

# **Energy Conservation in Wireless Sensor Networks**

**Ph.D. Thesis**

By  
**Rupendra Pratap Singh Hada**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

**March 2025**



# Energy Conservation in Wireless Sensor Networks

**A Thesis**

*Submitted in partial fulfillment of the  
requirements for the award of the degrees  
of*  
**Doctor of Philosophy**

*by*

**Rupendra Pratap Singh Hada**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

**March 2025**





# INDIAN INSTITUTE OF TECHNOLOGY INDORE

I hereby certify that the work which is being presented in the thesis entitled **Energy Conservation in Wireless Sensor Networks** in the partial fulfilment of the requirements for the award of the degree of **DOCTOR OF PHILOSOPHY** and submitted in the **Department of Computer Science & Engineering**, Indian Institute of Technology Indore, is an authentic record of my own work carried out during the time period from **August 2021** to **March 2025** under the supervision of **Prof. Abhishek Srivastava**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other institute.

07/07/2025

Signature of the student with date

(Rupendra Pratap Singh Hada)

-----  
This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

07/07/2025

Signature of Thesis Supervisor with date

(Prof. Abhishek Srivastava)

-----  
**Rupendra Pratap Singh Hada** has successfully given his Ph.D. Oral Examination held on 07/07/2025.

07/07/2025

Signature of Thesis Supervisor with date

(Prof. Abhishek Srivastava)

-----

## ACKNOWLEDGEMENTS

First and foremost, I'd like to express my heartfelt gratitude to my advisor, **Prof. Abhishek Srivastava**, for introducing me to this exciting field of research and giving me the opportunity to work under his never-ending source of knowledge and experience. I continue to learn from his dedication to research, courage in exploring the unknown, enjoyment of research problems, and extensive knowledge in the field. His encouragement, support, and excellent suggestions throughout the research process helped me to complete my thesis successfully. His mentoring not only assisted me academically in completing this thesis, but he also provided me with the moral support and freedom I required to move forward. I could not have imagined having a more incredible advisor and mentor for my PhD studies.

Besides my advisor, I thank my PSPC members, **Prof. Amod C. Umarikar** and **Dr. Bodhisatwa Mazumdar**, for their helpful suggestions and comments during my progress report presentations. These helped me widen my research from various perspectives. I also thank **Prof. Somnath Dey** and **Dr. Ranveer Singh**, Head of the Department of Computer Science and Engineering, for their support and help.

My heartfelt gratitude to the director of the Indian Institute of Technology Indore, **Prof. Suhas Joshi**, for creating a competitive research environment at the institution. Special thanks to the staff of the Indian Institute of Technology Indore, particularly **Mr. Shailendra Verma**, **Mrs. Ujavala Gorakh Langhi** and **Mr. Jagat Singh** for their assistance with academics and for always being willing to support me in any official task. I'd also like to thank the Indian Institute of Technology Indore for the opportunity to pursue a PhD in Computer Science and Engineering.

I am grateful to my many student colleagues for the fascinating talks and all the fun we've had over the last three and a half years. My heartfelt gratitude goes to **Sneha Shukla**, who has always been a massive source of encouragement and celebrated each achievement with me. We also had several rounds of conversation prior to each submission, and her constructive feedback was highly beneficial to me.

I want to express my appreciation to my friend **Ms. Shrutika Chouhan**, whose encouragement was crucial and kept me going the whole time. I would also like to thank

my wonderful labmates, **Dr. Ankit Jain, Shekhar Tyagi, Shibani Das, Urvesh Trivedi, Anukarsh Pratap, Sudhanshu Trivedi, Money Bansal, and Himanshu Mishra**, who had, in their own ways, kept me going on my path to success, assisting me as per their abilities, in whatever manner possible. I want to extend my thankfulness to **Drishti Sharma** for always being there to motivate me in difficult times. I would like to thank my friend **Anup Kumar Gupta**, who helped me in the initial days of IITI and research. I would also like to thank my loving juniors (**Rupesh Kumar, Prasanna Bairagi, Prasad Choudhary, Azahar Shaikh, Ajinkya Kulkarni, Balaram Sarkar, Mithun Singh, Priyansh Singh, Deepak Prasad, Prajaval Dhiman, Shabbir Poswal, Mukul Jain, Sarvagya Kanoongo, Govind KM**) for constant support. Although they are my academic juniors, I got to learn so many things from them.

My appreciation would be inadequate until I thanked the most important source of my strength: **my family**. The blessings of my late **grandparents**, my parents, **Mrs. Shivkanta Singh** and **Mr. Sodan Singh**, as well as the affection and care of my brothers **Mr. Hemendra Singh Hada** and **Mr. Manish Dubey**, and my nephew **Bhavya Raj Singh**, have all played an essential role in my success. This would not have been possible without their constant and selfless love and support for me.

***Rupendra Pratap Singh Hada***

***Dedicated to My Family, Friends  
and Respected Teachers***



## ABSTRACT

A wireless Sensor Network (WSN) is a self-configured *ad hoc* network comprising nodes with built-in processing units, communication systems, and sensors. WSNs are deployed in mostly outdoor terrains and play a crucial role in monitoring such terrains through their sensing units and communication systems. Given the lack of accessibility in terrains that WSNs are deployed, these often work under extreme resource constraints with respect to energy, memory, and computational capabilities. WSN nodes mostly rely on alkaline batteries for energy and it becomes imperative that these be frugally used so as to extend their lifespan and in effect that of the WSN node.

In this thesis, we explore this aspect of WSN and make contributions towards preserving the energy of the nodes whilst simultaneously getting useful work done. We specifically look at the exercise of economically localising WSN nodes in vast deployments; and routing of communication signals through these networks.

Localisation of WSN nodes implies establishing the precise location of specific nodes within the network that have made interesting observations. This is non-trivial given the resource intensive nature of global positioning systems which makes them inappropriate for such resource constrained environments. We take two approaches to address this issue: the first comprises a hybrid approach that combines the Angle of Arrival (AoA) and Received Signal Strength Indicator (RSSI) of signals received at nodes of interest. The values of AoA and RSSI enables in establishing the location of unknown nodes *vis-a-vis* nodes with pre-determined locations.

The second approach to localisation comprises a machine learning-based method that employs an improvised random forest (RF) algorithm combined with RSSI values to determine the precise location of the nodes of interest.

The other major energy intensive exercise that a WSN has to indulge in is communication of signals. We attempt to alleviate this through a novel routing approach that effectively minimizes the energy expended in communication. The approach comprises a cluster-based routing strategy that dynamically creates clusters of nodes and selects a cluster head for effective funneling of signals from the nodes to the base station.

The proposed approaches for localisation and routing whilst preserving energy are rigorously tested and shown to be superior to existing state of the art approaches for the same. Prototypes of each of the ideas are practically tested in real-world deployments and demonstrated to be effective.

## LIST OF PUBLICATIONS

### (A) From PhD thesis work:

#### A1. Journal Articles:

##### Published/Accepted:

- J1. Rupendra Pratap Singh Hada**, and Abhishek Srivastava. “**A Hybrid Approach for Localisation of Sensor Nodes in Remote Locations**”, *ACM Transactions on Sensor Networks*, Vol. 21, issue 2, article 23, 1-33, 2025.
- J2. Rupendra Pratap Singh Hada**, and Abhishek Srivastava. “**Dynamic Cluster Head Selection in WSN**”, *ACM Transactions on Embedded Computing Systems*, Vol. 23, issue 4, article 64, 1-27, 2024.
- J3. Rupendra Pratap Singh Hada**, Uttkarsh Aggarwal, and Abhishek Srivastava. “**A Study and Analysis of a New Hybrid Approach for Localization in Wireless Sensor Networks**”, *Journal of Web Engineering* 22.2: 279-302, 2023.

### (B) Other publications during PhD:

#### B1. Journal Articles: Published:

- J1. Rupendra Pratap Singh Hada**, and Abhishek Srivastava, “**Priority Based Scheduler for Asymmetric Multi-core Edge Computing**”, *Journal of Web Engineering*, 22.6 (2023): 871-888.

#### B2. Conference Articles:

- C1. Rupendra Pratap Singh Hada**, and Abhishek Srivastava, “**A novel priority based scheduler for asymmetric multi-core edge computing**”, *International Conference on Web Engineering*, Cham: Springer Nature Switzerland, 2023.

# Contents

|                                                                            |             |
|----------------------------------------------------------------------------|-------------|
| <b>List of Figures</b>                                                     | <b>iv</b>   |
| <b>List of Tables</b>                                                      | <b>vii</b>  |
| <b>List of Abbreviations</b>                                               | <b>viii</b> |
| <b>1 Introduction</b>                                                      | <b>1</b>    |
| 1.1 Motivation . . . . .                                                   | 2           |
| 1.1.1 Thesis Contributions . . . . .                                       | 4           |
| 1.1.2 Preserving Energy in Localisation through simple computations . .    | 4           |
| 1.1.3 Preserving Energy in Localisation harnessing machine learning models | 5           |
| 1.1.4 Preserving Energy via Cluster-based Routing . . . . .                | 6           |
| 1.2 Thesis Organisation . . . . .                                          | 6           |
| <b>2 Literature Review</b>                                                 | <b>9</b>    |
| 2.1 Mathematics-based Localisation . . . . .                               | 9           |
| 2.2 ML-based Localisation . . . . .                                        | 14          |
| 2.3 Cluster-based Routing . . . . .                                        | 16          |
| <b>3 Preserving WSN Energy in Localisation using Mathematical</b>          |             |
| <b>    Model</b>                                                           | <b>19</b>   |
| 3.1 Understanding the Problem . . . . .                                    | 19          |
| 3.2 The Proposed Method . . . . .                                          | 20          |
| 3.2.1 RSSI and Distance calculation . . . . .                              | 21          |
| 3.2.2 Angle of Arrival (AoA) . . . . .                                     | 23          |
| 3.2.3 Multi-iteration . . . . .                                            | 25          |

|          |                                                             |           |
|----------|-------------------------------------------------------------|-----------|
| 3.2.4    | Network localisability                                      | 26        |
| 3.3      | Evaluation                                                  | 27        |
| 3.3.1    | Simulation Environment and parameters                       | 27        |
| 3.3.2    | Simulation Results                                          | 30        |
| 3.3.3    | Comparison with SOTA                                        | 46        |
| 3.4      | Limitations                                                 | 51        |
| 3.5      | Conclusion and Future Work                                  | 51        |
| <b>4</b> | <b>Preserving WSN Energy in Localisation using Machine-</b> |           |
|          | <b>Learning Model</b>                                       | <b>53</b> |
| 4.1      | Understanding the Problem                                   | 53        |
| 4.2      | The Proposed Method                                         | 55        |
| 4.2.1    | Localisation using Machine Learning                         | 58        |
| 4.2.2    | Localisation through Multilateration                        | 62        |
| 4.2.3    | The Hybrid Approach to Localisation                         | 64        |
| 4.3      | Evaluation                                                  | 66        |
| 4.3.1    | Dataset and Simulated Environment                           | 66        |
| 4.3.2    | Machine Learning (Random Forest) Localisation               | 68        |
| 4.3.3    | Multilateration Localisation                                | 73        |
| 4.3.4    | The Hybrid Localisation Approach                            | 76        |
| 4.4      | Limitations                                                 | 79        |
| 4.5      | Conclusion and Future Work                                  | 79        |
| <b>5</b> | <b>Preserving WSN Energy via Cluster Based Routing</b>      | <b>81</b> |
| 5.1      | Understanding the Problem                                   | 81        |
| 5.2      | The Proposed Method                                         | 82        |
| 5.2.1    | Determining the optimal cluster numbers                     | 84        |
| 5.2.2    | Initial Clustering                                          | 85        |
| 5.2.3    | Cluster Head Selection                                      | 86        |
| 5.2.4    | Secondary clustering                                        | 87        |
| 5.2.5    | Residual energy threshold                                   | 88        |

|       |                                          |     |
|-------|------------------------------------------|-----|
| 5.2.6 | Energy model for data transmission       | 89  |
| 5.3   | Evaluation                               | 90  |
| 5.3.1 | Simulation Environment and parameters    | 91  |
| 5.3.2 | Comparison with State-of-the-art methods | 95  |
| 5.4   | Limitations                              | 104 |
| 5.5   | Conclusion and Future Work               | 105 |
| 6     | Conclusions and Future Works             | 107 |

# List of Figures

|                                                                     |    |
|---------------------------------------------------------------------|----|
| 1.1 Flow Graph depicts the organisation of the thesis.              | 7  |
| 3.1 Position calculation of any unknown node using the beacon node. | 25 |
| 3.2 localisation using multi-iteration approach.                    | 26 |
| 3.3 Multi-iteration Process in WSN.                                 | 26 |
| 3.4 Node Localisation: Random Deployment                            | 35 |
| 3.5 Node Localisation: Equidistant Deployment on Boundary           | 36 |
| 3.6 Node Localisation: Random Deployment on Boundary                | 36 |
| 3.7 Impact of varying area                                          | 38 |
| 3.8 Impact of varying the initial unknown nodes                     | 40 |
| 3.9 Impact of varying deployment type                               | 42 |
| 3.10 Impact of varying the environments.                            | 43 |
| 3.11 Impact of Accumulative Error                                   | 45 |
| 4.1 Localization using RSSI                                         | 55 |
| 4.2 Proposed hybrid localisation approach                           | 56 |
| 4.3 A generic random forest                                         | 59 |
| 4.4 High-level depiction of multilateration                         | 63 |
| 4.5 Hybrid localisation comprising ML and multilateration           | 65 |
| 4.6 Overlap of predicted and actual locations of sensor nodes       | 69 |
| 4.7 Localisation accuracy vs size of 'region of interest'           | 71 |
| 4.8 Localisation accuracy vs number of beacon nodes                 | 72 |
| 5.1 Flow Diagram: Proposed Cluster Head Selection method            | 83 |
| 5.2 Optimal number of cluster selection methods                     | 86 |

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| 5.3 Parameters used for clustering and cluster head selection. . . . .           | 87  |
| 5.4 Secondary routing . . . . .                                                  | 88  |
| 5.5 Clustering-based routing . . . . .                                           | 90  |
| 5.6 Tuning parameters and Weights . . . . .                                      | 92  |
| 5.7 Homogeneous Protocols: Number of Rounds at percent of nodes died . . .       | 96  |
| 5.8 Heterogeneous Protocols: Number of Rounds at percent of nodes died . . .     | 97  |
| 5.9 Homogeneous Protocols: Average Distance per Round . . . . .                  | 98  |
| 5.10 Heterogeneous Protocols: Average Distance per Round . . . . .               | 98  |
| 5.11 Homogeneous Protocols: Number of Data Packets sent . . . . .                | 99  |
| 5.12 Heterogeneous Protocols: Number of Data Packets sent . . . . .              | 100 |
| 5.13 Homogeneous Protocols: Number of times clustering process initiated . . .   | 100 |
| 5.14 Heterogeneous Protocols: Number of times clustering process initiated . . . | 101 |
| 5.15 Homogeneous Protocols: Number of Cluster Heads Changed . . . . .            | 101 |
| 5.16 Heterogeneous Protocols: Number of Cluster Heads Changed . . . . .          | 102 |
| 5.17 Homogeneous Protocols: Number of Hops . . . . .                             | 103 |
| 5.18 Heterogeneous Protocols: Number of Hops . . . . .                           | 103 |

# List of Tables

|      |                                                                             |    |
|------|-----------------------------------------------------------------------------|----|
| 3.1  | Simulation Parameters                                                       | 28 |
| 3.2  | Beacon nodes deployed randomly in network (RD)                              | 33 |
| 3.3  | Beacon nodes deployed equidistance on boundary (EDB)                        | 33 |
| 3.4  | Beacon nodes deployed randomly on boundary (RDB)                            | 33 |
| 3.5  | Beacon nodes deployed randomly in network (RD)                              | 34 |
| 3.6  | Beacon nodes deployed equidistance on boundary (EDB)                        | 34 |
| 3.7  | Beacon nodes deployed randomly on boundary (RDB)                            | 34 |
| 3.8  | Network localisability for various deployment and area sizes                | 46 |
| 3.9  | Simulation Parameters for comparison                                        | 47 |
| 3.10 | Average Localisation Error (A-LE) for $R_1$ Region when Beacon nodes ran-   |    |
|      | domly deployed in Network                                                   | 48 |
| 3.11 | Average Localisation Error (A-LE) for $R_2$ Region when Beacon nodes ran-   |    |
|      | domly deployed in Network                                                   | 49 |
| 4.1  | Sensor nodes within communication range of beacon nodes                     | 67 |
| 4.2  | A small segment of the synthetic dataset                                    | 68 |
| 4.3  | X-Y coordinates, actual vs predicted by random forest                       | 70 |
| 4.4  | Deviation of predicted locations from actual values                         | 70 |
| 4.5  | localisation accuracies of various ML algorithms                            | 73 |
| 4.6  | X-Y coordinates, actual vs predicted by first iteration of multilateration  | 75 |
| 4.7  | X-Y coordinates, actual vs predicted by second iteration of multilateration | 75 |
| 4.8  | X-Y coordinates, actual vs predicted by third iteration of multilateration  | 76 |
| 4.9  | Comparison of localisation by random forest and multilateration             | 77 |
| 4.10 | Beacon nodes in the neighbourhood of sensor nodes                           | 78 |

|                                                                        |     |
|------------------------------------------------------------------------|-----|
| 4.11 X-Y coordinates, actual vs predicted by hybrid approach . . . . . | 78  |
| 5.1 Simulation Parameters . . . . .                                    | 91  |
| 5.2 Performances of the clustering algorithm's combinations . . . . .  | 93  |
| 5.3 Performance Gain . . . . .                                         | 104 |

## List of Abbreviations

|                     |                                                        |
|---------------------|--------------------------------------------------------|
| <b>AGG</b>          | Agglomerative clustering                               |
| <b>AoA</b>          | Angle of Arrival                                       |
| <b>BS</b>           | Base Station                                           |
| <b>BN</b>           | Beacon Node                                            |
| <b>CBR</b>          | Cluster Based Routing                                  |
| <b>CH_Threshold</b> | Cluster head threshold                                 |
| <b>CHP</b>          | Cluster Head Potential                                 |
| <b>Cluster1</b>     | Initial clustering                                     |
| <b>Cluster2</b>     | Secondary clustering                                   |
| <b>Cn</b>           | Connectivity (number of nodes connected to a node)     |
| <b>d</b>            | Distance between two nodes                             |
| $d_0$               | Path loss calculation reference distance               |
| $D_{BS}$            | Distance to base station                               |
| $D_C$               | Distance to cluster centroid                           |
| <b>DT</b>           | Decision Tree                                          |
| <b>EAMMH</b>        | Energy-Aware Multi-hop Multi-path Hierarchical         |
| <b>FCM</b>          | Fuzzy C-Means clustering                               |
| <b>GMM</b>          | Gaussian Mixture Model clustering                      |
| <b>GPS</b>          | Global Positioning System                              |
| $K$                 | Optimal number of clusters                             |
| <b>LE</b>           | Localisation Error                                     |
| <b>LEACH</b>        | Low Energy Adaptive Clustering Hierarchy               |
| <b>LEACH-C</b>      | Low Energy Adaptive Clustering Hierarchy - Centralised |
| <b>LT</b>           | Localisation Time                                      |
| <b>NIT</b>          | Number of Iterations                                   |
| <b>NN</b>           | Neural Network                                         |
| <b>RE</b>           | Residual Energy                                        |
| <b>RE_threshold</b> | Residual Energy threshold                              |
| <b>RF</b>           | Random Forest                                          |
| <b>RSSI</b>         | Received Signal Strength Indicator                     |
| <b>SEP</b>          | Stable Election Protocol                               |
| <b>SOM</b>          | Self Organizing Map clustering                         |
| <b>SVR</b>          | Support Vector Regression                              |
| <b>UAV</b>          | Unmanned Aerial Vehicles                               |
| <b>UN</b>           | Unknown Node                                           |
| <b>UNR</b>          | Unknown Nodes Remaining                                |
| <b>WSN</b>          | Wireless Sensor Network                                |
| <b>Z-SEP</b>        | Zonal - Stable Election Protocol                       |

# Chapter 1

## Introduction

A **Wireless Sensor Network (WSN)** comprises a network of spatially distributed sensor nodes that monitor physical and/or environmental conditions like temperature, humidity, noise, air quality, pressure, motion, and the like. Each node comprises several autonomous sensors that collect data from the environment and transmit the same to a central location (known as a base station) over a wireless ad-hoc network. The base station may process/analyze the data and/or communicate the same over the Internet to a back end cloud facilitating analysis and visualization of the same.

WSN is especially useful in scenarios where human access is difficult. The *modus operandi* largely comprises dropping WSN nodes in regions of interest using drones or other mechanisms and being able to subsequently monitor the region.

Use cases for WSN deployments include surveillance for floods and tsunamis, detection of forest fires early, advance detection of volcano eruptions; defense applications like monitoring movements of the enemy, early assessment of their strategies, and so on. WSNs are also useful in home environments for bolstering security and domestic services; in industrial set-ups for production monitoring; healthcare applications comprise monitoring of vital parameters of patients. WSN networks broadly comprise the following components:

sensor nodes, base stations, and a communication medium. There are a large number of sensor nodes in a WSN deployment, and each node comprises several sensors whose types are dictated by the kind of application that the WSN caters to (for example, a forest fire detection WSN would comprise nodes with heat sensors, CO sensors, humidity sensors); a data transmitting/receiving unit that transmits and receives data from other nodes; a power source (usually one of more batteries); and a microcontroller that is responsible for the overall management of these components. The base-station is itself a sensor node but is a little more energy liberal. These serve as a sink for data collection from the other nodes and orchestrate peripheral processing and analysis of data. Subsequently, based on requirements of the application, the base-station connects with the Internet and transmits the data to the back end cloud for more detailed analysis and inference.

An important limitation of WSN is that the energy sustaining the sensor nodes is finite and is derived from batteries. In most use cases these batteries, once exhausted, cannot be replaced as the nodes are inaccessible to humans. It is imperative, therefore, that the energy of batteries on the sensor nodes be preserved. Transmission and reception of signals by far require the most energy on WSN nodes and hence low power communication and routing protocols are adhered to in these environments. In addition to this, exercises like optimal routing of communication signals; effective localisation of regions of interest; and intelligent interventions like AI and machine learning; go a long way in promoting energy savings and improving the longevity of the network. In this thesis, we explore these approaches for energy conservation and realize specific instances of the same with appropriate validations.

## **1.1 Motivation**

Conserving energy is a primary concern in maintaining an effective wireless sensor network. This is because of the intrinsic limitations of sensor nodes and the environment in

which they are deployed. WSN networks are mostly deployed in remote, hazardous, and inaccessible terrains where it is almost impossible to replace/replenish the batteries powering the nodes. The following factors make conservation of energy in WSN deployments a significant concern and the main motivation for the work of this thesis:

1. **Compromised energy supply:** Nodes in WSN deployments are commonly powered by small batteries with a finite lifespan. Replacing or replenishing these batteries is usually not possible after deployment, especially when this is done in challenging terrains. It becomes imperative, therefore, that the nodes are utilised in a manner that prolongs the life of the batteries.
2. **Energy-intensive communication:** Communication between the sensor nodes in a WSN is by far the most energy-intensive task. The energy expended to transmit data between nodes is proportional to the distance between them. To reduce energy consumption in this respect, various routing techniques are employed that promote energy savings. In our work, we utilize an energy-efficient clustering-based routing method that helps increase the life of the network.
3. **Unbalanced energy depletion:** In WSN deployments, unbalanced workload is a major cause for energy depletion in nodes. Certain nodes end up doing much more work than others and die early. It is not uncommon, therefore, for a WSN to become ineffective owing to the death of a few critical nodes even while a large number is available.

It becomes very important, therefore, to make the functionalities supported by WSN nodes as lightweight as possible; to route messages over WSN deployments in a manner that the communication overhead is minimized; and to balance the usage of nodes; all of which effectively leads to prolonging the life of batteries and hence the nodes.

### 1.1.1 Thesis Contributions

Given the energy constraints within which WSN deployments need to work, we look at two aspects that are critical to making WSN efficacious. These are:

- Localisation, and
- Routing

Localisation and routing are both essential for the proper functioning of WSN and making them useful; and both are traditionally energy intensive activities.

Localisation is the exercise of identifying and establishing the location of a node of interest, one that has made an interesting observation. In theory, localisation is trivial and is done through the use of appropriate sensors, most commonly Global Position System (GPS) sensitive sensors. In practice, this is not feasible as GPS sensors are energy intensive and given the energy constraints of WSN nodes would exhaust one within hours if not minutes. In this thesis, therefore, we present novel approaches of localisation of WSN nodes that are light and energy efficient. These are briefly described as follows.

### 1.1.2 Preserving Energy in Localisation through simple computations

We propose a novel hybrid localisation method that can efficiently localise WSN nodes in outdoor environments using simple computations of distances and angles between nodes. The key contributions of this work are summarised as:

- Localisation of an unknown WSN node using just one other node through the use of two factors: angle and distance from a known node.
- An iterative approach to localisation that makes it possible to cover a very large area.

- An innovative and inexpensive approach to computing the angle of arrival is presented in this work and used for validation wherein a stepper motor is customised to move 360 steps per revolution and is used for measuring the angle with an accuracy of  $1^\circ$ .

### **1.1.3 Preserving Energy in Localisation harnessing machine learning models**

This approach involves the use of conventional random forest methods and multi-iteration for localisation of nodes. The approach is effective over expansive areas, where the number of beacon nodes (nodes whose locations are known in advance and which provide reference for further localisation) is small. The main contribution of this work is as follows:

- A hybrid machine-learning method for outdoor localisation. The method helps localise the unknown nodes more efficiently.
- The proposed work uses a multi-iteration process, which localises unknown nodes within the communication in each iteration, and then this newly localised node will act as a beacon node. Due to the multi-iteration method, the extensive region can be localised in a few iterations.

Routing is the other major aspect in WSN deployments that we explore in this thesis. Routing implies the mapping of an appropriate path between nodes that signals take to reach the base station. Routing involves perhaps the most energy intensive exercise, communication (transmission and reception) of signals, and is imperative that this be done in an energy conservative manner. In this thesis, we make the following contribution towards energy efficient routing.

### 1.1.4 Preserving Energy via Cluster-based Routing

The routing is an energy-intensive process in Wireless Sensor Networks. We proposed a Dynamic Cluster Head selection method, which can efficiently form the cluster and select the cluster heads. Our main contributions in this direction are:

- The adaptive clustering is used, so the number of clusters can be changed anytime based on the requirement.
- The method finds the optimal number of clusters with the help of a machine learning-based approach (using the Silhouette method).
- The method changes the cluster head based on the remaining energy of the head. If the remaining energy becomes less than some threshold, then the new node will work as cluster head, and this threshold is also selected using the optimisation method.
- Dynamic clustering is used for forming the clusters; with the help of that, the method can work efficiently with homogeneous, heterogeneous, as well as stable and movable nodes.

