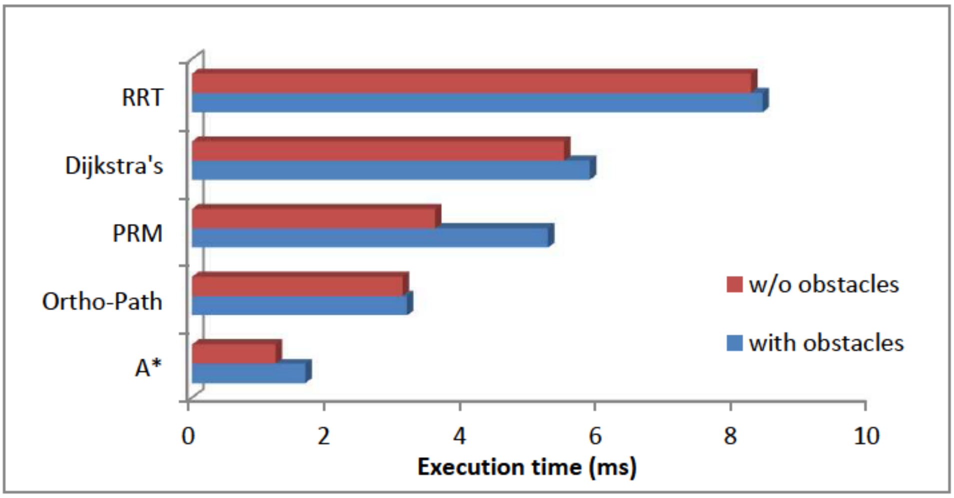


Figure 4.19: Execution time taken by A*, RRT, Dijkstra's and Ortho-PATH

In a uniform grid, it is wasteful to traverse all grid nodes at a time. Dijkstra's algorithm expands outward from the start node until it reaches the goal node. Figure 4.20 indicates that Dijkstra's algorithm traverses the maximum number of nodes. The A* algorithm balances the number of nodes with a selection of neighbouring nodes. Sampling-based algorithms create a graph or tree based on the space, with comparatively few nodes. However, as shown in Figure 4.20, the PRM algorithm failed with few nodes. There is a high chance that the algorithm would not generate a valid path with fewer nodes, even if such a path existed. A naïve uniform sampling-based algorithm requires many sample nodes; hence, it requires more computational time. The Ortho-PATH algorithm outperformed with a minimum number of traversed nodes by reducing the rectangular symmetry.

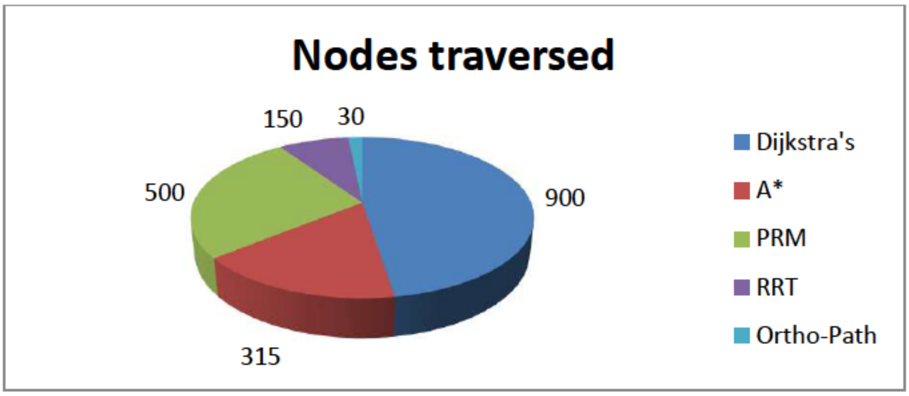


Figure 4.20: Nodes traversed by A*, RRT, Dijkstra's and Ortho-PATH

Figure 4.21 shows the possibility of generating a path with the risk factor of colliding into an unknown obstacle suddenly appearing. Danger and risk percentage were calculated as the ratio of turns passing through the intersection over the total number of turns. Ortho-PATH suggested a path that did not pass any intersection in an open area. Moreover, it offered a line-shore approach along the walls—ideal for VI people. Dijkstra's, RRT and PRM are algorithms that may suggest a path with a high-risk factor, both with and without obstacles.

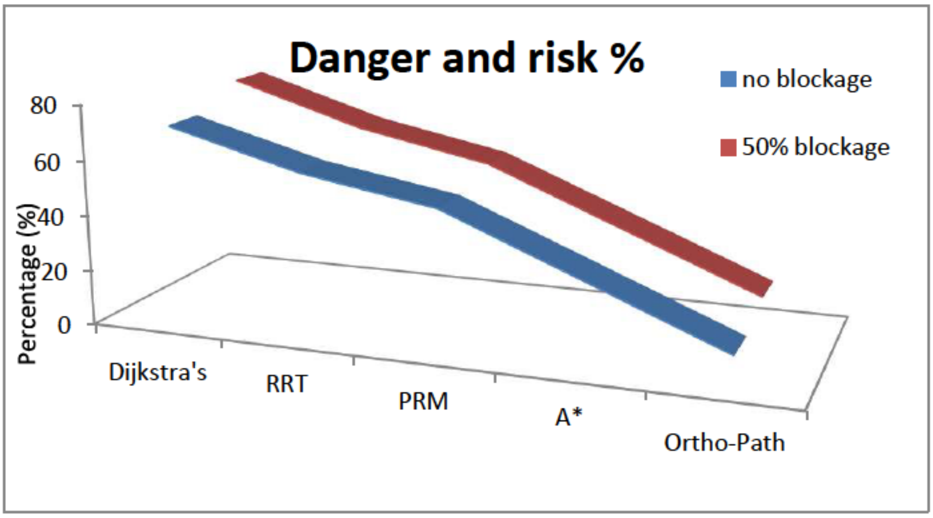


Figure 4.21: Risk and danger percentage involved in executing A*, RRT, Dijkstra's and Ortho-PATH for VI people

Table 4.2 shows the rank-wise analysis of the implemented pathfinding algorithms based on parameters suitable for VI people. For each simulation parameter, based on our results, the algorithms were individually ranked out of five. For example, considering the path length, Dijkstra's algorithm is ranked one out of five based on the minimum length path. The Ortho-PATH and A* algorithm outperformed for four of the criteria, including the minimum number of traversed nodes. However, Dijkstra's algorithm performed best in terms of path length. Based on this, the algorithm is ranked at the top when focusing on path length. The Ortho-PATH algorithm outclassed the other algorithms to generate a more VI-like with only orthogonal moves.

Table 4.2: Rank analysis of pathfinding algorithm for VI people

	VI-like path		**High-quality path**			**Rank**
Algorithm	Line-shore path	Danger and risk %	Path length	Execution time	Nodes traversed	
Dijkstra	3/5	5/5	1/5	4/5	5/5	3.6 ~ 5
A*	2/5	2/5	2/5	1/5	4/5	2.2 ~ 2
PRM	5/5	3/5	3/5	3/5	3/5	3.4 ~ 3
RRT	4/5	4/5	5/5	5/5	2/5	4 ~ 4

Ortho-PATH	1/5	1/5	4/5	2/5	1/5	1.8 ~ 1

4.6. Summary

The pathfinding process plays a vital role in generating a navigation path in an indoor environment. Proper indoor space representation, better heuristic to reach a goal location, and pathfinding algorithm help generate an efficient and optimal path. This chapter has discussed the design of the floor plan and the technique used to obtain obstacle information of the indoor space. Researchers worldwide are working to improve pathfinding algorithms, yet very few have investigated the needs of low-vision people. This chapter has summarised recent progress in pathfinding in an indoor environment by discussing discrete and sampling-based algorithms, with their results in a simulation platform implemented in MATLAB. Considering the limitations of algorithms and requirements of VI people, this work proposed Ortho-PATH—a novel pathfinding algorithm that provides a safe and shore-line path that avoids obstacles.

Discrete-based pathfinding algorithms, both A* and Dijkstra's, provide a suboptimal and optimal path, keeping the user away from static obstacles. Dijkstra's algorithm works well to find the shortest route; however, it wastes time exploring directions that are not promising or goal-oriented. Sampling-based approaches work well for complex environments and high-dimensional configuration space. Unlike RRT, the PRM algorithm may fail to generate maps with narrow gaps, as it does not create nodes in long narrow gaps or between obstacles. This failure arises because of incremental construction of the path and eliminating edges, making connections with obstacles. As a result of the randomness of the sample points in sampling-based algorithms, the generated path makes detours that need to be optimised and post-processed before actual navigation.

This chapter has summarised recent progress in the field of pathfinding in an indoor environment for VI people. It began by discussing the most popularly used pathfinding algorithms, and the simulation results of the algorithms demonstrated their strengths and weaknesses. Their results were analysed and compared with the needs of VI people, such as determining an optimal and safe path with shore-lining. Ortho-PATH outclassed the other pathfinding algorithms and generated a more VI-friendly path with only orthogonal moves. The trajectory path generated by the Ortho-PATH algorithm gave greater priority to safety, as it did not involve any angular movements. The next chapter proposes a novel approach to track and generate the trajectories of a VI person using a handheld smartphone with inertial sensors to help guide the indoor journey.

Chapter 5

BVIP: Indoor Tracking Technique

This chapter proposes an indoor tracking framework using a fusion of iBeacon and inertial sensors from a smartphone. The chapter discusses and demonstrates the implementation and experimental results of the fusion algorithm on the extracted inertial sensor's data of a customised profile for three different users. The framework and fusion algorithm proposed in this chapter ensure that the VI user correctly follows the path provided by the Orth-PATH algorithm.

Some parts of the work reported in this chapter have previously appeared in:
P.T. Mahida, S. Shahrestani, and H. Cheung, "An improved positioning method in a smart building for visually impaired user," *iCIOTRP2019*, Sydney, NSW, Australia, 2019, pp. 7–12.
P. T. Mahida, S. Shahrestani, and H. Cheung, "Indoor positioning framework for visually impaired people using Internet of Things," in *Int. Conf. Sens. Tech.*, Sydney, NSW, Australia, 2019, pp. 198–203.

5.1. Overview

This chapter introduces the framework used in the proposed model for tracking a VI person in an indoor environment. The IoT has become a backbone for autonomous tracking applications that can assist users within IoT-equipped smart buildings. An autonomous and independent tracking system must ensure the correct movements of a VI user in a complex indoor environment. This chapter introduces the proposed novel framework, BVIP, based on inertial sensors—mainly the accelerometer, gyroscope and magnetometer of a smartphone and iBeacon—in a building to help a pedestrian navigate between two locations. Contemporary smartphones have several sensors, including an accelerometer, gyroscope, magnetometer, GPS, gravity sensor, barometer and ambient light sensor [134].

Section 5.2 introduces the applicability and feasibility of using iBeacon and smartphone inertial sensors to track a VI user in an indoor environment. Section 5.3 discusses the proposed BVIP framework and its components for monitoring a VI user in an indoor environment. Section 5.3.1 presents the three phases of the proposed BVIP framework. Given the lack of an inertial sensor-based public dataset mapped with a user's absolute movement positions, an Android-based mobile application is developed to extract the data from the inertial sensor when the user is moving. Section 5.3.2 discusses the adaptive distance estimation algorithm, heading inference and turn detection algorithm applied on the generated dataset for three users. Section 5.4 demonstrates the emulation results and experiments to evaluate the proposed framework and fusion algorithm's performance. The section discusses the experimental results and walking trace tests, which show position estimation with 1.5 to 2 metres of mean positioning error. Finally, Section 5.5 summarises the chapter.

5.2. Introduction

The primary issue unresolved in indoor navigation is the difficulty in accurately estimating a moving user's location. Location-aware services have rapidly grown with the increased use of IoT devices in smart buildings [123]. Most navigation solutions require a structured environment with IoT devices, such as proximity sensors, RFIDs and smartphones [14], [135]. These IoT based systems can sense, predict and estimate the surrounding environment and communicate to the connected IoT devices over the Internet [7]. As discussed in Chapter 3, this study proposes a VI-friendly, safe and obstacle-free path for a VI person [136] with beacons placed on doorways. This approach demonstrates that external beacon sensors, when placed near the doors/entrances of a building, can provide a VI-friendly path with the minimum number of resource-extensive beacons. However, there is a need for a convenient device that can detect the motion of a VI person and overcome the challenge of navigating independently between landmarks/points along the provided path. Following a given path and becoming lost on the suggested way is common among VI people while traversing a complex building. In addition, a VI person may encounter difficulty following a route in an open space, such as a large hallway, given the unavailability of external physical devices due to increased infrastructure cost and flexibility.

Nowadays, MEMS technologies have developed energy-efficient and affordable inertial measurement systems in many consumer electronic products, including smartwatches, smartphones, tablets, gaming systems and wearable sensors. The rapid growth of low-cost and energy-efficient MEMS sensors has enabled rapid growth in applications related to fitness, emergency fall detection, entertainment and indoor navigation [137]. Indoor tracking is a technique to determine the position, velocity and altitude of a moving object with respect to a known reference [138]. A recent survey shows that 28% of the works published in IEEE International Conference on Indoor Positioning and Indoor Navigation uses inertial sensors for positioning and navigation in an indoor environment [139]. Table 5.1 discusses the popular technologies used for indoor positioning of VI users with the best factors regarding accuracy, coverage and cost.

Table 5.1: Comparison of indoor positioning systems [88]

Technology	Accuracy	Coverage	Cost
Wi-Fi	5–15 m	Building	Low
RFID	1–5 m	Room	High
IR	5–50 m	Room	High
UWB	15 cm	Building	High
Bluetooth/iBeacon	30 cm to 5 m	Building	Low
Inertial sensor	2 m	–	Low

Indoor Positioning systems using Wi-Fi, RFID, IR or UWB technologies require specific infrastructure, incurring the increased overall cost. To minimize cost in infrastructure-based environments and supporting the increased growth of sensors have boosted research interest in developing applications through built-in sensor’s function. An inertial-based navigation system (INSs) comprises an accelerometer, gyroscope, magnetometer, barometer, and proximity sensors are available in smartphones. However, most INS systems have used an accelerometer and gyroscope to monitor spatial position, which is insufficient to locate a mobile user [140]. Given the small size and low weight of inertial sensors on a mobile phone, they suffer large drifts and inaccuracies with increasing measurement time. The increasingly significant drift in the inertial sensor values affects the tracking system's performance and may generate erroneous information [141]. Therefore, there is a need to employ fusion strategies to correct the sensor deviations and improve results.Most of the literature uses accelerometer and gyroscope signals to estimate position indoors [80]. However, our approach involves a unique fusion of gyroscope, accelerometer and magnetometer with an IoT device (iBeacon) to track movement indoors.

The proposed framework suggests deploying the Bluetooth-based iBeacon to generate a smart indoor environment to provide absolute position information. MEMS, including accelerometer, gyroscope and magnetometer, acts as a relative positioning system in areas where external beacon signals are weak. This suggested work combines the absolute beacon positioning and relative positioning using adaptive step length, with heading and turns to reduce the drift in positioning error. The approach customises the user's profile with their measured step length and mode. This process uses adaptive threshold-based step detection to estimate the distance travelled by a VI user. Based on each user's evaluated adaptive threshold, the approach combines the tracked distance with the user's turns and headings. Based on prior knowledge of the initial position, the proposed fusion algorithm can estimate the user's positional status and can act as a practical, robust component to track a VI user's movement in an end-to-end indoor navigation system.

5.3. Proposed algorithm for tracking

This section presents a novel approach, BVIP, which combines learning techniques on MEMS and beacon-based positioning to locate a VI user in an indoor environment. The three stages of the proposed approach include data pre-processing and feature extraction, learning model and fusion algorithm, and position inference. Figure 5.1 displays the BVIP technique for an indoor tracking system using IoT devices and smartphone inertial sensors. BVIP integrates relative positioning from an inertial sensor in a smartphone and absolute positioning based on optimally located beacons/marker points. The BVIP system helps track the primary navigation and movement activities of a VI user. A person's walking movement is a specific type of periodic mechanical movement, defined as an individual step initiated with a heading angle followed by different orientations. A user with normal eyesight visualises the path to the destination in their brain. The user then follows the visualised path from a start location, with a specific distance and heading direction concerning the goal location. Correcting their imaginary vision with landmarks, the user steps forward in a particular direction and follows the intermediate landmarks/milestones to reach the destination accurately. The proposed approach is similar to the concept where optimally positioned marker points serve as landmarks to correct navigation errors.

The technique's three phases comprise data pre-processing and feature extraction, learning model and fusion algorithm, and position inference. Data pre-processing and feature extraction take input from the three axes of inertial sensors, including the accelerometer, gyroscope and magnetometer of a smartphone, to calibrate and filter it. The first phase extracts and pre-processes the inertial sensor data of a smartphone and the information from the iBeacon to extract the features of the data. The extracted data are passed to the learning model and fusion algorithm with beacon information in the second stage. The data are processed to calculate the absolute and relative position of the VI user in the second phase. Finally, the third phase calculates the estimated position of the user. This study collected and integrated sensor data periodically for a constantly moving VI user to estimate their position.

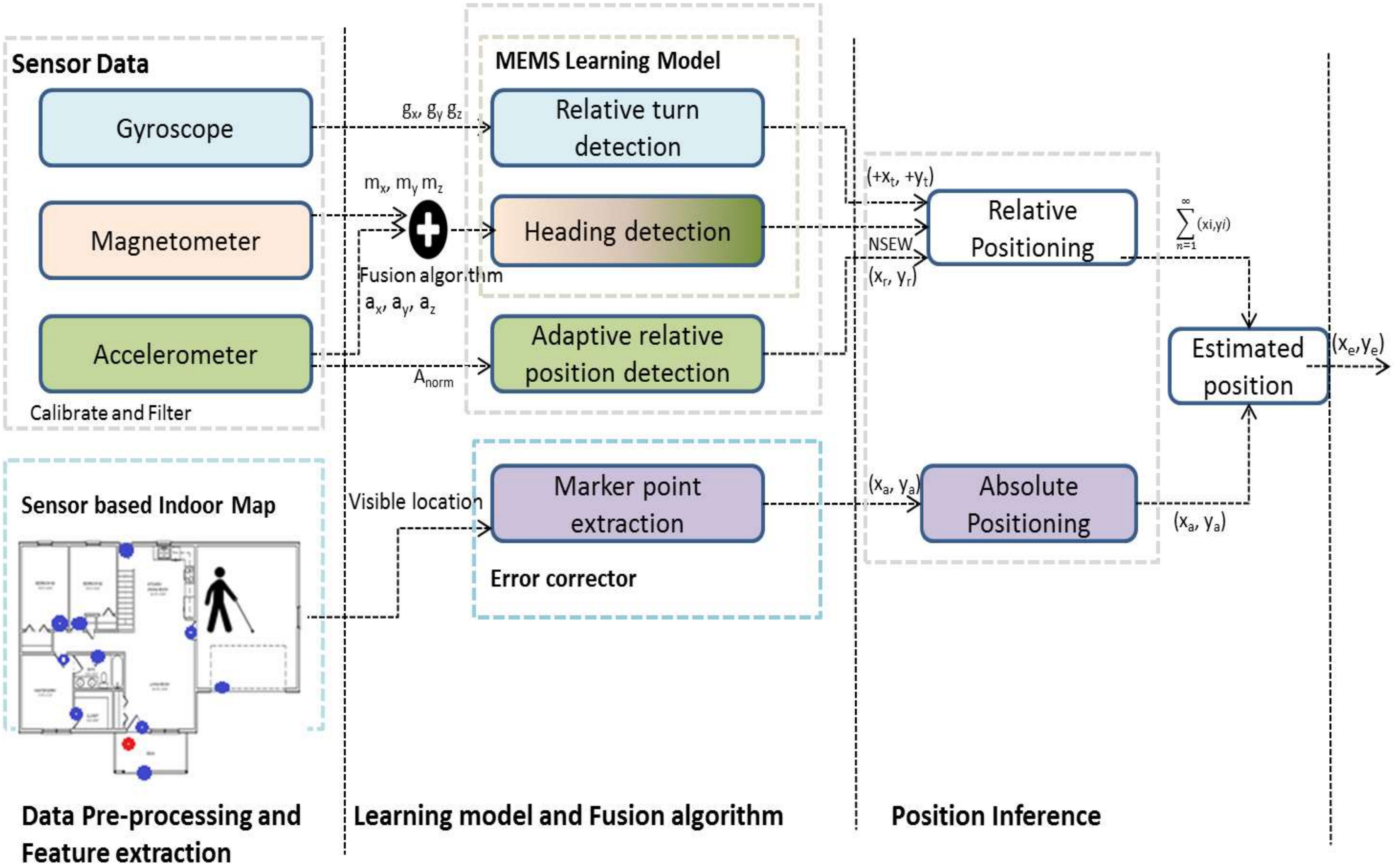
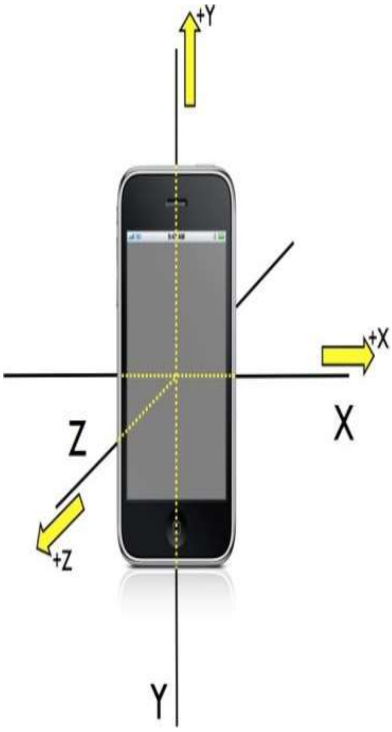


Figure 5.1: BVIP approach for indoor tracking of VI user [142]

Figure 5.2 (a) shows the accelerometer axes of a smartphone (x, y and z-axis), and Figure 5.2 (b) is a graphical representation of the three-axes accelerometer measurements (A_x, A_y and A_z). The accelerometer is a sensor that measures linear acceleration generated by the movement of an object and the change of the velocity holding the sensor. It measures the external force at the centre of mass. When the smartphone is at rest, the acceleration force may be the continuous force of gravity acting on the device (A_x and A_y). The linear acceleration is measured when the attached device is in motion to sense movements or vibrations [143].



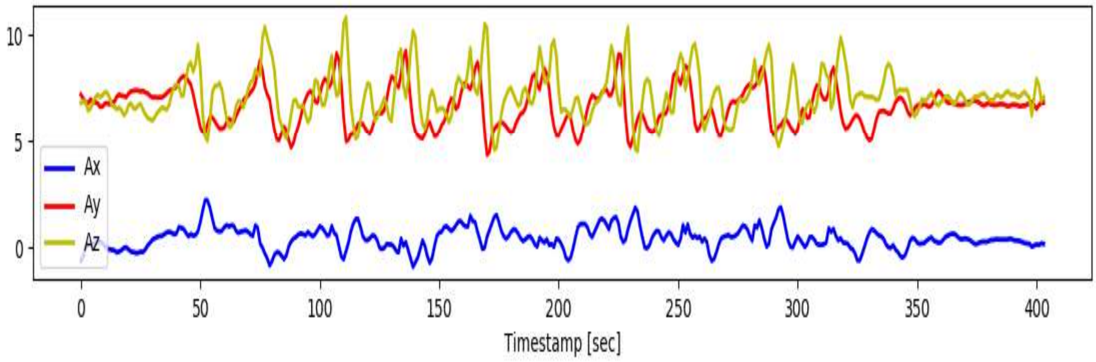


Figure 5.2: (a) Accelerometer axes of smartphone and (b) graphical representation of x-, y- and z-axes of accelerometer

MEMS gyroscopes are devices mounted on smartphones and smartwatches that can sense the earth's gravity to determine the body's orientation to which they are attached. They generally use a vibrating mechanical element as a sensing element to detect angular velocity. Smartphones have three-axis gyroscopes that deliver rotation values in each of the three axes (G_x, G_y and G_z). Rotation values are negative or positive, based on the direction of the rotation. The magnetometer is a sensor used to measure the magnetic field's strength and direction in the vicinity of the device in three axes (M_x, M_y and M_z) [144]. In aeronautics, the magnetometer can measure the position of an aircraft body, such as an aeroplane or satellite. A marine magnetometer detects the position of shipwrecks and other submerged objects. Magnetometers have the characteristics of being able to work under severe and limited conditions. Therefore, the magnetometer can be used as a digital compass to obtain all four directions (north, south, east and west) headings in indoor environments [145].