## 1.2 Thesis Organisation

This thesis is divided into six chapters, with the first serving as the introduction. The thesis is organised into chapters, as seen in the flow diagram below [1.1](#). **Chapter 1** introduces the wireless sensor network and its importance. It also highlights the need for energy management to extend the life of the network.

The **chapter 2** describes the technical work that is now being done in the field of wireless sensor networks to improve localisation and routing, as well as their limits.

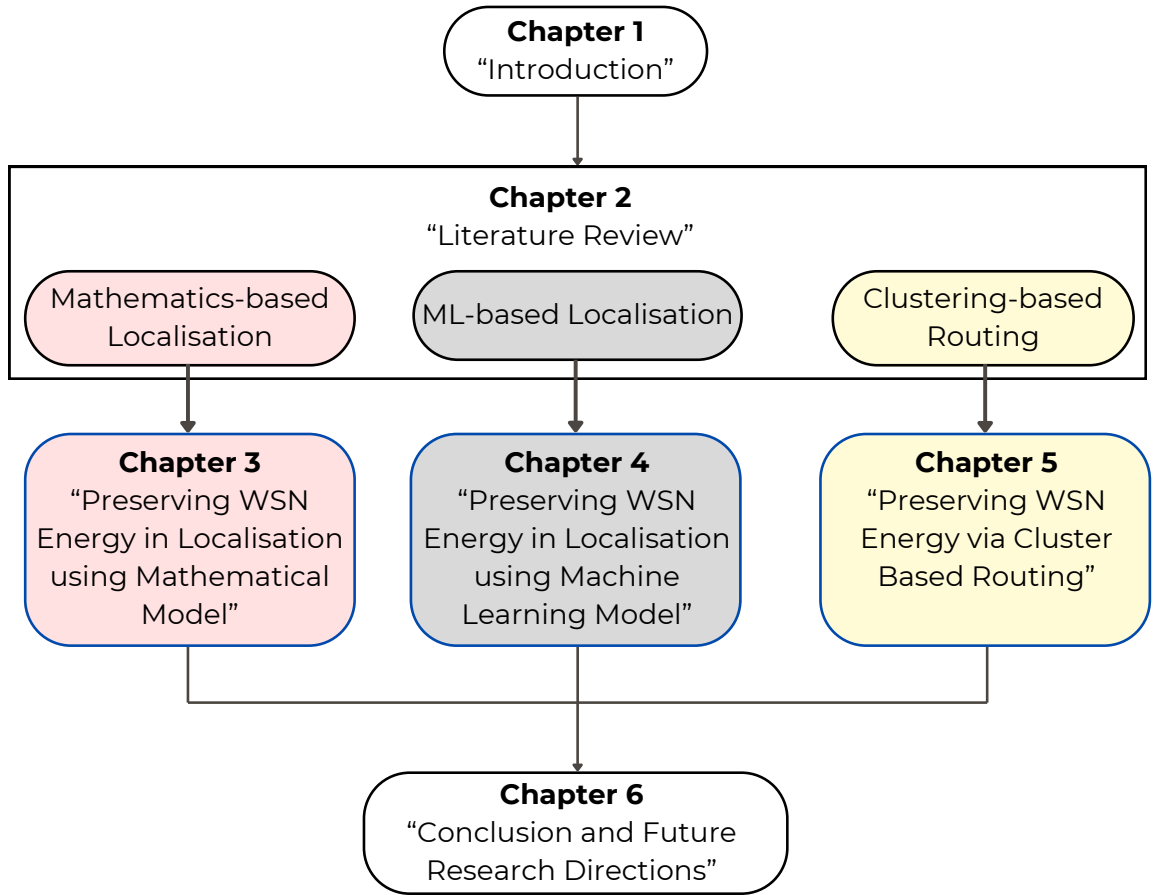


Figure 1.1: Flow Graph depicts the organisation of the thesis.

**Chapter 3** describes an energy-efficient outdoor localisation method based on a mathematical approach. The chapter emphasises the relevance of outdoor localisation. In that chapter, we attempt to achieve outdoor localisation by combining the Received Signal Strength Indicator (RSSI) and Angle of Arrival (AoA) approaches. The simulation and real-world experiments are carried out on different region sizes, adjusting the number of beacon nodes, unknown nodes, and communication range.

In **Chapter 4**, we suggest an outdoor localisation method based on the improvised random forest and iterative multi-iteration techniques. The method is evaluated on several datasets and compared to current state-of-the-art methods.

**Chapter 5** describes energy-efficient routing in WSNs. In this work, a new dynamic

cluster head selection approach is described. Using grid search, the algorithm determines the best clustering methodology. The various factors are used to form the clusters and pick cluster heads, resulting in energy efficiency when relaying data from sensor nodes to the base station.

In **Chapter 6**, we summarise the study described in this thesis and considered potential future research possibilities.

# Chapter 2

## Literature Review

In this chapter, we provide an exhaustive survey of current publications relating to the contributions made in this thesis. This chapter provided a survey of several mathematics and machine-learning strategies used for outdoor localisation and routing in wireless sensor networks. In section [2.1](#), we begin by reviewing the literature on existing localisation methods that employ mathematical methodologies. Then, in section [2.2](#), we give a literature review of different existing localisation approaches that leverage machine learning-based methodologies to achieve accurate outdoor localisation while also being more energy efficient. In the second to last section, [2.3](#), we reviewed a literature review of existing Cluster head selection algorithms in wireless sensor networks.

### 2.1 Mathematics-based Localisation

There is significant work done on localisation approaches for WSNs that are based on RSSI and AoA. In this section, we discuss a few prominent endeavours.

The work described in [\[1\]](#) is closely related to ours. An RSSI and AoA-based method is proposed in the paper for localising multiple unknown targets (unknown nodes). Different from existing algorithms that rely on complex mathematical tools that do not always

yield feasible solutions, a more straightforward approach was taken here: the paper utilises available AoA measurements and transitions from Cartesian to Spherical coordinates which allows the approximation of initially non-convex measurement models to linear ones. The iterative methods first localise some unknown nodes with a localisation error; then, these newly localised (with some localisation error) nodes help localise other unknown nodes. The localisation error propagated from one localised to another in subsequent rounds is known as accumulative error. Our proposed method works iteratively and considers the accumulative errors from the previous rounds (just like feasible solutions), representing the clear idea of waste areas. Also, our proposed approach is able to localise multiple nodes per iteration, leading to much fewer iterations. In addition to this, the work in [1] uses signals for the calculation of AoA, which is susceptible to disturbance and interference adversely affecting accuracy. To overcome this, we propose the use of a simple stepper motor-based mechanism, which is the most effective in free space models [2] (without any obstructions) for 2-dimensional spaces. We also study the effectiveness of our approach in different types of terrains that include: sandy, long grassy, and sparse tree terrains; which is again something that the earlier approach does not look at.

In [3], the authors address the node localisability problem in a network where a few nodes are localisable. The authors propose the concept of partially localisable networks by presenting theoretical conditions for a node to be considered uniquely localisable. Using this, localisable nodes are identified by dividing the network into redundantly rigid and reconnected components. The method uses 10% of the nodes as beacon nodes (BN), whereas our proposed approach uses only 1% of nodes as BNs. In spite of this, we are able to localise nodes over an extensive area and with greater accuracy. In [4], the authors propose an iterative multilateration method that localises the unknown node (UN) using trilateration (using three BNs) repetitively. The time difference of arrival (ToA) used for distance

measures have several disadvantages and obstacles in crowded areas when compared with RSSI-based methods [5]. The proposed method takes advantage of iterative multilateration to a great extent through its ability to localise a UN using a single BN. In the work cited, a UN node needs to be within the communication range of at least three BNs. The method, therefore, requires a very large number of beacon nodes when the density of unknown nodes is large, which is quite cost-ineffective when deployed in the real world. In [6], the authors propose the sequential Monte Carlo localisation method and exploit the node's mobility to improve localisation accuracy and precision. The localisation accuracy is established for three scenarios: when UNs are static and BNs are movable; when UNs are moving and BNs are stable; and when both UNs and BNs are moveable. The approach utilises range-free methods for distance estimation. Our proposed work uses a range-based approach for calculating the parameters, which is much more accurate than range-free methods. Furthermore, the above method evaluates the results based on simulations only and, even in these, uses a high BN density, which is hardly possible in realistic deployments. In [7], the authors address an important issue in trilateration wherein the latter marks a localisable graph as non-localisable. The approach can appropriately localise UNs that are one hop away from participating in the trilateration exercise. The method uses trilateration and three BNs to determine the location. The method uses trilateration and requires at least three BNs to get the location of every unknown node. [8] proposes a novel localisability algorithm (i.e. Patch and Stitching) wherein the whole network is initially divided into small localisable networks, and subsequently joined together to form a global localisable network. Using an algebraic approach, the work proposes a subset of localisable network merging conditions for 2D/3D networks. The method uses trilateration for node localisability and can localise 90% of the network with 5% of BNs. Our proposed method, on the other hand, takes a hybrid approach for localisation, and in simulations is able to localise the whole network

with only 1% of nodes acting as BNs. In [9], the authors work on the localisability of the network to answer the following two questions: whether it possible to say that a node is localisable or not while using a network graph; how many nodes of the network are localisable and whether these can be identified? The localisability method uses more nodes to localise unknown nodes in the network, where the proposed method takes a hybrid approach and is able to efficiently localise most nodes using a very small number of known nodes thus producing significantly superior localisability results. [10] deals with localisation accuracies affected by outliers in range-based localisation methods. Earlier solutions, in general, use triangle inequalities to deal with noisy data with outliers. The method proposed, on the other hand, utilises a theoretical model based on graph embeddability and rigidity theory for the same. The paper designs a bilateration generic cycles-based outlier detection algorithm and tries to evaluate its effectiveness through simulations and practical deployments. The method uses trilateration and cycles-based outlier detection. The detection method is ineffective for more extensive settings and likely fails to produce good results for larger outdoor areas. The proposed method discards the outliers if they are out of the predefined region of the network and is thus more robust.

In [11], a machine learning-based localisation approach for outdoor settings is proposed. The approach employs virtual nodes to widen the dataset to address the issue of limited training data. The method collects information from multiple access points and uses this information to appropriately localise other unknown nodes. The approach is tested using simulations that vary parameters like the number of sensors and anchor nodes, radio transmission power, and wireless signal quality. In [12], the ARBL algorithm for outdoor localisation is proposed. This approach uses trilateration and reference node selection to determine the locations of sensor nodes. A reference triangle through triangulation is formed in this approach, and using this, the ranging inaccuracies are evaluated. The sensor node

positions are based on these inaccuracies. Notably, the innovation in this work lies in selecting the optimal anchor node for the localisation process. The approaches in [11] and [12] are relatively impracticable. The former is because it is pretty challenging to implement, and the latter is because it requires at least three beacon nodes to localise an unknown node, which is hard to get in outdoor environments. In [13], the AoA and RSSI differences-based localisation method (ALRD) is proposed. In this method, the AoA is estimated by comparing the RSSI values of beacon signals received by two perpendicularly oriented directional antennas installed at the same place. Subsequently, two methods (maximum point minimum diameter and maximum point minimum rectangle) are proposed to minimise the ALRD localisation error. In [14], the authors propose a simplified combination of the AoA and RSSI methods. The calculations of the AoA require complex antenna arrays. The proposed approach (1AoA/nRSSI) uses the AoA values from only one anchor node in combination with  $n$  RSSI values to estimate the location of an unknown node. The method is suitable for shadowing environments and when more precise RSSI values are available.

In [15], a fingerprint-based WSN localisation technique is proposed for indoor and outdoor use. The objective of this work is to enhance fault tolerance and system efficiency. A precision of 5 m and 10 m was attained, respectively, in this technique for pedestrian and driving tests using phones for evaluation. In [16], multiple indoor and outdoor localisation approaches for WSN are analysed. The authors illustrate the applicability of Multidimensional Scaling (MDS) approaches in modern technologies like WSN-IoT, cognitive radios, and 5G networks. Centralised and distributed MDS techniques for indoor and outdoor localisation are discussed in the paper. In [17], a localisation technique for mobile nodes in wireless sensor networks is proposed. The strategy targets short beaconing intervals and localisation deviations stemming from radio propagation. It solves these issues through geometric least square curve fitting. In [18], a technique is proposed that employs curve fitting

for wireless sensor networks and a range-based approach for outdoor localisation is proposed. The techniques discussed in [15], [16], [17], and [18] are applicable in both indoor and outdoor environments. However, their localisation processes were tested on limited confines, potentially constraining their effectiveness in large areas.

For indoor localisation, [19] suggests a hybrid approach using particle swarm optimisation (PSO) and the global best local neighbourhood approach. This approach employs three anchor nodes to locate an unknown node. It achieves a localisation inaccuracy of 0.44 m in a simulated environment. RSSI values are used for distance estimation between nodes, and the mean error varies with network size changes. In [20], a range-free localisation technique is proposed using a fusion of Harris Hawks optimisation and area minimisation. The technique categorises neighbours into incoming and outgoing for heterogeneous wireless sensor networks. It employs area minimisation to reduce the node's predicted region. The technique is simulated for small areas in 2D and 3D environments. In [21], a heuristic approach is introduced to tackle anisotropy-related challenges in WSN, which can lead to localisation errors. The proposed range-free approach incorporates geometric constraints and hop-based strategies to mitigate this problem. The techniques in [19], [20], and [21] are indoor localisation strategies that pose challenges when applied to larger areas. Implementing these for expansive spaces is arduous and mostly not feasible.

## 2.2 ML-based Localisation

[22] is an important contribution at localization wherein the radar location system is used for determining the distance between access points and mobile terminals. The location is calculated by triangulation using the RSSI values of received signals at multiple receiver locations. The importance of this work stems from the fact that it is first contribution in the direction of using RSSI values for localization.

The use of machine learning in localization was first proposed in [23]. Here RSSI values at sensor nodes are used with a kernel-based learning algorithm to define the basis function which is used to localize other sensor nodes.

[24] propose a Support Vector Machine (SVM) algorithm which regards the localization of nodes in a WSN as a regression problem. RSS values are used as inputs to train the model. The position prediction model is developed in an offline manner using Support Vector Regression (SVR).

An Artificial Neural Network (ANN) based localization algorithm is proposed in [25]. Here RSSI values between the grid sensors and anchor nodes are used as inputs to train the neural network. ANN develops a mapping between the RSSI values and the locations of the node. This approach is based on the assumption that all sensor nodes can directly communicate with all anchor nodes.

Similarly, two groups of algorithms for localizing sensor nodes using RSSI values of signals from anchor nodes are proposed in [26]. The first class of algorithms uses fuzzy logic and genetic algorithms, while the second class uses neural networks with the RSSI values.

The idea of using a lightweight SVR implementation is proposed in [27] wherein the original problem of regression is split into 13 sub-problems. The algorithm progresses by splitting the entire network into a series of sub-networks, such that each regression algorithm (i.e. the sub-predictors of SVR) needs to process a small amount of data.

[28] propose a submarine detection solution for underwater surveillance systems that locates randomly deployed nodes in space using anchor node location co-ordinates. Every monitoring unit has a sensor node connected to a surface buoy with a cable and data is collected using the buoys. The collected data is subsequently transmitted to the central control unit after transmission. A decision tree classifier is finally used to identify submarines at the

sites monitored.

[29] propose a novel and simple indoor positioning solution. The technique harnesses an indoor visible light positioning system and a dual-function machine learning (ML) algorithm.

Low-Power Wide-Area Network (LPWAN) technologies have lately emerged as a viable alternative to scalable wireless connections in smart city applications. On a training dataset collected in two different environments: indoors and outdoors, [30] investigate the use of intelligent machine learning techniques such as support vector machines, spline models, decision trees, and ensemble learning for RSSI-based ‘ranging’ in LoRa networks. An appropriate ranging model is subsequently utilized to test the accuracy of the trilateration-based localization and tracking endeavours.

[31] use finger-printing to train a neural network to develop a median accuracy (of about 16 m to 100 m) model for outdoor localization using the very little information available over pre-5G base stations with active multi-beam antenna systems.

## 2.3 Cluster-based Routing

In recent decades, work in the cluster-based routing protocol has been pervasive. Researchers have used various protocols to identify appropriate *CHs* through the use of parameters like *RE*, distance, node degree and *CH\_Threshold*. The main objective of all protocols is to prolong the network lifetime by reducing overheads with efficient routing. In this section, we discuss the related work in this area.

Low Energy Adaptive Clustering Hierarchy (LEACH) is the first cluster-based routing protocol that works with homogeneous nodes and selects the *CHs* randomly. The method works in setup and steady-state phases. All the non-CH nodes join the *CH* with the strongest signal for cluster formation [32]. A progression of this is the LEACH-Centralised (LEACH-

C) approach. Here, a centralised system (i.e. *BS*) is used for *CH* selection. Two parameters: residual energy (RE) and distance of a node from the *BS*, are used for *CH* selection. Only nodes with RE greater than average can participate in the *CH* selection exercise. The *CH* selection process is entirely managed by the *BS* [33]. Both these approaches are meant for a homogeneous network. In addition to this limitation, the *CH* selection process in LEACH is mainly random, often leading to an ineffective selection. LEACH-C, on the other hand, does not prescribe random selection of *CH*, but its clustering process totally depends on *BS*, which is not desirable and runs the risk of falling into the ‘single point of failure’ trap.

The Stable Election Protocol (SEP) [34] assumes that all the nodes are stable and distributed randomly, and the positions of nodes are already known. A similar protocol, the Energy-Aware Multi-hop Multi-path Hierarchical protocol (EAMMH) [35] uses multiple hops to transmit data from the *CH* to the *BS*. This protocol works in three phases: setup, initialization and sleep-awake phase. The node transmits the data only if its residual energy is greater than the threshold energy; otherwise, the node will be in sleep mode. An advanced version of SEP, the Zonal-stable election protocol (Z-SEP) divides the nodes into *zone\_0* (only normal nodes), *head\_zone\_1* (half of the advanced nodes), and *head\_zone\_2* (half of the advanced nodes). In Z-SEP, the nodes from *zone\_0* can directly transmit the data to *BS*, but in *head\_zone\_1* and *head\_zone\_2*, transmission is performed via *CH*. [36]. All three of these approaches work on the realistic assumption that the network is heterogeneous. SEP is a rather delicate protocol as it becomes vulnerable after the death of the first node whereas Z-SEP mandates the rather unrealistic requirement of synchronization for maintaining the sleep-awake mode.

The PBC-CP protocol [37] is used to select optimal cluster heads and transfer data from the *CH* to the *BS* using energy-efficient paths. The main objective of the protocol is to reduce the energy use in the network. The use of a heuristic method on each round increases

the algorithm's complexity. In the Novel-LEACH-PSO approach [38], the particle swarm optimization (PSO) method selects *CHs* on the basis of the residual energy of a node and its distance from the *CH*. The *CH* distribution is non-uniform here, and there are no criteria for managing the energy imbalance in the network. The TTDFP [39] uses multiple parameters (like node connectivity, distance to the base station, and remaining energy) for cluster head selection. It uses the remaining energy with relative distance for route selection and fuzzy logic for handling the uncertainties during these phases. This method mainly covers the data aggregation problem in WSNs where multiple hops are required for data transmission. In DFLBCHSA [40], a fuzzy algorithm-based method selects the optimal *CH* using residual energy, mean distance, location and neighbour count. Here, GPS sensors are used to detect the location of nodes, which significantly increases the energy expended. The CLONALG-M [41] is a metaheuristic method and an advanced version of CLONALG, enhancing the performance of rule-based clustering methods. It improves the performance of fuzzy clustering algorithms and approximates the fuzzy output functions to the optimum. This method performs well in comparison with other clustering methods based on genetic algorithms. In HGWSFO [42], the authors propose a hybrid meta-heuristic technique based on energy and distance. The algorithm works on the Grey Wolf Optimization (GWO) and the Sunflower algorithm. The major concern with the GWO and Sunflower algorithms is that they tend to fall into a local optimum trap and are slow to converge, respectively. In the DRE-LEACH approach [43], a non-meta-heuristic technique is proposed for selecting the *CH* using centrality, positions, and the residual energy of nodes. Finally, in NCHR [44], the authors propose a non-meta-heuristic technique based on residual energy and distance that can work with homogeneous or heterogeneous nodes. The method involves topological changes to address cluster head failures and employs a distributed system, leading to a significant increase in the computation load on individual nodes.

## **Chapter 3**

# **Preserving WSN Energy in Localisation using Mathematical Model**

### **3.1 Understanding the Problem**

Wireless sensor networks (WSNs) consist of multiple sensor-equipped nodes designed to monitor their surroundings and relay data to a base station (BS) using various ad-hoc network methods. These sensors are especially useful in hazardous terrains where human presence is difficult. They detect anomalies and alert the BS. However, the limited energy supply of WSNs is a significant concern, particularly in remote outdoor areas where nodes rely on non-rechargeable batteries, making replacements impractical. Maximising battery life is crucial, as it restricts the use of GPS for energy-intensive tasks like localisation. Localisation aims to determine the exact/approximate coordinates of nodes using mathematical or machine learning algorithms.

For instance, in a project focused on early forest fire detection in the Melghat Tiger Reserve, WSN nodes equipped with heat sensors are deployed via unmanned aerial vehicles (UAVs) in inaccessible areas. While these nodes can detect fires and send alerts, they struggle to accurately pinpoint the fire's location over the 3,000 square kilometre reserve.

Beyond forest fires, localisation has applications in detecting environmental hazards (such as floods and volcanoes), healthcare (for patient tracking), agriculture (for soil and crop monitoring), military operations (for surveillance), and enterprises (for asset tracking).

Localisation typically uses two types of nodes: beacon and unknown. The beacon nodes are the nodes whose location is already known (using GPS or tagged a-priori at the time of deployment), and unknown nodes are the nodes whose location is to be determined using some algorithms in combination with the locations of neighbouring beacon nodes. However, these localisation algorithms often need help in large areas where multiple beacon nodes may not be within range, requiring significant time for accurate localisation. Existing techniques often work only for small areas, requiring nodes to be within communication range. Additionally, many methods need at least two beacon nodes to estimate the position of unknown nodes, a difficult requirement in vast regions.

## **3.2 The Proposed Method**

The proposed approach for node localisation in wireless sensor networks (WSNs) is a significant step forward and, to our knowledge, the first to address large-scale areas effectively. Previous localisation methods have assumed that all nodes are within communication range of each other, which is unrealistic for vast outdoor environments like forests or glaciers. These regions often span thousands of square kilometres, while WSN nodes have a communication range limited to 100 meters. Our proposed method can potentially work in such areas.

Our method overcomes this critical limitation by starting with a more practical assumption: only a few nodes (beacon nodes) locations need to be known initially. It is feasible, as certain parts of large regions, particularly along the periphery, are typically accessible. The known node's coordinates are determined using external GPS devices. In contrast, the nodes

that need localisation are often located deep within hostile, inaccessible areas. Examples include forests where the outer edges are reachable. Still, the inner core remains difficult to access, and glaciers where the perimeter can be navigated are difficult to traverse, but deeper regions are challenging to traverse. The proposed method follows these steps:

1. Randomly distribute the unknown nodes across the region.
2. Choose the deployment type for the beacon nodes and deploy them accordingly.
3. Determine the distance and angle between each unknown node and any beacon nodes within the communication range and localise the unknown nodes.
4. Use the newly localised nodes from step 3 as additional beacon nodes in the next iteration.
5. Repeat steps 3 and 4 until all unknown nodes are localised, except for a few outliers.

The proposed method employs a hybrid approach for localising unknown nodes. It combines the Received Signal Strength Indicator (RSSI) for distance calculation between nodes and the Angle of Arrival (AoA) technique for determining angles. This iterative, multi-step process enhances localisation, making it particularly effective for large-scale areas. A detailed explanation of the RSSI, AoA, and multi-iteration techniques is provided later in this section.

### **3.2.1 RSSI and Distance calculation**

The Received Signal Strength Indicator (RSSI) measures the power level of a received radio transmission. It estimates the strength of a wireless link, typically in dBm (decibel-milliwatts), where a higher value indicates a stronger signal and better link quality. Besides

detecting wireless interference, RSSI is crucial for estimating the distance between a wireless device and its connected access point. The RSSI value is determined using the following equation 3.1:

$$RSSI(dBm) = 10 * \log_{10} (P_r/P_0) \quad (3.1)$$

In this equation,  $P_r$  represents the received power level at the receiver. In difference,  $P_0$  is the reference power level, typically measured one meter from the transmitter. It's important to note that distance, obstacles, and interference significantly affect the RSSI value. As a result, the exact RSSI value fluctuates based on these conditions. Additionally, different wireless technologies and manufacturers use varying reference values, influencing the calculated RSSI.

The proposed approach also incorporates the path loss specific to different environmental conditions when calculating RSSI values. These environments include the free space model (without obstructions), sandy terrain, long grassy terrain, and areas with sparse trees. By using distinct equations to compute path loss for each type of terrain, the aim is to establish a correlation between these environments and their corresponding RSSI values:

$$L_p = E_0 = PL_0 + 10 * \alpha * \log_{10} (P_r/P_0) + X_\sigma \quad (3.2)$$

$$E_1 = 60.97 + 10 * 3.42 * \log_{10} (P_r/P_0) + 3.02 \quad (3.3)$$

$$E_2 = 59.42 + 10 * 2.56 * \log_{10} (P_r/P_0) + 3.84 \quad (3.4)$$

$$E_3 = 60.98 + 10 * 3.33 * \log_{10} (P_r/P_0) + 7.30 \quad (3.5)$$

In this context,  $PL_0$  represents the path loss at a reference distance  $d_0$ , measured in dB, while  $\alpha$  is the path loss exponent, indicating how quickly path loss increases with the logarithm of distance.  $X_\sigma$  is a normally distributed random variable with a mean of zero

and standard deviation  $\sigma$ . The path loss for sandy, long grassy, and sparse tree terrains are denoted as  $E_1$ ,  $E_2$ , and  $E_3$ , respectively.

The proposed algorithm considers all the environments ( $E_1$ ,  $E_2$ , and  $E_3$ ) and the free space path transmission model for evaluating the RSSI values.