5.3.1. Data pre-processing and feature extraction

The data pre-processing stage processes data gathered from the three different sensors—the accelerometer, gyroscope and magnetometer of a smartphone. The raw data are pre-processed and fed as input to the next phase of the BVIP indoor navigation system. Feature selection and extraction are critical to identify and remove unneeded, irrelevant and redundant features from collected inertial sensor data and create an accurate predictive model. This stage helps in the selection of information required to incur knowledge from the original filtered signals. To estimate the relative position, the client device (a smartphone) continuously sends the sensor data to the Cloud device. A systematic step detection algorithm with known orientation and heading is required. MEMS sensors help evaluate the relative distance travelled, direction and heading of the connected smartphone.

Given the lack of recordings of accelerometer, gyroscope and magnetometer sensors, this study collected data on three different users with different moving styles and speeds. The dataset included 50 sets of data for each user, with the three axes of inertial sensor signals. The sample values of azimuth, pitch and roll and three-axis accelerometer, magnetometer and gyroscope readings are shown in Table 5.2.

Table 5.2: Sample readings of data collected using Android-based application

acc_X	acc_Y	acc_Z	mag_X	mag_Y	mag_Z	Azimuth	Pitch	Roll	gyro_X	gyro_Y	gyro_Z
-0.656	2.41	9.078	-16	14.2	-28.2	40.72	-14.825	-3.992	-0.68951	-1.05994	0.368591
0.148	2.602	9.48	-16	13.6	-28.2	42.962	-12.337	-4.027	-0.02515	-0.61026	0.315186
0.474	2.468	9.202	-16.4	13	-28.2	37.137	-14.982	2.842	0.301697	-0.20224	0.336548
0.052	3.282	9.135	-16.4	11.8	-28.2	38.238	-19.758	0.3	-0.04758	0.095764	0.395294
-0.474	2.966	8.8	-16.4	11.8	-28.2	39.383	-16.733	0.947	-0.67242	0.406586	0.497833
-0.561	2.487	9.346	-16.8	11.8	-28.2	44.676	-14.876	-3.309	-1.00674	0.292297	0.479675
-0.589	2.583	10.064	-16.8	12.6	-28.2	42.278	-14.361	-1.279	-0.93732	0.170532	0.41452
-0.292	2.544	9.997	-16.8	13	-27.8	42.171	-14.272	-1.615	-0.68951	-1.05994	0.368591
0.11	2.477	9.413	-16.8	13	-27.8	39.762	-15.828	0.237	-0.02515	-0.61026	0.315186
0.11	2.707	8.608	-16.8	13.9	-27.8	37.375	-17.446	0.688	0.301697	-0.20224	0.336548
0.215	2.611	8.915	-16.8	14.9	-27.8	36.179	-16.324	1.325	-0.04758	0.095764	0.395294
-0.388	2.19	9.537	-16.8	15.7	-27.8	37.32	-14.881	-0.369	-0.67242	0.406586	0.497833
-0.178	2.295	9.154	-16.8	15.7	-27.8	38.272	-14.067	-1.069	-1.00674	0.292297	0.479675

In most instances, when a device is sitting on a table and not accelerating, the accelerometer reads a magnitude of $g = 9.81$ m/s^2. When a device is in free-fall and accelerates towards the ground at 9.81 m/ s^2, its accelerometer reads a magnitude of 0 m/ s^2. Equation (5.1) [137] depicts the simplified error model of the acceleration output measurement of the i^{th} axis (where i = x, y, z).

$$A_i = a_i + B_f + n_i + S_i a_i \tag{5.1}$$

where A_i is the acceleration output for the i^{th} axis, a_i is the acceleration value applied along the i^{th} axis, S_i is the scale factor error (usually presented as a polynomial to include the non-linear effects), B_f is the zero-offset bias of the measurement, and n_i is the random noise.

Table 5.3 shows a comparison of the three-axis accelerometer data in different positions of the Android smartphone. The gravitational acceleration is usually read as -g towards earth, given the construction of an accelerometer. The ideal values represent the theoretical values expected for the $A(t_i)$ in each position of an Android smartphone. The real-time values represent our reading in their given position. The values indicate that the generated data have noise.

Table 5.3: Accelerometer readings in different smartphone positions

Smartphone position	Ax (m/s^2)		A_y (m/s^2)		A_z (m/s^2)	
	Ideal value	Real-time values	Ideal value	Real-time values	Ideal value	Real-time values
Up	0	(0.010, 0.115)	9.81	(9.700, 9.899)	0	(0.283, 0.410)
Down	0	(0.010, 0.115)	-9.81	(-9.700, -9.899)	0	(0.283, 0.410)
Left	9.81	(9.700, 9.899)	0	(0.010, 0.110)	0	(0.283, 0.410)
Right	-9.81	(-9.70, -9.899)	0	(0.010, 0.110)	0	(0.283, 0.410)
Front up	0	(0.010, 0.115)	0	(0.283, 0.410)	9.81	(9.70, 9.899)

Instead of using the individual measurements of A_x, A_y and A_z, the process calculates normalised accelerometer A_{norm}. The advantage of using the A_{norm} versus any axis is that it is impartial to the device's orientation and can handle dynamic directions. Equation (5.2) presents the formula to evaluate normalised accelerometer A_{norm}:

$$A_{norm} = \sqrt{{A_x}^2 + {A_y}^2 + {A_z}^2} \tag{5.2}$$

where A_{norm} is the normalised magnitude of accelerometer and A_i is the i^{th} axis of three-axis accelerometer.

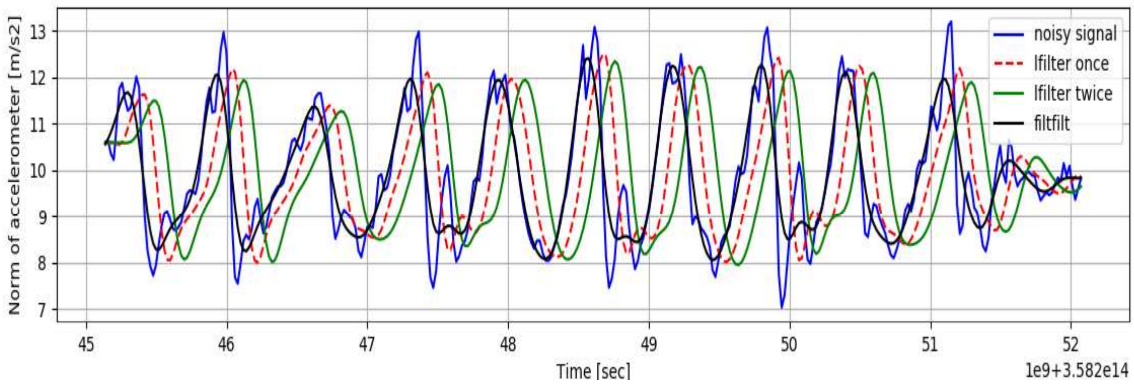


Figure 5.3: Unfiltered vs filtered accelerometer magnitude using low pass filter [142]

As discussed earlier, the accelerometer measurements suffer from time-varying biases and noise. The noise leads to an unreliable measure, known as drift. Therefore, it is not advisable to rely on the raw measurements of a smartphone's accelerometer signals. A digital filter can remove the high-frequency noise from raw accelerometer data. The Butterworth filter's frequency response rolls out the higher frequencies beyond the threshold limit down to zero. This study also applied an interactive Fourier filter and zero-phase digital filtering technique to remove noise. Figure 5.3 shows the noisy and filtered accelerometer signals over a span of time. The blue signal is the original noisy signal, while red, green and black represent the iFilter signal applied once and twice, and filtfilt. It shows that the output of filtered signal using iFilter (once and twice) introduced delay. Zero-phase filtering using the filtfilt function compensated the effect of phase distortion.

The gyroscope sensor in the smartphone measures the rate of rotation by detecting tiny shifts in pulses' timing arriving at a sensor. Figure 5.4 shows the output of gyroscope z-axis data for left and right turns. The gyroscope was unstable, as the low angular velocities were not registered on the sensor. It would be impossible to verify a gyroscope reading when the orientation of the smartphone changed slowly over time. Thus, a fourth-order low pass filter was applied to the gyroscope (G_x, G_y and G_z) data to estimate the turns a VI person takes during movement.

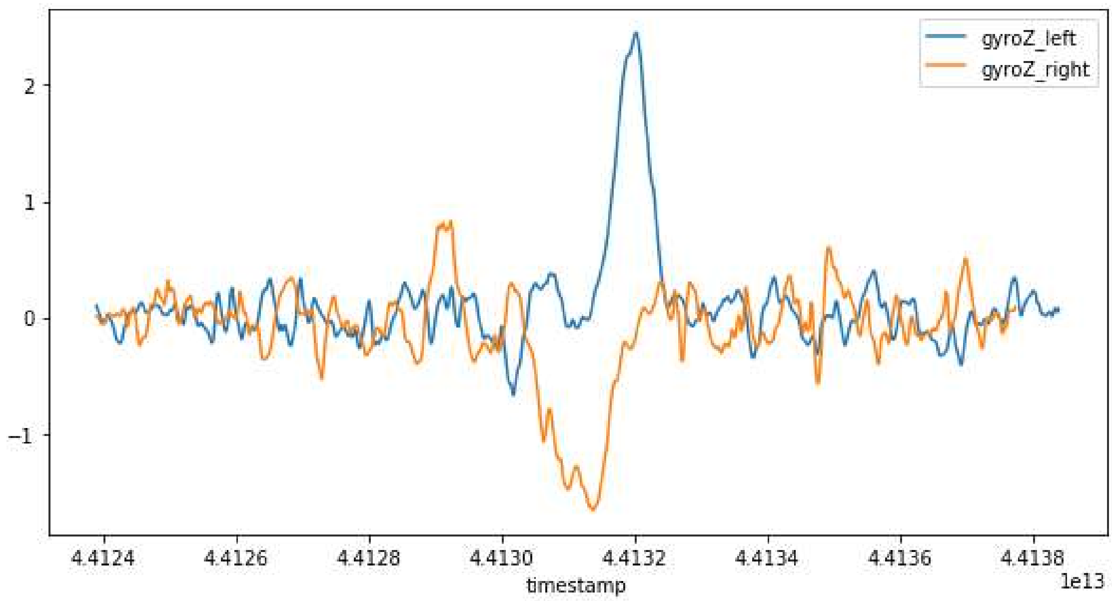


Figure 5.4: Gyroscope data for different turns [146]

To extract important features in the filtered three-axis gyroscope, data were input to the MATLAB decision tree-based supervised machine learning algorithm. Accuracy, training time and prediction speed were evaluated to predict an important feature for orientation estimation, as shown in Figure 5.5 (a). The figure depicts the precision of the z-axis measurement of the gyroscope, G_z, is a critical feature to estimate the user's orientation. Our results in Section 5.3.2.3 further support this finding. Based on the selected features, advanced features are extracted that incurs knowledge and information used in the indoor positioning system, as shown in Figure 5.5 (b).

Feature	Accuracy (%)	Training time (sec)	Prediction speed (obs/sec)
G_x	88.9	1.1273	49000
G_y	91.6	1.0784	46000
G_z	99.9	0.9933	48000

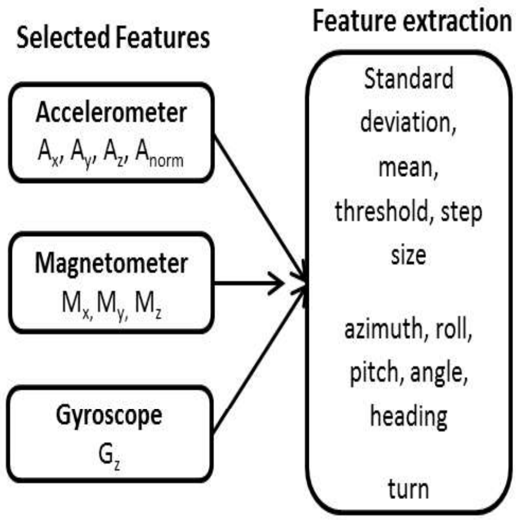


Figure 5.5: (a) Feature selection and (b) list of extracted features [142]

Figure 5.5 (b) shows the features extracted from inertial sensors to estimate a VI person's indoor position. Gyroscope data helped predict the orientation/turns of the user. The direction of the user's movement holding a smartphone can be determined via roll, pitch and yaw angles from magnetometer data. These angles help predict the device's azimuth angle, knowing the change in the device's orientation from the original position. Therefore, advanced features—such as step size, orientation, standby position, movement, roll, pitch, yaw, heading angle and relative turn—were extracted to estimate the user's position. The angle at which the user is heading can be evaluated using the three-axis magnetometer measurements. Azimuth angle/heading and turn were fed as a feature into the step detection algorithm to predict and ensure the user's correct movement.

The extracted features acted as input to the MEMS-based learning model, discussed in the next section. First, each user's profile is extracted and maintained as a training phase by recording three-axis inertial sensor data based on the user's walking pattern. The model expected to know essential features, such as the standard deviation and mean of the user's step size from the accelerometer readings. Based on this activity, the adaptive threshold features were estimated to evaluate the user's actual distance, as discussed in Section 5.3.2.1.

5.3.2. Micro-electromechanical system learning model and evaluation results

This section discusses an improved positioning technique based on an inertial sensor of a smartphone. The fusion algorithm consisted of three phases: adaptive and relative distance estimation, heading inference, and turn detection algorithms.

5.3.2.1. An adaptive distance estimation algorithm

This subsection proposes an adaptive and relative distance estimation algorithm with a smartphone's accelerometer sensors. An accelerometer is a sensor that can measure both static and dynamic acceleration forces. By measuring dynamic acceleration, the sensor can analyse the way the device is moving. To detect the distance travelled, it is necessary to know the steps taken by the user. The algorithm adapts to time-varying changes in normalised accelerometer data, Anorm, to evaluate steps travelled. Every user has a different walking pattern with varying step sizes. The proposed distance estimation algorithm is shown in Figure 5.6. The algorithm includes further steps, including adaptive threshold extraction, step decision algorithm, average step length estimation, and relative distance estimation.

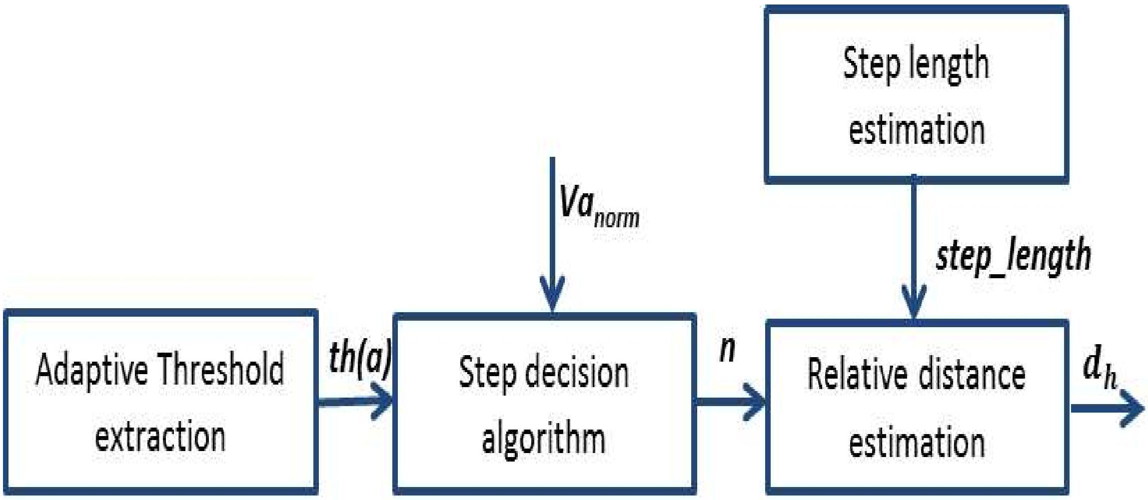


Figure 5.6: Proposed adaptive distance estimation algorithm

Adaptive threshold extraction: Conventional pre-configured fixed threshold-based step estimation algorithms may provide accurate results for the same user [80]. However, different walking styles may vary in their peak values; therefore, step detection based on peak values cannot estimate accurate steps, given varying peak values in different activities. Figure 5.7 shows various normalised accelerometer threshold reading patterns collected for detecting a step phase over some time. The figure shows five thresholds—the top, upper, lower, bottom and adaptive threshold for three steps in the period. The threshold's top and bottom are labelled based on the highest and lowest value in the reading. The next step's upper and lower threshold values are slightly less than the same user's first step. These thresholds are not standard values, but represent diversity in the thresholds for a user.

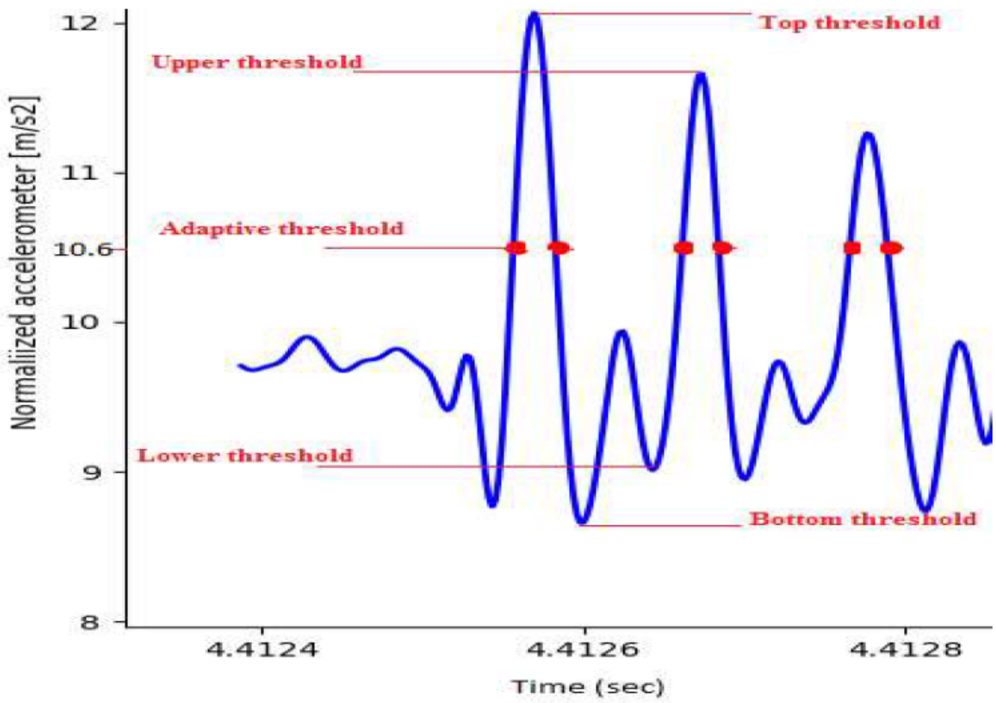


Figure 5.7: Different thresholds for a user

There are three phases of a step: (a) lifting, (b) swinging and (3) stepping. The lifting phase of the step involves acceleration towards the gravity direction, which causes lower threshold values. The swinging phase is the transient phase between the lifting and stepping phases. The swinging phase is detected when the current phase is the lifting phase, and the acceleration value passes upward through the lower threshold. The stepping phase leads to a high threshold, as the acceleration is in the direction opposite to gravity. The threshold value evaluated varies based on the step size and length of the user. Each user may have a different threshold value based on their walking pattern. A real-time training phase calculates the threshold of the user while walking for a predefined time. A proper training phase evaluates the adaptive threshold for each user, as illustrated in Figure 5.8. For the same user in a different context, the threshold value $Th(a)$ needs to be updated periodically. Equation (5.3) calculates the adaptive threshold $Th(a)$ for each user:

$$Th(a) = \frac{\sum_{i=1}^{m} A_i}{m} + \sqrt{\frac{\sum_{i=1}^{m} (A_i - \bar{A})^2}{m-1}} \tag{5.3}$$

where $\bar{A}$ represents the mean value of m readings of normalised accelerometer data A_{norm} shown in Equation (5.2) and A_i is the normalised accelerometer value at the i^{th} axis. The system extracts the user's movement pattern by estimating the adaptive threshold.

Step decision algorithm: The adaptive threshold *Th(a)* for a user is set as an input to the step decision algorithm. The algorithm counts the number of times the normalised accelerometer readings cross the adaptive threshold *Th(a)*. Figure 5.8 is a pictorial representation of normalised accelerometer magnitude A_{norm} over a short period, with several spikes that have changed in the signal measurement, including valid and fake ones. The spike data have different peak/maxima and valley/minima values. Each red dot in Figure 5.8 represents a crossing of the adaptive threshold *Th(a)*, evaluated based on Equation (5.3) and the actual A_{norm} values. However, three valid spikes cross an adaptive threshold *Th(a)* for each user.

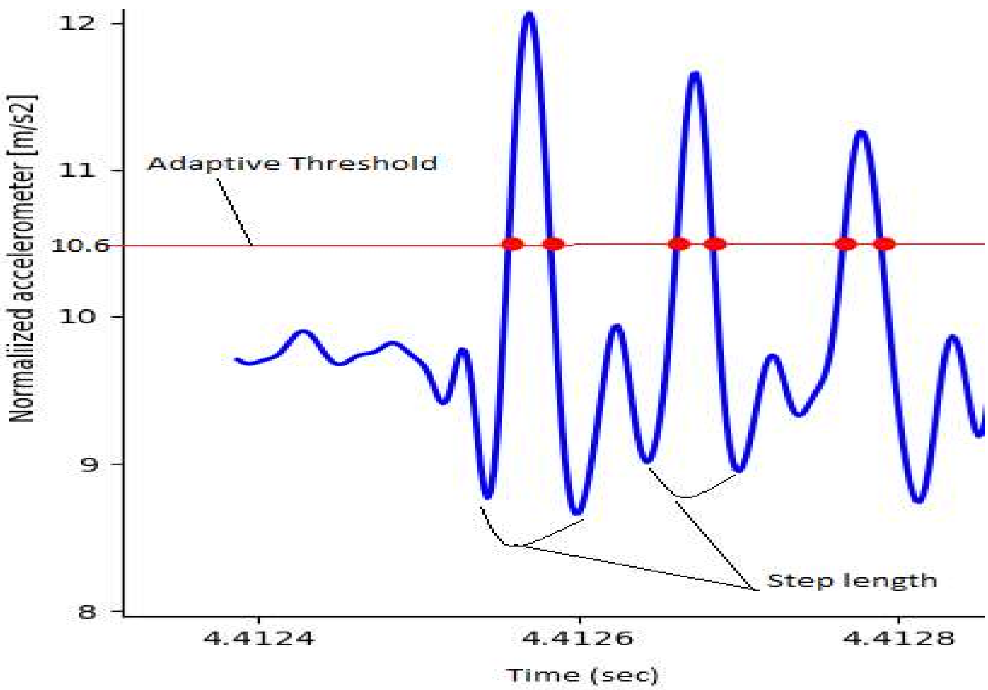


Figure 5.8: Representation of adaptive threshold and step length

Given the total number of crossings ($threshold_{crossings}$) for a walking user from an initial starting location, the number of steps, *n_steps*, for each user is equal to the total number of crossings divided by two, as depicted in Equation (5.4):

$$n_{steps} = threshold_{crossings}/2 \tag{5.4}$$

Using Python's code snippet, as shown in Figure 5.9, *n_steps* is calculated and stored in an array for the specific user based on the adaptive threshold and the given accelerometer data at a specified timestamp.

```
def step_decision_adaptive_threshold(data, timestamps, threshold):
    last_state = 'below'
    # below - less than threshold
    # above - above the threshold
    crest_troughs = 0
    crossings = []

    for i, datum in enumerate(data):
        current_state = last_state
        if datum < threshold:
            current_state = 'below'
        elif datum > threshold:
            current_state = 'above'
        if current_state is not last_state:
            if current_state is 'above':
                crossing = [timestamps[i], threshold]
                crossings.append(crossing)
            else:
                crossing = [timestamps[i], threshold]
                crossings.append(crossing)

            crest_troughs += 1
        last_state = current_state
    return np.array(crossings)
```

Figure 5.9: Snippet of step detection based on adaptive threshold

Step length estimation: The step length is the distance travelled by a single leg that counts the heel print from one foot to the other foot's heel print during a walking stride. Our approach's average step length ($step_{length}$) was predetermined from the training set for a user. The average $step_{length}$ is the difference between the pair of consecutive valleys or the minimum value of a spike, as shown in Equation (5.5):

$$step_{length} = \frac{\sum_{i=1}^{m}(v_{i+1} - v_i)}{m} \tag{5.5}$$

Relative distance estimation: With the computed number of steps, distance displacement is provided for the user over time for each moving step. Distance d_h travelled by a user in a specific heading h is evaluated by Equation (5.6):

$$d_h = n_{steps} * step_{length} \tag{5.6}$$

where d_h is the distance travelled by the user in direction h, n_steps denotes the number of steps computed by step detection algorithm, and $step_{length}$ represents the average value of each user's step length. Figure 5.10 shows the snippet code to calculate the estimated distance travelled based on the step size of the given user.

```
def evaluate_distance(title, gps_data, actual_distances, steps, est_dists):
    act_dests = []
    act_dest_cum = []
    act_dest_sum = vincenty((0, 0), (0, 0))

    est_dests = []
    est_dest_cum = []
    est_dest_sum = vincenty((0, 0), (0, 0))

    for i in range(0, len(gps_data) - 1):
        ts = gps_data[i]['milli_ts']
        act_dest = actual_distances[i]
        act_dests.append([ts, act_dest.meters])
        act_dest_sum += act_dest
        act_dest_cum.append([ts, act_dest_sum.miles])

    print(len(steps), len(est_dists))
    for step, est_dest in zip(steps, est_dists):
        ts = step[0]
        est_dests.append([ts, est_dest.meters])
        est_dest_sum += est_dest
        est_dest_cum.append([ts, est_dest_sum.miles])
# Actual
    act_dest_cum = np.array(act_dest_cum)
    plt.plot(act_dest_cum.T[0], act_dest_cum.T[1], 'b-', linewidth=2)
    # Estimated
    est_dest_cum = np.array(est_dest_cum)
    plt.plot(est_dest_cum.T[0], est_dest_cum.T[1], 'r-', linewidth=2)
```

Figure 5.10: Snippet to calculate distance

5.3.2.2. Heading inference algorithm

For a low-vision person moving in an unknown indoor environment, it is essential to know where they are heading. A magnetometer in a smartphone can measure the earth's magnetic field and detect a user's heading. However, it is not easy to know if the device's angle is horizontal or vertical, or both, with magnetic field readings. The proposed heading measurement approach is based on the fusion of magnetometer and accelerometer to overcome this challenge. An accelerometer can measure the static acceleration force to find the angle at which the device is tilted with respect to the earth. Therefore, using the fusion of the magnetometer and accelerometer reading can provide the user's heading/orientation relative to the device's orientation. Figure 5.11 represents the local earth magnetic field *H*, with a fixed component H_h pointing to the earth's magnetic north. It represents the mathematical model of an approach used to estimate the heading that involves measuring the two orthogonal components of the magnetic vector, H_x and H_y. The magnetic compass heading, or yaw, or azimuth *H* angle can be determined from H_x and H_y.