In localisation, RSSI values are measured using various hardware, and to determine the location of a node, the algorithm needs to convert these values into distances. The distance between an unknown node and a beacon node can be estimated by calculating RSSI over different terrains. These distance and angle measurements help pinpoint the unknown node's position. The distance,  $d$ , is derived from the RSSI value using the following equation:

$$d = 10^{\left(\frac{P_m - RSSI}{10 * N}\right)} \quad (3.6)$$

Here,  $RSSI$  represents the signal strength received from the beacon node at the unknown node. The measured power,  $P_m$ , is the RSSI value recorded at a distance of 1 meter from the beacon node, while  $N$  is a constant ranging from 2 to 4, depending on environmental conditions.

### 3.2.2 Angle of Arrival (AoA)

The Angle of Arrival (AoA) refers to the angle between a wave's direction of propagation and a reference direction, typically measured in degrees clockwise from North. When the orientation is 0 (pointing North), the AoA is considered absolute; otherwise, it is relative. AoA can be determined using radio waves, Bluetooth, RFID, or WiFi signals. Historically, AoA was measured through radio waves by positioning four antennas in a square configuration on each beacon node.

This enables Angle of Arrival (AoA) calculations [45] by adjusting the signal phases across antennas. A rotating beacon is created by focusing maximum radiation from the an-

tenna array using beamforming. This scanning phased array technique ensures constructive interference of radio waves in a specific direction [46]. The demand for cost-effective indoor localisation has recently driven the use of technologies like Bluetooth Low Energy (BLE) and Radio Frequency Identification (RFID) for AoA computation. BLE determines AoA through two methods: 1) Switched Beam Systems (SBS), which scan the azimuth plane to detect the most vital signal direction, and 2) Adaptive Array Systems (AAS), which steer beams in any direction by adjusting antenna weights [47]. In RFID-based systems, UHF transponders and directional antenna arrays are used. Given the narrow bandwidth of RFID, AoA becomes crucial in multi-path environments, where it isolates the strong line-of-sight path to estimate the transponder's position accurately by analysing phase differences between signals [48].

For simulation, the Equation 3.7 shows the calculation of AoA values between three points:

$$Angle = degrees(atan2(N_y - BN_y, N_x - BN_x) - atan2(UN_y - BN_y, UN_x - BN_x)) \quad (3.7)$$

Here,  $UN$ ,  $BN$ , and  $N$  represent the unknown node, beacon node, and pointer towards the north axis, respectively. The two-argument arctangent function,  $atan2(y, x)$ , computes the angle in radians between the positive x-axis and the line connecting the origin to the point  $(x, y)$  in the Cartesian plane, and the resulting angle is  $-\pi < \theta \leq \pi$ . Equation 3.7 outlines the method used to calculate the AoA during simulations. Typically, the AoA of signals from at least two beacon nodes is needed to localise an unknown node. However, the proposed approach overcomes this limitation by effectively localising an unknown node using only a single beacon node. This is achieved by combining the RSSI value with the AoA measurement. Figure 3.1 illustrates how the distance calculated from the RSSI value is combined with the AoA to determine the location of the unknown node. The X and

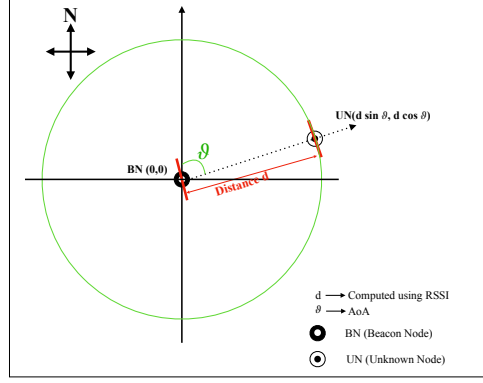


Figure 3.1: Position calculation of any unknown node using the beacon node.

Y coordinates of the unknown node are derived using simple trigonometry, as shown in Equation 3.8.

$$X_{UN} = d_{RSSI} \sin(AoA) \quad (3.8)$$

$$Y_{UN} = d_{RSSI} \cos(AoA) \quad (3.9)$$

### 3.2.3 Multi-iteration

The combination of RSSI and AoA values from a beacon node enables the accurate calculation of the location of an unknown node within its communication range. This hybrid approach allows precise localisation using just one beacon node [49]. Once an unknown node is localised, it becomes a “new” beacon node. Then, the RSSI and AoA values from this newly localised node are used to localise other unknown nodes within its range. This iterative process continues until all nodes in the region of interest are localised or until no unknown nodes remain within the range of any beacon node. Figure 3.2 illustrates the step-by-step progression of the localisation process. In the multi-iteration process, unknown nodes are initially localised (with some error) using beacon nodes. These newly localised nodes then act as beacon nodes in subsequent iterations, continuing the localisation process. However, the errors from initial localisations, combined with errors from previous itera-

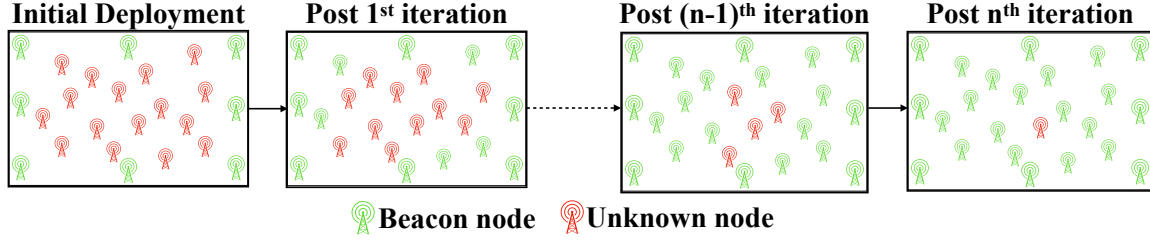


Figure 3.2: localisation using multi-iteration approach.

tions, lead to an accumulation of localisation inaccuracies over time, as shown in Figure

3.3

Since our method relies on both angle and distance to determine the location of an unknown node using just one beacon, the initial localisation error remains small. The ability to localise with a single beacon node accelerates the process and leads to faster network convergence. The result is that most localisations happen during the initial iterations, and the error is significantly minimised. Many localisation iterations would be inadequate as the localisation errors accumulate with progressive iterations.

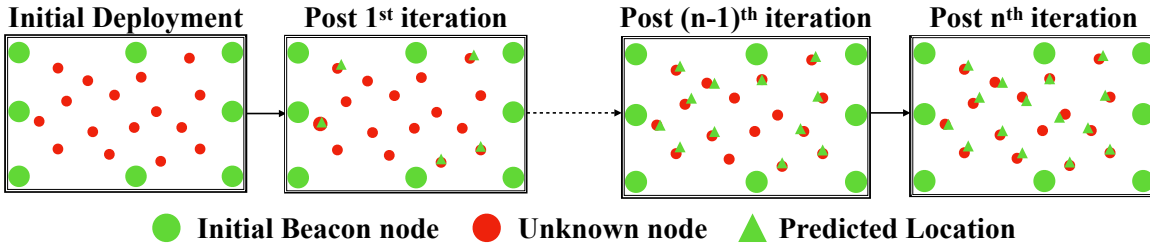


Figure 3.3: Multi-iteration Process in WSN.

**Note:** The nodes with green and circles represent the initial beacon nodes and unknown nodes, respectively. The green triangles represent the predicted location of a nearby unknown node. In multi-iteration, some unknown nodes are localised in every iteration with some error, and this will increase with the iteration round.

### 3.2.4 Network localisability

The WSN network can be represented as a distance graph  $G = (V, E)$ , where  $V$  denotes the WSN nodes and  $E$  consists of unweighted edges  $(i, j)$  in  $E$  connecting nodes  $i$  and  $j$ .

A graph or network is considered localisable if each node has a unique location, and the distance  $d(i, j)$  between any two nodes  $i$  and  $j$  satisfies specific conditions, such as being within the communication range ( $CR$ ) of the nodes. Network localisability is helpful for applications like network deployment, routing, energy management, and mobility control [50].

A successive localisability algorithm can assess whether a graph is localisable through iterative localisation. Current methods often use trilateration and require four connected components ( $K_4$ ) to verify localisability. In contrast, the proposed algorithm simplifies this by using just one beacon node, which means only two connected components ( $K_2$ ) are needed to determine the location of an unknown node. The essential condition for node localisability is that the node, represented by vertex  $v$ , must be within one hop of a beacon node.

### 3.3 Evaluation

This section is dedicated to evaluating the effectiveness of the proposed localisation approach through experiments. The evaluation involves thoroughly validating the method in a controlled environment using simulations. The results first assess the performance of the proposed method in different scenarios, followed by a comparison with state-of-the-art techniques.

#### 3.3.1 Simulation Environment and parameters

The proposed protocol is implemented in Python language and tested on a Kaggle Kernel using a system with an Intel Xeon CPU (Haswell family), 16 GB RAM, and four 2.30 GHz CPUs. A GPU is used for computationally intensive tasks like converting RSSI values to distance and performing Angle of Arrival calculations. Since these tasks involve large matrices and can be processed independently, the GPU can efficiently handle them in parallel

Table 3.1: Simulation Parameters

| Parameters     | Values for small regions ( $R_1$ )                            | Values for extensive regions ( $R_2$ )                           |
|----------------|---------------------------------------------------------------|------------------------------------------------------------------|
| Size of region | $1000 \times 1000$ , $2000 \times 2000$ to $4000 \times 4000$ | $5000 \times 5000$ , $6000 \times 6000$ , and $7000 \times 7000$ |
| $IUN^*$        | 6000, 7000, and 8000                                          | 10000, 11000, 12000 to 15000                                     |
| $OBN^\dagger$  | 1, 1.5, 2.5, and 5 %                                          | 1, and 1.5 %                                                     |
| $CR^\ddagger$  | 50, 75, and 100 m                                             | 75, and 100 m                                                    |
| $EN^§$         | $E_0$ , $E_1$ , $E_2$ , and $E_3$                             | $E_0$ , $E_1$ , $E_2$ , and $E_3$                                |

**Note:** Size of the region is measured in square metres,  $IUN^*$  denotes the number of unknown nodes initially deployed,  $OBN^\dagger$  denotes the number of original beacon nodes (in percent of the number of  $IUN$ ), and  $CR^\ddagger$  denotes the communication range of a node (in metres), and  $EN^§$  denotes the effect of noise occurred due to various environmental factors like free space ( $E_0$ ), sandy terrain ( $E_1$ ), long grassy ( $E_2$ ), and sparse tree ( $E_3$ )

and increase the performance. The above Table 3.1 represents the parameters and their values used for evaluating the results throughout the section. The detailed description of the parameters used in this section are as follows:

- Region of Interest ( $R_x$ ): Experiments are conducted in various regions in square shapes of different sizes. The regions of interest are categorised into two types:  $R_1$  for smaller regions and  $R_2$  for larger ones. The specific sizes of  $R_1$  and  $R_2$  are mentioned in Table 3.1. Testing with various region sizes helps us understand how the size of the region affects our algorithm's performance.
- Initial Unknown Nodes (IUN): IUNs are nodes with initially unknown coordinate values that must be localised using the proposed approach. The number of IUN (Table 3.1) is varied extensively throughout the section, and it serves as a valuable metric for assessing the effectiveness of the proposed approach.
- Original Beacon Nodes (OBN): A small number of Original Beacon Nodes (OBN) with known locations are used. The number of OBNS is expressed as a percentage of

Initial Unknown Nodes (IUN) and ranges between 1% and 5%. Testing with such a small percentage of OBNs shows how effectively the method can localise a large area with minimal known nodes. The OBNs are deployed in three ways: Equidistant Deployment on the Boundary (EDB), where they are evenly spaced along the boundary; Random Deployment at the Boundary (RDB), where they are placed randomly on the boundary; and Random Deployment (RD), where they are distributed randomly throughout the region.

- Communication Range ( $CR$ ): The communication range of a node represents the distance travelled by the radio signal of the nodes. This range is important for determining the distance and angle of an unknown node relative to a known beacon node. As a result, the CR plays a vital role in the effectiveness of the approach. We can understand how well the method performs in a heterogeneous node deployment by varying the CR and evaluating its impact. To better understand the effect of varying the communication range on the localisation process, we experimented with different values (as shown in Table 3.1) for both small and extensive regions.
- Environment Noise ( $E_x$ ): Environment noise interrupts the smooth transmission of radio signals between nodes, which makes it a critical factor in determining the accuracy of node localisation. A noise value of 0 indicates a disturbance-free medium, while higher values represent increasing levels of interference. Noise can arise from obstacles like sand, tall grasses, and trees. In our simulations, we experiment with different noise levels corresponding to surfaces with sand [Eq. 3.3], grass [Eq. 3.4], and sparse trees [Eq. 3.5]. The impact of these factors on the localisation performance is carefully evaluated.

### 3.3.2 Simulation Results

We tested all possible parameter combinations from Table 3.1, using the following factors to evaluate and understand the performance and effectiveness of the proposed localisation method:

- **Number of Iterations (NIT):** This factor indicates how often the multi-iteration algorithm runs to localise the network as effectively as possible. Once the NIT iterations are completed, no additional nodes are localised. The NIT parameter is important, as the fewer iterations speed up the whole localisation process.
- **Unknown Nodes Remaining (UNR):** The UNR shows the number of initial unknown nodes that remain unlocalised at the completion of the localisation process. These nodes remain unlocalised mainly due to not being in communication range to any beacon node. The smaller the count of UNR, the better the algorithm.
- **Localisation Error (LE):** The localisation error represents the Euclidean distance between the position of actual and predicted values of the newly localised nodes, which is calculated as:

$$Localisation\ Error\ (LE) = \sqrt{(X_{actual} - X_{predicted})^2 + (Y_{actual} - Y_{predicted})^2} \quad (3.10)$$

- **Average localisation error (ALE):** The ALE can be determined as:

$$Total\ Localisation\ Error\ (TLE) = \sum_{i=1}^n LE_i \quad (3.11)$$

$$ALE = \frac{TLE}{(IUN - UNR)} \quad (3.12)$$

- **Localisation time (LT):** This parameter represents the time (in seconds) required by

the algorithm for localising the network up to a maximum extent. This parameter provides valuable information about the algorithm because, in many applications, the time beyond a particular limit is unacceptable for the localisation process.

### 3.3.2.1 Experimental environment

Experiments were conducted in simulated environments to assess the effectiveness of the proposed localisation approach under different conditions. We varied the size of the Region of Interest (ROI) from  $1000 \times 1000$  meters to  $7000 \times 7000$  square meters. These large outdoor areas help us determine if the approach works well in various sizes of outdoor locations. We limited the simulation to this size range due to the increased time and computation resources required to compute them. We measured localisation error (LE), the number of iterations (NIT) needed, the number of nodes left unlocalised (UNR), and localisation time (LT) for different ROI sizes. Additionally, we tested the approach on various terrains by introducing different environmental noises, such as  $E_0$ ,  $E_1$ ,  $E_2$ , and  $E_3$ . Where  $E_0$  represents a perfectly smooth terrain for benchmarking,  $E_1$  denotes sandy terrain,  $E_2$  indicates grassy terrain, and  $E_3$  represents terrain with sparse trees. Each terrain includes more noise than the previous one, interrupting the signal transmission between WSN nodes, which helps in understanding how the signals are affected due to noise levels and affect the localisation accuracy. The simulation is done like it simulates the real-world environment, and initially, unknown nodes (IUN) whose location is to be calculated are deployed randomly throughout the region of interest (ROI). The deployment of initial or original beacon nodes (OBN) (nodes whose location is already known) are deployed in three manners: a) Randomly throughout the ROI (RD), b) Equidistance along the boundary (EDB), and c) Randomly on the boundary. The RD deployment type is used in the location where the whole region is GPS accessible, and beacon nodes can be deployed anywhere throughout

the region. The EDB deployment type is helpful in regions where only the whole periphery or boundary of the region is accessible for beacon node deployment. In RDB deployment, beacon nodes are also positioned along the boundary but randomly, without equal spacing. The tables [3.2-3.4] and [3.5-3.7] show the comparative results between three deployment types for the beacon nodes on various comparative measurements like localisation error, number of iterations, unknown nodes remaining, and localisation time. The results are calculated for small and extensive regions and all four obstacle levels (environments). The localisation results obtained with all the combinations are very large and cannot be included here but can be accessible using the following [url](#).

Table 3.2: Beacon nodes deployed randomly in network (RD)

| $\mathbf{EN}^*$ | $\mathbf{Size}^\dagger$ | $\mathbf{A-LE}^\dagger$ | $\mathbf{A-NIT}^\S$ | $\mathbf{A-UNR}^\ddagger$ | $\mathbf{A-LT}^\parallel$ |
|-----------------|-------------------------|-------------------------|---------------------|---------------------------|---------------------------|
| $E_0$           | $S_1$                   | 0.0495                  | 2.89                | 0.00                      | 38                        |
|                 | $S_2$                   | 0.0498                  | 5.64                | 0.08                      | 67                        |
|                 | $S_3$                   | 0.0526                  | 11.56               | 40.44                     | 136                       |
|                 | $S_4$                   | 0.0579                  | 23.00               | 1333.08                   | 634                       |
| $E_1$           | $S_1$                   | 0.0643                  | 3.76                | 0.00                      | 41.88                     |
|                 | $S_2$                   | 0.0648                  | 7.33                | 0.09                      | 73                        |
|                 | $S_3$                   | 0.0684                  | 15.02               | 42.47                     | 150                       |
|                 | $S_4$                   | 0.0752                  | 29.90               | 1399.74                   | 697                       |
| $E_2$           | $S_1$                   | 0.1712                  | 5.63                | 1.52                      | 51                        |
|                 | $S_2$                   | 0.1811                  | 11.00               | 1.75                      | 103                       |
|                 | $S_3$                   | 0.1856                  | 22.53               | 34.02                     | 200                       |
|                 | $S_4$                   | 0.1911                  | 65.88               | 1413.64                   | 942                       |
| $E_3$           | $S_1$                   | 0.3413                  | 7.04                | 2.90                      | 63                        |
|                 | $S_2$                   | 0.3719                  | 15.28               | 3.75                      | 144                       |
|                 | $S_3$                   | 0.4123                  | 31.77               | 71.78                     | 266                       |
|                 | $S_4$                   | 0.4454                  | 87.82               | 1540.87                   | 1271                      |

Table 3.3: Beacon nodes deployed equidistance on boundary (EDB)

| $\mathbf{A-LE}^\dagger$ | $\mathbf{A-NIT}^\S$ | $\mathbf{A-UNR}^\ddagger$ | $\mathbf{A-LT}^\parallel$ |
|-------------------------|---------------------|---------------------------|---------------------------|
| 0.0655                  | 7.97                | 0.00                      | 213                       |
| 0.0579                  | 16.97               | 0.00                      | 286                       |
| 0.0576                  | 30.81               | 30.11                     | 507                       |
| 0.0591                  | 35.97               | 1747.83                   | 1108                      |
| 0.0851                  | 9.81                | 0.00                      | 235                       |
| 0.0753                  | 20.88               | 0.00                      | 309                       |
| 0.0748                  | 37.89               | 31.62                     | 542                       |
| 0.0769                  | 44.25               | 1835.23                   | 1178                      |
| 0.1874                  | 14.71               | 1.97                      | 290                       |
| 0.1924                  | 31.31               | 2.18                      | 432                       |
| 0.1966                  | 56.84               | 44.92                     | 721                       |
| 0.1989                  | 65.88               | 1908.02                   | 1591                      |
| 0.3718                  | 18.39               | 3.75                      | 356                       |
| 0.4019                  | 43.53               | 4.66                      | 605                       |
| 0.4383                  | 80.14               | 94.78                     | 958                       |
| 0.4513                  | 87.82               | 2079.74                   | 2148                      |

Table 3.4: Beacon nodes deployed randomly on boundary (RDB)

| $\mathbf{A-LE}^\dagger$ | $\mathbf{A-NIT}^\S$ | $\mathbf{A-UNR}^\ddagger$ | $\mathbf{A-LT}^\parallel$ |
|-------------------------|---------------------|---------------------------|---------------------------|
| 0.0601                  | 8.00                | 0.00                      | 209                       |
| 0.0567                  | 16.92               | 0.06                      | 300                       |
| 0.0565                  | 31.17               | 38.11                     | 529                       |
| 0.0593                  | 33.97               | 1892.64                   | 956                       |
| 0.0782                  | 10.40               | 0.00                      | 230                       |
| 0.0737                  | 21.99               | 0.06                      | 330                       |
| 0.0735                  | 40.52               | 40.02                     | 582                       |
| 0.0770                  | 44.16               | 1987.27                   | 1052                      |
| 0.1874                  | 15.60               | 2.10                      | 283                       |
| 0.1914                  | 32.99               | 2.78                      | 462                       |
| 0.1966                  | 60.78               | 48.91                     | 774                       |
| 0.2111                  | 65.88               | 1951.56                   | 1420                      |
| 0.3824                  | 19.50               | 3.99                      | 348                       |
| 0.4183                  | 45.85               | 5.95                      | 646                       |
| 0.4423                  | 85.69               | 103.20                    | 1030                      |
| 0.4609                  | 87.82               | 2127.20                   | 1917                      |

**Note:**  $\mathbf{EN}^*$  denotes the noise included by external environmental factors while calculating the RSSI value such as for free space ( $E_0$ ), sandy terrain ( $E_1$ ), long grassy terrain ( $E_2$ ), and sparse tree terrain ( $E_3$ ).  $\mathbf{Size}^\dagger$  denotes the area of the network in the square of metres,  $\mathbf{A-LE}^\dagger$  denotes Average error per node for localisation (in metres),  $\mathbf{A-NIT}^\S$  denotes the average number of iterations,  $\mathbf{A-UNR}^\ddagger$  denotes the average number of nodes remains unknown,  $\mathbf{A-LT}^\parallel$  denotes Average execution time (in seconds).

Table 3.5: Beacon nodes deployed randomly in network (RD)

| $EN^*$ | $Size^\dagger$ | $A-LE^\dagger$ | $A-NIT^s$ | $A-UNR^\ddagger$ | $A-LT^{  }$ |
|--------|----------------|----------------|-----------|------------------|-------------|
| $E_0$  | $S_5$          | 0.0740         | 12.75     | 8.42             | 650         |
|        | $S_6$          | 0.0740         | 18.63     | 150.96           | 1043        |
|        | $S_7$          | 0.0740         | 27.63     | 1188.29          | 2130        |
| $E_1$  | $S_5$          | 0.0962         | 16.58     | 8.84             | 715         |
|        | $S_6$          | 0.0962         | 24.21     | 158.51           | 1147        |
|        | $S_7$          | 0.0962         | 35.91     | 1247.71          | 2343        |
| $E_2$  | $S_5$          | 0.2089         | 24.86     | 7.74             | 943         |
|        | $S_6$          | 0.2109         | 32.44     | 91.36            | 1560        |
|        | $S_7$          | 0.2292         | 48.12     | 1517.19          | 3234        |
| $E_3$  | $S_5$          | 0.5012         | 32.82     | 9.52             | 1245        |
|        | $S_6$          | 0.5901         | 44.12     | 103.23           | 2122        |
|        | $S_7$          | 0.6442         | 68.33     | 1618.84          | 4463        |

Table 3.6: Beacon nodes deployed equidistance on boundary (EDB)

| $A-LE^\dagger$ | $A-NIT^s$ | $A-UNR^\ddagger$ | $A-LT^{  }$ |
|----------------|-----------|------------------|-------------|
| 0.0756         | 38.08     | 5.33             | 2946        |
| 0.0745         | 52.42     | 110.79           | 4170        |
| 0.0740         | 71.00     | 1883.75          | 5912        |
| 0.0983         | 46.84     | 5.60             | 3103        |
| 0.0969         | 64.47     | 116.33           | 4396        |
| 0.0962         | 87.33     | 1977.94          | 6227        |
| 0.2112         | 70.26     | 9.19             | 4096        |
| 0.2289         | 86.39     | 129.02           | 5978        |
| 0.2268         | 117.02    | 2011.94          | 8593        |
| 0.5231         | 92.75     | 11.31            | 5407        |
| 0.6501         | 117.49    | 145.79           | 8131        |
| 0.6834         | 166.17    | 2146.74          | 11859       |

Table 3.7: Beacon nodes deployed randomly on boundary (RDB)

| $A-LE^\dagger$ | $A-NIT^s$ | $A-UNR^\ddagger$ | $A-LT^{  }$ |
|----------------|-----------|------------------|-------------|
| 0.0758         | 37.96     | 6.46             | 2939        |
| 0.0749         | 54.46     | 136.63           | 4318        |
| 0.0748         | 73.58     | 1985.17          | 6701        |
| 0.0985         | 49.35     | 6.78             | 3233        |
| 0.0974         | 70.80     | 143.46           | 4750        |
| 0.0973         | 95.66     | 2084.43          | 7372        |
| 0.2232         | 74.02     | 11.56            | 4267        |
| 0.2289         | 94.87     | 135.02           | 6460        |
| 0.2312         | 128.18    | 2034.16          | 10173       |
| 0.5313         | 97.70     | 14.22            | 5633        |
| 0.6622         | 129.02    | 152.57           | 8785        |
| 0.6914         | 182.02    | 2170.45          | 14039       |

**Note:**  $EN^*$  denotes the noise included by external environmental factors while calculating the RSSI value such as for free space ( $E_0$ ), sandy terrain ( $E_1$ ), long grassy terrain ( $E_2$ ), and sparse tree terrain ( $E_3$ ).  $Size^\dagger$  denotes the area of the network in the square of metres,  $A-LE^\dagger$  denotes Average error per node for localisation (in metres),  $A-NIT^s$  denotes the average number of iterations,  $A-UNR^\ddagger$  denotes the average number of nodes remains unknown,  $A-LT^{||}$  denotes Average execution time (in seconds).

### 3.3.2.2 Visualisation of the Localisation Process:

In this section, we visualise the localisation process in the simulation setup with a large number of initially unlocalised nodes (IBN) and a small number of original beacon nodes (OBN). Three visualisations are presented, each showing a different deployment scheme for the OBNs, with parameters set as ROI as  $3000 \times 3000 m^2$ , IUN as 4000, and OBN as 1%. The visualisations depict the step-by-step localisation of unknown nodes over multiple iterations, eventually localising most or all of the IUN nodes. Figure 3.4 shows the localisation process with OBNs randomly deployed across the region. The OBNs (blue) are few, while the unlocalised IBNs (red) change to green as they are localised in successive iterations.