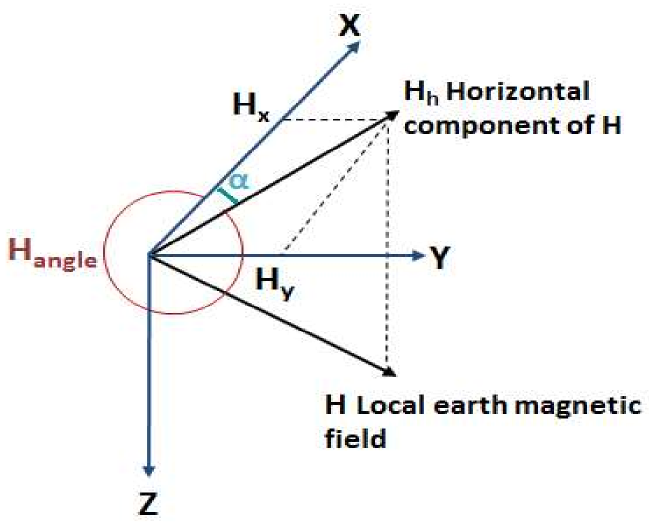


Figure 5.11: Heading inference

Roll angle (θ) and pitch angle (ϕ) are defined as rotation angles around the x- and y-axis, respectively. The pitch and roll angles in the position of Figure 5.11 are evaluated as 0'. Using a calibrated magnetic sensor, the heading angle H_{angle} can be measured using the two components M_x and M_y, where $H_x = M_x$ and $H_y = M_y$, as in Equation (5.7):

$$H_{angle\ (in\ radians)} = \tan^{-1}\left(\frac{H_x}{H_y}\right) \tag{5.7}$$

Equation (5.7) is valid only if the magnetic sensor embedded in a smartphone is precisely levelled. Magnetic sensors work accurately when they are flat and horizontal to the ground [145]. Any other positions, such as holding straight or tilted on an angle, make the reading inaccurate. This problem occurs because the approach only considers the x- and y-axis of the earth's magnetic field. Therefore, when the magnetic sensor is not parallel to this axis, the amount of magnetism felt will change based on the alignment of the sensor.

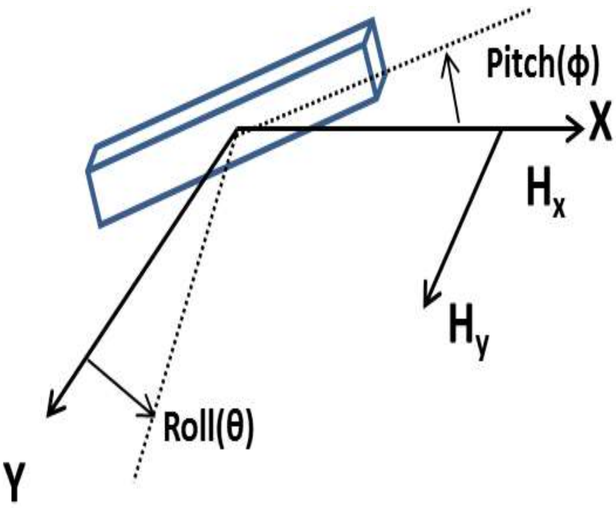


Figure 5.12: Heading inference with tilt compensation

Given that VI users may not hold the smartphone in their hands, the H_x and H_y measured values might change. Therefore, additional information about the compass space orientation is required. The x- and y-axis of the magnetic sensor alone do not help calculate correct heading values in varying device positions. It is essential to know how much the device has tilted, and there is a need to integrate the M_z axis measurement and accelerometer. We must know the orientation by incorporating the three-axis accelerometer into our heading inference system. The pitch and roll angles are not 0' with the tilted device, as shown in Figure 5.12. Based on the three-axis accelerometer values, H's local vector coordinates are converted to a reference coordinate to measure the heading. Roll and pitch angles are measured by a three-axis measurement of the accelerometer using Equations (5.8) and (5.9):

$$\theta = \tan^{-1}\left(\frac{A_x}{A_z}\right) \tag{5.8}$$

$$\phi = \tan^{-1}\left(\frac{A_y}{A_x * \sin(\theta) + A_z \cos(\theta)}\right) \tag{5.9}$$

$$H_x = M_x * \cos(\phi) + M_z * \sin(\phi) \tag{5.10}$$

$$H_y = M_y * \cos(\theta) + M_x * \sin(\theta) * \sin(\phi) - M_z * \sin(\theta) * \cos(\phi) \tag{5.11}$$

where θ and ϕ are the roll and pitch angles, respectively, evaluated from the three axes (A_x, A_y and A_z) of an accelerometer. The values of H_x and H_y are evaluated using Equations (5.8) and (5.9) in Equations (5.10) and (5.11). The values of H_x and H_y from Equations (5.10) and (5.11) are added to Equation (5.12) to estimate heading angle H_{angle} in a tilted smartphone position:

$$H_{angle\ (in\ degrees)} = \tan^{-1}\left(\frac{H_x}{H_y}\right) * \frac{180}{\pi} \tag{5.12}$$

$$\alpha = \left[\begin{cases} 360 - H_{angle}\, , for\ H_{angle} < 0 \\ H_{angle} \qquad\quad , for\ H_{angle} > 0 \end{cases}\right] \tag{5.13}$$

where α is the azimuth angle of the VI person holding the smartphone in any position. Equation (5.12) converts the H angle into degrees. Then, Equation (5.13) evaluates the azimuth angle α. Based on the estimated angle α, the user's direction and azimuth angle at each step are measured. Figure 5.13 presents the mapping of the azimuth angle (in degrees) to direction. The integration of tilt compensation with accelerometer and magnetometer data can predict a moving or steady VI user's accurate heading in an indoor environment.

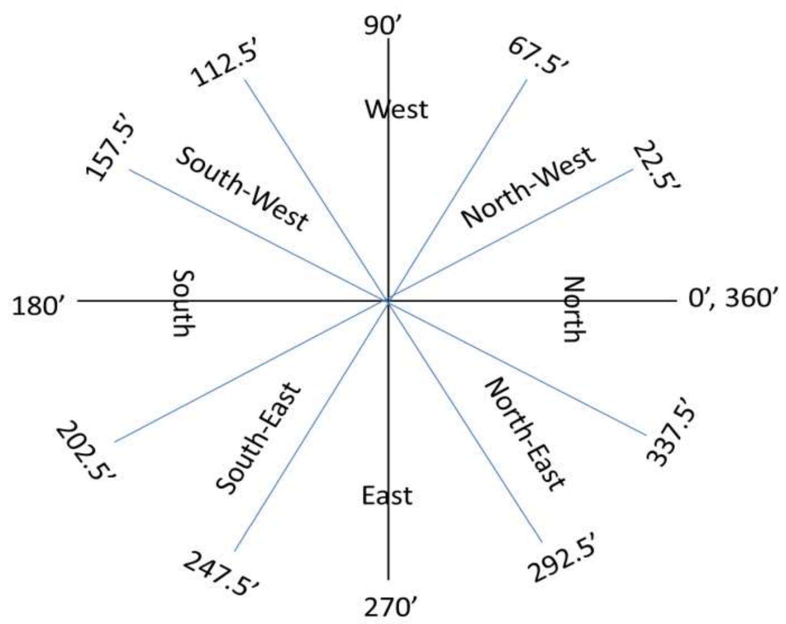


Figure 5.13: Azimuth—degree to direction

5.3.2.3. Turn detection algorithm

The smartphone measures the user's movement parameters: moving or steady position, steps travelled, heading and turn/orientation (if any). Based on the path-planning algorithm [136], the algorithm proposes a VIP-friendly path. To track the user indoors if they are following the given route, our system depends on determining the change point or the time the user takes a turn. The gyroscope sensor in the smartphone is the device that measures the rate of rotation by detecting tiny shifts in the timing of pulses arriving at the gyroscope sensor. This process of turn determination acts as a validation of the heading measurement presented in Section 5.3.2.2. The critical issue in the process is determining landmarks or points at which the user takes a turn. Machine learning techniques, including support vector machine (SVM), KNN, linear discriminant and classification decision tree methods, were applied to design the prediction models. Figure 5.14 presents the accuracy of these machine learning techniques when applied to the gyroscope data.

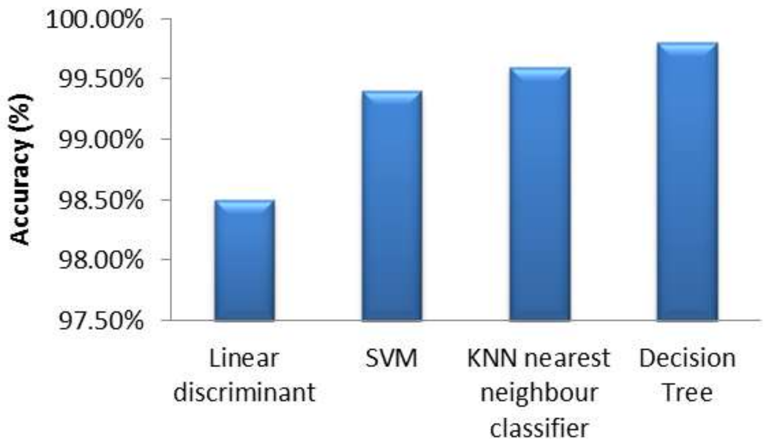


Figure 5.14: Performance of machine learning techniques on gyroscope data

Based on the performance results, a decision tree was applied to build a tree with a set of hierarchical decisions to predict orientation, as shown in Figure 5.15. The figure presents the rule format generated of the proposed Classification and Regression Tree (CART) decision tree algorithm using the Graphviz library. A decision tree in machine learning involves building tree nodes and branches to help make hierarchical decisions to generate the result. Each tree has nodes representing the unique features, while each link or branch of the tree represents a rule, and the leaf represents the result or outcome. The decision made by this machine learning technique aims to achieve high classification accuracy.

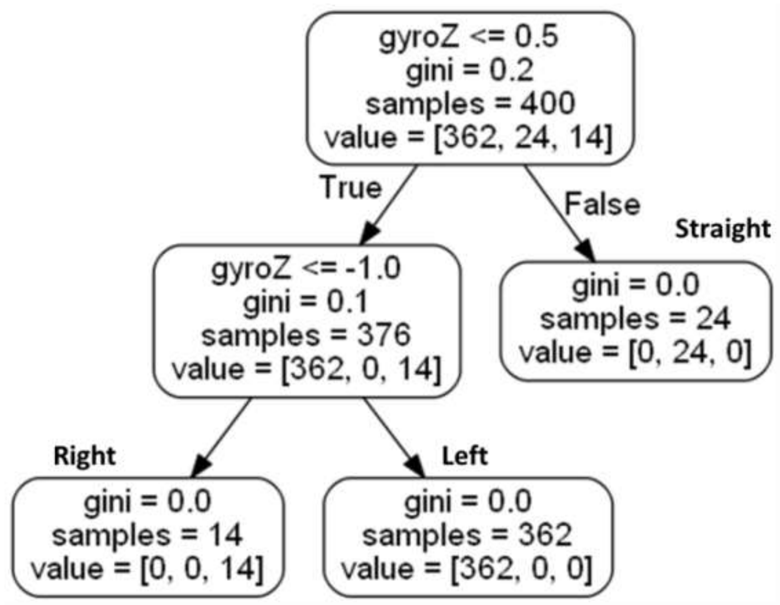


Figure 5.15: Classification decision tree for turn detection

This model helps understand how the gyroscopes z-axis decides the turn taken by the user holding the device. The parameters used to build the model are the maximum depth of the tree, the minimum number of samples to split and the Gini index. Gini impurity measures how often a randomly chosen element from the set would be incorrectly labelled. Supervised learning relies on training samples to achieve higher accuracy [147]. For this experiment, 400 samples of training data were used. The z-axis of gyroscope data frames the classification tree to detect the turns made by the user. The turn estimation algorithm detects all the turns with their respective moves (left, right, straight, forward or backward).

5.3.2.4. Fusion algorithm

The primary goal in developing a fusion algorithm is to provide location information and track a VI user in an indoor environment. Sections 5.3.2.1, 5.3.2.2 and 5.3.2.3 discussed the algorithm individually. As discussed in Section 5.3.1, the proposed system obtains the user's initial location via iBeacon. The beacons are placed at specific landmark locations, such as the entry/exit points of a room, stairs, elevators and landmarks. The beacons are configured before deployment to create a network connected to the Cloud server that stores the absolute location of the beacons mapped with the unique beacon identification parameter. When in proximity to a beacon, the smartphone finds the user's absolute location with respect to the environment. The BVIP system deployed on the smartphone, scans and detects Bluetooth signals to locate the absolute position of the user. This section discusses the absolute positioning and relative positioning-based learning model with the appropriate sequence diagram. The sequence diagram describes the flow of messages and the message content between the entity (smartphone, Cloud server and database server). Figure 5.16 illustrates the sequence diagram of the learning model in this approach.

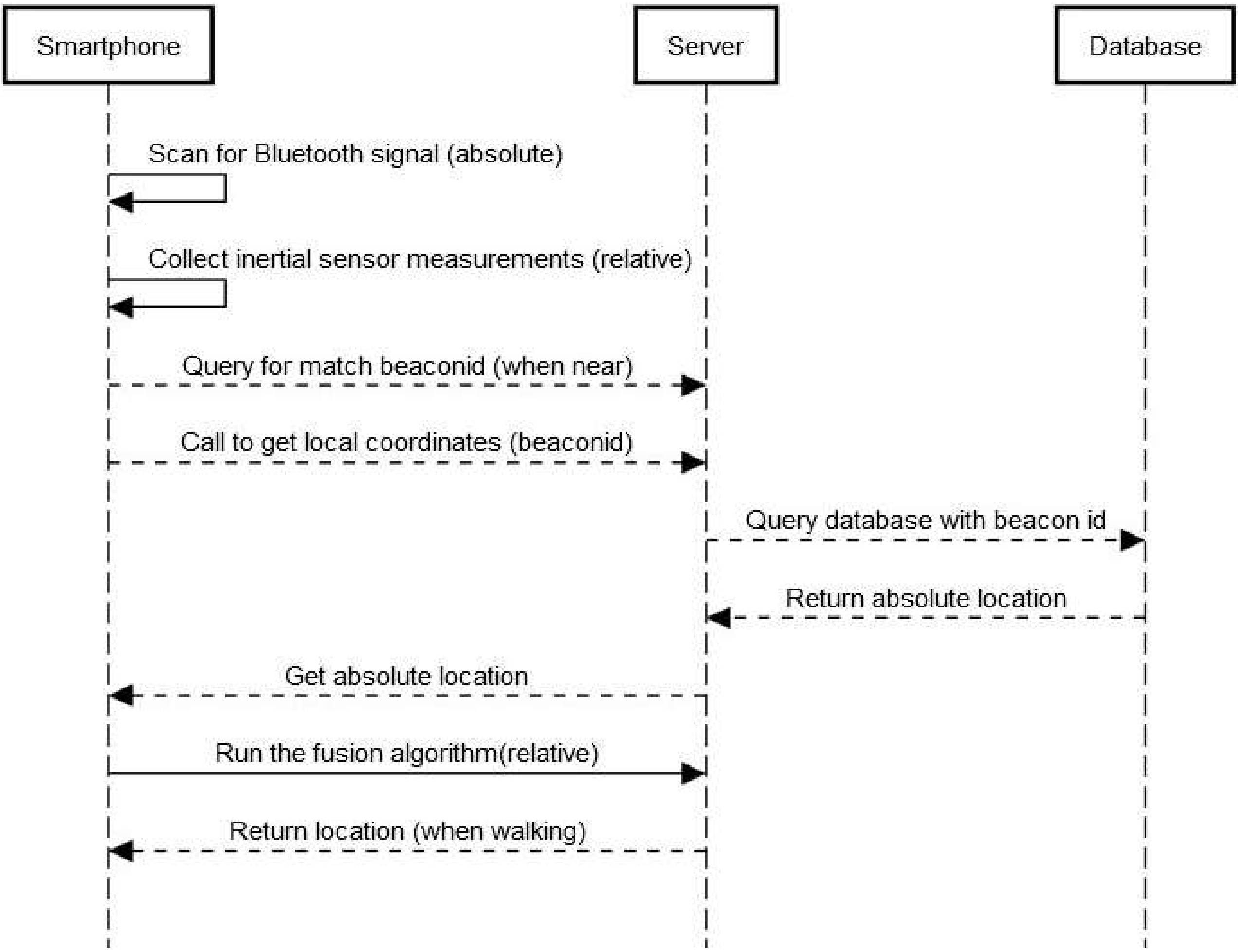


Figure 5.16: Sequence diagram of BVIP technique

The learning phase starts with the repeating loop of scanning the iBeacon Bluetooth to estimate the user's absolute position with a smartphone. The smartphone also continuously measures the reading of the inertial sensors of a smartphone to estimate relative position in situations where no beacon signals are available. If the Bluetooth beacon signal is found, the smartphone queries the Cloud server for the strongest *beacon_id* to obtain the absolute location of the moving user. The Cloud server queries the database server and returns the local coordinates of the indoor space based on the scanned *beacon_id*. The process of obtaining the absolute location is only executed at the start of the navigation or to verify the correct landmarks after a predefined time. Later, the fusion algorithm with calibrated MEMS measurements of the smartphone determines the position of the VI person in a building. The readings of the inertial sensors are continuously scanned and read to estimate the relative position of the moving user with the fusion algorithm.

The fusion algorithm helps overcome the difficulty when the user is in a location outside the beacon range. The fusion algorithm works to estimate the relative position with minimal beacon devices in a building. One of the options to improve this situation is to increase the number of beacons in multiple locations. However, this option increases the resource cost, including message exchange between the beacons and smartphones. For hollow areas, an additional system could be applied to estimate the location of the moving user. Therefore, this study proposes a fusion algorithm that fuses the data incurred from a travelled distance, heading and turns to estimate the current position of a VI user with respect to the known absolute location from beacons.

Algorithm 5.1: BVIP fusion algorithm

```
Input:  A, G,M, (x_a,y_a)
A, G, M ← dataAcquireandFilter()
step_length, Th(A) ← getStepParameters()
        α ← findInitialHeading(A, M)
        S_s ← getStepCount (A)
for each turn t:
       if(direction(t)==direction (h)
            D_s(s)   ← getDistance(S_s)
            h(s)     ← estimateHeading(A,M)
            t(s)     ← estimateTurn(G)
P_e ← generateFinalPosition(D_s,t,h,P_i)
```

Algorithm 5.1 is the fusion algorithm for positioning the user in an indoor environment from the inertial sensor measurements. The position is estimated from distance d travelled, turns t and heading information h, from the accelerometer A, gyroscope G and magnetometer M readings at a fixed frequency. The accelerometer A, magnetometer M and gyroscope G samples are acquired as the user moves in an indoor environment. The noises caused by low-cost sensor interference, such as frequency introduced by side movements, should be eliminated. The absolute initial position (x_a, y_a) from the beacon is fed as input to the algorithm. Different users may have a different walking style, leading to a different threshold *Th(A)* and length of step, *step_length*. The suggested algorithm feeds the calculated threshold and steps length to determine the distance travelled by the VI user. These values are updated at a regular interval for a specific profile and stored in the Cloud.

The function *findInitialHeading()* helps find the initial heading angle α of the VI user. As the user moves, the method *getStepCount()* evaluates whether a step event occurred and detects the steps moved by the user in the α direction from the filtered accelerometer A readings. It is unnecessary for the system to estimate the heading constantly. To be more concrete, the heading needs to be calculated whenever the user makes a turn. Steps 4 to 8 are repeated to for each turn the user makes to calculate the travelled distance and heading. Whenever the user makes a turn, the distance travelled by the user is evaluated. To verify the feasibility of the heading estimation method with varying smartphone positions, the heading direction based on the turn taken by the user is estimated. The current position of the VI person is estimated by the last known sensor location (x_a, y_a), travelled distance d, i turns with t directions, and α heading angle. The known location of the VI user is denoted by (x_s, y_s). The estimated position Equation (5.14) can be modelled as the function of d, h and t. The path-planning algorithm Ortho-PATH [136] discussed in section 4.4 suggests orthogonal paths that make either horizontal or vertical movements of the user. The proposed system BVIP discussed in this chapter helps to track the user whether he/she is moving on the suggested path:

$$(x_e, y_e) = (x_a, y_a) + \sum_{i=1}^{n} f(d, \alpha, t_i) \tag{5.14}$$

$where\ f(d, \alpha, t_i) = \{(d\sin(\alpha), d\cos(\alpha))\}$, and d, α and t represent the distance, azimuth angle and turns travelled by the user, respectively.

Figure 5.17 presents the pictorial flow of Algorithm 5.1 with intermediate positions evaluated based on turns taken by the user. The figure represents the floor plan of a building with a beacon (red dots) placed at the entrance of each room. The beacons are the visible marker points, as represented in the technique. These beacons provide the absolute location of the user with respect to the local coordinates. The position (16, 2) in the figure is the start absolute location of the user. The living area in the floorplan has no beacons, and the fusion algorithm estimates the position of the user with respect to the absolute start location. The intermediate relative positions evaluated by the algorithm are marked as (0, +14) and (+7, 0). The relative position is the evaluated movement in the respective axis with the respective turns. The turns in the algorithm are verified with the estimated heading movement of the user. The proposed algorithm ideal movement calculation is represented in the flow of the algorithm.

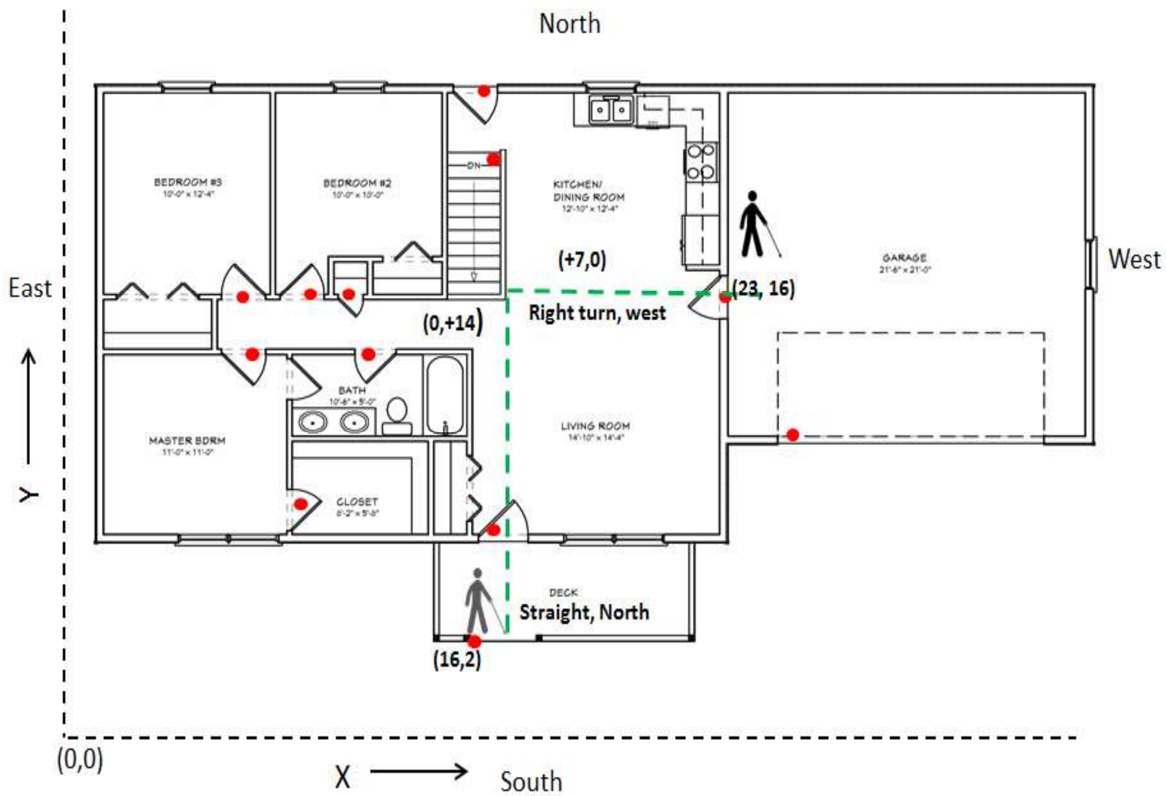


Figure 5.17: Flow of fusion algorithm

5.4. Emulation, experiments and results

A thorough performance analysis was conducted in the experiments by comparing the proposed system BVIP with commercial products. An app was developed and installed on an Android smartphone to evaluate the number of steps travelled in a specific direction by a user. The inertial sensor data were collected at the sampling frequency of 20 Hz. The app developed in Android creates a .csv file that contains the timestamp and the raw data of the sensors. Walking experiments on three participants with different walking styles were conducted, which gathered almost 100 sample datasets. Each sub-stage for step detection, heading and turn detection was individually tested to evaluate the performance of the individual algorithms and sensors. Then, the fusion algorithm was implemented and used to calculate the position of the user.

The step and distance estimation algorithm discussed in Section 5.3.2.1 was implemented with calibrated and filtered accelerometer, gyroscope and magnetometer sensor data. Each participant walked by following pre-specified paths that comprised 10, 15, 20, 25, 30, 35, 40, 45 and 50 steps in a straight line during this experiment. Table 5.4 presents the actual steps and average error estimated by the proposed adaptive step detection algorithm for the three participants (u1, u2 and u3).

Table 5.4: Root mean squared error (RMSE) for users (u1, u2 and u3)

Actual steps	RMSE (u1)	RMSE (u2)	RMSE (u3)
10	0	0	0
20	1	1	0
30	2	0	1
40	0	1	2
50	1	2	1

The accuracy of the experimental data was determined by calculating the RMSE with Equation (5.15):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(PS_i - SA_i)^2}{n}} \tag{5.15}$$

where n represents the walking experiments conducted by each user, PS_i denotes the number of predicted steps by the proposed algorithm and SA_i denotes the actual steps. RMSE for u1, u2 and u3 was calculated as 1.9, 2.0 and 2.1, respectively. Typically, the step length of a person walking with average speed varies between 0.50 and 0.70 metres. The overall distance RMSE was calculated as 0.65 metres.

As discussed in Section 5.3.2.2, the heading inference algorithm was thoroughly tested for the correct heading directions of a walking user. Figure 5.18 presents the heading angles calculated by the proposed heading inference method in blue, with the actual heading in orange. There were instances of a minor deviation in the heading angle; however, this may not affect evaluating azimuth direction. This research evaluated the heading inference algorithm with 100 samples of gyroscope data, which achieved an accuracy of 99.99% with the proposed heading inference algorithm.

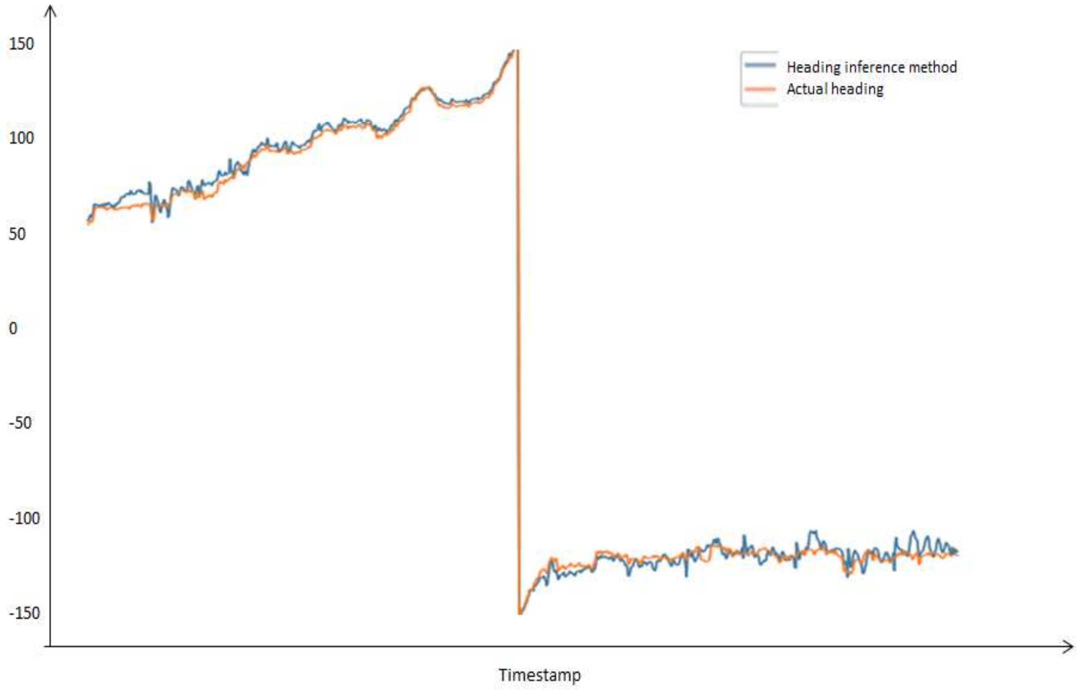


Figure 5.18: Heading inference for actual heading vs heading inference method

Detecting turns plays a vital role in predicting the absolute position of the user. As discussed in Section 5.3.2.3, the results of the prediction model of gyroscope data were implemented and tested. Figure 5.19 presents the scatter plot of results achieved by applying the decision tree prediction model on the gyroscope data. The prediction model was implemented to detect three classes: 0 (straight), 1 (left turn) and 2 (right turn). The proposed approach achieved a classification accuracy of 99.99%. The approach used in this study provides satisfactory performance for detecting turns.

After implementing and testing the algorithms individually, the fusion algorithm proposed in Section 5.3.2.4 to track the walking user was tested. Figure 5.20 displays a screenshot of the output of the fusion algorithm in an Android application to demonstrate the working of the system with a person holding the smartphone.

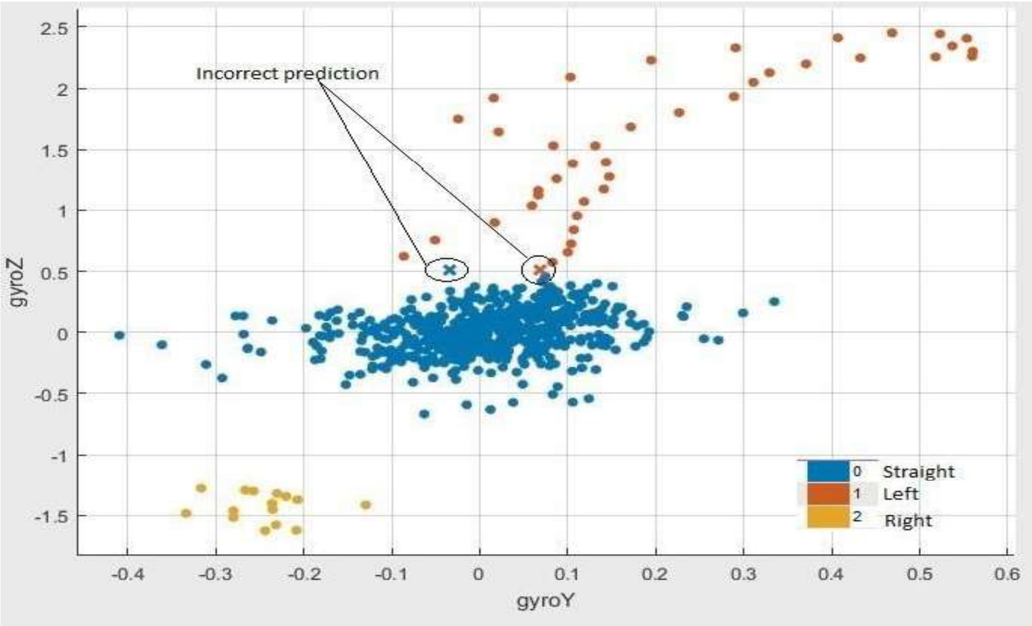


Figure 5.19: Result of prediction model on gyroscope data

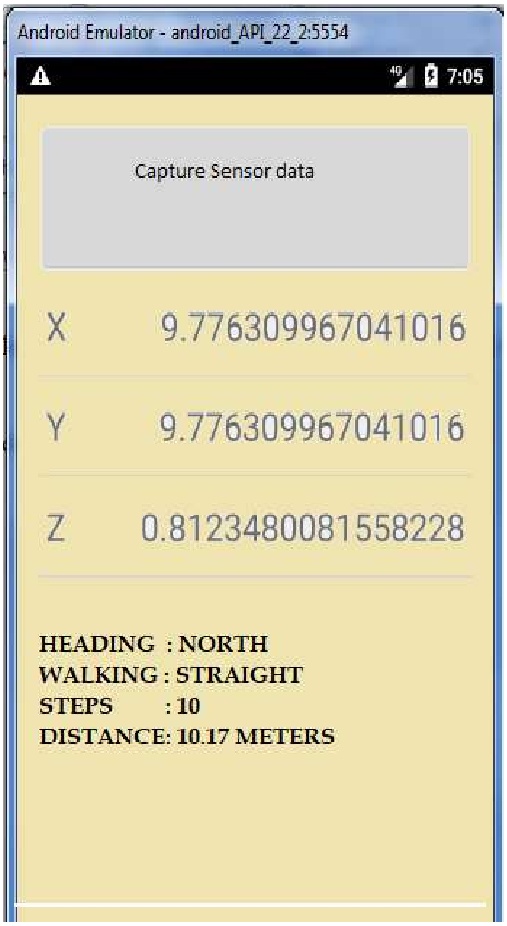


Figure 5.20: Screenshot of Android application

Figure 5.21 presents the trajectories of the estimated in blue and actual path in red of the user on the given floor plan, with iBeacons at the entry/exit of the doors of each room displayed by red dots.

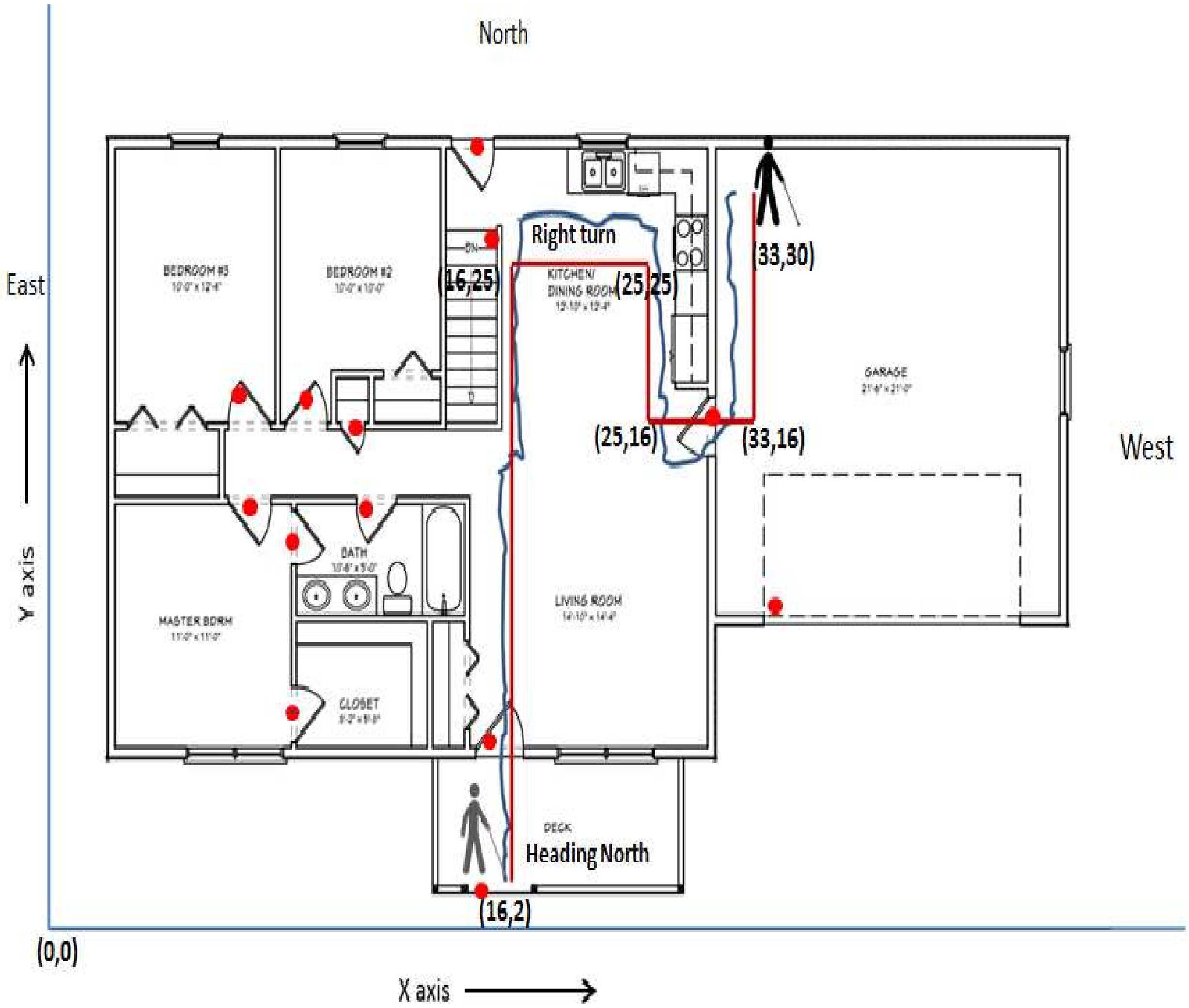


Figure 5.21: Position estimation (in blue) vs actual position (in red)

Figure 5.21 indicates the traces of a walking user with a smartphone with the newly developed Android-based fusion algorithm application installed. The red lines on the floor plan show the actual walking movements of the user. The estimated traces of the trajectories evaluated by the implemented fusion algorithm are presented in blue. The value of RMSE of the actual and estimated position of the user had an error of 1.5 to 2 metres.

5.5. Summary

This chapter has proposed a reliable and adaptive approach, BVIP, with a fusion positioning algorithm to track the walking movements of a VI person. IoT devices, including smartphones and smartwatches, with lightweight sensors, are a promising low-cost technique and convenient device for the long-term continuous tracking of a walking user. The proposed fusion algorithm can be implemented in a wearable device, such as a smartwatch or smartphone, to track the trajectory movement of the user in an indoor environment. With the path to follow evaluated based on the Ortho-PATH pathfinding algorithm, the fusion technique discussed in the chapter helps a VI person move independently in an indoor environment without assistance.

Nowadays, the smartphone has become a convenient IoT device in many people's lives. This chapter has discussed the foundation of working with the inertial sensors' filtration technique, including the accelerometer, gyroscope and magnetometer, from a smartphone's raw data. The chapter proposed a novel approach, BVIP, for positioning a VI user in an indoor environment with beacons as IoT devices at the entrance/exit of any location. The proposed technique minimises the use of external sensors travelling between two mark locations and the loss of signals in dark areas. The optimised system minimises the number of beacons, as the calibrated and filtered inertial sensor data evaluate the user's relative position in the indoor environment.

For areas such as significant hallways/corridors with no or minimal external positioning solutions, this system acts as a visual device to track the VI person in the indoor environment. The presented fusion algorithm demonstrates using the smartphone's inertial sensors (accelerometer, gyroscope and magnetometer) to determine the user's absolute position. Experiments were performed in real-time with an Android-based smartphone application. The system integrated the data from the three sensors (accelerometer, gyroscope and magnetometer) to position the user. The system was tested for varying behaviours of the user holding the device. The overall results illustrated that the mean error of the proposed system was 1.5 to 2 metres. Although the positioning error is not suitable for VI people but this solution works with minimum infrastructure to help VI know his/her direction primarily. The model overcomes the major disadvantage of inertial sensors of accuracy deteriorating over increasing measurement time. The current system estimates the user's position offline based on inertial sensor readings and the marker points' absolute position. However, there is a need to improve the coverage in case of a lack of signal of external signals from the iBeacon. In the next step, the intention is to extend the functionality to estimate the user's position online via the smartphone or Cloud.

Chapter 6

Deep Learning-based Indoor Positioning

This chapter proposes a regression-based deep learning technique to position a VI user in an indoor environment. The chapter presents a grid-based representation of an indoor floor plan and relates it to the public dataset, IPIN2016, of a smartphone's inertial sensors. The contribution of this chapter is to provide a self-directed, accurate and audio-aided standalone positioning system, considering the constraints of a VI person. The deep learning-based regression model discussed in the chapter provides a complementary solution using smartphones to achieve satisfactory results without external devices.

This chapter is derived from:
P. Mahida, S. Shahrestani, and H. Cheung, "Deep learning-based positioning of visually impaired people in indoor environments," *Sensors*, vol. 20, no. 21, p. 6238, Oct. 2020.

6.1. Overview

This chapter presents the deep learning-based MLP model to position a VI user in an indoor environment. Section 6.2 introduces the concept of absolute positioning and briefly discusses the problems faced, with the solution provided in Chapter 5. Further, the section discusses the 2-D representation of indoor space maps to the multivariate IPIN2016 dataset. Section 6.3 discusses the usability of the public dataset IPIN2016 to evaluate the 2-D position of an indoor user. Section 6.4 examines different versions of the regression-based DNN training and parameters for experimentation. It also discusses the deep network structure with the hyperparameters used in the experiments. Section 6.5 discusses the experimental platform and evaluation results, with Section 6.5.1 presenting the deployment platform with a selection of deep network architecture and suitable hyperparameters, considering prediction accuracy and localisation error. Section 6.5.2 presents the performance metrics and results of the models. Finally, Section 6.6 summarises the chapter, with a discussion of limitations.

6.2. Introduction

Accessible location-based information for navigating a complex indoor environment is a requirement of every individual [81]. Navigation in complex infrastructures, such as shopping malls, airports and hospitals, is aided through the proliferation of visual maps, digital maps and kiosks. However, VI people can find it challenging to use such aids effectively. The lack of a robust technology hinders the navigation of VI people, given issues regarding layout complexity, accessibility, connectivity and temporal changes in the environment [113]. Technologies must ease VI people's navigation processes by solving challenging issues, such as providing suitable indoor positioning, tracking moving users, obstacle avoidance, and pathfinding [14]. A variety of wireless technologies are currently available for indoor positioning and navigation, relying on Zigbee, RFID, iBeacon, Bluetooth, UWB, magnetic field and pedestrian dead reckoning, as discussed in previous Chapter 2, Section 2.2 [14], [148].

In previous chapters, we reported on providing solutions for movements of VI people in a smart environment using interconnected IoT devices [142]. A robust approach involves using BLE beacon sensors in the building to help a VI person navigate indoors. A developed algorithm, DynaPATH, generates VI-friendly safe routes to a destination, with considerations such as walking along walls and ensuring a straight path with minimum turns. Unlike solutions that choose the shortest path [110], DynaPATH proposes a safe path by considering the limitations of VI people [136]. However, VI people may encounter difficulty positioning themselves in an open space, such as a large hallway, because of the unavailability of external physical devices to provide location information. As a result of the possible loss of external signals, the system must maintain the position of VI people when other external devices are out of range. This chapter addresses this positioning issue and investigates the use of inertial sensors to provide a complementary solution that can be integrated into our work. The main idea behind the work is to demonstrate minimum infrastructure usage to aid VI people to overcome the challenge of positioning themselves independently between landmarks.

This section discusses a deep learning approach to position a VI user in a grid-based indoor environment, as discussed in Chapter 2. The indoor area is divided into microcells, each of which is assigned a unique region/place identifier that acts as a recognition layer. Each room in the indoor space is given a room identifier. The vertex of the microcell has 2-D (x, y) local coordinates. Figure 6.1 shows a representation of the sample floor plan that has undivided and divided areas. The solid black lines represent the walls of the indoor environment, while obstacles in the rooms are solid filled rectangles. The indoor space is divided into grids of cells, as depicted in the lower part of the figure. The shaded grey rectangle is a unique microcell with four vertices. Local coordinates are allocated manually and stored for each vertex in the building, resembling the latitude and longitude used in a GPS.

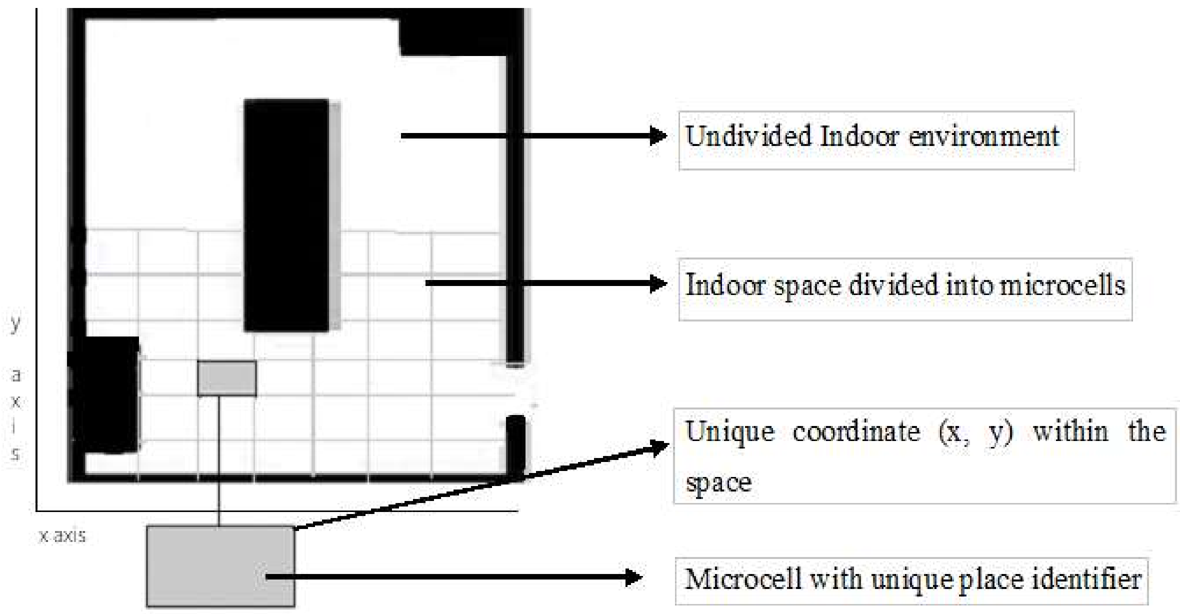


Figure 6.1: Grid distribution of indoor environment [149]

In this work, we propose a positioning technique based on mapping the inertial sensor measurements of a smartphone into position coordinates, using regression-based training of a DNN. The work discusses the results of various experiments to validate the suitability of the proposed approach. The experiments used a publicly available dataset containing records that resemble the walking and movement data of a VI person, such as walking straight along walls. The novelty of this approach stems from the use of regression-based MLP neural network training to determine the position of VI people in a building accurately.