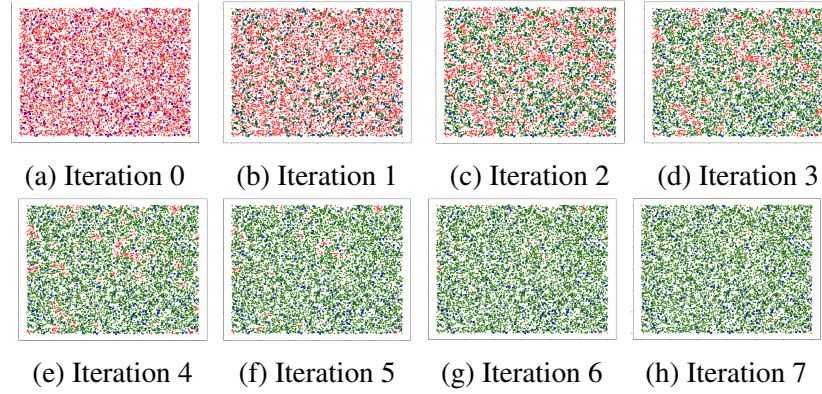


Figure 3.4: Node Localisation: Random Deployment

**Note:** Node localisation for Random original beacon nodes (OBNs) deployment. Here, blue nodes represent OBNs, while red and green nodes represent unlocalised and localised nodes in that iteration, respectively.

Figure 3.5 shows the localisation process with beacon nodes placed evenly along the boundary of the region of interest. In contrast, Figure 3.6 illustrates the process when OBNs are randomly placed at the boundary. These visualisations are designed to give the reader a clearer understanding of the process and its development.

The number of iterations required to localise the unknown nodes (UNs) depends on how the original beacon nodes (OBNs) are deployed. Random deployment (RD) of OBNs across the whole area typically requires fewer iterations than placing OBNs only on the boundaries

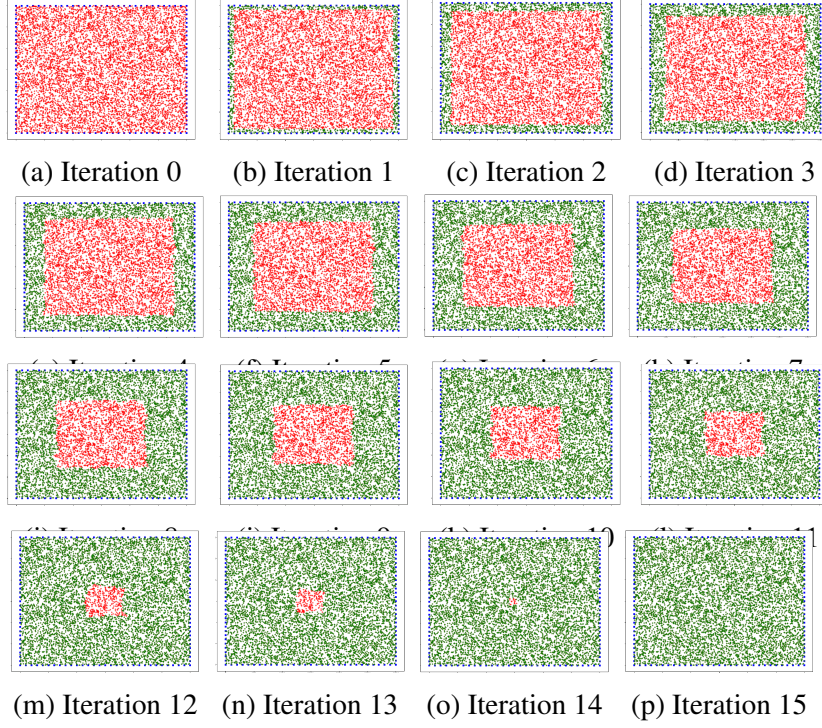


Figure 3.5: Node Localisation: Equidistant Deployment on Boundary

**Note:** Node localisation for original beacon nodes (OBNs) deployment at boundary. Here, blue nodes represent OBNs, while red and green nodes represent unlocalised and localised nodes in that iteration, respectively.

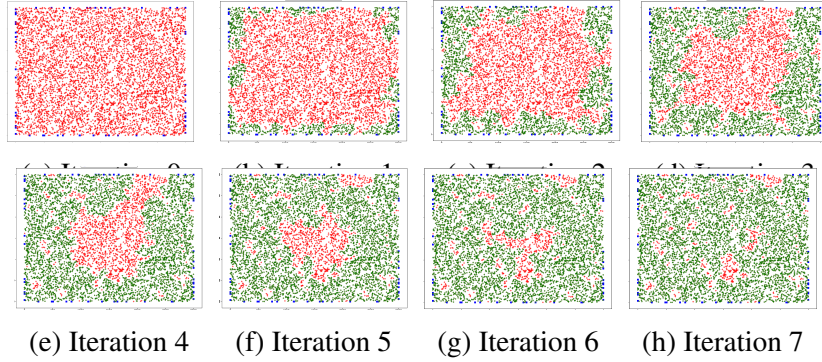


Figure 3.6: Node Localisation: Random Deployment on Boundary

**Note:** Node localisation for original beacon nodes (OBNs) deployed randomly at the boundary. Here, blue nodes represent initial beacon nodes, while red and green nodes represent unlocalised and localised nodes in that iteration, respectively.

(EDB and RDB). This is because the OBNs are spread nearer to the unknown nodes in random deployment (RD), which allows faster localisation in each iteration. Similarly, the OBNs randomly placed on the boundary (RDB) also require fewer iterations than those

arranged in a strict equidistant pattern (EDB).

In Figures 3.4, 3.5, and 3.6, the unknown nodes (IUNs) and OBNs are randomly deployed within a  $3000 \times 3000 \text{ m}^2$  area. The EDB deployment, shown in Figure 3.5 follows a specific pattern, resulting in fewer unresolved nodes (UNRs, marked in red) than RDB. However, in RDB, the localisation is irregular and can leave more unknown nodes (UNRs), increasing the number of iterations required to fully localise the area.

### 3.3.2.3 Impact of varying the area of the region of interest

In this section and the subsequent ones, we explore how different characteristics of the region of interest (ROI) affect the performance of the proposed localisation method. This analysis is essential for establishing the method as a reliable solution across varying regions. The first factor we examine is the size of the ROI. Figure 3.7 shows how variations in the area of ROI impact the average localisation error (A-LE), the number of unresolved nodes (UNR), the number of iterations (NITs) required to localise most nodes, and the total localisation time (LT).

For a more precise understanding, we used two ROI sizes: smaller regions ( $R_1$ ) ranging from  $1000 \times 1000$  to  $4000 \times 4000$  square meters, and larger regions ( $R_2$ ) from  $5000 \times 5000$  to  $7000 \times 7000$  square meters. Each experiment starts with 15,000 unknown nodes (IUN), and 1% of them (150 nodes) are beacon nodes (OBNs), deployed using three different types: random deployment (RD), random boundary deployment (RDB), and equidistant boundary deployment (EDB). To simulate real-world scenarios, we also tested different communication ranges (CR) for wireless sensor nodes: 50 m, 75 m, and 100 m for small regions ( $R_1$ ), and 75 m and 100 m for extensive regions ( $R_2$ ). We excluded the 50 m range for large regions due to the sparse node distribution in extensive regions, which makes 50 m too narrow for effective localisation. The effect of varying the ROI area on each factor is discussed

below:

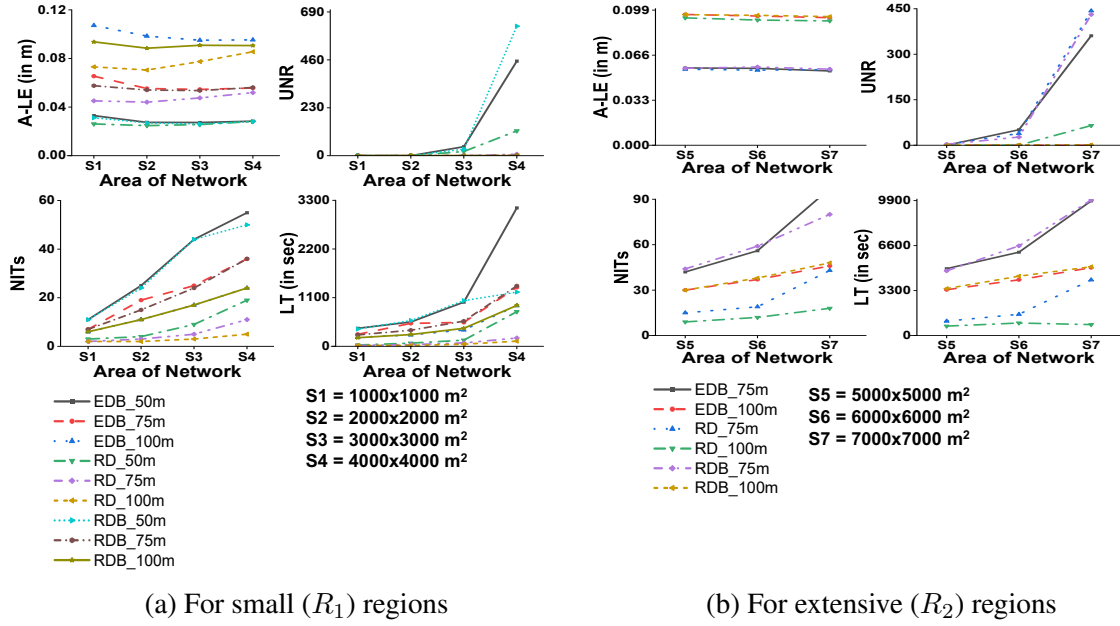


Figure 3.7: Impact of varying area

**Note:** The plot labels represent the communication range and deployment type (like 50m\_EDB represents the EDB type of deployment with a 50m communication range of a node).

- **Average Localisation Error (A-LE):** The average localisation error changes slightly as the size of the ROI increases, ensuring that the proposed model works well for localisation in outdoor areas of any size with reliable accuracy. Interestingly, sometimes, there is a slight reduction in error with larger ROIs, indicating the system performs even better in extensive regions.
- **Unknown Nodes Remaining (UNR):** The number of unlocalised nodes is nearly zero by the end of the localisation process for the smaller regions. However, for extensive regions, the number increases sharply. This happens because, in extensive regions, more unknown nodes remain outside the communication range of any beacon node.
- **Number of Iterations (NITs):** The number of iterations generally increases with the

ROI size. However, in some cases (as shown in Figure 3.7), the NITs remain constant beyond a specific area size. This occurs because, as the UNR increases in extensive regions, the number of iterations doesn't change, even though more nodes remain unresolved.

- Localisation Time (LT): The time required to localise the whole network increases with the size of ROI due to the higher number of iterations required. However, the LT decreases when the communication range (CR) is larger because more nodes are localised per iteration. In some cases, LT may also decrease with extensive area sizes, probably due to more unknown nodes remaining ( $UNR$ ), which reduces the total number of nodes localised and, consequently, the time required.

#### 3.3.2.4 Impact of varying the number of initial Unknown Nodes

In this section, we analyse how varying the number of Initial Unknown Nodes (IUNs) affects key parameters like Average Localisation Error (A-LE), the number of Unknown Nodes Remaining (UNR), the total number of iterations (NITs) needed to localise the entire region of interest (ROI), and the localisation time (LT). Figure 3.8 depicts the effect of varying IUNs on these parameters. We conduct experiments in two ROI sizes: a smaller region ( $R_1$ ) of  $3000 \times 3000$  square meters with 1.5% of the nodes being known beacon nodes (OBNs). For example, when the IUN count is 6000, there are 90 OBNs. We tried with IUN numbers 6000, 7000, and 8000, with communication ranges of 50m, 75m, and 100m, to observe the method's behaviour as IUNs change.

Similarly, in the extensive region ( $R_2$ ) of  $5000 \times 5000$  square meters, we maintain 1.5% OBNs while experimenting with IUN counts between 10,000 and 15,000. Communication ranges of 75m and 100m are used for this region, as 50m proves insufficient due to the greater node distances in extensive areas.

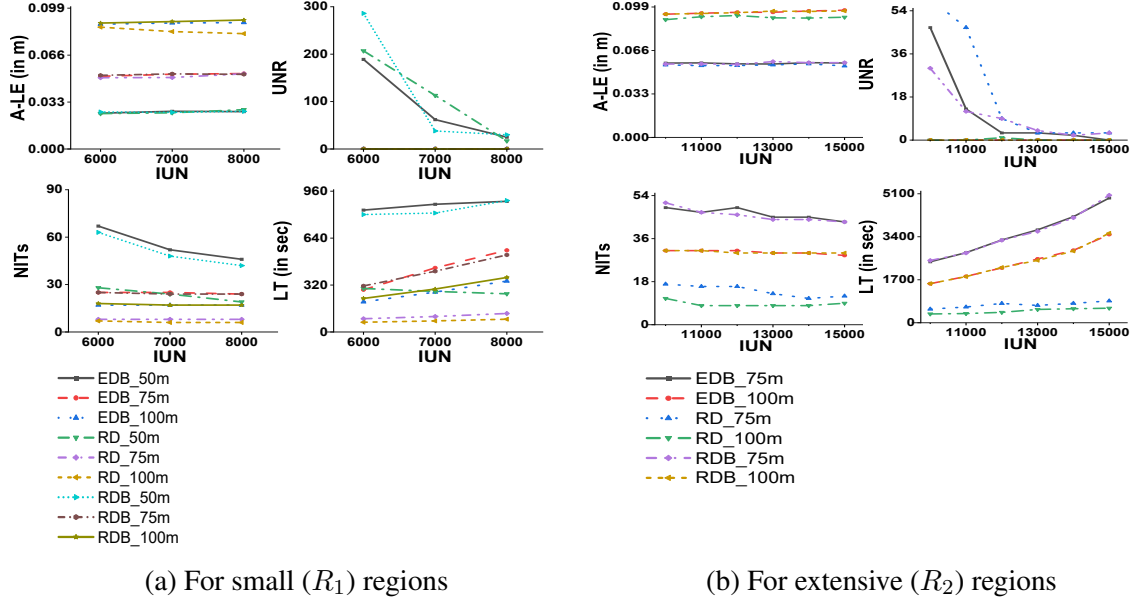


Figure 3.8: Impact of varying the initial unknown nodes

**Note:** The plot labels represent the communication range and deployment type (like 50m\_EDB represents the EDB type of deployment with a 50m communication range of a node).

The following things can be concluded from varying the number of IUNs:

- Average Localisation Error (A-LE): The A-LE increases gradually as the number of IUNs rises for small and extensive regions. This is anticipated since adding more unknown nodes slightly reduces accuracy when the number of known beacon nodes (OBNs) remains almost unchanged.
- Unknown Nodes Remaining (UNR): The number of unlocalised nodes (UNR) falls sharply as the number of IUN increases. This is because a higher density of IUNs within a fixed area means nodes are more closely deployed, making it easier for most of them to communicate with beacon nodes, leaving fewer unlocalised at the end.
- Number of Iterations (NITs): The number of iterations remains stable for higher communication ranges (75m and 100m). However, for lower ranges like 50m, the number of iterations required decreases significantly as the number of IUN increases. This

is because a higher node density helps nodes within smaller ranges connect more efficiently and reduces the number of iterations required.

- Localisation Time (LT): The time needed to localise the entire network increases as the number of IUNs and node density increases. This is due to the additional computations required to localise more nodes.

### 3.3.2.5 Impact of varying Deployment type

This section explores how various deployment strategies impact key aspects of our localisation approach. We focus on three strategies: a) Equidistant Deployment at the Boundary (EDB), the original beacon nodes (OBNs) are placed along the edges of the region at equal distances from each other; b) Random Deployment (RD), the OBNs are scattered randomly throughout the region of interest (ROI); and c) Randomly at the Boundary (RDB), where the OBNs are placed along the boundary but randomly, without maintaining equal distances. EDB and RDB are common in inhospitable terrains, where it is easier to establish node locations at the boundary using GPS or other means. RD is less common but can occur in environments like forests where random points with limited GPS connectivity allow for node deployment. These strategies were selected to cover a range of deployment possibilities. Section 3.3.2.2 provides helpful visualisations of the localisation process for each strategy.

The experiments to assess the impact of the deployment method were conducted in both small ( $R_1$ ) and extensive ( $R_2$ ) regions. In the small region ( $R_1$ ), there were 8000 IUNs, with 1% (or 80) being OBNs placed. For the extensive region ( $R_2$ ), there were 15,000 IUNs, with 1% (or 150) as OBNs. The communication range for all nodes in both regions was set to 100 meters. Figure 3.9 illustrates how the localisation approach changes with different network sizes and ROIs. The outcomes from the above experiments are as follows:

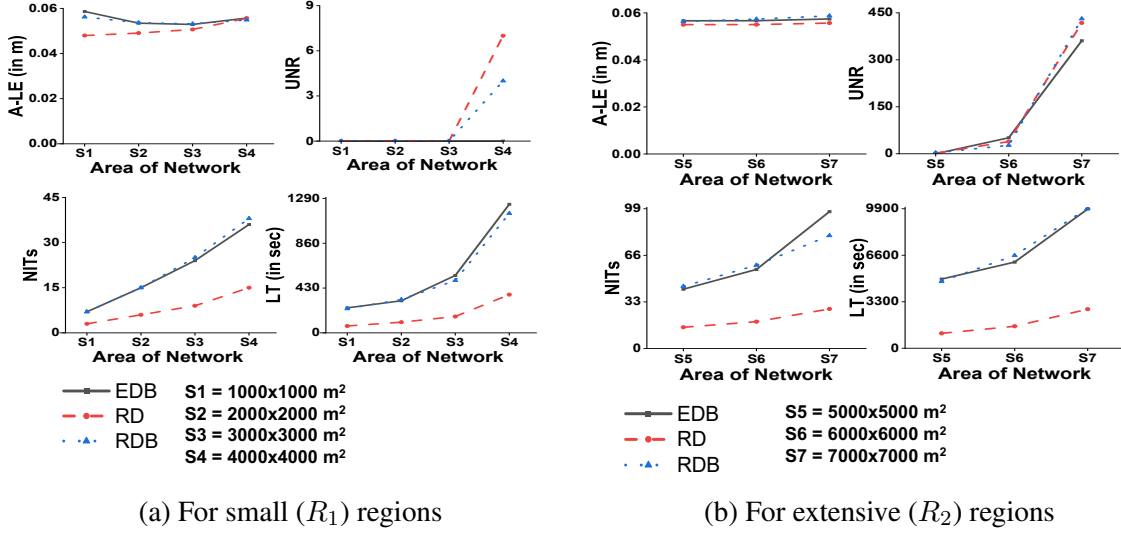


Figure 3.9: Impact of varying deployment type

- Average Localisation Error (A-LE): The A-LE remains consistently low across all three deployment types, even when we increased the size of the ROI for both small and large regions, backing that the proposed approach is practical regardless of the deployment type or the size of the area.
- Unknown Nodes Remaining (UNR): In the smaller ROIs, the number of unlocalised nodes (UNR) remains almost near zero. However, when the ROI size increases with the same number of IUNs, the density of the node decreases, which causes some IUSs to be too far from beacon nodes to be localised. This leads to a sharp increase in UNR as the ROI grows.
- Number of Iterations (NITs): The number of iterations increases proportionally to the size of ROI. Because the localisation process typically starts at the edges and moves inward in EDB and RDB deployments. In the RD strategy, fewer iterations are needed because the presence of OBNs throughout the region allows localisation to progress from multiple points, speeding up the process.

- Localisation Time (LT): Similar to the NITs, the localisation time (LT) increases with the size of the ROI. However, in the RD deployment, the time is much shorter since localisation can coincide from various points across the region.

### 3.3.2.6 Impact of varying Terrain

In this section, we aim to find the effectiveness of the proposed method in different types of terrain. We tried it in four different environments: open space (ENV1), sandy terrain (ENV2), grassy terrain (ENV3), and terrain with sparse trees (ENV4). Simulations were run for small ( $R_1$ ) and extensive regions ( $R_2$ ). In the  $R_1$  region, 8,000 unknown nodes (IUN) were deployed, with 1% (i.e. 80) of them as original beacon nodes (OBN), with a communication range (CR) of 75 meters. For the  $R_2$  regions, 15,000 IUNs were deployed, with 1% (150) OBNs, keeping the CR at 75 meters. Results were recorded for all the terrain types mentioned.

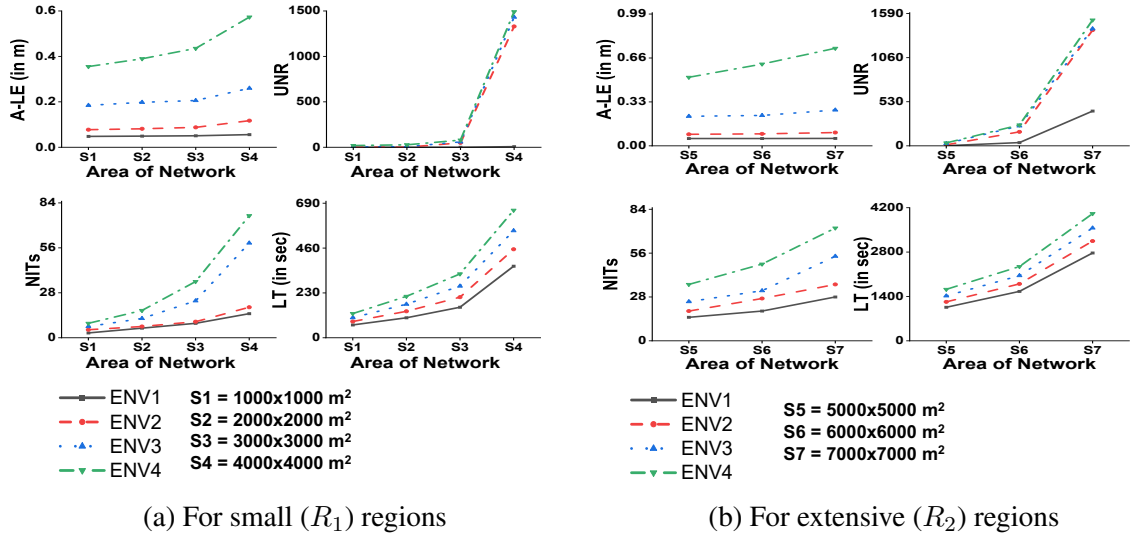


Figure 3.10: Impact of varying the environments.

- Average Localisation Error (A-LE): Localisation error rises as obstacles (environmental noise) increase, making it more difficult for the RSSI method to accurately

estimate the distances.

- **Unknown Nodes Remaining (UNR):** The number of unknown nodes that remain un-localised increases significantly as environmental noise becomes more intense due to obstacles.
- **Number of Iterations (NITs):** Environmental noise weakens signal strength (e.g., a signal reaching 75m in open space may only reach 25-30m in sparse trees). This reduction in communication range (CR) requires more iterations.
- **Localisation Time (LT):** As environmental noise grows, the time to localise the network also increases. More iterations (NITs) result in longer localisation time.

### 3.3.2.7 Impact of Accumulative Error

In this section, we aim to find how the cumulative localisation varies from one iteration to another. We tried to test the impact for both R1 and R2 regions, ranging in size from  $4000 \times 4000$  to  $7000 \times 7000 \text{ m}^2$ . The number of original beacon nodes (OBNs) was fixed at 1% of the initial unknown nodes (IUNs), with a communication range (CR) of 100 meters. The extensive region size and CR were used to assess the impact of cumulative error better. The results (Figure 3.11) indicate that while the cumulative error increases over iterations, the rise is gradual and not very pronounced.

### 3.3.2.8 Node localisability

In WSN, localisation can become easier when there are many nodes in a network (dense network). However, in extensive areas, ranging from  $1000 \times 1000 \text{ m}^2$  to  $7000 \times 7000 \text{ m}^2$  and beyond, deploying a dense network is challenging. For such areas, it's important to assess node localisability before deployment. This helps estimate the minimum number of nodes

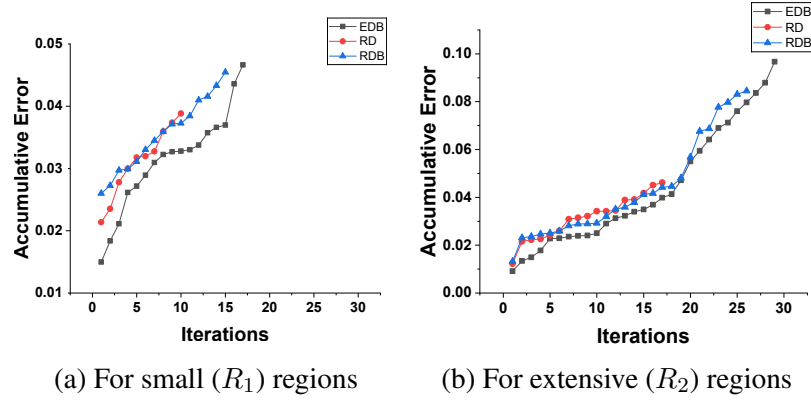


Figure 3.11: Impact of Accumulative Error

**Note:** Effect of Accumulative Error with respect to the number of iterations. For  $R_1$  and  $R_2$  regions, area sizes  $4000 \times 4000 \text{ m}^2$  and  $7000 \times 7000 \text{ m}^2$  are considered, respectively, with CR of 100 m. Here EDB, RDB means the beacon nodes are deployed equidistance and randomly on the boundary of the region. Where RD represents the beacon nodes are deployed randomly throughout the region.

needed to cover the area, determine how many can be localised, and identify which nodes are localisable. To establish this, we calculate the minimum number of nodes required to localise at least 95% of the unknown nodes, with a communication range (CR) of 100 meters. Simulations were run across areas of different sizes using three observing node (OBN) deployment methods: Random (RD), Equidistant on the Boundary (EDB), and Randomly on the Boundary (RDB). Table 3.8 summarises the minimum number of nodes needed for network localisation.

The simulation results were obtained for environment  $E_0$ , with no external interference, while keeping the number of observing nodes (OBNs) fixed at 80 for the  $S_1$  region and 100 for the  $S_2$  region. The OBNs were kept constant to simplify the interpretation of the results. The outcomes demonstrate that the proposed method performs effectively in both regions, achieving a high percentage of node localisability.