Our contributions in this work aim to achieve the research objective stated in Chapter 1 (subsection 1.3.1, RO4) to provide a robust independent inertial guidance tool to position a VI person in the indoor environment. The work aims to develop an audio assistant app deployed on a smartphone to help VI people move independently in a complex building. To the best of the author’s knowledge, this work is the first to propose and recommend regression-based neural network training for estimating the position of a VI person moving in an indoor environment with a smartphone. A DNN model is proposed to predict the position of an indoor user as a complementary system to our existing navigation framework using external sensors [136], [142].

6.3. Characteristics of multivariant IPIN2016 dataset

Using an appropriate dataset for training and testing the model is an essential step in a DNN. Despite many works seeking to solve the indoor localisation issue, there is a lack of public datasets with inertial sensor data for a controlled environment. With the limited number of datasets, we used the multivariate IPIN2016 dataset [150] in this work to test the proposed approach. Although the pedestrian collecting the inertial sensor data was not VI, the user's movement had similar steps, including walking along the wall and walking at the same pace. As with our design discussed in Chapter 3, the dataset splits the indoor environment into cells mapped with the inertial sensor data of a smartphone. This section discusses the dataset with usability for a controlled indoor environment for VI people. The dataset has different types of movement fingerprints, including magnetic readings from smartphone/smartwatches in the divided spaces. Magnetic readings are data captured by the magnetometer, accelerometer and gyroscope of a smartphone/smartwatch. The multivariate IPIN2016 dataset has captured the records of the moving user in 325 different locations [150]. This dataset includes 36,795 continuous samples over two scenarios of one hour at 10 Hz, which resulted in 6,500 discrete samples in 325 locations.

The dataset was created on the first floor of the Institute of Information Science and Technologies, inside the Italian National Council building. The dataset covered movements over a surface of 185.12 m^2. Figure 6.2 depicts the overall map of the building, with the top view and trajectory path [150]. The top left corner of the figure is the top view of the floorplan. The middle portion of the figure is the highlighted corridor of the given floor. Further, the trajectory path followed by the users is shown at the bottom with dots. Each dot in the map corresponds to a detection point, and each dot is 0.6 metres from another. The dots represent different locations at which two users acquired inertial sensors data on their smart devices. Thus, each combination of four dots occupies an area of 0.6 × 0.6 m. Given the fixed size of the microcell in the given dataset, our experiments used the same grid size. However, there was further scope to observe the effects of different grid sizes on the results.

The dataset consists of two scenarios with a combination of zigzag and straight-path trajectory performed by two different users holding a smartphone to cover the entire target area. The walking speed of each user was 0.6 m/s on average. Each sample was collected about every 100 milliseconds, and the collection time was short. The dataset employs a unique combination of Wi-Fi signals and inertial sensor data of smartphones and smartwatches. This study did not consider data from the Wi-Fi access points or smartwatch data. During the acquisition, the smartphone was kept at chest level, with the screen facing up.

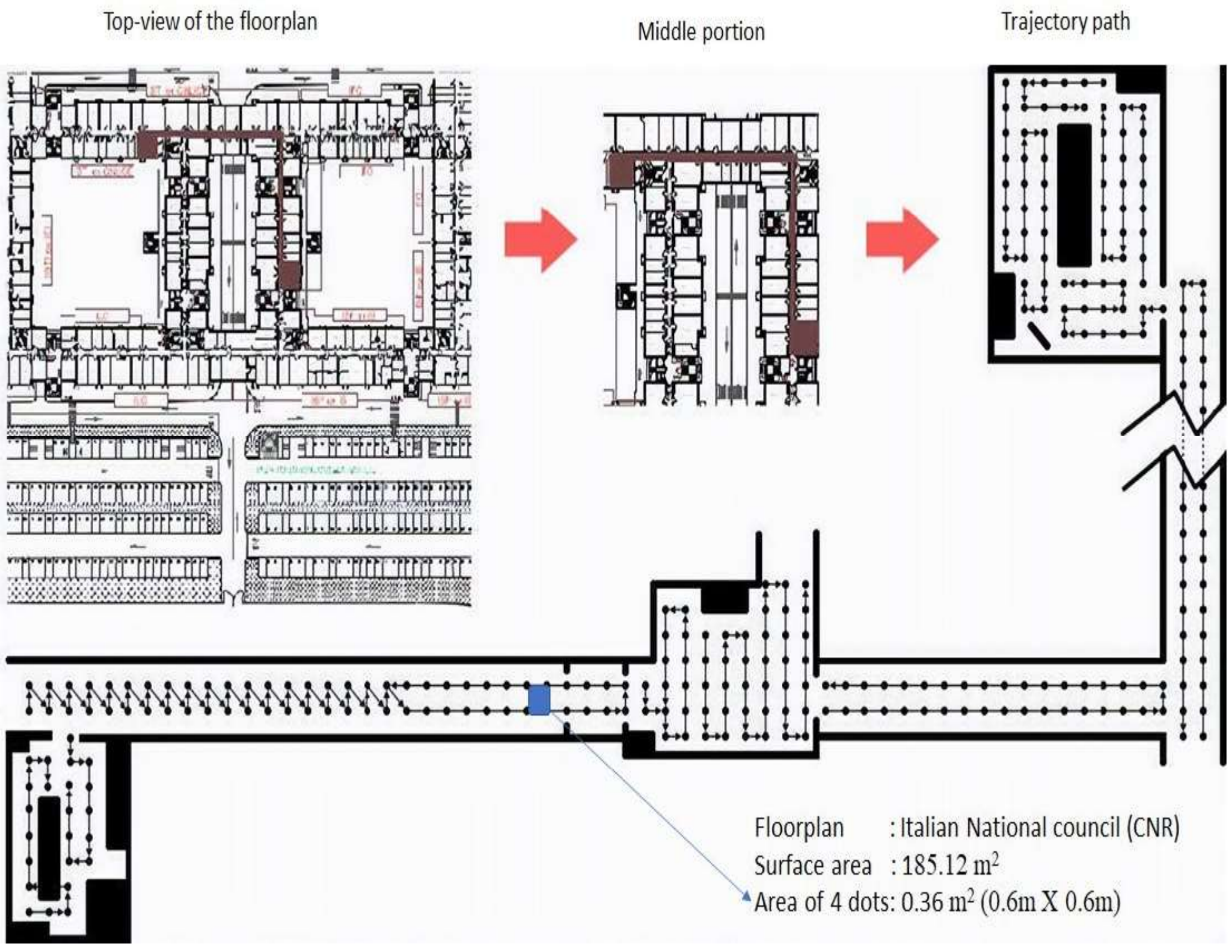


Figure 6.2: Indoor floorplan with top view and trajectory path from IPIN2016 [150]

Each time the user was at a specific location, the device recorded the data as mentioned below at each dot location of the indoor space. The recorded data included the following readings at each dot, represented as PlaceID with their local coordinates (x, y) at a given timestamp:

- x-, y- and z-axis values of the accelerometer sensor
- x-, y- and z-axis values of the magnetometer
- x-, y- and z-axis values of the gyroscope
- roll, pitch and azimuth values of the inertial sensor.

The work focused on the magnetic field signals with the accelerometer and gyroscope of the smartphone. Figure 6.3 presents a graphical representation of values from the x-, y- and z-axis of the magnetometer. The normalised magnitude M_{mag} of the magnetometer is calculated by Equation (6.1):

$$M_{mag} = \sqrt{{M_x}^2 + {M_y}^2 + {M_z}^2} \tag{6.1}$$

where M_{mag} is the normalised magnitude of the magnetometer and M_i is the value of the i^{th} axis of the three-axis accelerometer.

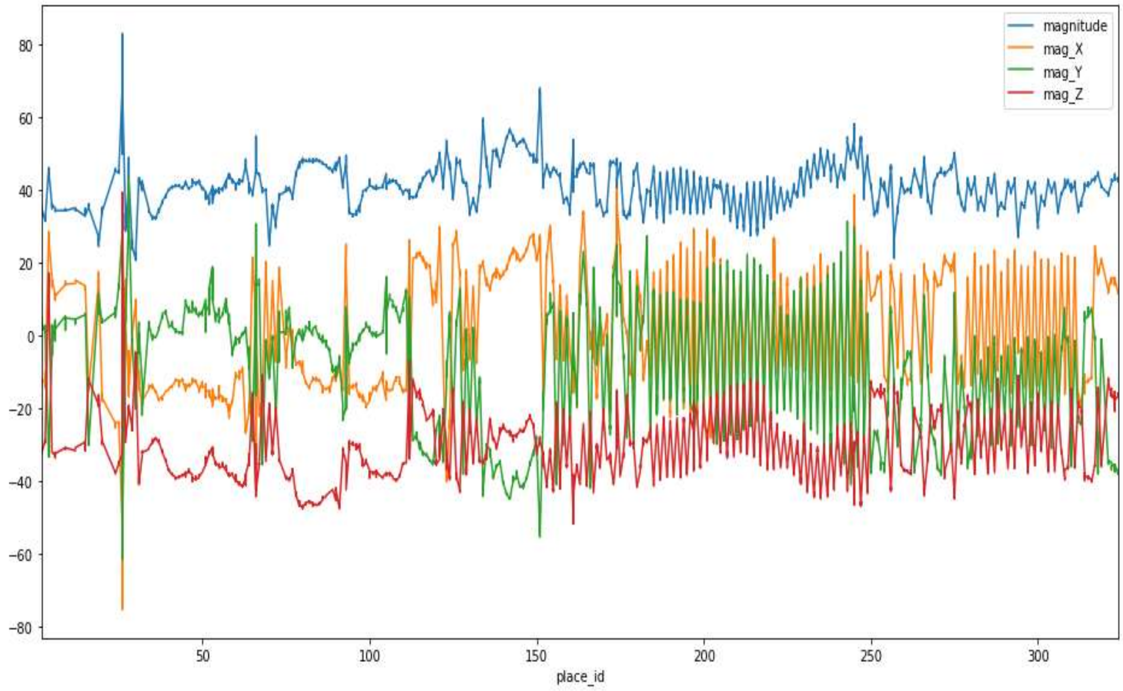


Figure 6.3: Graphical representation of x, y, z and magnitude of magnetometer readings

Figure 6.4 displays the magnetic field heatmap of each location on the trajectory of the corridor, followed by user 1. The number represented in each cell of the grid on the heatmap is normalised, as evaluated in Equation (6.1). The value of the magnitude varied from 21 to 68 for the given dataset. The indoor magnetic field may be distorted over time locally because of the steel-reinforced concrete in the structures. However, the study revealed that the magnetic field's distortion pattern remains static [151].

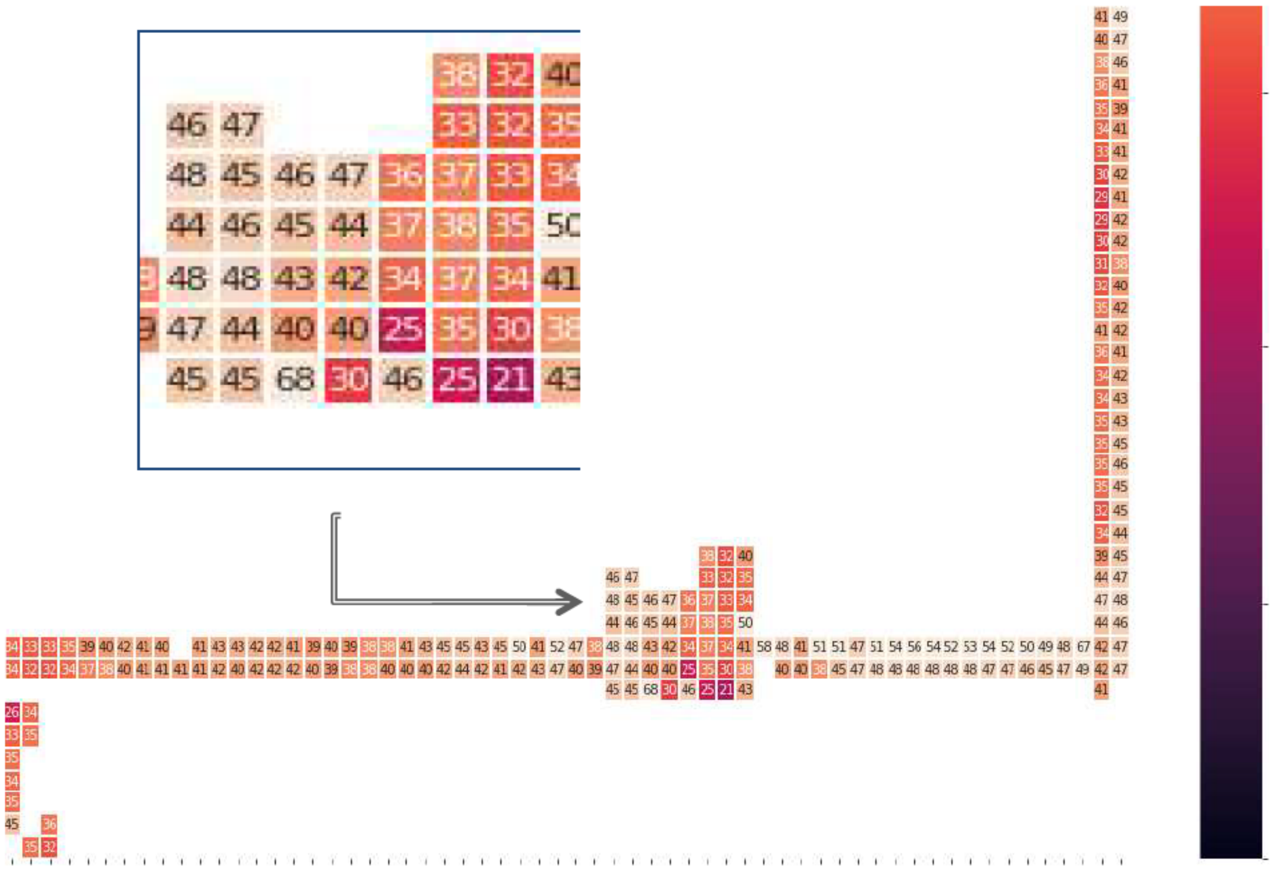


Figure 6.4: Heatmap of magnitude of magnetic field in each location in the building [149]

6.4. Simulation parameters of deep learning-based positioning

The data were split into two subsets to train the MLP and validate the learning. Testing of the performance involved the use of an independent dataset not used to train the model. The size of the dataset in our simulation was comparatively small and intricate. Therefore, we used the K-fold cross-validation technique. The K-fold method is a resampling procedure used to evaluate a deep learning model based on a limited number of data samples [152]. It is popular because it is a less biased and less optimistic estimate of the model over a simple train/test split. This technique involved a random division of the dataset into K groups, or folds, of approximately equal size. The first blue fold was treated as validation data, and the model was trained on the remaining K-1 training data, as depicted in the first of the K iterations in Figure 6.5. A validation fold was used to monitor the performance during training and was not used in training the model. In the second iteration, the second fold was used as validation data, while the rest were used in the training process.

The distribution of the training and validation data in the experiments was with $K = 5$ in the K-fold cross-validation technique. The dataset was equally distributed in five parts, including the first 7,359 records as the validation data and the remaining 29,436 records as training data in the first iteration. Five iterations were completed over the total samples each time, in which 7,359 data samples were treated as validation and the remaining as training data.

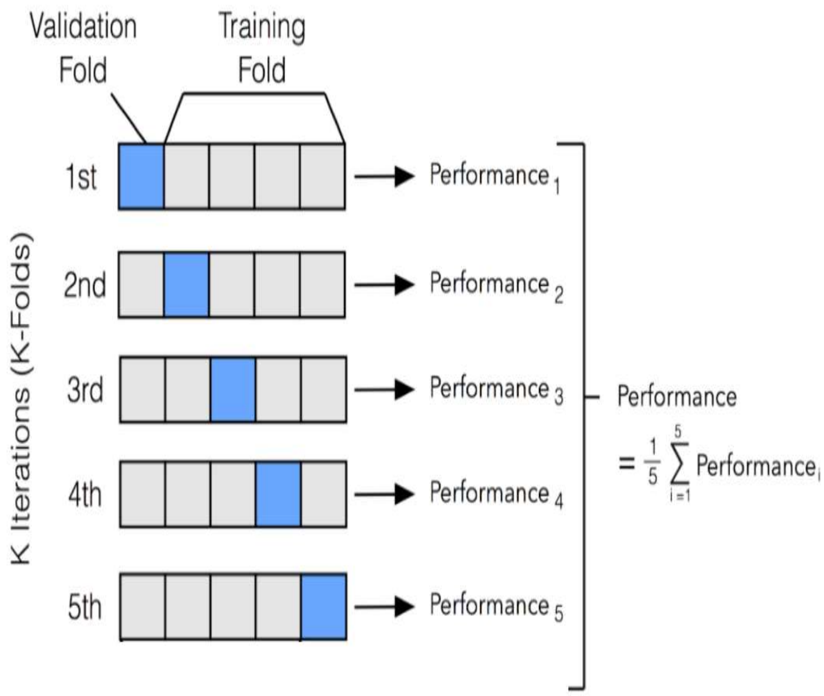


Figure 6.5: K-fold cross-validation technique [149]

The training data were used in each iteration with fixed hyperparameters. The hyperparameter is the higher-level properties of the data model that improves its performance and expresses the capacity of the model for learning the data complexity [153]. To improve the performance of the model, we involved hyperparameters, including several layers, epochs, mini-batch size, activation function, dropout, regularisation and optimisers [154]. The experimental model was implemented in Python with Keras and TensorFlow libraries with different settings and hyperparameters, as listed in Table 6.1.

Table 6.1: Experimental model with hyperparameter values

Parameter	Hyperparameter values in proposed deep MLP
Software and libraries	Python, Keras, TensorFlow
Training data Validation data	29,436 7,359
Epochs	60 to 140
Batch size	20, 40, 60, 80
Layers with hidden neurons (with batch normalisation)	3 layers—128, 64 and 128 neurons 5 layers—256, 128, 64 and 128 and 256 neurons 7 layers—512, 256, 64, 128 and 256 neurons
Dropout rate	0.2 to 0.8
Activation	SELU, ELU, Softplus, ReLU
Optimiser	Adam, Adamax, RMSprop, Adagrad
Loss function	MAE, MSE, RMSE

The experiments were performed with different hyperparameter settings making a different version of the MLP model. Further, the training data and labels were tuned to select the final model. The training data were tuned with the best hyperparameters and learning algorithms. The following section discusses the experimental results of the different settings with the best-suited hyperparameters.

6.5. Setup of experiments and analysis of results

The experimental platform used to test the performance of the proposed neural network is presented in Section 6.5.1, and the performance metrics and evaluation results are discussed in Section 6.5.2.

6.5.1. Setup of experiments

Figure 6.6 presents the experimental platform in estimating the user's position using a deep MLP algorithm.

The experimental platform takes a sequence of inertial sensor recordings as input, including the accelerometer, gyroscope and magnetometer readings of the smartphone. The collected inertial sensor readings from the smartphone are archived and passed as the training set to build the DNN model. The platform continuously updates the model using the archived and stream data from a smartphone. Further, the features are extracted and passed to the model for predicting the position. In this work, a sequence of inertial sensor sample values from a dataset—including those from an accelerometer, gyroscope and magnetometer—for a moving user were collected and fed as input to the neural network as training data.

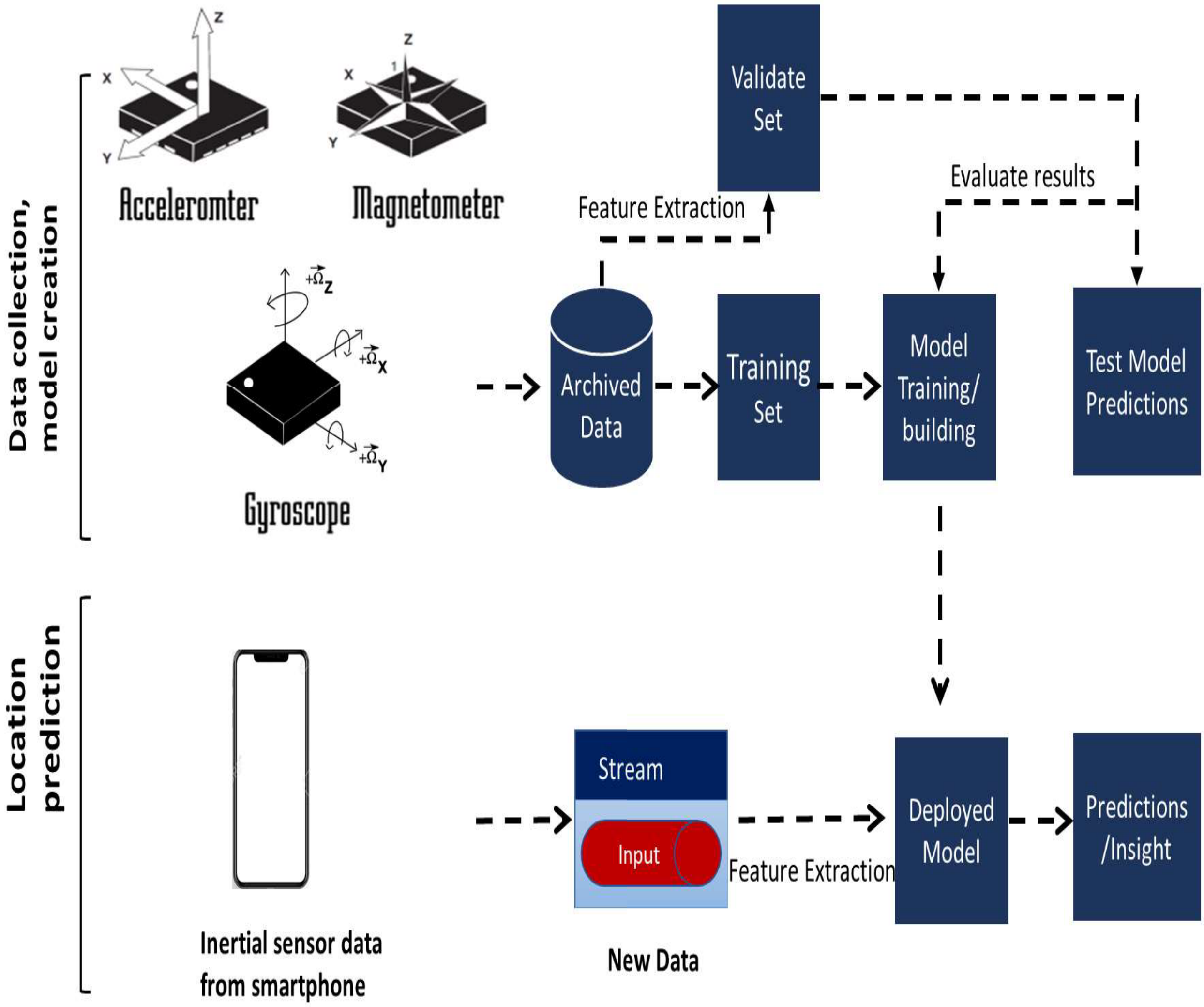


Figure 6.6: Experimental platform for the proposed model

We trained the neural network as a regression problem to learn the 2-D location of the user based on the input information. After the training, a model was established and could be used to estimate a user's location based on real-time sensor data. To test the performance of the model, a test set based on the K-fold technique was used to evaluate the prediction accuracy of the proposed model.

6.5.2. Performance metrics and evaluation

The deep learning model was implemented using Python with libraries such as TensorFlow and scikit-learn. The performance was measured using the MAE, RMSE and MSE between the ground truth and predicted location. The model evaluation process assessed the localisation error and prediction accuracy of the model on the multivariate IPIN2016 dataset, as described in Section 6.3. The MAE is the mean of the absolute value of the errors, as shown in Equation (6.2):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|PS_i - SA_i| \tag{6.2}$$

The MSE measures the average of the squares of the errors. It is the average squared difference between the actual value and estimated value, as shown in Equation (6.3):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(PS_i - SA_i)^2 \tag{6.3}$$

The RMSE is a measure of the average deviation of the predicted from the actual values. It is used to measure the difference between the values predicted by a model and the values observed from the modelled environment. The average localisation error of the calculated distance travelled of the proposed approach can be evaluated by calculating the RMSE as the square root of the residuals, with Equation (6.4):

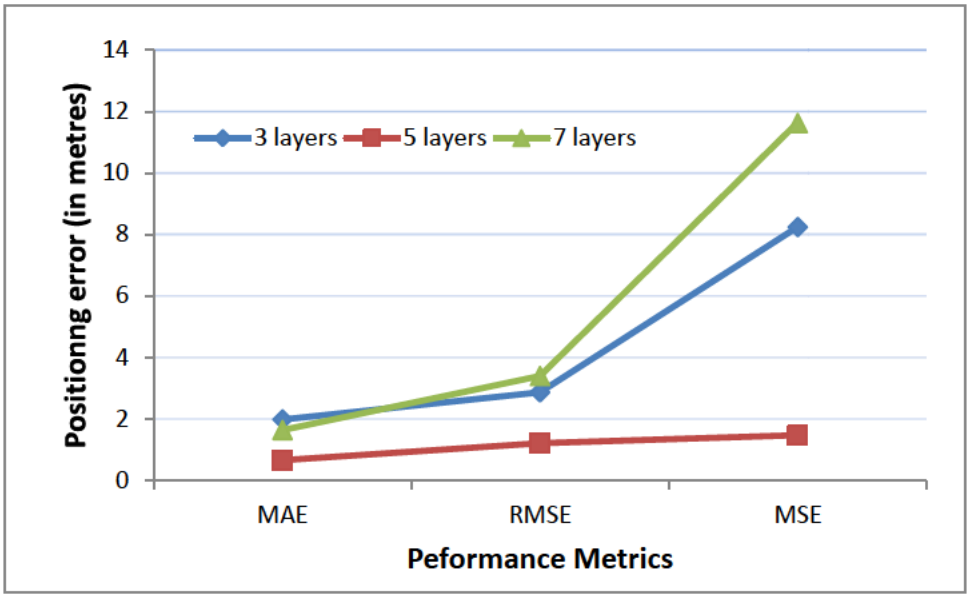
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(PS_i - SA_i)^2}{n}} \tag{6.4}$$

where n represents the walking experiments conducted by each user along the given path, PS_i denotes the predicted final location by the proposed algorithm and SA_i denotes the actual final location in the i^{th} experiment.

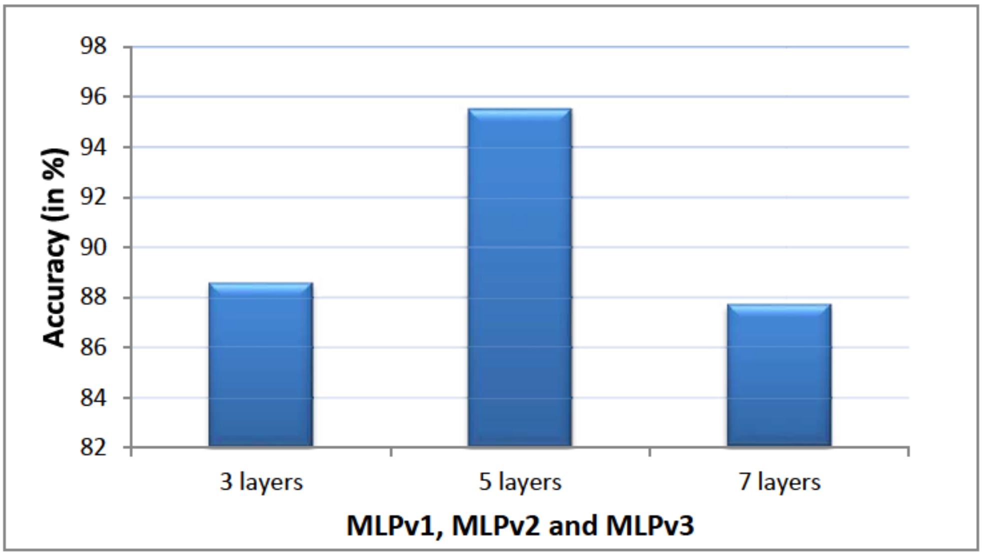
We implemented three normalised batch versions of the MLP algorithm, as discussed in Chapter 2: MLPv1, MLPv2 and MLPv3, with three, five and seven layers. All three versions were implemented to evaluate the performance of the best-fit model and investigate the effects of a different number of hidden layers. The performance metrics, including MAE, MSE and RMSE, are shown in Figure 6.7 (a). The prediction accuracy of the model was evaluated by calculating the ratio of the number of correct prediction occurrences to the total number of predictions. The prediction accuracy is shown in Figure 6.7 (b).

The MAE positioning error for MLPv1 with three layers was 1.99 metres, with a prediction accuracy of 88.57%. When the number of layers increased to seven, the positioning error was reduced to 1.64 metres, with a prediction accuracy of 87.71 %. The performance of the MLPv2 with five hidden layers had a positioning error of 0.66 metres, with a prediction accuracy of

95.54%. The results indicated that the average positioning error and prediction accuracy of the five-layered network was better than the other two networks. As such, we used the network with five hidden layers in the rest of this work.



(a)



(b)

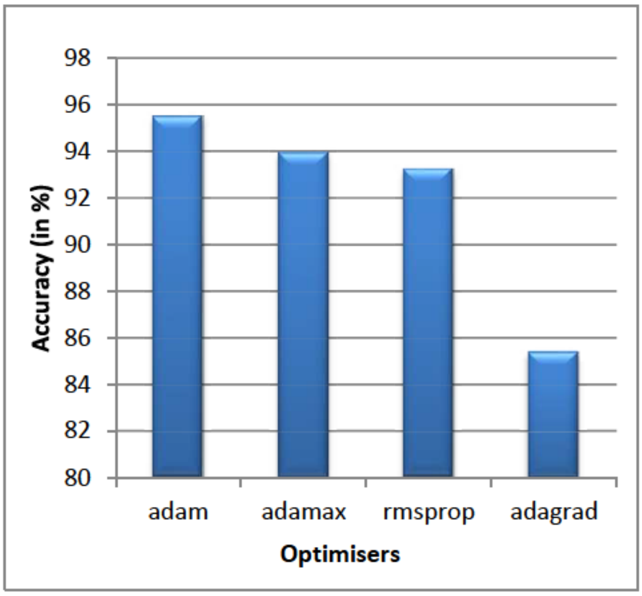
Figure 6.7: Comparison of (a) positioning error (MAE, MSE and RMSE) and (b) accuracy for some layers of the model [149]

Different permutations of hyperparameters were further implemented with a five-layered network. Given that the input data were continuous and differentiating, we tested the model with non-linear (Tanh, SELU, ReLU, Softmax and ELU) activation functions. In the tuning process, we applied optimisers, including Adam, Adamax, RMSprop and Adagrad. Table 6.2 shows the positioning errors for the variously implemented optimisers. For each optimiser, an appropriate learning rate was proposed.

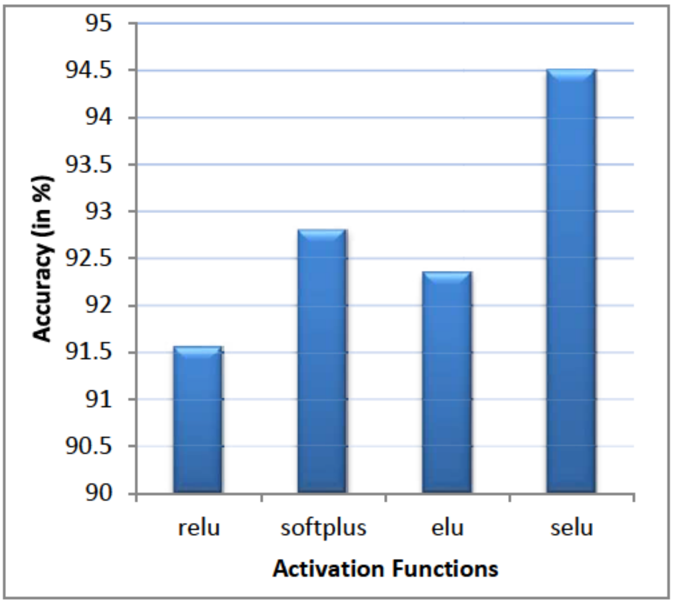
Table 6.2: Positioning error (in metres) with different optimisers and activation functions

Optimiser	MAE (m)	RMSE (m)	MSE (m)
Adam	0.71	1.30	1.70
Adamax	0.84	1.35	1.81
RMSprop	1.04	1.84	3.39
Adagrad	5.59	3.25	3.62
Activation	**MAE (m)**	**RMSE (m)**	**MSE (m)**
ReLU	1.24	2.61	6.84
Softplus	1.35	2.45	6.01
ELU	0.92	1.85	3.45
SELU	0.65	1.29	1.67

As shown in Figure 6.8 (a), the implementation of the Adam optimiser provided the best performance, with an average of 0.71 m MAE error and 95.5% prediction accuracy. Thus, we continued keeping the adaptive moment estimation (Adam) with $\beta1 = 0.9$, for $\beta2 = 0.999$ and $\epsilon = (10 \text{ x } \exp(-8))$. The positioning accuracies using activation functions, such as ReLU, Softplus, ELU and SELU, are shown in Figure 6.8 (b). All activation functions were performed with the Adam optimiser. The results indicated that the SELU activation function outperformed the other activation functions, with a positioning error of 0.65 metres and 94.51% prediction accuracy. Table 6.2 shows that the minimum positioning error was found with the SELU activation function, with 0.65 metres MAE, 1.67 metres MSE and 1.29 metres RMSE. Figure 6.8 (b) and Table 6.2 show that the SELU activation function outperformed the ReLU, Softplus and ELU activation functions.



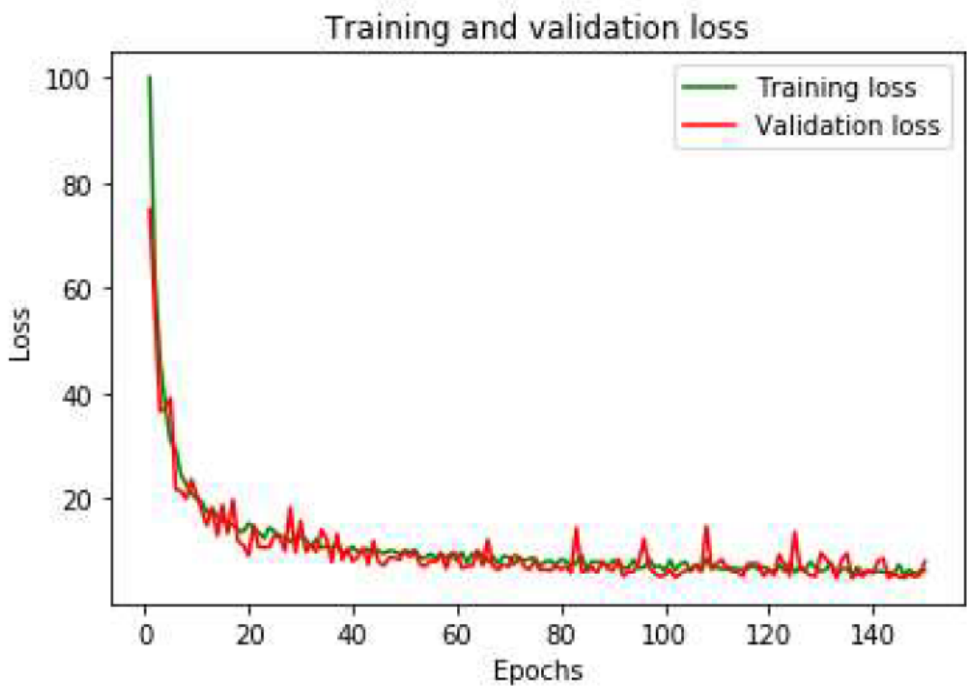
(a)



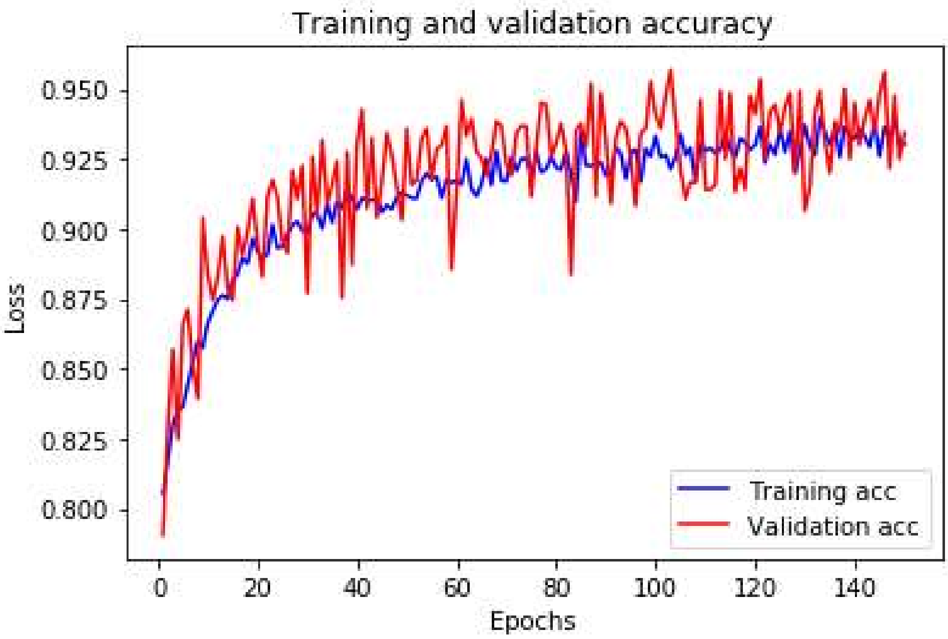
(b)

Figure 6.8: Prediction error with (a) different optimisers and (b) different activation functions [149]

The loss function declining curve with different epochs on the training and validation dataset is plotted in Figure 6.9 (a) and 6.9 (b), showing that the curve became stable after 60 epochs, yet we continued the observation to the 140th epoch.



(a)



(b)

Figure 6.9: Training and validation (a) loss and (b) accuracy for deep MLP model

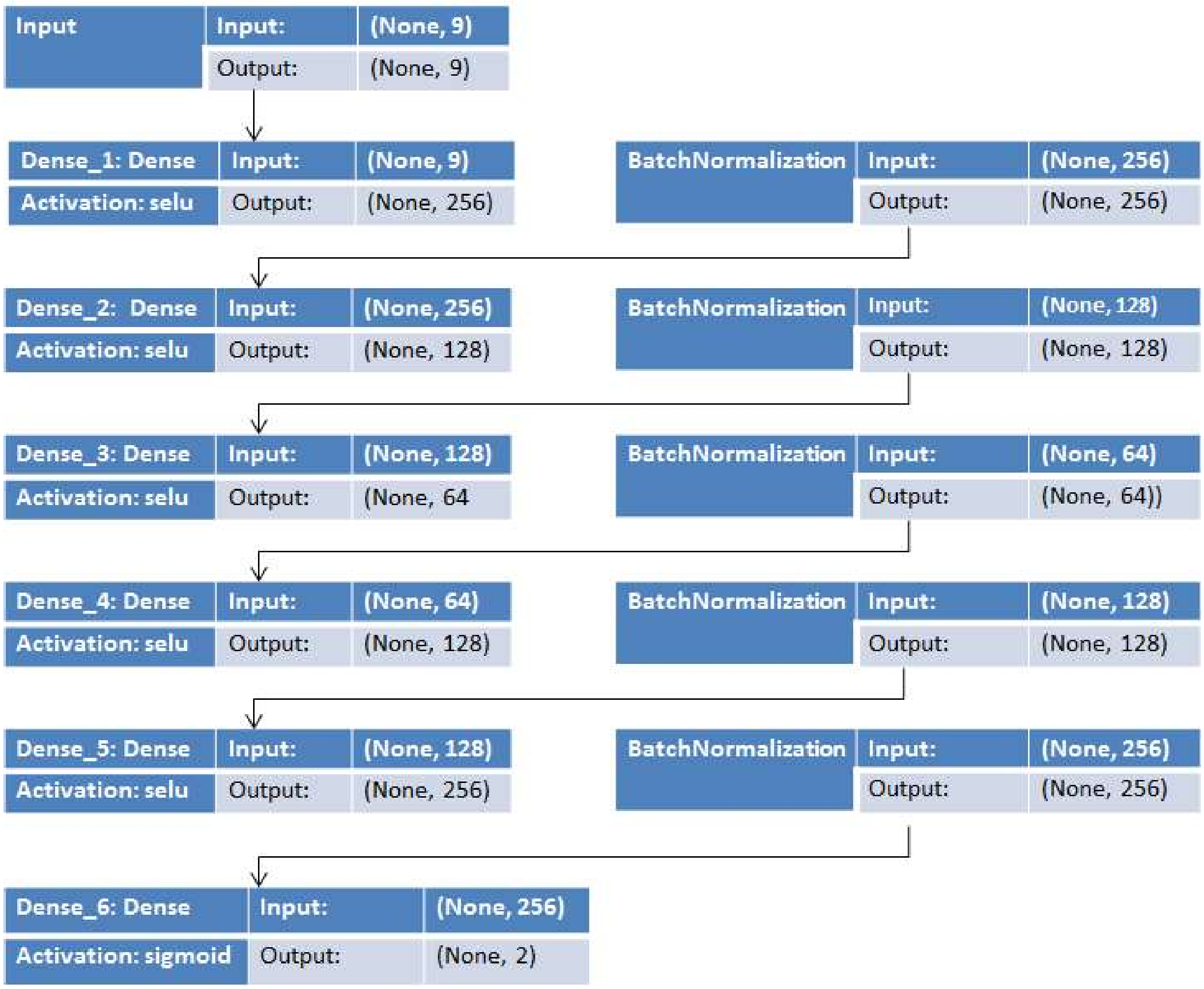


Figure 6.10: Best-suited regression-based DNN MLP model [149]

Figure 6.10 presents the best-suited regression-based DNN model used for the predictions, while Figure 6.11 presents the traces of the actual position (x, y) from the IPIN2016 dataset for user 1. Figure 6.12 displays the calculated predicted positions (x', y') using the best-suited regression MLP model.

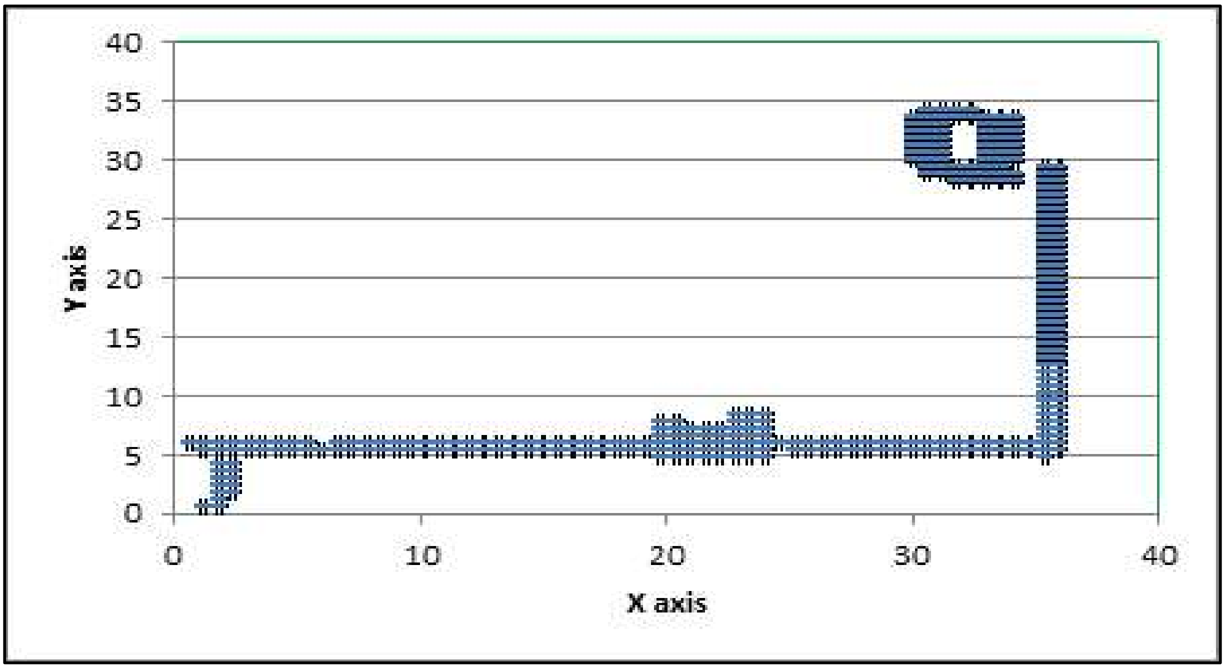


Figure 6.11: Actual (x, y) position based on dataset

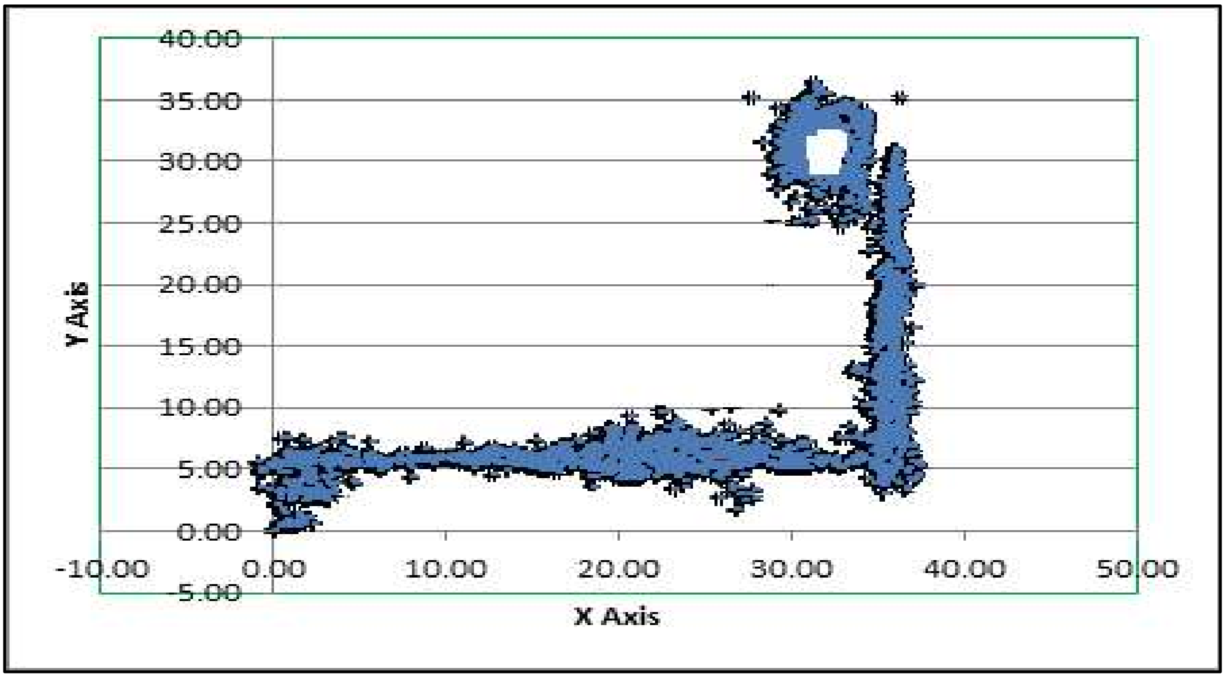


Figure 6.12: Predicted (x', y') position based on deep MLP model

The results demonstrate that the proposed model achieved a considerable prediction accuracy of 94.51%, with a 0.65-metre positioning error. The highest positioning error was no higher than 0.89 metres. The training time for the given model was approximately 16 seconds and the prediction time for a given sample once trained was five milliseconds. Our previous work based on an improved positioning algorithm [146] applied to the dataset provided an almost equal prediction accuracy of 95%. However, the positioning error was evaluated as 1.5 to 2 metres. The work requires an absolute position from additional devices, such as those capable of generating beacon signals. As such, the variation of the error is too high compared with the proposed model. Moreover, the computation time is almost doubled to 10 to 12 milliseconds.