Table 3.8: Network localisability for various deployment and area sizes

| $S^\dagger$ | $MNR^\ddagger$ | EDB      |         | $MNR^\ddagger$ | RD       |         | $MNR^\ddagger$ | RDB      |         |
|-------------|----------------|----------|---------|----------------|----------|---------|----------------|----------|---------|
|             |                | $OBN^\S$ | $PL^\P$ |                | $OBN^\S$ | $PL^\P$ |                | $OBN^\S$ | $PL^\P$ |
| $S_1$       | 2273           | 80       | 95.78%  | 1210           | 80       | 95.88%  | 1650           | 80       | 95.83%  |
| $S_2$       | 5551           | 80       | 95.99%  | 2951           | 80       | 97.44%  | 4627           | 80       | 96.64%  |
| $S_3$       | 6837           | 80       | 95.36%  | 4553           | 80       | 96.42%  | 5910           | 80       | 95.58%  |
| $S_4$       | 7050           | 80       | 95.12%  | 5732           | 80       | 95.76%  | 6307           | 80       | 95.43%  |
| $S_5$       | 8527           | 100      | 95.31%  | 6845           | 100      | 96.06%  | 7839           | 100      | 95.65%  |
| $S_6$       | 9786           | 100      | 95.05%  | 7918           | 100      | 96.03%  | 8798           | 100      | 95.55%  |
| $S_7$       | 11003          | 100      | 95.56%  | 9423           | 100      | 96.53%  | 10385          | 100      | 96.35%  |

**Note:** The above table shows the minimum number of nodes required to localise the network. Here, **RD**, **EDB**, and **RDB** represent the deployment of *OBN* randomly, equidistant on the boundary, and randomly on the boundary, respectively. The  $S^\dagger$  and  $MNR^\ddagger$  imply the area of the network, and the minimum nodes (*IUN*) required to localise the network, respectively.  $OBN^\S$ ,  $PL^\P$  stand for the original beacon nodes deployed and percentage of *IUN* localised, respectively.

### 3.3.3 Comparison with SOTA

In this section, we compare the localisation accuracy of the proposed method with well-known techniques such as trilateration [7], Angle of Arrival (AoA) [13], and advanced machine learning methods [49] like neural networks, random forest, and support vector regression. The received signal strength is used in the trilateration (TL), random forest (RF), neural network (NN), and support vector regression (SVR) methods. For the AoA approach, two angles from different beacon nodes are used to localise an unknown node. An iterative process was applied to generate the results. Table 3.9 presents the simulation parameters used for comparison with state-of-the-art methods.

The experiments were conducted by varying the parameters listed in Table 3.9. Unlike the proposed method, the other five techniques found it hard to localise the initial unknown nodes (*IUN*) when the number of original beacon nodes (*OBN*) was limited to 1%, 1.5%, and 2.5% of the total *IUN*, with communication ranges (*CR*) of 50 m and 75 m. In these

Table 3.9: Simulation Parameters for comparison

| Parameters    | Values for small regions ( $R_1$ )                                        | Values for extensive regions ( $R_2$ )                                      |
|---------------|---------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| ROI           | $1000 \times 1000, 2000 \times 2000$<br>to $4000 \times 4000 \text{ m}^2$ | $5000 \times 5000, 6000 \times 6000,$<br>and $7000 \times 7000 \text{ m}^2$ |
| $IUN^*$       | 8000                                                                      | 15000                                                                       |
| $OBN^\dagger$ | 1, 1.5, 2.5, and 5 %                                                      | 1, 1.5, 2.5, and 5 %                                                        |
| $CR^\ddagger$ | 50, 75, and 100 m                                                         | 75, and 100 m                                                               |
| $EN^\S$       | $E_0, E_1, E_2,$ and $E_3$                                                | $E_0, E_1, E_2,$ and $E_3$                                                  |

**Note:** The size of the region is measured in square metres,  $IUN^*$  denotes the number of unknown nodes initially deployed,  $OBN^\dagger$  denotes the number of original beacon nodes (in terms of percent of the number of IUNs),  $CR^\ddagger$  denotes the communication range of a node (in metres), and  $EN^\S$  denotes the effect of noise due to environmental factors like free space ( $E_0$ ), sandy terrain ( $E_1$ ), long grassy terrain ( $E_2$ ), and terrains with sparse trees ( $E_3$ ).

cases, the number of unknown nodes remaining (UNR) was significantly high because these methods require at least two or more beacon nodes for localisation. The results were compared on four parameters: average localisation error, average number of iterations, average unknown nodes remaining, and average localisation time for both the  $R_1$  and  $R_2$  regions. Evaluations were also conducted based on four different environments, like open space ( $E_0$ ), sandy terrain ( $E_1$ ), grassy terrain ( $E_2$ ), and sparse tree terrain ( $E_3$ ).

### 3.3.3.1 Average Localisation Error

The average localisation error is an essential parameter for comparing different localisation algorithms. In 2D space, it is calculated using the Euclidean distance between the actual and predicted coordinates of the nodes. Tables 3.10 and 3.11 present the average localisation error for the  $R_1$  and  $R_2$  regions, respectively. The proposed method significantly outperforms the five state-of-the-art methods. In the  $R_1$  region, 8,000 IUNs were deployed, with 5% of them acting as OBNs, with a communication range (CR) of 100 meters. As noted earlier, we can only compare with this higher percentage of OBNs because the existing methods

struggled with fewer beacon nodes.

Table 3.10: Average Localisation Error (A-LE) for  $R_1$  Region when Beacon nodes randomly deployed in Network

| $EN^*$ | $Size^\dagger$ | $Proposed^\ddagger$ | $RF^§$ | $NN^¶$ | $SVR^  $ | $TL^{**}$ | $AoA^{\dagger\dagger}$ |
|--------|----------------|---------------------|--------|--------|----------|-----------|------------------------|
| $E_0$  | $S_1$          | 0.0332              | 0.0359 | 0.0376 | 0.0407   | 0.4524    | 0.4691                 |
|        | $S_2$          | 0.0345              | 0.0366 | 0.0389 | 0.0414   | 0.5198    | 0.6026                 |
|        | $S_3$          | 0.0383              | 0.0414 | 0.0427 | 0.0459   | 0.5130    | 0.5393                 |
|        | $S_4$          | 0.0466              | 0.0498 | 0.0528 | 0.0569   | 0.5325    | 0.6256                 |
| $E_1$  | $S_1$          | 0.0470              | 0.0502 | 0.0532 | 0.0575   | 0.5918    | 0.7409                 |
|        | $S_2$          | 0.0485              | 0.0526 | 0.0550 | 0.0590   | 0.6527    | 0.7645                 |
|        | $S_3$          | 0.0571              | 0.0615 | 0.0636 | 0.0689   | 0.6975    | 0.8405                 |
|        | $S_4$          | 0.0609              | 0.0658 | 0.0689 | 0.0748   | 0.7026    | 0.7641                 |
| $E_2$  | $S_1$          | 0.1599              | 0.1734 | 0.1820 | 0.1989   | 0.8564    | 0.8645                 |
|        | $S_2$          | 0.1648              | 0.1789 | 0.1875 | 0.2060   | 0.9157    | 0.9972                 |
|        | $S_3$          | 0.1733              | 0.1887 | 0.1972 | 0.2174   | 0.9356    | 1.0382                 |
|        | $S_4$          | 0.1768              | 0.1922 | 0.2016 | 0.2213   | 0.9646    | 1.0567                 |
| $E_3$  | $S_1$          | 0.3240              | 0.3530 | 0.3700 | 0.4073   | 1.1580    | 1.2435                 |
|        | $S_2$          | 0.3566              | 0.3881 | 0.4075 | 0.4476   | 1.1709    | 1.3130                 |
|        | $S_3$          | 0.3920              | 0.4272 | 0.4476 | 0.4919   | 1.3652    | 1.4929                 |
|        | $S_4$          | 0.4341              | 0.4724 | 0.4960 | 0.5443   | 1.5772    | 1.5939                 |

**Note:**  $EN^*$ , and  $Size^\dagger$  denotes the environmental noise and size of the network.  $Proposed^\ddagger$ ,  $RF^§$ ,  $NN^¶$ ,  $SVR^||$ ,  $TL^{**}$ ,  $AoA^{\dagger\dagger}$  denotes proposed method, random forest, neural network, support vector regression, trilateration, and angle of arrival, respectively.

Similarly, for  $R_2$  regions, 15000  $IUN$ s are deployed in different regions of various sizes, with the  $OBN$  constituting 5% of the total nodes and with a  $CR$  of 100 m. The proposed method performs better than existing state-of-the-art methods for regions of all sizes and environments.

Table 3.11: Average Localisation Error (A-LE) for  $R_2$  Region when Beacon nodes randomly deployed in Network

| $EN^*$ | $Size^\dagger$ | $Proposed^\ddagger$ | $RF^§$ | $NN^¶$ | $SVR^  $ | $TL^{**}$ | $AoA^{\dagger\dagger}$ |
|--------|----------------|---------------------|--------|--------|----------|-----------|------------------------|
| $E_0$  | $S_5$          | 0.0357              | 0.0430 | 0.0787 | 0.0487   | 0.5705    | 0.6606                 |
|        | $S_6$          | 0.0396              | 0.0481 | 0.0877 | 0.0536   | 0.5632    | 0.5918                 |
|        | $S_7$          | 0.0482              | 0.0570 | 0.1052 | 0.0653   | 0.5846    | 0.6859                 |
| $E_1$  | $S_5$          | 0.0505              | 0.0599 | 0.1105 | 0.0672   | 0.7150    | 0.8366                 |
|        | $S_6$          | 0.0595              | 0.0694 | 0.1289 | 0.0778   | 0.7639    | 0.9193                 |
|        | $S_7$          | 0.0635              | 0.0740 | 0.1375 | 0.0842   | 0.7696    | 0.8365                 |
| $E_2$  | $S_5$          | 0.1739              | 0.1939 | 0.3678 | 0.2218   | 1.0011    | 1.0897                 |
|        | $S_6$          | 0.1828              | 0.2043 | 0.3872 | 0.2339   | 1.0228    | 1.1344                 |
|        | $S_7$          | 0.1865              | 0.2083 | 0.3948 | 0.2382   | 1.0546    | 1.1548                 |
| $E_3$  | $S_5$          | 0.3785              | 0.4171 | 0.7955 | 0.4759   | 1.2786    | 1.4331                 |
|        | $S_6$          | 0.4160              | 0.4586 | 0.8747 | 0.5226   | 1.4901    | 1.6289                 |
|        | $S_7$          | 0.4607              | 0.5068 | 0.9675 | 0.5780   | 1.7209    | 1.7390                 |

**Note:**  $EN^*$ , and  $Size^\dagger$  denotes the environmental noise and size of the network.  $Proposed^\ddagger$ ,  $RF^§$ ,  $NN^¶$ ,  $SVR^||$ ,  $TL^{**}$ ,  $AoA^{\dagger\dagger}$  denotes proposed method, random forest, neural network, support vector regression, trilateration, and angle of arrival, respectively.

### 3.3.3.2 Average Number of Iterations

This parameter represents the number of iterations needed to complete the localisation process of the network. The smaller number of iterations signifies a more efficient algorithm. This parameter is essential in real-world implementations because changing the topological information of the beacon and unknown known nodes is difficult and crates an extra overhead (in terms of energy) on the network. This parameter also helps to analyse the number of nodes localised per round, which is a key parameter for node localisability.

Experiments were conducted using 8,000 IUNs, 5% of IUN as OBNs, with a communication range (CR) of 100 meters across four environments for  $R_1$ . Similarly, for  $R_2$  regions, 15,000 IUNs were deployed, with 5% of IUNs being OBNs and a CR of 100 meters in all

four environments. The experimental results clearly demonstrate that the proposed method outperforms existing approaches in terms of iteration efficiency. While not all results are included here, interested readers can access them through the following [Link](#).

### 3.3.3.3 Average Unknown Nodes Remaining

The Average Unknown Nodes Remaining parameter shows the average number of initially unknown nodes (IUNs) that remain unlocalised at the end of the localisation process. An IUN may stay unlocalised if it falls outside the communication range of beacon nodes or acts as an outlier within the network. A smaller number of unlocalised nodes indicates better algorithm performance. Network density, defined as the number of IUNs deployed, is inversely related to the number of unknown nodes remaining.

For R1 regions, the average number of unknown nodes was evaluated by deploying 8,000 IUNs, 5% of IUN as OBNs, with a communication range (CR) of 100 meters. Similarly, for R2 regions, 15,000 IUNs were deployed, with 5% of IUNs as OBNs and a CR of 100 meters. These results, computed across four distinct simulation environments, consistently show that the proposed method outperforms existing state-of-the-art approaches. For a detailed breakdown of the quantitative results, please refer to the following [URL](#).

### 3.3.3.4 Average Localisation Time

The average localisation time measures the duration required to localise all possible nodes within the network. This parameter is crucial for estimating the time needed to execute the algorithm in real-world scenarios such as forest fire detection or battlefield surveillance, where timely localisation is crucial.

The average localisation time was calculated by deploying 8,000 IUNs across regions ranging in size from  $1000 \times 1,000$  to  $4000 \times 4,000 \text{ m}^2$ , with 5% of the IUNs being OBNs and

a communication range of 100 meters. In extensive regions, ranging between  $5000 \times 5000$  and  $7000 \times 7000 \text{ m}^2$ , 15000 IUNs were deployed. Across all environments, the proposed method consistently outperforms state-of-the-art approaches. Interested readers can access the detailed results from the [URL](#).

### **3.4 Limitations**

The localisation algorithms are very useful in critical situations where it is challenging to detect the profound location of the object by human presence. The proposed localisation approach can be suitable for the 2-dimensional environments where the obstacles are as minimal as possible. For the practicability of the approach, we tried to calculate the Angle of Arrival values using some mechanical solutions, where we used stepper motor and laser light. This solution is not more accurate for the 3D environment where it is hard to penetrate the objects using the light rays, and Received Signal Strength (RSSI) values also result poorly.

### **3.5 Conclusion and Future Work**

This work proposed a novel technique for identifying the location of nodes in a Wireless Sensor Network (WSN) without the need for GPS and using the coordinates of only one beacon node whose location is known in advance. The technique proposed comprises the use of a hybrid combination of the angle of arrival (AoA) and received signal strength indicator (RSSI) to identify the location of unknown nodes. This hybrid localisation approach permeates through the entire region of interest, irrespective of its size, through an iterative approach. Here, the initial beacon nodes localise unknown nodes, which in turn serve as beacon nodes in the subsequent iteration. The uniqueness of our approach is that it is

the first of its kind that only requires one beacon node to localise an unknown node. This is made possible through the use of a combination of RSSI and AoA. Furthermore, our approach overcomes the unrealistic assumption that all unknown nodes are within the communication range of all beacon nodes to start with. This is made possible through adopting an iterative approach to cover the entire region of interest. The effectiveness of the proposed localisation approach was thoroughly validated through extensive simulation studies. This was necessitated owing to the lack of access to a sufficiently large WSN deployment. Subsequently, the real-world efficacy of the approach was validated through a prototypical implementation.

## **Chapter 4**

# **Preserving WSN Energy in Localisation using Machine-Learning Model**

### **4.1 Understanding the Problem**

Wireless Sensor Network (WSN) is an infrastructure-less, self-configured network of sensor nodes that communicate with each other via radio signals. Each node in a WSN is laden with sensors of various kinds, and these are often deployed in dangerous terrains that are inaccessible to humans. The sensors on these nodes send back relevant sensed data via an ad-hoc network of nodes that constitutes a WSN to a back-end cloud for analysis. Once deployed in such terrains, a sensor node is on its own with limited energy and computational resources and no means of replenishing these. The aim, therefore, is to minimise energy expenditure and prolong the useful life of nodes. In such circumstances, the localisation of sensor nodes in WSN is a crucial issue. This is because the usual localisation approach in outdoor locations using Global Positioning Systems (GPS) is infeasible. GPS comprises resource-intensive modules, and deploying these over WSN nodes significantly shortens the latter's life. In addition to this, the geographical locations in which such nodes are deployed often do not facilitate the proper functioning of GPS modules. This is displayed in our

project wherein we are deploying a WSN in a Tiger Reserve, which is a thickly forested area, to detect forest fires. Although the WSN works well and warns of a fire effectively, determining the location of the fire is non-trivial. GPS modules not only work here owing to their heavy nature but also because of the thickly wooded environment that disrupts GPS signals and makes them ineffective.

There are mainly two types of solutions for detecting the approximate location of the nodes: a) *range-free* [51], and b) *range-based* [52] localisation. Unlike the previous solution (in Chapter 3), both these localisation schemes work on the premise that there are specific nodes in the network whose correct locations are known. Such nodes are called *beacon nodes*, and based on these, the locations of the other nodes are computed. In realistic scenarios, like our project on forest fires, beacon nodes are usually deployed in parts of the terrain that are more accessible (for example, the periphery of the forested area in our project), where a GPS device can be used to determine the correct location. In most cases, such beacon nodes are few and far between and must be utilised effectively to localise most remaining nodes. In range-free localisation, the approach utilises simple data like the ‘number of hops’ between the beacon nodes and the node being localised to get a rough idea of the location of the node. The important point is that no additional hardware is utilised at any of the nodes to facilitate range-free localisation. The advantage of this approach is its simplicity and cost-effectiveness. The downside, however, is the low accuracy of localisation. A few examples of approaches employing range-free localisation are Centroid [53], DV-hop [54], and APIT [55]. Range-based localisation, on the other hand, requires additional hardware for transmitting and receiving signals at each node. In a WSN network, this hardware is already available at each node, and hence, range-based localisation becomes convenient. Range-based localisation involves an assessment of the signals received at unknown nodes from beacon nodes, and the strength, angle, and arrival time of such signals are utilised to

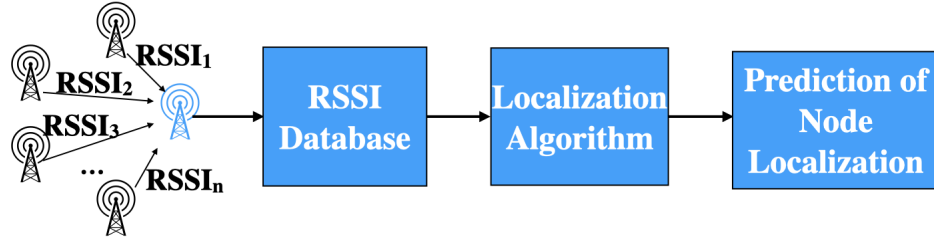


Figure 4.1: Localization using RSSI

assess the position of the node. Range-based localisation is much more accurate than the range-free approach and is more commonly employed for localisation. The angle of Arrival [56], Time of Arrival [57], and Received Signal Strength Indicator (RSSI) [58] are popular approaches that utilise range-based localisation techniques.

In this work, we utilise a range-based technique, more specifically, the Received Signal Strength Indicator (RSSI) technique, for localisation. The RSSI is a measure of the power present in a received radio signal, which can be used to estimate the distance between the transmitter and receiver. A high-level depiction of the use of RSSI for localisation is shown in Figure 4.1. Beacon nodes whose locations are known in advance to transmit signals that are received by the node to be localised. The strength of the received signals from different beacon nodes is analysed using various algorithms. Based on this, the position of the node is determined.

## 4.2 The Proposed Method

The method proposed in this work is meant for localisation of unknown nodes, without the use of a GPS device, in a WSN that is spread over a large area. ‘Large area’ here implies that most nodes in the WSN are not within the communication range of most other nodes owing to the large size of the area of interest. It is important to specify this as most existing localisation techniques work on the assumption that each node in the WSN is within the

communication range of every other node.

In this large area, we assume that the locations of a few sensor nodes called *beacon nodes*, are known in advance. These beacon nodes are located at the periphery of the area of interest. This is a realistic assumption as the sensor nodes at the periphery of the WSN are usually accessible and within the reach and range of a GPS device. The sensor nodes located deep within the area of interest are usually not accessible by a GPS device because of a hostile geographical terrain and/or the presence of disrupting structures like trees, and tall buildings. It is these nodes that need to be localised.

This work proposes a hybrid approach to localise such sensor nodes that comprise a Machine Learning (ML) approach combined with a more conventional multilateration approach. The ML algorithm harnessed here is a random forest, and it localises a large number of unknown nodes by analysing the RSSI values of communication signals received at the unknown nodes from one or more beacon nodes. Subsequently, these newly localised unknown nodes now serve as the ‘new’ beacon nodes and are used to localise nodes deeper inside the area using multilateration. The multilateration approach is usually harnessed for more than one iteration until all unknown nodes are localised. Figure 4.2 is a high-level depiction of the steps followed for localisation.

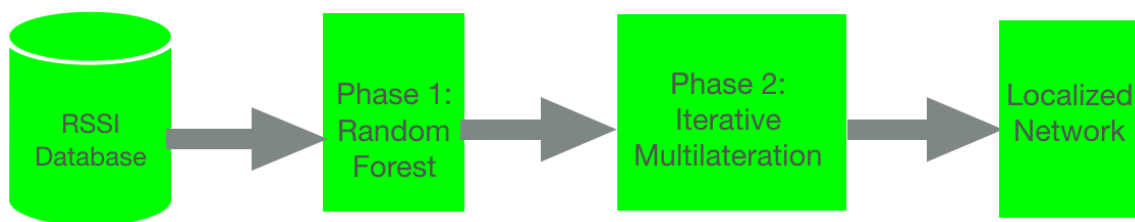


Figure 4.2: Proposed hybrid localisation approach

We now discuss the proposed approach, comprising localisation using RSSI in general, analysis of RSSI using an ML algorithm (random forest), and the use of multilateration with

RSSI for localisation, in more detail.

#### 4.2.0.1 Localisation using RSSI

Localisation through RSSI values comprises sending low-power signals from the transmitter at an beacon node (a node whose location is known) and receiving the signal using a receiver at an unknown node. The strength of the signal as received at the unknown node is assessed and analysed, and conclusions are drawn on the position of the unknown node relative to the beacon node that sends the signal. Usually, signals sent from multiple beacon nodes are received and analysed at the unknown node, and more accurate localisation is achieved. The intensity of signals received at the unknown node decreases with increasing distance from the transmitting beacon node i.e., an unknown node close to the beacon node receives a strong signal, while a distant unknown node receives a weak signal.

Equation 4.1 is Frii's Free Space Transmission Equation [59] and shows that the received signal strength decreases quadratically with distance from the transmitter.

$$P_r = \frac{P_t G_t G_r \lambda^2}{4\pi d^2} \quad (4.1)$$

Where  $P_r$  represents the power of a signal as received at an unknown node,  $P_t$  shows the power of a signal as transmitted at the beacon node.  $G_t$  and  $G_r$  represent the gain of the transmitter at the beacon node and the receiver's gain at the unknown node.  $d$  and  $\lambda$  show the distance between the beacon and the unknown node and the wavelength of the signal, respectively.

The power of the signal received at the unknown node is roughly interpreted as the Received Signal Strength Indicator (RSSI) value after incorporating factors specific to the communication technology in use. The RSSI values for signals received at unknown nodes from the various beacon nodes are collected and stored in a database. A matrix template for RSSI

values obtained at each node from every other node in the region of interest is shown in the matrix in Equation 4.2. Most of these RSSI values are assigned values of  $-200$  db, indicating that the receiving node is beyond the communication range of the sending node.

$$R = \begin{bmatrix} RSSI_{11} & RSSI_{12} & \dots & RSSI_{1k} \\ RSSI_{21} & RSSI_{22} & \dots & RSSI_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ RSSI_{n1} & RSSI_{n2} & \dots & RSSI_{nk} \end{bmatrix} \quad (4.2)$$

The RSSI values so collected are subsequently analysed by an ML algorithm (random forest in this case) and a multilateration technique for localisation.

### 4.2.1 Localisation using Machine Learning

The Machine Learning (ML) approach to localisation involves training an algorithm with data on a large number of sensor nodes. The data comprises the RSSI values of signals received at each node and the relative location of the node. The algorithm is trained in such a manner that it is able to accurately localise a node that receives relevant signals from at least three beacon nodes (beacon nodes, as mentioned earlier, are nodes whose locations are known). The larger the number of beacon nodes, the better the accuracy of localisation. The algorithm is trained in an ‘off-line’ manner such that it is trained before it is put to use for localising sensor nodes.

A large number of ML algorithms can be employed for localisation. We assessed several algorithms and, based on experiments, chose to use random forest in our work as it gave the best localisation accuracy. A comparison of the localisation accuracies of the ML algorithms that we experimented with is shown in Section 4.3, which discusses the experiments conducted.

Random forest [60] is an ensemble technique that can perform both regression and classi-

fication tasks [61]. A random forest comprises several decision trees, which are tree-like structures that divide a dataset based on the decisions taken at each node. The decision point or split value at a node is determined as one that provides the maximum information gain. Intuitively, a large information gain implies splitting the data in a manner that the data subsets formed as a result of splitting are more homogeneous i.e. datapoints in each subset formed are closer to each other in terms of attribute values. The decision tree establishes the best split amongst its variables with the intent of maximising information gain. This process of splitting begins at the root, and each node applies its own split function to the new input. This is repeated recursively until a terminal node is reached. A detailed discussion on forming a decision tree is beyond the scope of this work. The interested reader is pointed to [62]. Once trained, a decision tree is able to provide an appropriate value to a new data point. The random forest comprises several such decision trees as shown in Figure 4.3 and an average of the value assigned by each decision tree is assigned to the new point.

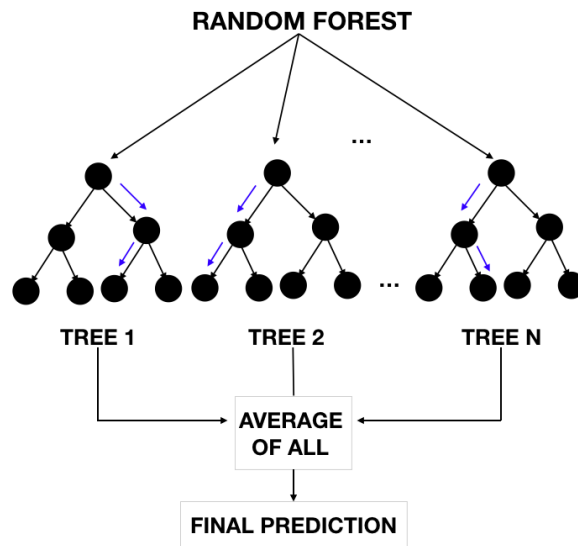


Figure 4.3: A generic random forest

#### 4.2.1.1 Data for the Random Forest

The first step in localisation using the random forest algorithm is the collection of data for training the model. The training entails teaching the model to correctly map RSSI values of signals received at a node with the 2D coordinates expressing the location of the node. The 2D coordinates of the nodes constitute the output of the random forest model. The input data consists of the RSSI values at the unknown nodes from various beacon nodes. At each unknown node  $N_i$ , we represent the RSSI value of the signal received from beacon node  $A_j$  as  $RSSI_{ij}$ . The input data and output of the model are in the formats shown in matrices 4.2 and 4.3, respectively, for  $k$  unknown sensor nodes and  $n$  beacon nodes.

$$O = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix} \quad (4.3)$$

In the matrix 4.2, the input data  $RSSI_{nk}$  corresponds to the RSSI value of the signal received at the  $n^{th}$  sensor node from the  $k^{th}$  beacon node; whereas in the matrix 4.3,  $(x_n, y_n)$  represent the coordinates of the  $n^{th}$  sensor node.