6.6. Summary

This chapter has proposed a novel approach to achieve the positioning of a moving VI person as part of an indoor navigation system. The approach was based on feeding the data from the inertial sensors of a typical smartphone to a trained MLP, which mapped them into 2-D local coordinates of the microcell corresponding to the person's position holding the phone. The proposed approach was tested with data from a publicly available multivariant dataset, IPIN2016. The dataset contains data from movements that resemble walking by a VI person. The performed experiments showed that the proposed approach could achieve positioning accuracy close to the step size of a typical user—around 0.65 metres. The achieved accuracy for the DNN based approach is without the help of dedicated hardware of the infrastructure in the building. Regarding positioning accuracy, it is safe for a VI person to use the system with the known positioning error equal to the average step size. The previous Chapter 4 discussed the working and efficiency of pathfinding algorithm Orth-PATH, suggesting a reliable and VI-friendly path to the VI person in an indoor space. Chapter 5 presented a fusion tracking system based on IoT devices that helps generate the walking user trajectories. This chapter aims to complement our navigation system with the proposed positioning approach to facilitate easy and independent indoor positioning and movements of a VI person.

Chapter 7

Conclusion and Future Research Directions

This chapter summarises the research and contributions of the thesis to improve the indoor navigation of VI people using IoT and deep learning techniques. It concludes the thesis by illustrating the work conducted to answer the research questions and highlights the directions of future work in this area.

7.1. Overview

Among essential activities in daily life, travelling safely and independently in different and unknown environments is a challenging task for a VI person. This research work proposes a safe assistive Android application with features including VI-friendly paths, centimetre level absolute positioning and tracking a mobile VI person to travel indoors independently. Section 7.2 summarises the major research activities undertaken in this thesis, while Section 7.3 presents limitations, future research directions and concluding remarks.

7.2. Conclusion

This doctoral research investigated the usability of current technologies, including IoT and deep learning techniques, to facilitate the navigation experience of VI people in an indoor environment. VI people may attain assistance to sense environment cues from devices such as a white cane to navigate an outdoor environment. However, previous research findings reveal that navigation inside buildings with unfamiliar features and barriers discourages the independent movement of a VI person. Many navigation systems have proposed solutions to mitigate some challenges, such as localisation, pathfinding and landmark detections. However, to navigate indoors, VI people require granular context-aware information via turn-by-turn directions, providing orientation and mobility features, including centimetre-level positioning accuracy. Some existing navigation solutions may require complex infrastructure installations or do not provide a complete solution that meets the needs of VI people [22]. This thesis addressed many of the challenges faced by VI people in the preceding chapters.

Chapter 1 defined the context and scope of the research. The chapter situated the research in navigation for VI people and using advanced technologies and devices to enhance VI people's indoor experience. The chapter presented the research objectives and questions, with the methodology addressed and applied in the following chapters of the thesis. This chapter answered the first part of the primary research question discussed in Section 1.3.2 by discussing the difficulties faced by VI people while navigating an indoor environment. The chapter discussed the methodology followed to resolve significant gaps in the existing systems to create effective and smart navigation.

Chapter 2 presented the theoretical background with a literature review on indoor navigation for VI people. It identified and classified the needs of a VI person travelling in an unknown indoor environment. Further, it discussed the existing indoor navigation systems for VI people, highlighting context-aware architecture, such as RSNAVI. Further, it highlighted the advantages and disadvantages of the existing navigation systems for VI people. The chapter addressed the second half of the primary research question, RQ1, as stated in Section 1.3.2. The analysis and study of the literature helped identify and address research gaps, including accurate positioning, reliable pathfinding and tracking techniques for VI people. The chapter summarised critical issues, such as self-positioning, finding an appropriate path to the destination that considers vision limitations, and generating the user trajectory with minimal devices to carry. The chapter discussed various map representation techniques and proposed a grid-based occupancy map representation for VI people to enhance accessibility. Finally, the chapter underlined and discussed the high-level building blocks of an indoor navigation system with suitable map representation techniques.

Chapter 3 discussed the proposed framework, Indoor-Nav, highlighting the scope and major problems resolved in the thesis to enhance the indoor journey of VI people. The chapter presented the primary components of the proposed framework, including indoor map setup, path estimation and tracking and positioning. The chapter explored the possibility of including IoT and deep learning techniques with grid-based OGMs in the indoor navigation framework. The approach used in the framework addressed the RQ2 and RQ3 initiated in Chapter 1 (Section 1.3.2) by illustrating the use of IoT and deep learning to develop an intelligent navigation guide for VI people. Further, the chapter derived the use of a grid-based occupancy map from the study in Chapter 2 to ensure collision-free paths and avoid significant barriers, including fixed furniture and other environmental changes in a dynamic environment. All of the simulations and experiments assumed the use of OGM for the given indoor space. Finally, the chapter discussed the high adaptive learning model used in the thesis to track and position the moving user in an indoor environment, proposing the use of IoT devices and deep learning.

Chapter 4 discussed the popular pathfinding algorithms, including Dijkstra’s, A*, PRM and RRT, used in an indoor environment with grid-based indoor maps to target the research gap related to path generation. The framework proposed embedding UHF RFID on the interiors and obstacles in the environment to display obstacles in the indoor map. The updated OGM grid-based map is provided as the input to the pathfinding algorithms. Dijkstra’s algorithm is mainly used over the expanding grids and provided paths from the source location to all destination locations. Unlike Dijkstra's algorithm, A* does not explore all possible paths. However, it sought a better path by using an admissible heuristic function.

Nevertheless, in cases where obstructions in the environment increased, the PRM and RRT algorithms helped to generate paths more rapidly. However, the paths generated by the PRM and RRT pathfinding algorithms were found non-smooth and unfavourable for VI users. The popular pathfinding algorithms were compared based on computation time and path quality, considering the limitations of low-vision users. Overall, A* provided better results but primarily focused on distance. For a VI person, travelling a few metres more is acceptable if it enables a safer route. These algorithms may produce routes with less distance to travel, but the path quality and obstacles are unsuitable for VI users. The results of this research show there is still scope for developing more efficient pathfinding algorithms. Considering the limitations of VI people, the chapter proposed a novel pathfinding algorithm, Ortho-PATH, to provide line-shore, safe paths that avoid obstacles. The algorithm's orthogonal path generation technique prioritises safety by providing paths that do not involve any angular movements. This study answered RQ4 (Section 1.3.2) concerning essential criteria for a pathfinding algorithm for VI agents. It also addressed RQ5 by proposing the Ortho-PATH pathfinding algorithm to provide an orthogonal path without the burden of post-processing and having computation time approximately the same as A*, the optimal pathfinding algorithm.

Chapter 5 proposed a framework to use IoT devices and fusion algorithms as a step forward in guiding VI users and generating trajectories to help users when lost in an unknown environment. The chapter discussed the details of the indoor tracking technique, BVIP, which uses a beacon and fusion algorithm that incorporates adaptive learning from inertial sensors, including an accelerometer, magnetometer and gyroscope. The dynamic threshold value used to evaluate the travelled distance by a moving user considers a user's profile, including step length and stance length. Therefore, the proposed fusion algorithm is customised to the registered user and is less error-prone. The chapter answered RQ5, suggesting a fusion algorithm to track a VI person in a beacon- and RFID-embedded environment. Smartphones are currently everyday devices for many users; thus, a smartphone-based application is an appropriate choice for tracking a moving user. Experimentation with a smartphone-based Android application was employed in several scenarios with three users to demonstrate the efficiency of the proposed method and provide accurate trajectories of the users with minimal devices.

Chapter 6 presented the most remarkable contribution to emerge from the experimentation results of the thesis. The idea of deep learning techniques is implemented in smartphone applications that emulate human-like sensory organs to extract high-level information and discern essential features and patterns of the surrounding environment. These features of deep learning hold great promise to fulfil the challenging needs of humans with disabilities. The chapter discussed the multivariant public dataset, IPIN2016, used in the experiments, consisting of inertial sensor records of a smartphone and smartwatch. This study applied a regression-based MLP DNN model on the normalised dataset on an indoor map with a fixed grid size of 0.6×0.6 cm. The experimental results of the deep learning model provided performance accuracy of 0.65 metres by discerning the pattern of inertial sensor data and providing local coordinates of the indoor space. The chapter addresses the research question RQ6 discussed in section 1.3.2 of Chapter 1 by proposing a deep learning-based mechanism to self-position a VI user. The chapter has also responded to RQ7 with a Cloud-based deployment platform for the DNN in an indoor IoT-equipped environment. We also intend to test our approach using a much larger and temporal dataset to observe the effect of magnetic intensity variations on prediction error. The system enables VI people to know their direction and exact position, recognising nearby landmarks.

The contributions of the thesis, discussed in Section 1.5, have been addressed and verified throughout the research. The research goals are achieved by answering the research questions, as discussed above. In summary, the thesis contributes to boosting the confidence of VI people by enhancing their indoor travel with minimal equipment requirements.

7.3. Limitation and future directions

Although this thesis contributes to providing a complementary solution for VI people to navigate an indoor environment, this section presents the study limitations and research opportunities for future work that this thesis has revealed.

Deploying infrastructure such as UWB may lead to enhanced accuracy and reduced positioning error. However, it can be costly. Our framework proposes use of a smartphone—a widely accepted device—as the thesis proposes a non-vision-based technique that does not require high computational technology. The suggested system achieves reasonable accuracy. However, the thesis performed unit testing not on VI people but on people pretending to have vision impairment. Thus, the system must be employed with VI people to enable enhanced testing of the system.

This work does not involve the generation of OGMs from the floorplan but assumes that the structured OGM of the indoor space is provided as an input mapped with local coordinates of the indoor grid space. However, in future, we may seek to automate and resolve this issue. The smartphone-based Android application discussed in Chapter 5 tests two particular scenarios: (1) holding the phone in hand and (2) keeping the phone in the pocket. Although the fusion algorithm considers the smartphone's displacement, the work did not test various smartphone poses. Further, we plan to test the hybrid integrated system with IoT devices and a deep learning framework to assist VI people to move freely in the indoor world using minimal devices. The proposed approach requires Internet connectivity, as it relies on attaining position estimates from the trained model residing in the Cloud. We intend to explore whether this shortcoming can be addressed by using pre-trained models in a smartphone application in the future work. Finally, the proposed framework executed on a 2-D floor plan entailed the limitation of not detecting different building floors.

The simulation and experimental results discussed in the previous chapters provide a proof-of-concept of the proposed framework for an indoor navigation system for VI people. However, exhaustive integrated testing of the work has not been performed. Hence, the proposed framework can be further optimised in certain areas. In this context, some ideas are presented below:

- The grid size in the OGM may affect the positioning error and prediction accuracy. Given the limitation of the inertial sensor public dataset mapped local coordinates, the work focused on a fixed grid size of 0.6×0.6 cm. Further research and investigation of an appropriate grid size may result in better and more reliable accuracy than those received with the given grid size.
- The DNN suggested in this work achieves positioning errors that are less than 65 cm. However, there is scope to improve the positioning accuracy by implementing the deep neural model on the data received from advanced technologies, such as Lidar and UWB. Researchers may wish to create public datasets using these technologies to generate and test the DNN model.
- Privacy and security concerns related to the IoT devices proposed in the system must be addressed. Given the highly interconnected nature of IoT devices, these devices are

prone to attacks, and the attacker can further propagate through the IoT devices. Thus, this thesis can be extended to protect user privacy and secure data from being compromised.

Bibliography

[1] Z. Farid, R. Nordin, and M. Ismail, “Recent advances in wireless indoor localization techniques and system,” *J. Comput. Networks Commun.*, vol. 2013, no. 185138, p. 12 pages, 2013, doi: 10.1155/2013/185138.

[2] P. Doctoral and W. Bengal, “Internet of Things : New Promises for Persons with Disabulities,” *Global initiative for inclusive information and communication technologies*, vol. I, no. I, pp. 50–63, 2015.

[3] A. W. F. the Blind, “Blindness: Challenge and Achievement | World Access for the Blind,” *World Access for the Blind*, 2015. [Online]. Available: https://waftb.net/blindness-challenge-and-achievement. [Accessed: 18-Aug-2020].

[4] A. Brock and C. Jouffrais, “Interactive audio-tactile maps for visually impaired people,” *ACM SIGACCESS Access. Comput.*, no. 113, pp. 3–12, Nov. 2015, doi: 10.1145/2850440.2850441.

[5] G. E. Legge *et al.*, “Indoor Navigation by People with Visual Impairment Using a Digital Sign System,” *PLoS One*, vol. 8, no. 10, pp. 1–15, 2013, doi: 10.1371/journal.pone.0076783.

[6] K. C. Konstantinos Papadopoulos Marialena Barouti, “A University Indoors Audio-Tactile Mobility Aid for Individuals with Blindness,” in *Springer International Publishing Switzerland*, vol. 8548, 2014.

[7] M. C. Domingo, “An overview of the Internet of Things for people with disabilities,” *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 584–596, 2012, doi: 10.1016/j.jnca.2011.10.015.

[8] K. W. Fawzi Behmann, *Collaborative Internet of Things (C-IoT) : For Future Smart Connected Life and Business*, 1st ed. Online: John Wiley & Sons, 2015.

[9] F. Zafari, I. Papapanagiotou, and K. Christidis, “Microlocation for internet-of-things-equipped smart buildings,” *IEEE Internet Things J.*, vol. 3, no. 1, pp. 96–112, 2016, doi: 10.1109/JIOT.2015.2442956.

[10] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, “A realistic evaluation and comparison of indoor location technologies,” in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks - IPSN ’15*, 2015, no. Table 1, pp. 178–189, doi: 10.1145/2737095.2737726.

[11] A. S. Martinez-Sala, F. Losilla, J. C. Sánchez-Aarnoutse, and J. García-Haro, “Design, Implementation and Evaluation of an Indoor Navigation System for Visually Impaired People.,” *Sensors (Basel).*, vol. 15, no. 12, pp. 32168–87, 2015, doi: 10.3390/s151229912.

[12] H. K. Parikh and W. R. Michalson, “An RF System Design for an Ultra Wideband Indoor Positioning System,” Worcester Polytechnic Institute, 2008.

[13] G. R. Opshaug and P. Enge, “GPS and UWB for indoor navigation,” *Time*, vol. 25, no. September, pp. 11–14, 2001.

[14] P. T. Mahida, S. Shahrestani, and H. Cheung, “Localization techniques in indoor navigation system for visually impaired people,” *2017 17th International Symposium on Communications and Information Technologies, ISCIT*, IEEE, Cairns, Queensland, Australia, pp. 243–248, 01-Jul-2017.

[15] Mapsted, “Indoor Navigation - Beacon-free Location Technology,” *Mapsted*, 2020. [Online]. Available: https://mapsted.com/indoor-navigation. [Accessed: 13-Mar-2021].

[16] D. Plikynas, A. Žvironas, A. Budrionis, and M. Gudauskis, “Indoor navigation systems for visually impaired persons: Mapping the features of existing technologies to user needs,” *Sensors (Switzerland)*, vol. 20, no. 3, p. 636, Feb. 2020, doi: 10.3390/s20030636.

[17] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, “Enhancing the map usage for indoor location-aware systems,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2007, vol. 4551 LNCS, no. PART 2, pp. 151–160, doi: 10.1007/978-3-540-73107-8_17.

[18] J. Liu and R. Jain, “Survey of Wireless Based Indoor Localization Technologies,” *IEEE Communications Surveys & Tutorials*, Washington University St. Louis, pp. 1–

17, 2014.

[19] S. Alghamdi, R. Van Schyndel, and A. Alahmadi, "Indoor navigational aid using active RFID and QR-code for sighted and blind people," *Proceedings of the 2013 IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing: Sensing the Future, ISSNIP 2013*, vol. 1, Melbourne, VIC, Australia, pp. 18–22, 2013.

[20] J. C. and D. H. Ron Apelt, *Wayfinding design guidelines*. Brisbane, Queensland, Australia: CRC for Construction Innovation, 2007.

[21] D. Peraković, M. Periša, and A. B. Prcić, "Possibilities of Applying ICT to Improve Safe Movement of Blind and Visually Impaired Persons," in *Cutting Edge Research in Technologies*, 1st ed., V. Constanin, Ed. Croatia: IntechOpen, 2015, pp. 1–26.

[22] R. Ivanov, "RSNAVI: An RFID-based context-aware indoor navigation system for the blind," *Proceedings of the 13th International Conference on Computer Systems and Technologies - CompSysTech '12*, Ruse Bulgaria, pp. 313–320, 2012.

[23] H. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," in *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Aplications and reviews*, 2007, vol. 37, no. 6, pp. 1067–1080.

[24] K. Duarte, J. Cecilio, J. S. Silva, and P. Furtado, "Information and Assisted Navigation System for Blind People," *Int. J. Smart Sens. Intell. Syst.*, vol. 7, no. 5, pp. 1–4, 2014, doi: https://doi.org/10.21307/ijssis-2019-062.

[25] Barrows Karin, "Beginner's Guide to Blindsquare | Paths to Technology | Perkins eLearning," *BlindSquare*, 12-Nov-2016. [Online]. Available: https://www.perkinselearning.org/technology/posts/beginners-guide-blindsquare. [Accessed: 01-Jul-2021].

[26] A. F. for Blinds, "The Nearby Explorer Blindness-Focused Navigation App from APH Comes to iOS | American Foundation for the Blind | AccessWorld |," 2016. [Online]. Available: https://www.afb.org/aw/17/11/15388. [Accessed: 01-Jul-2021].

[27] A. Alarifi *et al.*, "Ultra Wideband Indoor Positioning Technologies: Analysis and

Recent Advances," *Sensors*, vol. 16, no. 5, p. 707, 2016, doi: 10.3390/s16050707.

[28] D. Jain, "Path-guided indoor navigation for the visually impaired using minimal building retrofitting," *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility - ASSETS '14*, Rochester New York USA, pp. 225–232, 2014.

[29] M. Al-Qutayri, J. Jeedella, and M. Al-Shamsi, "An integrated wireless indoor navigation system for visually impaired," *2011 IEEE International Systems Conference, SysCon 2011 - Proceedings*, Montreal, QC, Canada, pp. 17–23, Apr-2011.

[30] A. Ganz, J. Schafer, S. Gandhi, E. Puleo, C. Wilson, and M. Robertson, "PERCEPT indoor navigation system for the blind and visually impaired: Architecture and experimentation," *Int. J. Telemed. Appl.*, vol. 2012, no. Article ID 894869, p. 12 page, 2012, doi: 10.1155/2012/894869.

[31] M. Nakajima and S. Haruyama, "New indoor navigation system for visually impaired people using visible light communication," *EURASIP J. Wirel. Commun. Netw.*, vol. 2013, p. 1, 2013, doi: 10.1186/1687-1499-2013-37.

[32] L. a. Guerrero, F. Vasquez, and S. F. Ochoa, "An Indoor Navigation System for the Visually Impaired," *Sensors*, vol. 12, no. 6, pp. 8236–8258, 2012, doi: 10.3390/s120608236.

[33] M. Kiers, E. Krajnc, M. Dornhofer, and W. Bischof, "Evaluation and Improvements of an RFID Based Indoor Navigation System for Visually Impaired and Blind People," *2011 International Conference on Indoor Positioning and Indoor Navigation*, Guimaraes, Portugal., pp. 1–4, Sep-2011.

[34] V. A. Kulyukin, J. Nicholson, V. Kulyukin, and J. Nicholson, "RFID in Robot-Assisted Indoor Navigation for the Visually Impaired RFID in Robot-Assisted Indoor Navigation for the Visually Impaired," *Proceedings of 2004 IEEE International conference on intelliigent robots and systems*, no. August 2004, Sendai, Japan, pp. 1979–1984, 2004.

[35] E. Di Giampaolo, "A passive-RFID based indoor navigation system for visually impaired people," *2010 3rd International Symposium on Applied Sciences in*

Biomedical and Communication Technologies, ISABEL 2010, Rome, Italy, pp. 3–7, Nov-2010.

[36] L. Ran, S. Helal, and S. Moore, “Drishti : An Integrated Indoor / Outdoor Blind Navigation System and Service,” *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communication*, IEEE Computer, Orlando, FL, USA, Mar-2004.

[37] C. Feng *et al.*, “Anonymous indoor navigation system on handheld mobile devices for visually impaired,” *Int. J. Wirel. Inf. Networks*, vol. 19, no. 4, pp. 352–367, 2012, doi: 10.1007/s10776-012-0194-0.

[38] T. Gallagher, E. Wise, B. Li, A. G. Dempster, C. Rizos, and E. Ramsey-Stewart, “Indoor positioning system based on sensor fusion for the Blind and Visually Impaired,” *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, no. November, Sydney, NSW, Australia, pp. 1–9, Nov-2012.

[39] D. Ahmetovic, C. Gleason, K. M. Kitani, H. Takagi, and C. Asakawa, “NavCog: turn-by-turn smartphone navigation assistant for people with visual impairments or blindness,” *13th Web for All Conference, W4A 2016*, Montreal Canada, pp. 4–5, Apr-2016.

[40] N. STEIN, “LowViz Guide launched to help visually impaired people navigate,” *American foundation for Blinds*, 2017. .

[41] E. Di Giampaolo, “A passive-RFID based indoor navigation system for visually impaired people,” *2010 3rd Int. Symp. Appl. Sci. Biomed. Commun. Technol. ISABEL 2010*, pp. 3–7, 2010, doi: 10.1109/ISABEL.2010.5702800.

[42] Z. Tee, L. Ang, and K. Seng, “Smart Guide System to Assist Visually Impaired People in an Indoor Environment,” *IETE Tech. Rev.*, vol. 27, no. 6, p. 455, 2010, doi: 10.4103/0256-4602.68522.

[43] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer, “Indoor human navigation systems: A survey,” *Interact. Comput.*, vol. 25, no. 1, pp. 21–33, 2013, doi: 10.1093/iwc/iws010.

[44] M. A. Al-Ammar, H. S. Al-Khalifa, and A. S. Al-Salman, “A proposed indoor navigation system for blind individuals,” *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, Ho Chi Minh City Vietnam, pp. 527–530, Dec-2011.

[45] M. Worboys, “Modeling indoor space,” *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness*, no. January 2011, New York, United States, pp. 1–6, Nov-2011.

[46] A. Botea, B. Bouzy, M. Buro, C. Bauckhage, and D. Nau, “Pathfinding in Games,” *Artif. Comput. Intell. Games*, vol. 6, pp. 21–31, 2014, doi: 10.4230/DFU.Vol6.12191.21.

[47] J.-S. Kim, Y. Han, and K.-J. Li, “K-Anonymity in Indoor Spaces Through Hierarchical Graphs,” *Proceedings of the Fourth ACM SIGSPATIAL International Workshop on Indoor Spatial Awareness - ISA ’2012*, New York, United States, pp. 21–28, Nov-2012.

[48] S. Ma, Q. Liu, and H. Tang, “An Overview of Location Semantics Technologies and Applications,” *Int. J. Semant. Comput.*, vol. 9, no. 3, SI, pp. 373–393, 2015, doi: 10.1142/S1793351X15500051.

[49] C. S. Jensen, H. L. H. Lu, and B. Y. Bin Yang, “Graph Model Based Indoor Tracking,” *2009 Tenth International Conference on Mobile Data Management Systems Services and Middleware*, Taipei, Taiwan, pp. 122–131, Jun-2009.

[50] K. F. Richter, S. Winter, and S. Santosa, “Constructing Hierarchical Representations of Indoor Spaces,” *Environ. Plan. B Urban Anal. City Sci.*, vol. 38, no. 6, pp. 1052–1070, 2011, doi: 10.1068/b37057.

[51] Z. Abd Algfoor, M. S. Sunar, and H. Kolivand, “A comprehensive study on pathfinding techniques for robotics and video games,” *Int. J. Comput. Games Technol.*, vol. 2015, no. Article ID 736138, p. 11 pages, 2015, doi: 10.1155/2015/736138.

[52] J. O. Wallgriin, “Hierarchical voronoi graphs: Spatial representation and reasoning for mobile robots,” in *Hierarchical Voronoi Graphs: Spatial Representation and Reasoning for Mobile Robots*, Verlag Berlin Heidelberg: Springer, 2010, pp. 1–218.

[53] A. Patel, “Map representations From Amit’s Thoughts on Pathfinding,” *Stanford CS Theory*, 2020. [Online]. Available: http://theory.stanford.edu/~amitp/GameProgramming/MapRepresentations.html. [Accessed: 12-Apr-2017].

[54] S. B. Balakirsky, *A Framework for Planning with Incrementally Created Graphs in Attributed Problem Spaces*, vol. 270. National Institute of Standards and Technology, Gaithersburg: IOS Press, 2003.

[55] N. R. Sturtevant, “Choosing a State Space Representation,” in *Game AI Pro 360*, 1st ed., vol. 1, no. c, CRC Press, 2019, pp. 253–258.

[56] M. Kneebone and R. Dearden, “Navigation Planning in Probabilistic Roadmaps with Uncertainty,” *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009*, Thessaloniki, Greece, Sep-2009.

[57] S. M. LaValle, “Rapidly-Exploring Random Trees: A New Tool for Path Planning,” *Annu. Res. Rep.*, vol. 129, pp. 1–4, 1998, doi: 10.1.1.35.1853.

[58] D. Nandini and K. R. Seeja, “A novel path planning algorithm for visually impaired people,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, no. 3, pp. 385–391, 2017, doi: 10.1016/j.jksuci.2017.03.005.

[59] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[60] R. Ammar and S. Samer, *Internet of Things from hype to reality The Road to Digitization*, Second. Springer Nature Switzerland, 2019.

[61] Rob Barton; Gonzalo Salgueiro; David Hanes, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*, 1st ed. Indianapolis, USA: Cisco Press, 2017.

[62] M. C. Mahto Holdowsky, JonathanMonika, Michael Raynor, “A primer on technologies building the Internet of Things | Deloitte Insights.” [Online]. Available: https://www2.deloitte.com/us/en/insights/focus/internet-of-things/iot-primer-iot-

technologies-applications.html. [Accessed: 01-May-2021].

[63] L. Ran, S. Helal, and S. Moore, “Drishti : An Integrated Indoor / Outdoor Blind Navigation System and Service,” *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communication*, Orlando, FL, USA, Mar-2004.

[64] A. Bensky, *Wireless Positioning Technologies and Applications, GNSS Technology and Applications*, 2nd ed. MA, United States: Artech House Publishers, 2016.

[65] R. Mautz, “Indoor positioning technologies,” ETH Zurich, Department of Civil, Environmental and Geomatic Engineering, Institute of Geodesy and Photogrammetry, Zurich, 2012.

[66] H. D. Chon, S. Jun, H. Jung, and S. W. An, “Using RFID for Accurate Positioning,” *Jorunal Glob. Position. Syst.*, vol. 3, no. 1, pp. 32–39, 2005, doi: 10.5081/jgps.3.1.32.

[67] S. Singh and A. Aggarwal, “Survey on Localization Techniques of RFID for IOT,” *Int. J. Comput. Appl.*, vol. 137, no. 12, pp. 975–8887, 2016, doi: 10.5120/ijca2016908989.

[68] Beaconstac, “Beacons.” [Online]. Available: https://www.beaconstac.com/what-is-a-bluetooth-beacon. [Accessed: 23-Mar-2020].

[69] D. Zhang, F. Xia, Z. Yang, and L. Yao, “Localization Technologies for Indoor Human Tracking,” *Proceedings 5th International Conference on Future Information Technology, FutureTech 2010*, no. 60903153, Busan, Korea (South), Jun-2010.

[70] F. Chollet, *Deep Learning with Python*, 2nd ed. Manning Publishers, 2020.

[71] M. Z. Alom *et al.*, “A State-of-the-Art Survey on Deep Learning Theory and Architectures,” *Electronics*, vol. 8, no. 3, p. 292, Mar. 2019, doi: 10.3390/electronics8030292.

[72] T. Van Haute *et al.*, “Performance analysis of multiple Indoor Positioning Systems in a healthcare environment,” *Int. J. Health Geogr.*, vol. 15, no. 1, p. 7, 2016, doi: 10.1186/s12942-016-0034-z.

[73] H. Hu and D.-L. Lee, “Semantic location modeling for location navigation in mobile environment,” *IEEE International Conference on Mobile Data Management, 2004.*

2004, HongKong, pp. 52–61, 2004.

[74] A. Alempijevic, R. Fitch, and N. Kirchner, “Bootstrapping navigation and path planning using human positional traces,” *Proceedings - IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, pp. 1242–1247, Oct-2013.

[75] Y. Xu, Z. Wen, and X. Zhang, “Indoor optimal path planning based on Dijkstra Algorithm,” *International Conference on Materials Engineering and Information Technology Applications (MEITA 2015)*, no. 28, Guilin, China, pp. 309–313, Aug-2015.

[76] S. A. Fadzli, S. I. Abdulkadir, M. Makhtar, and A. A. Jamal, “Robotic indoor path planning using dijkstra’s algorithm with multi-layer dictionaries,” *IEEE 2nd International Conference on InformationScience and Security, ICISS 2015*, no. 2, Seoul, Korea (South), pp. 45–56, Jan-2016.

[77] L. Liu and S. Zlatanova, “A ‘Door-To-Door’ Path-Finding Approach For Indoor Navigation,” *Proc. Gi4DM 2011 Geoinf. Disaster Manag. Antalya, Turkey, 3-8 May 2011*, pp. 3–8, 2011.

[78] R. Ivanov, “An approach for microscopic path finding and obstacle avoidance for blind and visually impaired people,” *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 - CompSysTech ’16*, no. June, Palermo, Italy, pp. 285–292, Jun-2016.

[79] A. Goyal, “Path-Finding Methodology for Visually-Impaired Patients Based on Image-Processing,” San Jose, California, 2017.

[80] M. Khedr and N. El-Sheimy, “A smartphone step counter using IMU and magnetometer for navigation and health monitoring applications,” *Sensors , MDPI*, vol. 17, no. 2573, 2017, doi: https://doi.org/10.3390/s17112573.

[81] P. Davidson and R. Piché, “A Survey of Selected Indoor Positioning Methods for Smartphones,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 1347–1370, 2017, doi: 10.1109/COMST.2016.2637663.

[82] K. Tumkur and S. Subbiah, “Modeling Human Walking for Step Detection and Stride

Determination by 3-Axis Accelerometer Readings in Pedometer," *2012 Fourth International Conference on Computational Intelligence, Modelling and Simulation*, IEEE, Kuantan, Malaysia, pp. 199–204, Sep-2012.

[83] H. H. Lee, S. Choi, and M. J. Lee, "Step detection robust against the dynamics of smartphones," *Sensors (Switzerland)*, vol. 15, no. 10, pp. 27230–27250, 2015, doi: 10.3390/s151027230.

[84] L. Ojeda and J. Borenstein, "Non-GPS Navigation for Security Personnel and First Responders," *J. Navig.*, vol. 60, no. 3, pp. 391–407, 2007.

[85] A. R. Jimenez, F. Seco, J. C. Prieto, and J. Guevara, "Indoor Pedestrian navigation using an INS/EKF framework for yaw drift reduction and a foot-mounted IMU," *Proceedings of the 2010 7th Workshop on Positioning, Navigation and Communication, WPNC'10*, no. April, Dresden, Germany, pp. 135–143, Mar-2010.

[86] Z. Xu *et al.*, "Utilizing High-level Visual Feature for Indoor Shopping Mall Navigation," *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Montreal, QC, Canada, pp. 1378–1382, Nov-2017.

[87] S. H. P. Won, W. W. Melek, and F. Golnaraghi, "Remote Sensing Technologies for Indoor Applications," *Handb. Position Locat. Theory, Pract. Adv.*, pp. 649–684, 2011, doi: 10.1002/9781118104750.ch20.

[88] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. Fangmeyer, "Evolution of Indoor Positioning Technologies: A Survey," *J. Sensors*, vol. 2017, pp. 1–21, 2017, doi: 10.1155/2017/2630413.

[89] C. Tsirmpas, A. Rompas, O. Fokou, and D. Koutsouris, "An indoor navigation system for visually impaired and elderly people based on Radio Frequency Identification (RFID)," *Inf. Sci. (Ny).*, vol. 320, pp. 288–305, 2015, doi: 10.1016/j.ins.2014.08.011.

[90] C.-C. Chen, C.-Y. Chang, and Y.-N. Li, "Range-Free Localization Scheme in Wireless Sensor Networks Based on Bilateration," *Int. J. Distrib. Sens. Networks*, vol. 10, no. Article ID 620248, pp. 1–10, 2013, doi: 10.1155/2013/620248.

[91] M. Nowicki and J. Wietrzykowski, "Low-effort place recognition with WiFi

fingerprints using deep learning," *Adv. Intell. Syst. Comput.*, vol. 550, pp. 575–584, 2017, doi: 10.1007/978-3-319-54042-9_57.

[92] H. Mehmood and N. K. Tripathi, "Optimizing artificial neural network-based indoor positioning system using genetic algorithm," *Int. J. Digit. Earth*, vol. 6, no. 2, pp. 158–184, 2013, doi: 10.1080/17538947.2011.606337.

[93] C. H. Hsieh, J. Y. Chen, and B. H. Nien, "Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information," *IEEE Access*, vol. 7, pp. 33256–33267, 2019, doi: 10.1109/ACCESS.2019.2903487.

[94] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Min. Knowl. Discov.*, vol. 33, pp. 917–963, 2019, doi: 10.1007/s10618-019-00619-1.

[95] A. Adege, H.-P. Lin, G. Tarekegn, and S.-S. Jeng, "Applying Deep Neural Network (DNN) for Robust Indoor Localization in Multi-Building Environment," *Appl. Sci.*, vol. 8, no. 7, p. 1062, Jun. 2018, doi: 10.3390/app8071062.

[96] L. Wu, C.-H. Chen, and Q. Zhang, "A Mobile Positioning Method Based on Deep Learning Techniques," *Electronics*, vol. 8, no. 1, p. 59, Jan. 2019, doi: 10.3390/electronics8010059.

[97] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent Neural Networks for Accurate RSSI Indoor Localization," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019, doi: 10.1109/JIOT.2019.2940368.

[98] W. Liu, H. Chen, Z. Deng, X. Zheng, X. Fu, and Q. Cheng, "LC-DNN: Local Connection Based Deep Neural Network for Indoor Localization with CSI," *IEEE Access*, vol. 8, pp. 108720–108730, 2020, doi: 10.1109/ACCESS.2020.3000927.

[99] L. Zhang, E. Ding, Y. Hu, and Y. Liu, "A novel CSI-based fingerprinting for localization with a single AP," *Eurasip J. Wirel. Commun. Netw.*, vol. 2019, no. 1, p. 51, Dec. 2019, doi: 10.1186/s13638-019-1371-y.

[100] K. P. Subbu, Brandon Gozick, and R. Dantu, "LocateMe : Magnetic-fields-based Indoor localization using smartphones," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no.

4, Article 73, p. 27 pages, 2013, doi: 10.1145/2508037.2508054.

[101] Y. Shu, K. G. Shin, T. He, and J. Chen, "Last-Mile Navigation Using Smartphones," in *MobiCom '15: Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 512–524, doi: https://doi.org/10.1145/2789168.2790099.

[102] H. J. Jang, J. M. Shin, and L. Choi, "Geomagnetic field based indoor localization using recurrent neural networks," *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Institute of Electrical and Electronics Engineers Inc., Singapore, pp. 1–6, Dec-2017.

[103] G. Ligorio and A. Sabatini, "Extended Kalman Filter-Based Methods for Pose Estimation Using Visual, Inertial and Magnetic Sensors: Comparative Analysis and Performance Evaluation," *Sensors*, vol. 13, no. 2, pp. 1919–1941, 2013, doi: 10.3390/s130201919.

[104] S. Shahidi and S. Valaee, "GIPSy: Geomagnetic indoor positioning system for smartphones," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–7, doi: 10.1109/IPIN.2015.7346761.

[105] D. Gonzalez-Arjona, A. Sanchez, F. López-Colino, A. de Castro, and J. Garrido, "Simplified Occupancy Grid Indoor Mapping Optimized for Low-Cost Robots," *ISPRS Int. J. Geo-Information*, vol. 2, no. 4, pp. 959–977, 2013, doi: 10.3390/ijgi2040959.

[106] D.-L. Almanza-Ojeda, Y. Gomar-Vera, and M.-A. Ibarra-Manzano, "Occupancy Map Construction for Indoor Robot Navigation," in *Robot Control*, 1st ed., IntechOpen, Ed. InTech, 2016, pp. 69–88.

[107] M. Salem, "Building an Efficient Occupancy Grid Map Based on Lidar Data Fusion for Autonomous driving Applications," Kth Royal Institute of Technology, 2019.

[108] S. Wirges, C. Stiller, and F. Hartenbach, "Evidential Occupancy Grid Map Augmentation using Deep Learning," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2018-June, Institute of Electrical and Electronics Engineers Inc., Changshu, Suzhou, China, pp. 668–673, 18-Oct-2018.

[109] T. Graichen, R. Schmidt, J. Richter, and U. Heinkel, "Occupancy Grid Map Generation from OSM Indoor Data for Indoor Positioning Applications," *Proceedings of the 6th International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2020)*, pp. 168–174, 2020.

[110] P. T. Mahida, S. S. Shahrestani, and H. Cheung, "Comparision of pathfinding algorithms for visually impaired people in IoT based smart buildings," *International Telecommunication networks and applications conference (ITNAC 2018)*, UNSW, Sydney, Australia, pp. 10–13, Nov-2018.

[111] N. R. Sturtevant and R. Geisberger, "A Comparison of High-Level Approaches for Speeding Up Pathfinding," *Proceedings of the 6th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Palo Alto, CA, pp. 76–82, 2010.

[112] H. H. Pham, T. L. Le, and N. Vuillerme, "Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor," *J. Sensors*, vol. 2016, no. Article ID 3754918, pp. 1–13, 2016, doi: 10.1155/2016/3754918.

[113] J. Cecio, K. Duarte, and P. Furtado, "BlindeDroid: An information tracking system for real-time guiding of blind people," *Procedia Comput. Sci.*, vol. 52, no. 1, pp. 113–120, 2015, doi: 10.1016/j.procs.2015.05.039.

[114] L. Mutlu and E. Uyar, "Indoor navigation and guidance of an autonomous robot vehicle with static obstacle avoidance and optimal path finding algorithm," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2012, vol. 45, no. 24, pp. 315–319, doi: 10.3182/20120912-3-BG-2031.00067.

[115] C. L. Lee, C. Y. Chen, P. C. Sung, and S. Y. Lu, "Assessment of a simple obstacle detection device for the visually impaired," *Appl. Ergon.*, vol. 45, no. 4, pp. 817–824, 2014, doi: 10.1016/j.apergo.2013.10.012.

[116] S. M. Lavalle, "Motion Planning," in *Planning Algorithms*, 1st ed., Urbana, Illinois, U.S.A: Cambridge University Press, 2006, pp. 373–416.

[117] N. Kumar, V. Zoltán, and M. S.-R. Zsolt, "Heuristic approaches in robot navigation," *2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems*

(INES), IEEE, Budapest, Hungary, pp. 219–222, Jun-2016.

[118] T. Podhraski, “How to Speed Up A* Pathfinding With the Jump Point Search Algorithm,” 2013. [Online]. Available: gamedevelopment.tutsplus.com/tutorials/how-to-speed-up-a-pathfinding-with-the-jump-point-search-algorithm--gamedev-5818. [Accessed: 31-Jul-2017].

[119] D. Ferguson, M. Likhachev, and A. Stentz, “A guide to heuristic-based path planning,” *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS)*, Monterey, California, U.S.A, pp. 1–10, Jun-2005.

[120] H. Wu, A. Marshall, and W. Yu, “Path planning and following algorithms in an indoor navigation model for visually impaired,” *Second International Conference on Internet Monitoring and Protection, ICIMP 2007*, San Jose, California, USA, pp. 38–44, Jul-2007.

[121] A. Patel, “Heuristics From Amit’s Thoughts on Pathfinding,” *Stanford CS Theory*, 2015. .

[122] M. Stuart Bruce and M. S. Branicky, “Sampling based planning for discrete spaces,” *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Japan, pp. 1938–1945, Sep-2004.

[123] J. V. Zegarra Flores, L. Rasseneur, R. Galani, F. Rakitic, and R. Farcy, “Indoor navigation with smart phone IMU for the visually impaired in university buildings,” *J. Assist. Technol.*, vol. 10, no. 3, pp. 133–139, 2016, doi: 10.1108/JAT-05-2015-0018.

[124] J. Dijkstra, B. De Vries, and J. Jessurun, “Wayfinding search strategies and matching familiarity in the built environment through virtual navigation,” *Transp. Res. Procedia*, vol. 2, pp. 141–148, 2014, doi: 10.1016/j.trpro.2014.09.018.

[125] S. M.MasudurRahmanAl-Arif, a. H. M. Iftekharul Ferdous, and S. Hassan Nijami, “Comparative Study of Different Path Planning Algorithms: A Water based Rescue System,” *Int. J. Comput. Appl.*, vol. 39, no. 5, pp. 25–29, 2012, doi: 10.5120/4817-7058.

[126] M. Miler, D. Medak, and D. Odobasic, “The Shortest Path Algorithm Performance Comparison in Graph and Relational Database on a Transportation Network,” *PROMET-Traffic&Transportation*, vol. 26, no. 1, pp. 75–82, 2013.

[127] U. Md. Ashraf and S. Ashraful Huq, “Shortest path finding and obstacle detection for visually impaired people using smart phone,” *2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, IEEE, Savar, Bangladesh, pp. 1–4, May-2015.

[128] M. W. Otte, “A Survey of Machine Learning Approaches to Robotic Path-Planning,” *PhD Thesis. University of Colorado at Boulder*, 2015. [Online]. Available: http://www.cs.colorado.edu/~mozer/Teaching/Computational Modeling Prelim/Otte.pdf. [Accessed: 12-May-2019].

[129] L. Jaillet, J. Cort, and T. Sim, “Sampling-Based Path Planning on Costmaps Configuration-space,” *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 635–646, 2010, doi: 10.1109/TRO.2010.2049527.

[130] S. Karaman and E. Frazzoli, “Sampling-based Algorithms for Optimal Motion Planning,” *Spec. Issue Robot. Sci. Syst. 2010*, vol. 30, no. 7, pp. 846–894, 2011, doi: 10.1177/0278364911406761.

[131] S. M. Lavalle, “Sampling-Based Motion Planning,” in *Planning Algorithms*, 1st ed., Urbana, Illinois, U.S.A: Cambridge University Press, 2006.

[132] M. Otte and N. Correll, “Path Planning with Forests of Random Trees : Parallelization with Super Linear Speedup,” *Science (80-.).*, no. April, 2011.

[133] D. Harabor and A. Grastien, “Online Graph Pruning for Pathfinding On Grid Maps.,” *AAAI Conference on Artificial Intelligence*, San Francisco, United States of America, pp. 1114–1119, Aug-2011.

[134] D. Zeinalipour-yazti, C. Laoudias, K. Georgiou, and G. Chatzimilioudis, “Internet-based Indoor Navigation Services,” *IEEE Internet Comput.*, vol. 21, pp. 54–63, 2015, doi: doi: 10.1109/MIC.2017.2911420.

[135] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,”

Computer (Long. Beach. Calif)., vol. 34, no. 8, pp. 57–66, 2001, doi: 10.1109/2.940014.

[136] P. T. Mahida, S. Shahrestani, and H. Cheung, “DynaPATH: Dynamic Learning Based Indoor Navigation for VIP in IoT Based Environments,” *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, IEEE, Sydney, NSW, Australia, pp. 8–13, Dec-2018.

[137] M. Park and Y. Gao, “Error and Performance Analysis of MEMS-based Inertial Sensors with a Low-cost GPS Receiver,” *Sensors*, vol. 8, no. 4, pp. 2240–2261, Mar. 2008, doi: 10.3390/s8042240.

[138] R. Ivanov, “Indoor navigation system for visually impaired,” *Proc. 11th Int. Conf. Comput. Syst. Technol. Work. PhD Students Comput. Int. Conf. Comput. Syst. Technol. - CompSysTech ’10*, vol. 12, no. 6, p. 143, 2010, doi: 10.1145/1839379.1839405.

[139] S. Adler, S. Schmitt, K. Wolter, and M. Kyas, “A survey of experimental evaluation in indoor localization research,” *Indoor Positioning and Indoor Navigation (IPIN), 2015 International Conference on*, no. October, Banff, AB, Canada, pp. 1–10, Oct-2015.

[140] J. Victor, Z. Flores, L. Rasseneur, R. Galani, J. Victor, and Z. Flores, “Indoor navigation with smart phone IMU for the visually impaired in university buildings,” *J. Assist. Technol.*, 2016, doi: 10.1108/JAT-05-2015-0018.

[141] A. Noureldin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, “Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost Navigation Applications,” *IEEE Trans. Veh. Technol.*, vol. 58, no. 3, pp. 1077–1096, Mar. 2009, doi: 10.1109/TVT.2008.926076.

[142] P. T. Mahida, S. Shahrestani, and H. Cheung, “Indoor positioning framework for visually impaired people using Internet of Things,” *International Conference on Sensing Technology*, Sydney, NSW, Australia, pp. 198–203, 2019.

[143] Lucien MADL, “Attitude Heading Reference System and Step Recognition for Cloud-Based Indoor Positioning – Android Client,” 2018.

[144] “Magnetometer - an overview | ScienceDirect Topics.” [Online]. Available:

https://www.sciencedirect.com/topics/engineering/magnetometer. [Accessed: 07-Apr-2019].

[145] K. Gade, "The Seven Ways to Find Heading," *J. Navig.*, vol. 69, no. 5, pp. 955–970, 2016, doi: 10.1017/S0373463316000096.

[146] P. T. Mahida, S. Shahrestani, and H. Cheung, "An improved positioning method in a smart building for visually impaired user," *International Conference on Internet of Things Research and Practice (iCIOTRP2019)*, IEEE, Sydney, NSW, Australia, pp. 7–12, 2019.

[147] A. E. Maxwell, T. A. Warner, and F. Fang, "Implementation of machine-learning classification in remote sensing: an applied review," *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018, doi: 10.1080/01431161.2018.1433343.

[148] Y. Lu, D. Wei, Q. Lai, W. Li, and H. Yuan, "A Context-Recognition-Aided PDR Localization Method Based on the Hidden Markov Model," *Sensors*, vol. 16, no. 12, p. 2030, 2016, doi: 10.3390/s16122030.

[149] P. Mahida, S. Shahrestani, and H. Cheung, "Deep Learning-Based Positioning of Visually Impaired People in Indoor Environments," *Sensors*, vol. 20, no. 21, p. 6238, Oct. 2020, doi: 10.3390/s20216238.

[150] P. Barsocchi, A. Crivello, D. La Rosa, and F. Palumbo, "A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting," *2016 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2016*, no. October, IEEE, Alcala de Henares, Spain, pp. 1–8, Oct-2016.

[151] W. Shao *et al.*, "Location Fingerprint Extraction for Magnetic Field Magnitude Based Indoor Positioning," *J. Sensors*, vol. 2016, no. Article ID 1945695, p. 16 pages, 2016, doi: https://doi.org/10.1155/2016/1945695.

[152] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," *MAchine learning Mastering*, 2020. [Online]. Available: https://machinelearningmastery.com/k-fold-cross-validation/. [Accessed: 21-Jan-2020].

[153] A. Mashlakov, V. Tikka, L. Lensu, A. Romanenko, and S. Honkapuro, “Hyper-parameter optimization of multi-attention recurrent neural network for battery state-of-charge forecasting,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11804 LNAI, pp. 482–494, doi: 10.1007/978-3-030-30241-2_41.

[154] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” *2nd International Conference on Computational Sciences and Technology, (INCCST) 2020*, Jamshoro, Sindh Pakistan, p. 20 pages, 08-Nov-2018.