Each input data point comprises the set of RSSI values of signals received at an unknown node from all the beacon nodes. These constitute the features of the data point. Row  $i$  of the matrix in Equation 4.2 corresponds to the RSSI values of signals received at sensor node  $i$  from all the beacon nodes. In case a signal from an beacon node cannot reach a node owing to a large distance between them, the corresponding RSSI value is set to  $-200$  db. The output data comprises the coordinates of each of the unknown nodes. The training part involves the creation of a random forest and this requires labelled data for a large number of unknown nodes and corresponding beacon nodes. Depending on availability, this training data is procured from actual deployments, from standard datasets comprising mapped RSSI values and 2D coordinates, or the data is artificially generated using Frii's Free Space

Transmission Equation [59] shown in Equation 4.1. In our experiments, we use artificially generated data for lack of access to an extensive deployment and the unavailability of standard datasets.

#### 4.2.1.2 Data Preprocessing

Prior to creation of the random forest, the data collected goes through a quick step of pre-processing. Here a new parameter called  $\gamma$  is considered for each unknown node. The  $\gamma$  parameter indicates the number of beacon nodes for which the RSSI value at the node is not  $-200$  db. In other words, the  $\gamma$  parameter provides information on the number of beacon nodes whose signals reach the particular sensor node. For example,  $\gamma = 4$  indicates that the sensor node is within the communication range of 4 beacon nodes.

Only data points whose  $\gamma \geq 3$  are considered for the creation of the random forest. Those with smaller values are removed from consideration. This is because at least 3 legitimate RSSI values are required for accurate localisation with random forest.

#### 4.2.1.3 Creation of the Random Forest

To create a random forest, small bootstrap samples from the input data with  $\gamma \geq 3$  are taken, and a decision tree is developed with each sample. A small subset of the RSSI values at a node is considered for each tree. From this small subset of RSSI values, one RSSI value is randomly selected for the root node of the decision tree. A split point of this RSSI value is so selected that it gives the best improvement in terms of variance. For brevity, we do not dwell into the procedure for variance calculation and the interested reader is pointed to the following resource [62].

Based on the ‘best’ split point of the feature, the data is divided into two or more parts and these form the child nodes of the root. At each child node, again, a feature value (in

this case, RSSI value) is randomly chosen from the small sample, and the best-split point for this feature value further divides the data. This is continued until a certain number of iterations or until the data is exhausted, whichever comes first. The decision tree so created is combined with a larger random forest that comprises all such decision trees created.

The number of decision trees created in the random forest, called the  $n$ -estimator, is an important parameter that impacts the performance of the model. We experimented with using  $n$ -estimator values of 1000, 2000, and 3000. We got the best results with 2000 decision trees and used this value for further computations.

#### 4.2.1.4 Testing Phase

Of the legitimate RSSI data with values of  $\gamma \geq 3$ , 90% was allocated for training the model whereas 10% was kept aside for testing the efficacy. To test the model as well as use it with our real world implementation, the test point is made to go through each of the 2000 decision trees in the random forest. As the test data point moves through each tree and converges at a node in the tree, the  $x - y$  coordinates of the data point at the node are allocated as the coordinates of the test point. This is repeated for all 2000 decision trees, and finally, an average of all the 2000  $x$  and  $y$  coordinates is computed and allocated to the test point.

### 4.2.2 Localisation through Multilateration

Multilateration [63] is a localisation technique popularly used to localise vehicles in a GPS system. Multilateration depends on the relation between the distance of nodes and their relative location coordinates. To localise one node using multilateration, at least three nodes with known locations (beacon nodes in our case) within the communication range of the unknown node are required. The distance between an beacon node and the unknown is calculated using Frii's Free Space Transmission Equation [59] shown in Equation 4.1 that

relates the received signal strength value at the unknown node with the distance from the beacon node from which the signal was sent. This distance (which is not the exact distance but a computed approximate distance) is calculated between all the beacon nodes within the communication range of the unknown node and the unknown node. The calculated distance, along with the 2D coordinates of the beacon nodes, are together employed in the Least Squares Method [64]. Figure 4.4 is a high-level depiction of the localisation process in multilateration.

Equation 4.4 shows the expression that needs to be minimised to compute the loca-

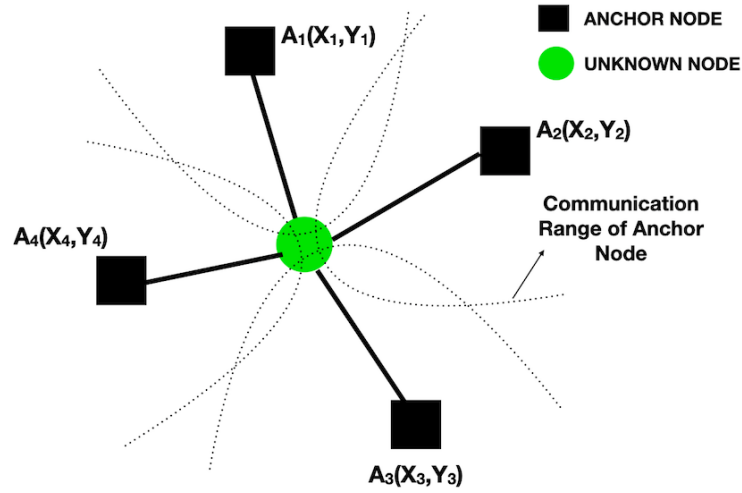


Figure 4.4: High-level depiction of multilateration

tion of the unknown node.  $\tilde{d}_i$  is the distance between the unknown node and the  $i^{th}$  beacon node as computed. The bar above  $d$  indicates that the value of the distance is not necessarily exact and is diluted by channel noise, obstacles, and other shadowing effects.

$$\text{Minimize } \varepsilon = \left| \sum_{i=1}^M \sqrt{(x_i - x)^2 + (y_i - y)^2} - \tilde{d}_i \right|^2 \quad (4.4)$$

$M$  denotes the number of beacon nodes within the communication range of the unknown node.  $M$  needs to be at least 3 for proper localisation. The square of the computed distance

between the unknown node and the beacon node is computed as follows:

$$\begin{aligned}
 (x_i - x)^2 + (y_i - y)^2 &= \tilde{d}_i^2 \\
 \forall i &= 1, \dots, M \\
 (x_i - x)^2 - (x_j - x)^2 + (y_i - y)^2 - (y_j - y)^2 \\
 &= \tilde{d}_i^2 - \tilde{d}_j^2 \\
 \forall i &= 1, \dots, M; i \neq j \\
 2x(x_j - x_i) + 2y(y_j - y_i) \\
 &= (\tilde{d}_j^2 - \tilde{d}_i^2) - (x_j^2 - x_i^2) - (y_j^2 - y_i^2) \\
 \forall i &= 1, \dots, M; i \neq j
 \end{aligned}$$

Expressing the equation in the form of a matrix:

$$\begin{bmatrix} 2(x_j - x_1) & 2(y_j - y_1) \\ \vdots & \vdots \\ 2(x_j - x_M) & 2(y_j - y_M) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \tilde{b}_j \\ \vdots \\ \tilde{b}_M \end{bmatrix}$$

Where

$$\begin{bmatrix} \tilde{b}_j \\ \vdots \\ \tilde{b}_M \end{bmatrix} = \begin{bmatrix} (\tilde{d}_j^2 - \tilde{d}_i^2) - (x_j^2 - x_i^2) - (y_j^2 - y_i^2) \\ \vdots \\ (\tilde{d}_j^2 - \tilde{d}_M^2) - (x_j^2 - x_M^2) - (y_j^2 - y_M^2) \end{bmatrix}$$

The form of the above equation is  $A.\bar{x} = \bar{b}$ . Using this, the location of the unknown node can be computed by minimising  $\|A.\bar{x} - \bar{b}\|^2$ . Using the Least Squares equation, the solution to the equation becomes  $\hat{x} = (A^T A)^{-1} A^T \bar{b}$ .

### 4.2.3 The Hybrid Approach to Localisation

We take a hybrid approach to localisation owing to limitations in the ML approach and the multilateration approach. The ML approach is effective in accurately localising a large number of sensor nodes harnessing the locations of just a few beacon nodes. However, the

ML approach's limitation is that it needs to be trained in advance and can only be employed for one iteration. It cannot be easily trained with the locations of the newly localised nodes and thus cannot be used for further iterations. The ML approach, therefore, is useful when all the unknown nodes are within the communication range of at least 3 beacon nodes. This is usually possible indoors and is seldom the case with large outdoor locations.

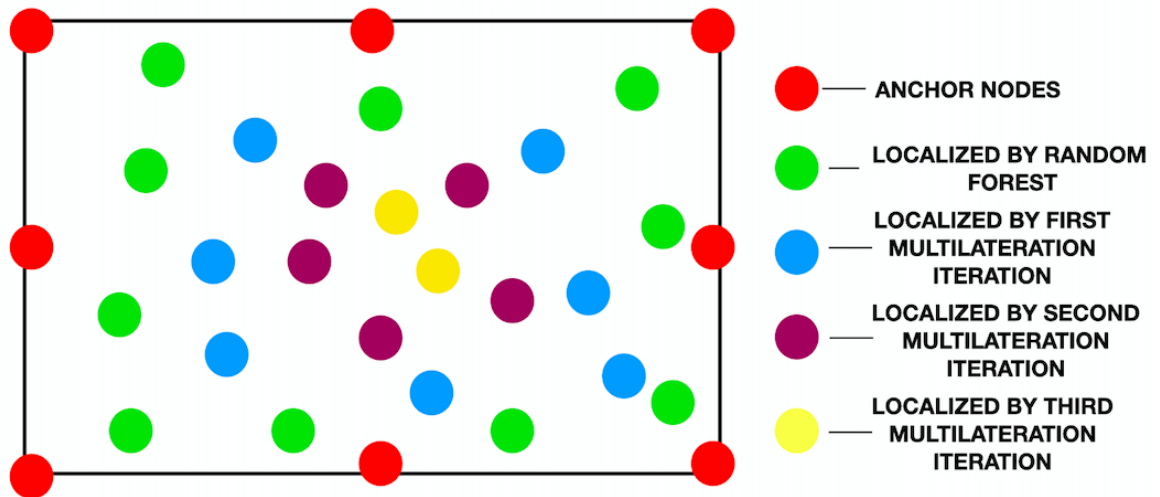


Figure 4.5: Hybrid localisation comprising ML and multilateration

The multilateration approach to localisation, on the other hand, can be readily employed for multiple iterations. Multiple iterations imply that the unknown nodes localised in an iteration become the new beacon nodes for subsequent iterations. The iterations continue until the entire area is covered. This is useful but has the drawback that localisations through multilateration are not very precise and this imprecision increases at every iteration. A very large number of iterations of multilateration localisation is therefore not advised.

The hybrid approach proposed in this work takes the best of both approaches. One iteration of ML localisation is first conducted. This results in significant number of unknown nodes getting accurately localised. These newly localised nodes become the new beacon nodes for subsequent localisations using multilateration. Combining the two approaches enables the coverage of most of the outdoor regions of interest. Figure 4.5 pictorially depicts

the hybrid approach proposed in this work. Algorithm 4.1 is a systematic description of the approach.

---

**Algorithm 4.1** Hybrid localisation

---

**Preconditions:**

```

1: Beacon nodes:  $A$ 
2: Unlocalised Sensor nodes:  $S$ 
3: function LOCALISATION( $A, S$ )
4:   RANDOM FOREST Localisation
5:    $S^{RF} \leftarrow$  nodes localised by random forest
6:    $S \leftarrow S - S^{RF}$ 
7:    $A \leftarrow A \cup S^{RF}$ 
8:   while  $num(S) \geq 0$  do
9:     MULTILATERATION Localisation
10:     $S^M \leftarrow$  nodes localised by Multilateration
11:     $S \leftarrow S - S^M$ 
12:     $A \leftarrow A \cup S^M$ 
13:   end while
14: end function

```

---

## 4.3 Evaluation

In this section, we experimentally assess the working of the random forest algorithm and the multilateration approach to localisation separately first and subsequently as a hybrid combination. We first create a simulated environment to comprehensively validate the approach and subsequently demonstrate its efficacy on a real-world set-up.

### 4.3.1 Dataset and Simulated Environment

To demonstrate the effectiveness of the proposed localisation approach, we create a simulated environment and a synthetic dataset. We need to synthesize the data as standard datasets for localisation over large areas do not exist. Also, we do not have access to real-world deployments of this scale.

We consider a  $130 \times 130 \text{ m}^2$  region. A dataset was synthesised with beacon nodes (nodes whose locations are known in advance) and sensor nodes (unknown nodes that need to be localised) deployed within this region. A total of 12,321 sensor nodes were created whose positions are along a  $1 \times 1 \text{ m}^2$  grid starting from a position of 10 m from the periphery of the region of interest and extending to a distance of 110 m. This is done along both the horizontal and vertical axes. 8 beacon nodes, whose locations are known, are placed at the periphery of the region of interest. This is a realistic scenario as nodes along the peripheries of real-world regions of interest are accessible, and their locations can be determined. The locations of the beacon nodes are as follows: (0,0), (60,0), (130,0), (0,60), (0,130), (60,130), (130,60), and (130,130). Each beacon node has a defined range to communicate with other sensor nodes.

Table 4.1 shows the number of sensor nodes that are within the range of communication

Table 4.1: Sensor nodes within communication range of beacon nodes

| Number of Sensor nodes | Neighbouring Beacon nodes |
|------------------------|---------------------------|
| 4666                   | 2                         |
| 3914                   | 3                         |
| 3622                   | 1                         |
| 119                    | 0                         |

**Note:** The above table shows the number of unknown nodes within the communication range of 3 and more, exact 2, exact 1, and none of the unknown nodes.

of different numbers of beacon nodes. For instance, 3914 out of 12321 sensor nodes are within the communication range of three beacon nodes. Each sensor node has an RSSI value associated with it based on their respective locations and distance from the beacon nodes. The RSSI value, as explained earlier, is the strength of signals received at a sensor node from various beacon nodes. The RSSI values for our synthetic dataset are computed using Frii's Free Space Transmission Equation [59] shown in Equation 4.1. Table 4.2 shows

the  $(x, y)$  coordinates and RSSI values at sensor nodes that form a part of the synthesised dataset. The uppermost row of the table shows the coordinates of the 8 beacon nodes, and below each beacon node's coordinates are the corresponding RSSI values at the various sensor nodes. In this table, several nodes have an RSSI value of  $-200$ . This indicates that the sensor node is not within the range of communication of the respective beacon node.

Table 4.2: A small segment of the synthetic dataset

| X   | Y   | (0,0)  | (60,0) | (130,0) | (0,60) | (0,130) | (60,130) | (130,60) | (130,130) |
|-----|-----|--------|--------|---------|--------|---------|----------|----------|-----------|
| 33  | 114 | -200.0 | -200.0 | -200.0  | -200.0 | -69.3   | -67.9    | -200.0   | -200.0    |
| 86  | 112 | -200.0 | -200.0 | -200.0  | -200.0 | -200.0  | -68.0    | -200.0   | -71.5     |
| 90  | 38  | -200.0 | -71.7  | -72.8   | -200.0 | -200.0  | -200.0   | -71.2    | -200.0    |
| 59  | 14  | -200.0 | -60.9  | -200.0  | -200.0 | -200.0  | -200.0   | -200.0   | -200.0    |
| ... | ... | ...    | ...    | ...     | ...    | ...     | ...      | ...      | ...       |
| 108 | 57  | -200.0 | -200.0 | -200.0  | -200.0 | -200.0  | -200.0   | -64.9    | -200.0    |
| 35  | 96  | -200.0 | -200.0 | -200.0  | -72.0  | -71.8   | -70.5    | -200.0   | -200.0    |

**Note:** The table shows the  $X$  and  $Y$  coordinate values of the unknown nodes (in  $X$  &  $Y$  columns). The rest of the columns show the measured RSSI values of this unknown node with respect to other beacon nodes placed on positions from  $(0, 0)$ ,  $(60, 0)$  to  $(130, 130)$ .

### 4.3.2 Machine Learning (Random Forest) Localisation

We choose random forest as the ML algorithm for the first localisation iteration. Of the total of 3914 sensor nodes that are within the communication range of 3 beacon nodes (you may recall that for localisation, a node needs to be receiving signals from at least 3 beacon nodes), 90% of the nodes or 3523 nodes are set aside for training of the random forest and 10%, or 391 is used for testing.

#### 4.3.2.1 Localisation accuracy

Figure 4.6 shows an overlap between the actual locations of the 391 test sensor nodes that are localised. The red triangles in the figure denote the actual locations of the nodes, and the blue squares denote the predicted locations. The figure indicates the precision of the random forest localisation as almost all the blue squares are hidden behind red triangles, implying almost perfect localisation.

Table 4.3 shows the localisation results of the random forest algorithm for 10 randomly

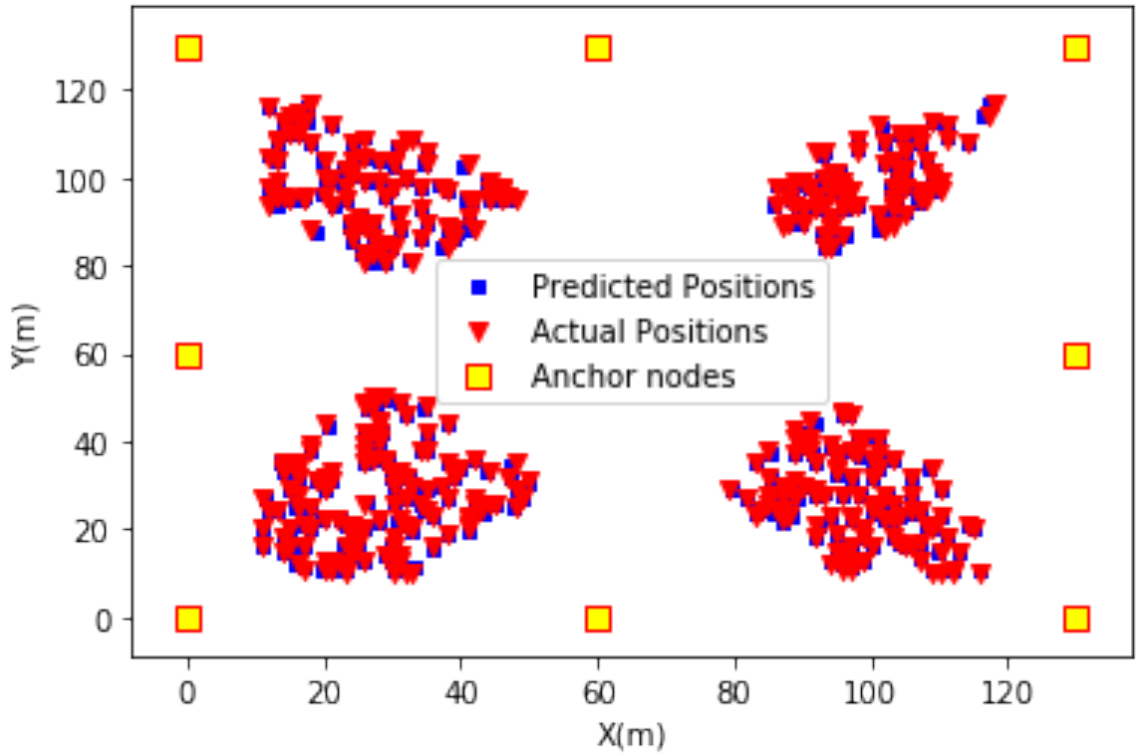


Figure 4.6: Overlap of predicted and actual locations of sensor nodes

selected data points. In this table ( $X_{actual}$ ,  $Y_{actual}$ ) are the actual coordinates of the data-points; ( $X_{pred}$ ,  $Y_{pred}$ ) are the predicted coordinates using random forest localisation; and Deviation indicates the distance between the actual and predicted locations. Table 4.4 further dwells into the precision of random forest localisation and shows the results of computing the average, minimum, and maximum change between the actual and predicted locations

Table 4.3: X-Y coordinates, actual vs predicted by random forest

| Xpred  | Ypred | Xactual | Yactual | Deviation (m) |
|--------|-------|---------|---------|---------------|
| 22.97  | 16.13 | 23.0    | 16.0    | 0.13          |
| 108.99 | 19.93 | 109.0   | 20.0    | 0.07          |
| 13.85  | 34.11 | 14.0    | 34.0    | 0.18          |
| 86.88  | 23.99 | 87.0    | 24.0    | 0.17          |
| 34.15  | 93.00 | 34.0    | 93.0    | 0.15          |
| 102.75 | 97.84 | 103.0   | 98.0    | 0.29          |
| 109.76 | 98.82 | 110.0   | 99.0    | 0.30          |
| 15.79  | 35.01 | 16.0    | 35.0    | 0.21          |
| 16.02  | 31.90 | 16.0    | 32.0    | 0.10          |
| 28.73  | 85.33 | 29.0    | 85.0    | 0.42          |

**Note:** The table shows the predicted position of  $x$  and  $y$  coordinates in  $X_{pred}$  and  $Y_{pred}$ . The  $X_{actual}$  and  $Y_{actual}$  shows the actual positions of  $X$  and  $Y$  coordinates, and deviation shows the Euclidean distance between actual and predicted data points.

for all the test sensor nodes. Equations 4.5, 4.6, and 4.7 respectively show the approach to computing these values.

Table 4.4: Deviation of predicted locations from actual values

| Compound of X and Y    | Deviation (m) |
|------------------------|---------------|
| Average Compound value | 0.20          |
| Minimum Compound value | 0.07          |
| Maximum Compound value | 0.30          |

$$\text{Average} = \frac{1}{\text{Datapoints}} \sum_{i=1}^{\text{Datapoints}} (\text{Distance}_i) \quad (4.5)$$

$$\text{Minimum} = \text{MIN}(\text{Distance}) \quad (4.6)$$

$$\text{Maximum} = \text{MAX}(\text{Distance}) \quad (4.7)$$

#### 4.3.2.2 Varying size of ‘region of interest’

We study the variation of the localisation accuracy of the random forest model by changing the size of the simulated area in Figure 4.7. It is seen that as the size of the simulated area increases, keeping the number of beacon nodes and the range of communication between the beacon nodes and the sensor nodes fixed; the localisation accuracy declines sharply. This is along expected lines and shows the impact that the size of the region within which localisation is done has on localisation accuracy. As the size of the region increases, it is imperative to increase the number of beacon nodes to maintain an acceptable level of localisation accuracy. This is vindicated in the following subsection where we experiment with increasing the number of beacon nodes.

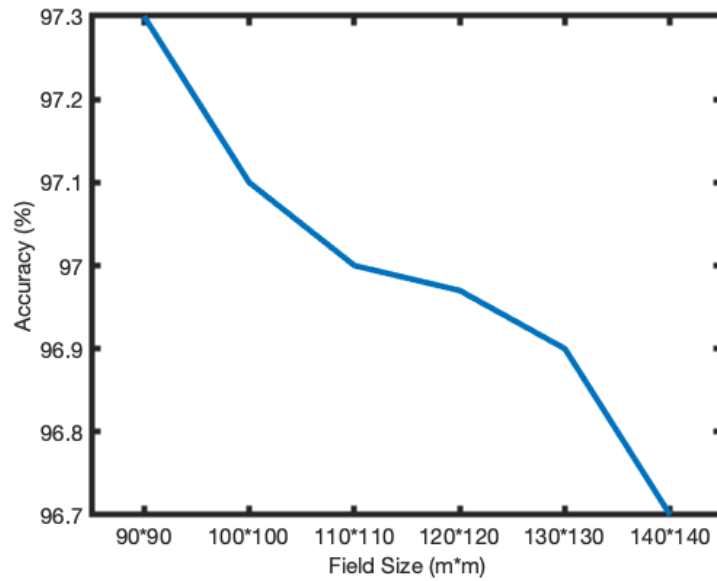


Figure 4.7: Localisation accuracy vs size of ‘region of interest’

#### 4.3.2.3 Varying number of beacon nodes

In Figure 4.8, we study the result of changing the number of beacon nodes in a fixed-size simulated area. An increase in the number of beacon nodes, with the range of communi-

cation between the beacon and the sensor nodes and the size of the simulated area (region of interest) fixed, results in a steady improvement in localisation accuracy. The number of beacon nodes becomes especially important for good localisation accuracy as we deal with larger regions of interest.

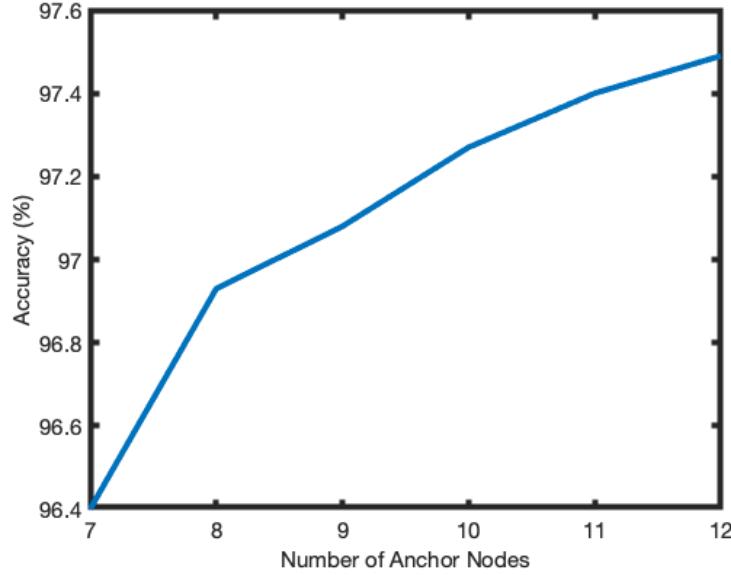


Figure 4.8: Localisation accuracy vs number of beacon nodes

#### 4.3.2.4 Comparison with other ML algorithms

We compare the localisation performance of random forests with other known machine learning algorithms on our simulated dataset.

Table 4.5, shows the localisation accuracy of the algorithm. The Error Margin,  $\alpha$  in the table provides an indication of the margin of error in localisation in the following manner: a datapoint  $(X_{actual}, X_{actual})$  is considered to be correctly predicted with an error margin  $\alpha$  if for the point, both the following are true:

$$X_{actual} * \alpha > |(X_{pred} - X_{actual})| \quad (4.8)$$

Table 4.5: localisation accuracies of various ML algorithms

| Algorithms           | Error Margin ( $\alpha=0.05$ ) | $\alpha=0.03$ | $\alpha=0.01$ |
|----------------------|--------------------------------|---------------|---------------|
| Neural Network       | 29.9%                          | 14.0%         | 10.2%         |
| SVR                  | 15.0%                          | 10.2%         | 8.1%          |
| Decision Tree        | 80.3%                          | 63.1%         | 24.5%         |
| <b>Random forest</b> | <b>96.9%</b>                   | <b>92%</b>    | <b>80.0%</b>  |
| XGBoost              | 97.18%                         | 92%           | 63.9%         |

$$Y_{actual} * \alpha > |(Y_{pred} - Y_{actual})| \quad (4.9)$$

If both equations 4.8 and 4.9 are true, then we consider that point to be a close point. For example 391 data points were considered as test points out of 3914 and  $\alpha$  is set to be 0.05 for which we got 379 close points out of 391 data points, so accuracy is calculated as:

$$\begin{aligned}
 \text{Accuracy} &= \text{Number of close points} * 100 / \text{Total test points} \\
 &= (379/391) * 100 \\
 &= 96.9\%
 \end{aligned}$$

The results clearly indicate the superiority of random forest in accurate localisation and vindicate our choice. XGBoost [65] does perform a little better when the margin of error permitted is large. However, the performance of XGBoost rapidly deteriorates with smaller permitted margins of error.

### 4.3.3 Multilateration Localisation

The other major localisation approach employed in this work is multilateration, as discussed earlier. Multilateration utilises the Least Squares Error technique to accurately localise nodes with distances computed from RSSI values. The advantage of the multilateration approach, in contrast to the ML localisation, is that it can be used for multiple iterations. This entails starting with a set of initial beacon nodes and using these to localise unknown sensor

nodes in the first iteration; the newly localised sensor nodes now become the new beacon nodes for the next iteration, localising further unknown nodes with this new set of beacon nodes; continuing this for multiple iterations. In this way, localisation is done over large ‘regions of interest’.

The downside of localisation with multilateration, however, is the inferior localisation accuracy as the iterations progress. The first iteration usually returns results that are acceptable and accurate. This deteriorates because the error in localisation at earlier iteration propagates through subsequent iterations. To explain this with an example: say an unknown node at coordinates  $(3, 4)$  is localised during the first multilateration iteration as  $(3.4, 4.2)$ . This node now becomes an beacon node for the subsequent iteration, and its already erroneous location further increases the error in subsequent localisations.

#### 4.3.3.1 Localisation over iterations

We conducted experiments to understand the extent of deterioration in localisation accuracy as the iterations of localisation with multilateration progress. To conduct this experiment, we use a  $50 \times 50 \text{ m}^2$  sized simulation environment with 8 beacon nodes positioned respectively at  $(0, 0)$ ,  $(25, 0)$ ,  $(50, 0)$ ,  $(25, 50)$ ,  $(50, 50)$ ,  $(0, 25)$ ,  $(0, 50)$ , and  $(50, 25)$ . The sensor nodes localised in the first iteration become the new beacon nodes for the next iteration and localise more sensor nodes. In this way, the nodes over the entire region of interest are localised in three iterations. Tables 4.6, 4.7, and 4.8 respectively show the localisation of sensor nodes after one, two, and three iterations of multilateration.  $(X_{\text{actual}}, Y_{\text{actual}})$  are the actual coordinates of the nodes localised and  $(X_{\text{pred}}, Y_{\text{pred}})$  are the coordinate values computed using multilateration. Deviation indicates the distance between the actual locations of the sensor nodes and the locations predicted by multilateration. The deviation values in the three tables indicate a trend towards deteriorating localisation accuracy as the

iterations progress.

Table 4.6: X-Y coordinates, actual vs predicted by first iteration of multilateration

| Xpred | Ypred | Xactual | Yactual | Deviation (m) |
|-------|-------|---------|---------|---------------|
| 10.36 | 13.36 | 11      | 14      | 0.90          |
| 14.24 | 11.30 | 15      | 12      | 1.03          |
| 35.76 | 12.21 | 35      | 13      | 1.09          |
| 37.61 | 12.30 | 37      | 13      | 0.92          |
| 9.60  | 9.60  | 10      | 10      | 0.56          |
| 11.25 | 35.75 | 12      | 35      | 1.06          |
| 13.36 | 39.63 | 14      | 39      | 0.89          |
| 38.94 | 10.06 | 39      | 10      | 0.08          |
| 9.27  | 39.49 | 10      | 39      | 0.87          |
| 36.78 | 36.78 | 36      | 36      | 1.10          |

**Note:** The table shows the predicted position of x and y coordinates in Xpred and Ypred. The Xactual and Yactual shows the actual positions of X and Y coordinates, and deviation shows the Euclidean distance between actual and predicted data points.

Table 4.7: X-Y coordinates, actual vs predicted by second iteration of multilateration

| Xpred | Ypred | Xactual | Yactual | Deviation (m) |
|-------|-------|---------|---------|---------------|
| 26.72 | 12.44 | 27      | 15      | 2.57          |
| 24.67 | 12.12 | 25      | 11      | 1.16          |
| 16.21 | 29.60 | 18      | 30      | 1.83          |
| 10.06 | 27.98 | 10      | 29      | 1.02          |
| 14.62 | 26.97 | 14      | 28      | 1.20          |
| 17.18 | 14.31 | 17      | 16      | 1.69          |
| 21.00 | 35.20 | 22      | 35      | 1.01          |
| 23.09 | 34.31 | 24      | 34      | 0.93          |
| 31.06 | 12.01 | 31      | 10      | 2.01          |
| 25.29 | 11.77 | 25      | 11      | 0.82          |

**Note:** The table shows the predicted position of x and y coordinates in Xpred and Ypred. The Xactual and Yactual shows the actual positions of X and Y coordinates, and deviation shows the Euclidean distance between actual and predicted data points.

Table 4.8: X-Y coordinates, actual vs predicted by third iteration of multilateration

| Xpred | Ypred | Xactual | Yactual | Deviation (m) |
|-------|-------|---------|---------|---------------|
| 30.99 | 34.61 | 32      | 35      | 1.08          |
| 32.97 | 32.94 | 34      | 33      | 1.03          |
| 32.01 | 36.07 | 33      | 37      | 1.35          |
| 33.99 | 34.85 | 35      | 35      | 1.02          |
| 31.14 | 34.24 | 33      | 35      | 2.00          |
| 34.57 | 36.46 | 35      | 36      | 0.62          |
| 37.01 | 34.75 | 38      | 34      | 1.24          |
| 38.03 | 34.64 | 39      | 33      | 1.90          |
| 35.99 | 35.17 | 37      | 34      | 1.54          |
| 31.11 | 36.0  | 31      | 39      | 3.00          |

***Note:** The table shows the predicted position of  $x$  and  $y$  coordinates in  $X_{pred}$  and  $Y_{pred}$ . The  $X_{actual}$  and  $Y_{actual}$  shows the actual positions of  $X$  and  $Y$  coordinates, and deviation shows the Euclidean distance between actual and predicted data points.*

#### 4.3.3.2 Comparison of multilateration and random forest

As stated earlier, localisation with multilateration has an advantage over random forest and other ML algorithms in terms of ease of conducting multiple iterations. The accuracy of localisation with multilateration, however, is inferior to that of random forest. We compare the localisation accuracy of multilateration and random forest in Table 4.9. The superiority of random forests in terms of localisation is clear from these results.

#### 4.3.4 The Hybrid Localisation Approach

In this work, we combine the localisation potential of random forest localisation and multilateration localisation seeking to harness the strengths of both. Random forest is utilised in the first iteration and it localises a large number of sensor nodes with a high degree of accuracy. These newly localised sensor nodes serve as the beacon nodes for the subsequent iterations of localisation which is done using multilateration. As discussed earlier, it is difficult to harness random forest for more than one iteration as it needs to be trained in advance

Table 4.9: Comparison of localisation by random forest and multilateration

| Xactual | Yactual | XR    | YR   | XM    | YM   |
|---------|---------|-------|------|-------|------|
| 23      | 16      | 22.9  | 16.1 | 22.01 | 18.9 |
| 109     | 20      | 108.9 | 19.9 | 109.1 | 19.8 |
| 14      | 34      | 13.8  | 34.1 | 13.8  | 33.8 |
| 16      | 35      | 15.7  | 35.0 | 16.0  | 35.0 |
| 87      | 24      | 86.9  | 23.9 | 88.8  | 21.7 |
| 27      | 50      | 27.6  | 49.3 | 26.4  | 49.7 |
| 104     | 99      | 104.1 | 99.1 | 103.8 | 98.9 |
| 97      | 16      | 97.0  | 15.9 | 97.0  | 16.2 |
| 28      | 41      | 28.0  | 41.0 | 27.8  | 40.5 |
| 102     | 88      | 101.1 | 88.3 | 101.9 | 88.0 |

**Note:** In the table ( $X_{actual}$ ,  $Y_{actual}$ ) are the actual coordinates of a random set of nodes; ( $XR$ ,  $YR$ ) are the coordinates computed by the random forest algorithm; and ( $XM$ ,  $YM$ ) are the coordinates computed by multilateration for the same nodes.

in an ‘offline’ manner. Table 4.10 is a segment of a larger table that shows the number of beacon nodes that sensor nodes are in the neighbourhood (neighbourhood implies being within the communication range). This is after the first iteration of the random forest is completed. It is interesting to compare Table 4.10 with Table 4.1 that shows the number of beacon nodes in the neighbourhood of unknown sensor nodes before localisation by random forest. The effect of the random forest algorithm is that in just one iteration, it makes subsequent multilateration iterations very effective by creating a large number of beacon nodes.

We observe in Table 4.10 that all the remaining sensor nodes are within the communication range of at least 12 beacon nodes. This significantly bolsters the localisation accuracy of the subsequent multilateration iteration. Table 4.11 shows the localisation results for 10 random sensor nodes in terms of the predicted coordinates ( $X_{pred}$ ,  $Y_{pred}$ ) and actual coordinates ( $X_{actual}$ ,  $Y_{actual}$ ). The Deviation column shows the distance between the actual locations of the nodes and the locations predicted by the hybrid approach. The results indi-

Table 4.10: Beacon nodes in the neighbourhood of sensor nodes

| Number of Sensor nodes | Neighbouring Beacon nodes |
|------------------------|---------------------------|
| 705                    | 75                        |
| 309                    | 73                        |
| 292                    | 74                        |
| 287                    | 91                        |
| 202                    | 103                       |
| ...                    | ...                       |
| 3                      | 122                       |
| 20                     | 12                        |
| 2                      | 13                        |
| 2                      | 14                        |
| 2                      | 12                        |

cate acceptable localisation with small deviations from actual locations owing to the initial boost provided to multilateration in terms of a large number of beacon nodes provided by random forest. The hybrid approach, therefore, is seen to be quite useful for localisation of nodes in large outdoor spaces.

Table 4.11: X-Y coordinates, actual vs predicted by hybrid approach

| Xpred  | Ypred | Xactual | Yactual | Deviation (m) |
|--------|-------|---------|---------|---------------|
| 86.01  | 47.70 | 86      | 48      | 0.30          |
| 68.02  | 31.26 | 68      | 32      | 0.74          |
| 75.06  | 98.07 | 75      | 98      | 0.09          |
| 95.40  | 73.73 | 95      | 74      | 0.48          |
| 77.80  | 69.20 | 78      | 70      | 0.82          |
| 39.90  | 71.40 | 39      | 72      | 1.08          |
| 49.00  | 75.26 | 48      | 76      | 1.24          |
| 56.94  | 78.91 | 57      | 79      | 0.10          |
| 63.81  | 19.54 | 64      | 20      | 0.49          |
| 109.80 | 47.60 | 110     | 48      | 0.44          |

## **4.4 Limitations**

The proposed method uses a multi-iteration approach for localising the nodes in an extensive outdoor environment. The approach first localises a few unknown nodes with the help of some beacon nodes with some localisation error. In the subsequent iterations, these newly localised nodes will act as the beacon nodes and help localise other unknown nodes. The primary concern of the proposed work is that the model needs help understanding or showing the effect of the localisation inaccuracies from the previous rounds. If the model runs for a smaller region and is not considered, the accumulative errors for a few rounds won't affect the model's effectiveness. However, if the large region is considered for localisation, this error will drastically affect the model's performance.

## **4.5 Conclusion and Future Work**

In this work, we proposed a hybrid technique for localisation of nodes in a Wireless Sensor Network (WSN) without the use of GPS. The major contribution of our approach is that it overcomes the simplifying assumption that every node in the WSN deployment is within the communication range of every other node. Our hybrid approach combines the capability of random forest, a Machine Learning (ML) algorithm, with a more conventional multilateration algorithm. The random forest algorithm is trained in advance and is able to accurately localise a large number of unknown nodes using just a small number of beacon nodes (nodes whose locations are known in advance). It is difficult to train random forest 'on the go' and hence it cannot be used for subsequent iterations. The nodes localised by random forest, however, are utilised as new beacon nodes and employed for localisation of the remaining nodes by the multilateration approach. Multilateration is not as accurate as ML algorithms but can be repeated several times and hence is effective in covering large deployments. In

spite of being a little compromised in terms of accuracy of localisation, multilateration does a fairly decent job within the hybrid set-up owing to the initial boost provided by random forest wherein a large number of beacon nodes are created.

# **Chapter 5**

## **Preserving WSN Energy via Cluster Based Routing**

### **5.1 Understanding the Problem**

The primary objective of WSN is to collect the data from various dispersed nodes throughout the region and transmit the data to the base station. Data collection and transmission is the most energy-intensive task, draining the energy level of external resources drastically. The transmission energy required to transmit the data from various nodes to the base station is directly proportional to the distance between them. Therefore, it is important to drag the transmission distance between them to a minimum via the appropriate routing. Three types of routing are used in the WSN: a) flat routing, b) location-based routing, and c) cluster-based routing.

In flat routing, each node transmits the data to the one transmission (hop) away nodes in the network, also known as flooding. This approach is simple and forthright but unnecessarily consumes too much energy and congests the network, making it unsuitable for networks with larger area sizes or nodes. The location-based routing invents various optimal paths through different mechanisms between the source and destination nodes. Such a type of

routing is suitable when the source and destination move frequently. Suppose the transmission occurs between the same source and destination. In that case, the method follows the same path for data transmission, leading to an unbalanced network with the nodes on the path getting exhausted and less energy remaining. In contrast, other nodes remain with high energy [66]. The cluster-based routing initially forms the clusters based on the locations of the nodes. Then, it selects one node from each cluster as its head (known as cluster head), which is responsible for collecting the data from all the other cluster nodes and transmitting it to the base station. This approach significantly lessens the communication overhead if the heads are appropriately selected.

## **5.2 The Proposed Method**

This section describes the proposed approach for determining the optimal cluster. The approach helps save energy, which prolongs the network's lifespan. The model uses several hyperparameters, which help determine the optimal clusters and cluster heads. The grid search method is used to find the value of the optimal parameter. The grid search method exhaustively explores all the possible combinations of the hyperparameters and selects the best result. This identifies the best combination of hyperparameters and 'locks' them for further uses. The proposed method works as follows:

1. The optimal number of clusters is selected using the Silhouette method.
2. Initial clustering is performed. In the beginning, the initial energy level of all the nodes is almost the same, so only spatial information is used for the clustering.
3. One node in each cluster is selected as cluster head (based on some parameters).
4. Nodes start the data transmission to their heads, and heads aggregate all the data and

transmit it to the base station. The data transmission through the cluster head process is continuous till the energy level of the nodes in any of the clusters remains less than a particular threshold.

5. Secondary clustering is performed. Now, some of the nodes in the network have less remaining energy, so they are restricted from participating in the cluster formation process to avoid uncertainty. The secondary clustering includes remaining energy with the spatial information of nodes.
6. Data transmission and secondary cluster head selection processes are continuous until all nodes in the cluster exhaust their energies beyond a set threshold.

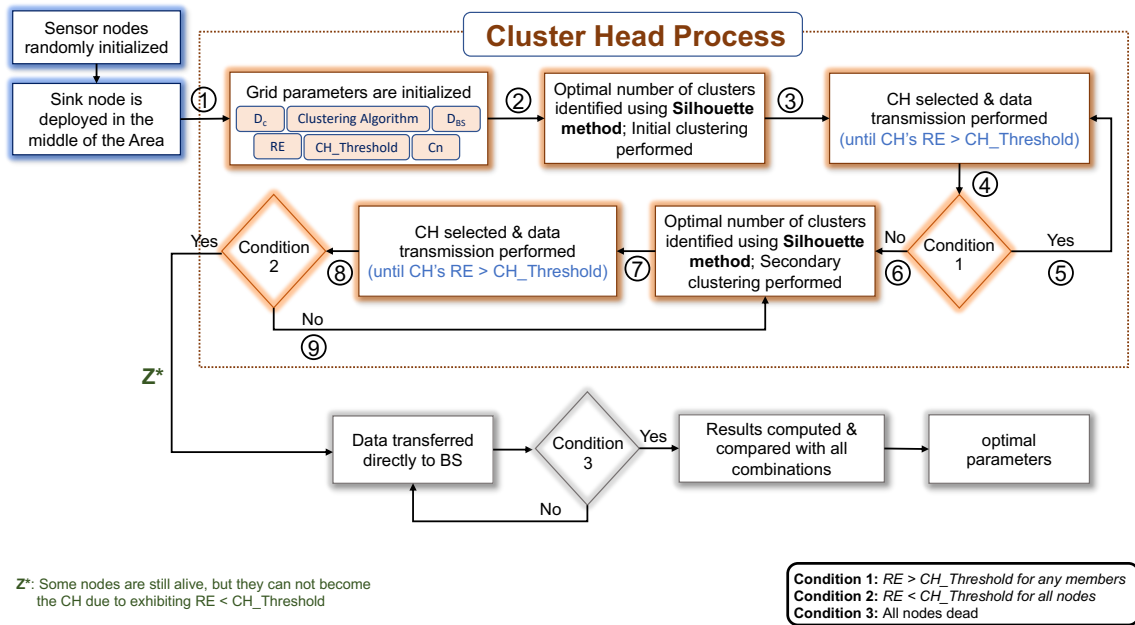


Figure 5.1: Flow Diagram: Proposed Cluster Head Selection method

**Note:** The proposed cluster head (CH) selection protocol initialises nodes randomly throughout the network, forms dynamic clusters and selects optimal CHs using a weighted objective function. The weights are updated using the Grid-search method. For initialising grid parameters, the method uses clustering algorithms, RE (residual energy), distance to the BS ( $D_{BS}$ ), distance to centroid ( $D_C$ ), Connectivity ( $C_n$ ), and cluster head threshold ( $CH_{Threshold}$ ). The Silhouette method is used to find the optimal number of clusters in the proposed method. The CHs transmit the data until the RE reaches a certain threshold.

There are several hyperparameters used for finding the optimal results, which include: a) The clustering algorithms (for both initial and secondary clustering), b) weights assigned for four cluster head selection parameters (residual energy, distance from node to base station and centroid of cluster, and node connectivity), c) threshold of residual energy for changing the cluster heads.

### 5.2.1 Determining the optimal cluster numbers

This approach transmits data from each node to its cluster head (CH) and from there to the base station (BS). The careful formation of clusters is essential to minimize the overall energy consumption during data transmission. The first step in cluster formation is determining the optimal number of clusters, denoted as  $K$  throughout the chapter. However, selecting the optimal number of clusters is a complex task with no definitive solution. The two most widely used techniques for this are the Elbow Method [67] and the Silhouette Method [68], both of which assess the Cohesion and Separation factors among nodes. Cohesion refers to the compactness within a cluster, while separation represents the distance between nodes in different clusters. High cohesion and separation values are key indicators of well-formed clusters.

#### 5.2.1.1 Elbow method

The Elbow Method considers a range of candidate values for the number of clusters ( $K$ ) by applying the K-Means algorithm to form clusters for each  $K$  value. Then, it calculates the average distance between each node and the centroid of its allotted cluster. This process is repeated for all possible  $K$  values, and the results are plotted, as shown in Figure 5.2 (a). The optimal  $K$  is identified when the average distance sharply decreases and starts to level

off, creating a curve resembling a hand's 'elbow' (as the name of the method). In Figure 5.2 (a), the optimal  $K$  is denoted as  $\alpha_3$ . While effective for smaller  $K$  values, the Elbow Method becomes less reliable as  $K$  increases. Larger  $K$  values lead to smaller clusters, causing the average distance to flatten and approach zero, making it difficult to determine the optimal  $K$ .

### 5.2.1.2 Silhouette method

The Silhouette method for identifying the optimal number of clusters is harnessed to overcome this limitation of the Elbow method. The Silhouette method considers the distance of a node from all the other nodes within its cluster (similar to the Elbow method). This is loosely termed as cohesion. In addition, the Silhouette method also uses the distance of each node from all the nodes in the neighbouring clusters. Here, a set of possible cluster numbers ( $K$ ) is also chosen, and clustering is performed using K-means for each of these  $K$  values. The average distance of each node  $i$  from all other nodes in its cluster is computed as  $a_i$ , and the average distance of  $i$  from all nodes in its closest neighbouring cluster is computed as  $b_i$ . Using  $a_i$  and  $b_i$ , a Silhouette co-efficient for node  $i$ ,  $s_i$  is computed following Equation 5.1. A value of  $s_i$  close to +1 indicates high cohesion and high separation, whereas a value close to -1 indicates low cohesion and separation and is undesirable. The average of all Silhouette coefficients provides the score and is plotted against the number of clusters  $K$ . The value of  $K$  that corresponds to the Silhouette score closest to +1 is chosen as the optimal value of  $K$ . In Figure 5.2 (b), the optimal value of  $K$  is  $\alpha_8$ .

$$\text{Silhouette coefficient (i)} = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (5.1)$$

## 5.2.2 Initial Clustering

Once the optimal number of clusters,  $k$ , is chosen, the first round of clustering is initiated, with the spatial location of the WSN nodes serving as the basis for cluster formation. We comprehensively explore various clustering algorithms to identify the most suitable one. This exploration follows a grid-search approach and tuning other appropriate hyperparam-

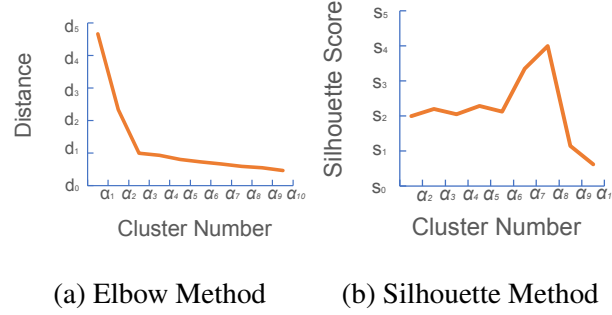


Figure 5.2: Optimal number of cluster selection methods

**Note:** Graphs for determining the optimal number of clusters. a) The Elbow method selects the point where the distance value starts decreasing rapidly. b) The Silhouette method selects the point with the highest silhouette score.

ters, allowing us to identify the best clustering algorithm for the task.

### 5.2.3 Cluster Head Selection

Once the  $K$  clusters are formed, a cluster head (CH) is selected within each cluster based on the following four essential parameters to ensure the most suitable CHs:

- **Residual Energy (RE):** RE is the most critical parameter in selecting a CH. Nodes with higher RE will likely stay longer, providing extended network operation.
- **Distance from Cluster Centroid ( $D_C$ ):** This parameter determines the distance between the node and the centroid of the respective cluster. The CH should be near the cluster centroid, as most nodes are typically close. This minimizes transmission energy, improving overall energy efficiency.
- **Distance from Base Station ( $D_{BS}$ ):** This shows the distance from a node to the base station (BS). A CH situated closer to the BS will consume less energy when transmitting data, as all communication with the base station occurs via the CH.
- **Connectivity (Cn):** Connectivity directs to the number of nodes for which the candidate node is the closest neighbour. A cluster head with high connectivity will transmit

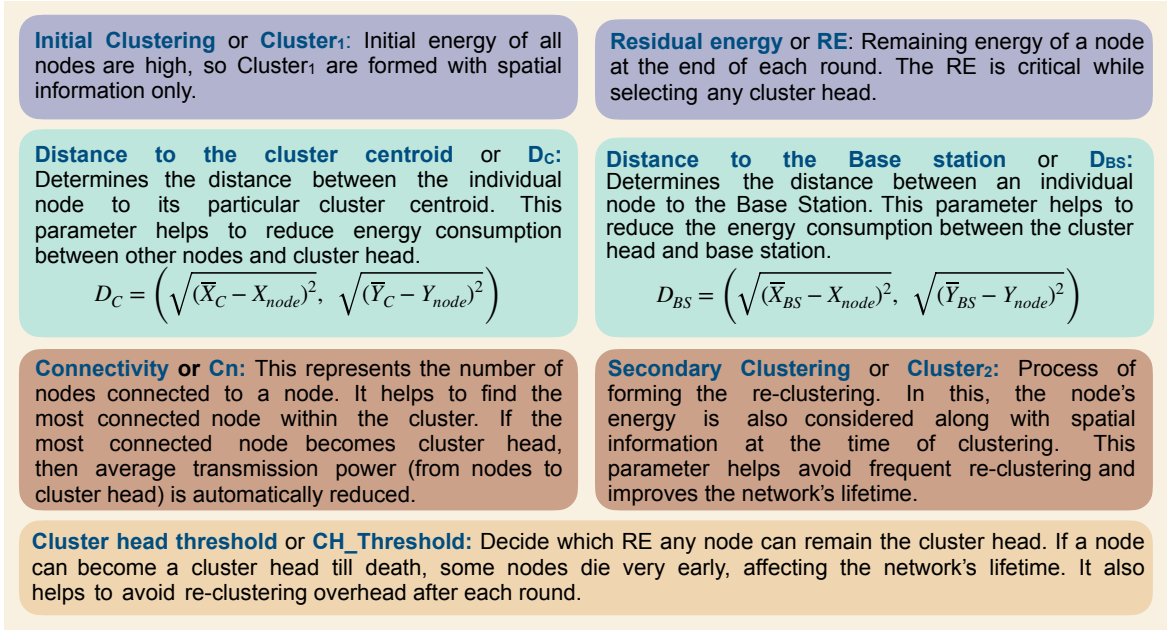


Figure 5.3: Parameters used for clustering and cluster head selection.

data more efficiently, helping in a longer network lifetime.

These four factors are weighted and combined to calculate each node's "Cluster Head Potential" (CHP) score. The node with the highest CHP is selected as the cluster head. The formula for calculating the CHP is illustrated in Equation 5.2:

$$CHP = \frac{RE^{w_1} * Cn^{w_2}}{D_C^{w_3} * D_{BS}^{w_4}} \quad (5.2)$$

The  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$  are weights of hyperparameters systematically optimised alongside other variables using grid search. The optimal values for these weights are determined and executed. For a more profound understanding of each factor, refer to the box in Figure 5.3.

## 5.2.4 Secondary clustering

The Cluster-head selection is performed whenever the energy level of the current cluster head drops below the predefined residual energy (RE) threshold. This process continues until no nodes in the cluster exceed the threshold. At that point, a new clustering of the nodes

is initiated. This secondary clustering process is similar to the initial clustering except that in addition to the spatial parameter, the residual energy of the nodes is also considered for the clustering. Various clustering algorithms are evaluated during this phase, with the most effective one being selected based on its combination with other hyperparameters within a chosen small sub-space. The newly formed clusters again go through the same iterative process of CH selection. When the energy level of all the nodes in a cluster drops below the RE threshold, the re-clustering is performed. The re-clustering process is performed until all the node's RE falls below the RE threshold. After that, the nodes transmit their data to the BS directly through some random path until they have some remaining energy.

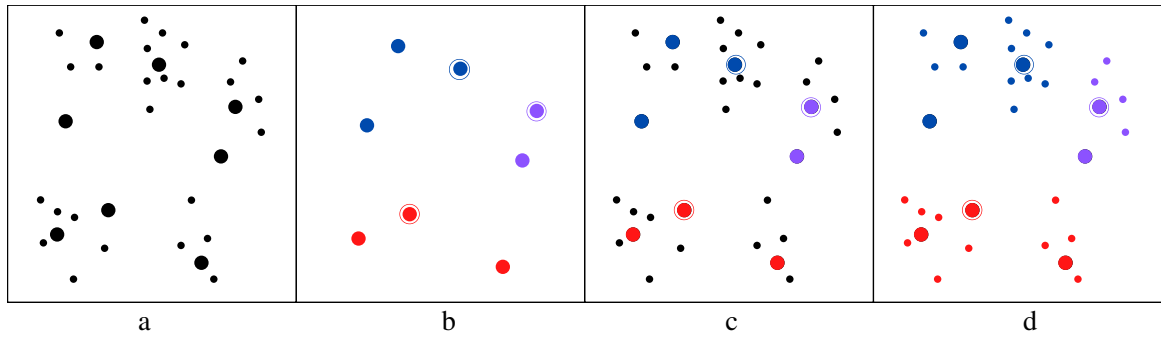


Figure 5.4: Secondary routing

**Note:** The figure shows the working of secondary clustering ( $Cluster_2$ ): a) The big circles represent nodes with residual energy (RE) more than the cluster head threshold ( $CH\_Threshold$ ) implying that they can potentially become cluster heads. The small circles represent nodes with RE less than  $CH\_Threshold$ . b) Clusters are formed only taking into account nodes with RE greater than  $CH\_Threshold$ , and from these, a cluster head (CH) is selected for each cluster. The big circle with an outer line represents the CH. c) The remaining alive nodes try to join the clusters. d) All alive nodes join clusters whose CH they are nearest to.

### 5.2.5 Residual energy threshold

An important hyperparameter that significantly impacts the lifetime of the WSN network is the Residual Energy (RE) threshold. This threshold defines the minimum percentage of residual energy required for a node to maintain its position as a cluster head (CH). If a CH's

RE drops below this threshold relative to its initial energy, it can no longer act as a CH. The optimal RE threshold is established through a comprehensive analysis of various threshold values in combination with other factors and hyperparameters, ultimately identifying the most effective value for network performance.

### 5.2.6 Energy model for data transmission

In the proposed work, the energy model used is similar to the LEACH protocol [32]. The energy required for transmitting and receiving the data packets from source to destination and the energy used for aggregation by the CHs. The energy required for receiving (eq. 5.3), aggregating (eq. 5.4) and transmitting (eq. 5.5) data is defined as:

$$E_{rx}(l) = lE_{elec} \quad (5.3)$$

$$E_{ag}(l) = lE_{da} \quad (5.4)$$

$$E_{tr}(l) = \begin{cases} \cdot E_{elec} + l \cdot \epsilon_{fs} \cdot d^2, & d \leq d_0 \\ \cdot E_{elec} + l \cdot \epsilon_{mp} \cdot d^4, & d > d_0 \end{cases} \quad (5.5)$$

Here, let  $d$  represent the distance between any two nodes, with  $d_0$  signifying the reference distance where communication occurs seamlessly. The energies required for processing  $l$  bits of data are denoted as  $E_{rx}(l)$  for receiving,  $E_{ag}(l)$  for aggregating, and  $E_{tx}(l)$  for transmitting. Communication relies on free space when the distance between devices is below the threshold (characterized by  $d^2$  power loss). However, the multipath fading model is used if the distance exceeds this threshold, resulting in  $d^4$  power loss. The energy consumed by electronic components,  $E_{elec}$ , is shaped by modulation techniques, signal spreading, and digital coding, while amplifier energy consumption varies based on distance, described by  $\epsilon_{fs} \cdot d^2$  for free space or  $\epsilon_{mp} \cdot d^4$  for multipath techniques. The reference distance  $d_0$  is calculated as  $\sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$ , where  $\epsilon_{fs}$  and  $\epsilon_{mp}$  represent the energy metrics for the free space and

multipath models, respectively [69].

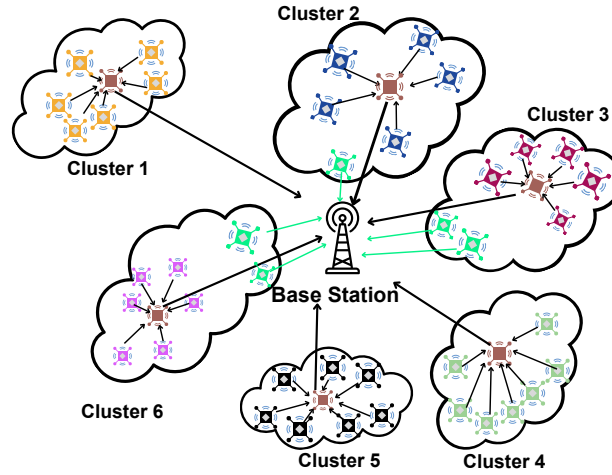


Figure 5.5: Clustering-based routing

**Note:** The proposed protocol follows both direct and cluster head (CH) communication. If the distance from the node to BS is nearer than CH (such nodes are represented using green colour in each cluster), the protocol follows direct communication; In other cases, protocol follows communication through CH.

### 5.3 Evaluation

This section compromises the comprehensive evaluation of the proposed data transmission approach to expand the WSN network's lifespan. The section includes the simulation assessment for both homogeneous (the WSN nodes having the same capability) and heterogeneous (the WSN nodes having different capabilities in terms of initial energy, computation power, communication range, etc.) nodes. The proposed approach is compared with our combination of the methods and the different state-of-the-art methods. The proposed method outperforms simulation results concerning the state-of-the-art methods compared to different parameters. Later, to check the practicability of the proposed method, real-world prototypical deployment is also carried out with the help of a small number of nodes.

### 5.3.1 Simulation Environment and parameters

The proposed approach and the compared state-of-the-art methods are implemented in Python programming language. The Google Colab platform is used to execute the programs. The platform uses Intel(R) Xeon(R) CPU, the Haswell CPU family, 12 GB RAM, and two CPUs with a Frequency of 2.30 GHz. The table 5.1 demonstrated the values of various parameters used in the experiments.

Table 5.1: Simulation Parameters

| Parameters               | Values              |
|--------------------------|---------------------|
| Data packet size         | 4000 bits           |
| $E_{elec}^*$             | 50 nJ/bit           |
| $E_{DA}^\dagger$         | 5 nJ/bit/signal     |
| $\epsilon_{fs}^\ddagger$ | 10pJ/bit/ $m^2$     |
| $\epsilon_{mp}^\S$       | 0.0013pJ/bit/ $m^4$ |
| $d_0$                    | 87m                 |

**Note:**  $E_{elec}$  denotes the energy used for transmitting and receiving 1 bit of data,  $E_{DA}$  denotes the energy used for data aggregation,  $\epsilon_{fs}$  denotes the energy used for calculating the received power in an unobstructed line-of-sight,  $\epsilon_{mp}$  denotes the energy dissipated by the amplifier for multipath, and  $d_0$  is  $\sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$  [69].

#### 5.3.1.1 Hyper-parameters Selection

This section explored all the combinations of hyperparameters using grid search and computed the simulation results. The optimal parameters are chosen for further computation with state-of-the-art methods based on the simulation results. The initial results are computed on the region of  $100 \times 100 m^2$ , with 100 nodes having 0.5 Joule initial energy, randomly placed throughout the region. The base station  $BS$  is situated in the centre of the region (i.e. at (50,50)). The five clustering methods (Kmeans [70], Fuzzy C-Means [71], Agglomerative [72], Gaussian Mixture Model [72], and Self-organising map [73]) are ex-

plored for both initial and secondary clustering. Figure 5.6 shows the values of all the hyperparameters on which the grid search method explored for the results. An extensive Grid

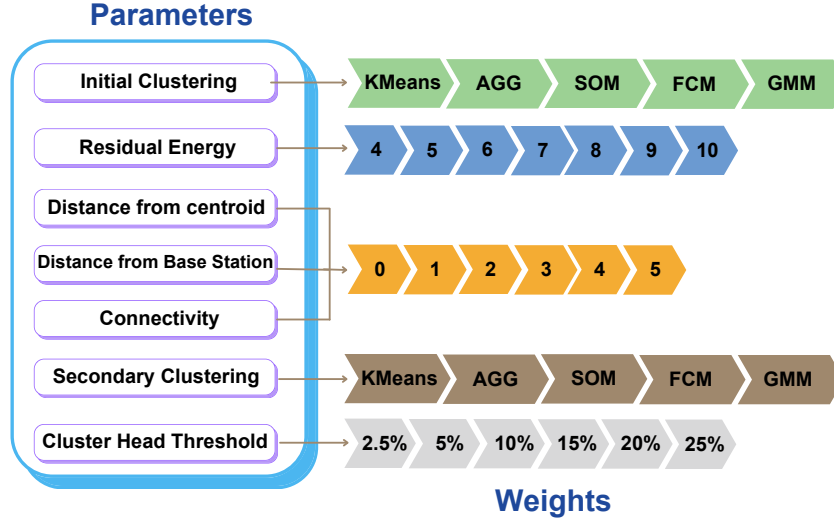


Figure 5.6: Tuning parameters and Weights

**Note:** Tuning values of different parameters are shown in this figure. The grid search method uses all the combinations of these parameters, executes the proposed approach, and evaluates the results.

Search was conducted, systematically exploring every possible combination of clustering algorithms, weights assigned to CH (Cluster Head) selection criteria, and threshold energy percentages. This comprehensive approach aimed to capture the effect of each combination on the network's lifetime. With two stages of clustering and five algorithmic options per stage, 25 unique combinations of clustering methods were evaluated. Table 5.2 presents the network performance for each of these 25 configurations, highlighting the impact of different clustering strategies on the network's longevity. The results are computed based on eight different parameters like: a) The number of transmission rounds at which one-fourth (25%) of the nodes become inactive, b) the number of transmission rounds at which half (50%) of the nodes become inactive, c) the number of transmission rounds at which three-fourth (75%) of the nodes become inactive, d) the number of transmission rounds at all (100%) of the nodes become inactive, e) The number of data packets transmitted throughout the

transmission process, f) distance travelled by the data packets (in meters), g) the number of hops, and f) the number of cluster heads changed. All combination results (initial and secondary clustering) are shown in Table 5.2, and the best result is highlighted in bold. In

Table 5.2: Performances of the clustering algorithm's combinations

|    | protocol          | 25%*        | 50%*        | 75%*        | 100%*       | $DPT^\dagger$ | $DT^\ddagger$ (in m) | Hops          |
|----|-------------------|-------------|-------------|-------------|-------------|---------------|----------------------|---------------|
| 1  | KMeans_KMeans     | 988         | 1187        | 1306        | 3076        | 81946         | 2023506              | 112221        |
| 2  | FCM_FCM           | 987         | 1139        | 1194        | 3079        | 84632         | 1779372              | 105271        |
| 3  | GMM_GMM           | 988         | 1160        | 1280        | 3085        | 84780         | 2086696              | 109583        |
| 4  | AGG_AGG           | 1041        | 1189        | 1472        | 3061        | 84407         | 2667696              | 103896        |
| 5  | SOM_SOM           | 988         | 1187        | 1306        | 3076        | 81946         | 2023506              | 112221        |
| 6  | KMeans_FCM        | 1078        | 1221        | 1345        | 3086        | 78410         | 2235971              | 112983        |
| 7  | FCM_KMeans        | 977         | 1115        | 1203        | 3083        | 84919         | 1799206              | 106899        |
| 8  | KMeans_AGG        | 1020        | 1233        | 1381        | 3079        | 81877         | 2312429              | 114101        |
| 9  | AGG_KMeans        | 930         | 1101        | 1447        | 3059        | 85621         | 2208149              | 103525        |
| 10 | FCM_AGG           | 958         | 1088        | 1338        | 3077        | 88676         | 1861678              | 109964        |
| 11 | AGG_FCM           | 1003        | 1204        | 1392        | 3072        | 83932         | 2388230              | 101406        |
| 12 | KMeans_GMM        | 976         | 1085        | 1247        | 3083        | 84698         | 1923743              | 105117        |
| 13 | GMM_KMeans        | 1011        | 1144        | 1249        | 3069        | 84354         | 1888125              | 107273        |
| 14 | <b>KMeans_SOM</b> | <b>1085</b> | <b>1243</b> | <b>1322</b> | <b>3088</b> | <b>79116</b>  | <b>2254810</b>       | <b>114788</b> |
| 15 | SOM_KMeans        | 740         | 1122        | 1565        | 3079        | 89366         | 1865606              | 101278        |
| 16 | FCM_SOM           | 1015        | 1112        | 1239        | 3088        | 86278         | 1910295              | 107738        |
| 17 | SOM_FCM           | 828         | 1055        | 1289        | 3081        | 86999         | 1987222              | 101120        |
| 18 | FCM_GMM           | 482         | 705         | 990         | 2774        | 79622         | 1989558              | 103453        |
| 19 | GMM_FCM           | 935         | 1419        | 1508        | 2756        | 82668         | 2029349              | 104524        |
| 20 | SOM_GMM           | 1196        | 1760        | 2257        | 3050        | 112742        | 1559751              | 105721        |
| 21 | GMM_AGG           | 1369        | 1970        | 2576        | 2980        | 142980        | 1685813              | 107837        |
| 22 | SOM_AGG           | 1079        | 1590        | 2013        | 3010        | 158030        | 1834176              | 104461        |
| 23 | GMM_SOM           | 963         | 1402        | 1775        | 3013        | 158480        | 1489400              | 113950        |
| 24 | AGG_GMM           | 1301        | 1742        | 2151        | 3030        | 95791         | 1383659              | 103991        |
| 25 | AGG_GMM           | 957         | 1520        | 2000        | 3035        | 124402        | 1430227              | 101305        |

**Note:** 25%\*, 50%\*, 75%\* and 100%\* denotes the number of rounds completed until the death of 25, 50, 75 and 100 percent of nodes.  $DPT^\dagger$  denotes the number of data packet travelled,  $DT^\ddagger$  denotes distance travelled by all packets during communication (in meters).

total, the proposed algorithm was executed 226,800 times in the test simulation to determine and finalise the following optimal hyper-parameter values and clustering options for further

experiments:

- **K-means** is selected as the best option for initial clustering.
- **Self-Organizing Map (SOM)** is the most effective for secondary clustering.
- The best weight for Residual Energy in the Cluster Head (CH) selection is **8**.
- The best weight for distance from the centroid is **1**.
- The best weight for distance from the Base Station (BS) is **1**.
- The best weight for node connectivity is **2**.
- The optimal threshold percentage for energy is **5%**.

These parameters are also tested across different region sizes and found that the optimal values remained almost the same as for a 100x100 m<sup>2</sup> area. Variations in parameter weights for a few sizes led to minimal changes in performance, with less than a 1.5% difference in the number of rounds completed (when all the nodes become inactive). Therefore, it's clear that investing resources into finding new optimal weights for different region sizes is largely unnecessary.

### 5.3.1.2 Impact of Varying Nodes

The results are computed with varying the number of nodes (75, 125, 175 and 225) for six area sizes (50×50, 100×100, 150×150, 200×200, 250×250, 300×300 m<sup>2</sup>) for verifying the efficacy of the proposed method in various node deployment and area sizes. The results are computed for both homogeneous and heterogeneous (with 10% of nodes with 5% extra energy) nodes. The weights of the hyperparameters are taken from the subsection [5.3.1.1](#).

The following conclusions are drawn from the above experiments:

- The number of rounds that nodes remain alive increases as the number of deployed nodes increases, mainly due to the formation of more efficient clusters as the number of nodes increases.
- The number of rounds decreases as the network size increases, as larger areas lead to sparser node deployment.
- As more nodes are deployed, the number of data packets transmitted, distance travelled, and hop count all increase.
- Generally, these metrics also rise with increasing network size. However, in some cases, unexpected decreases occur due to the randomness of node deployment, which are considered anomalies.

### **5.3.2 Comparison with State-of-the-art methods**

The proposed method is compared with two well-known homogeneous cluster head selection methods, LEACH [32] and LEACH-C [33], and three well-known heterogeneous methods, SEP [34], EAMMH [35], and Z-SEP [36]. The proposed approach to extending the lifetime of a WSN network is evaluated and compared with existing methods based on the following five key factors: a) Number of transmission rounds before a percentage of nodes die: A higher value indicates a more extended network lifetime and is desirable, b) Average distance per round (in meters) that data packets travel to reach the base station (BS): A shorter distance is preferable because it reflects a more compact and efficient network, c) Number of data packets transmitted: A higher number signifies greater network efficiency, which is desirable in networking, d) Frequency of clustering and Cluster Head (CH) changes: Fewer changes in cluster heads make the network more stable, e) Number of hops for data to reach the base station: As the distance travelled, fewer hops are better,

indicating efficient data routing. The later subsections show a detailed comparison based on each of the parameters.

### 5.3.2.1 Number of transmission rounds

This criterion measures the number of transmission rounds until a certain percentage of sensor nodes exhaust their energy. A higher value is desirable, as it represents a higher lifespan of the sensor nodes. The number of rounds is calculated for four cases: when 25%, 50%, 75%, and 100% of nodes die. Figures 5.7 and 5.8 illustrate the percentage of node deaths for homogeneous and heterogeneous approaches, respectively. The network's performance is ultimately evaluated based on the number of rounds after which all (100%) nodes die or the network stops functioning. Figure 5.7 clearly shows that the proposed

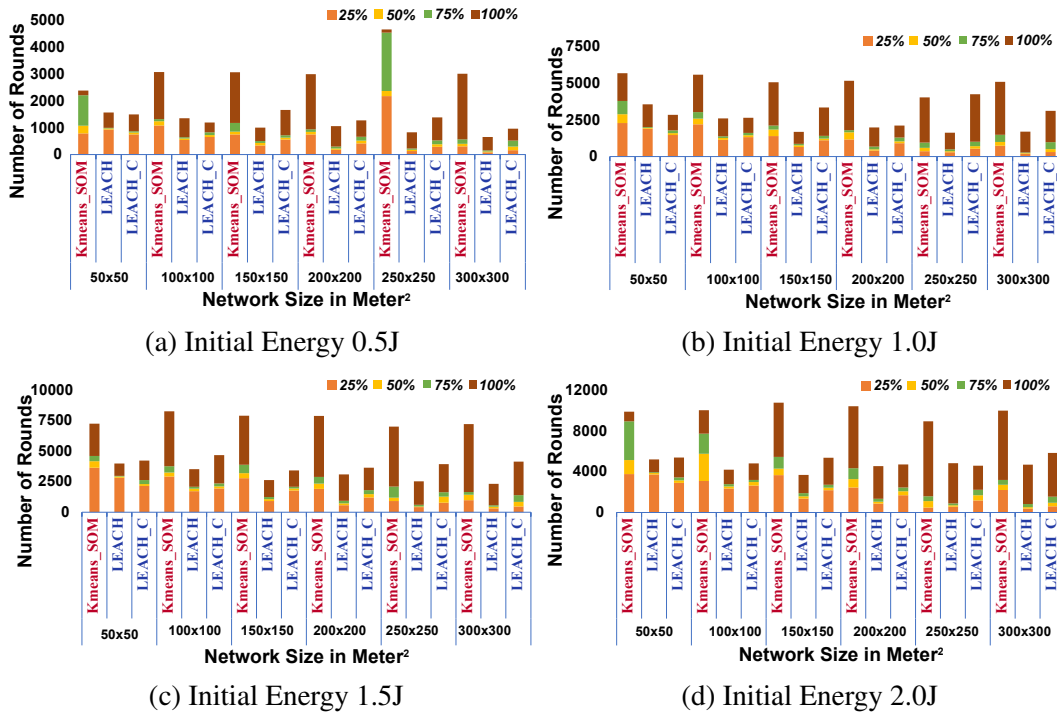


Figure 5.7: Homogeneous Protocols: Number of Rounds at percent of nodes died

method outperforms the two well-known homogeneous cluster head selection methods for initial energy levels 0.5, 1.0, 1.5, and 2.0 J for the area sizes from  $50 \times 50$  to  $300 \times 300$

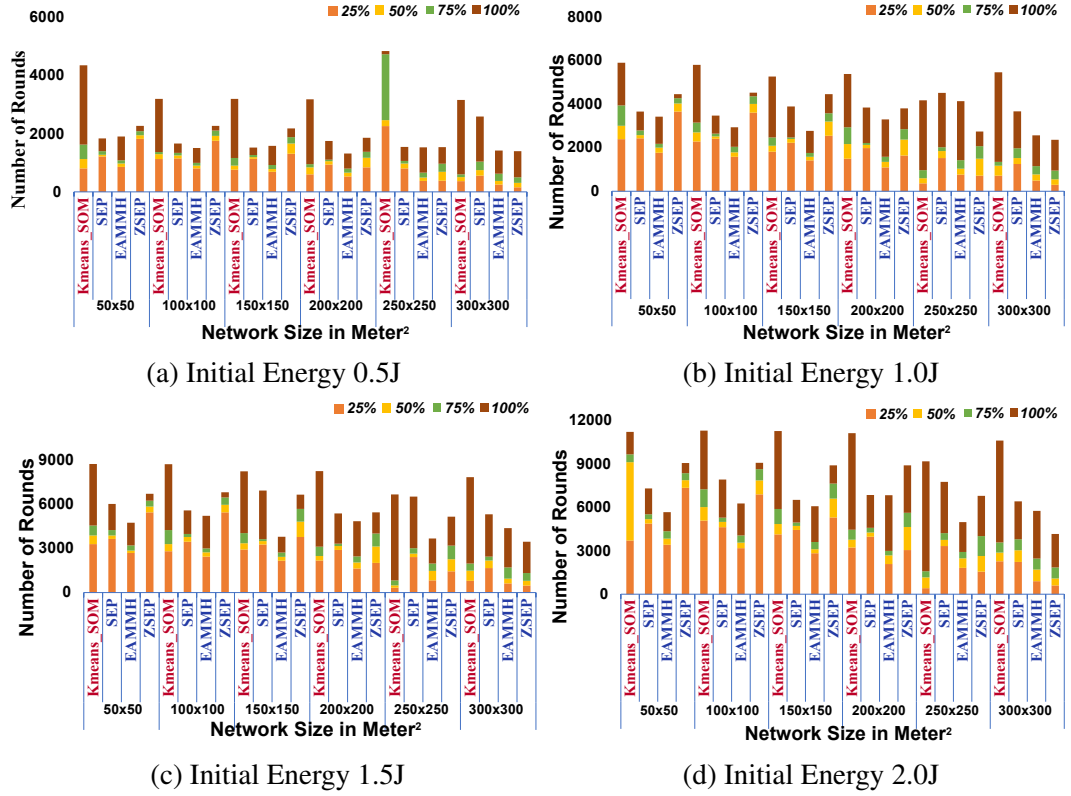


Figure 5.8: Heterogeneous Protocols: Number of Rounds at percent of nodes died

$m^2$ . Similarly, Figure 5.8 represents that, the proposed method surpasses the heterogeneous methods like SEP, EAMMH, and Z-SEP with an significant margin.

### 5.3.2.2 Average distance per Round

This criterion assesses the total distance (in meters) travelled by all the data packets. The total distance is then divided by the number of rounds for each protocol and calculated using Equation. 5.6.

$$= \frac{\sum_{i=1}^n (Dist(node, CH) + Dist(CH, BS))}{n} \quad (5.6)$$

Here,  $n$  shows the number of transmission rounds,  $Dist(node, CH)$  is the distance (in meters) between the node and the Cluster Head, and  $Dist(CH, BS)$  is the distance (in meters) between the Cluster Head and the Base Station. A smaller average distance is preferable, as

it reflects the efficiency of the approach. The proposed method optimises transmission by considering both direct communication and routing through the Cluster Head (CH), minimizing unnecessary transmissions and extending the lifespan of the network. Figures 5.9 and 5.10 illustrate the average distance (in meters) travelled by data packets per round for homogeneous and heterogeneous protocols, respectively. Figures 5.9(a)-(d) and 5.10(a)-(d)

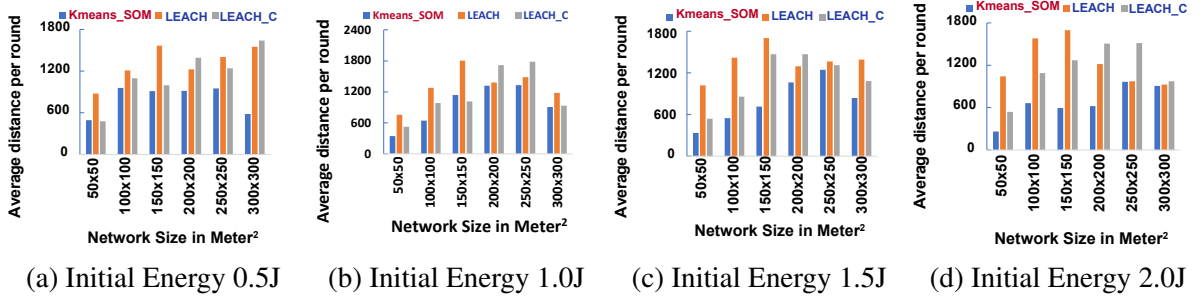


Figure 5.9: Homogeneous Protocols: Average Distance per Round

present the results for homogeneous and heterogeneous protocols with initial sensor node energies of 0.5 J, 1.0 J, 1.5 J, and 2.0 J, respectively. The results clearly demonstrate that the proposed approach (KMeans\_SOM) outperforms existing methods to reduce the average distance travelled by packets per round.

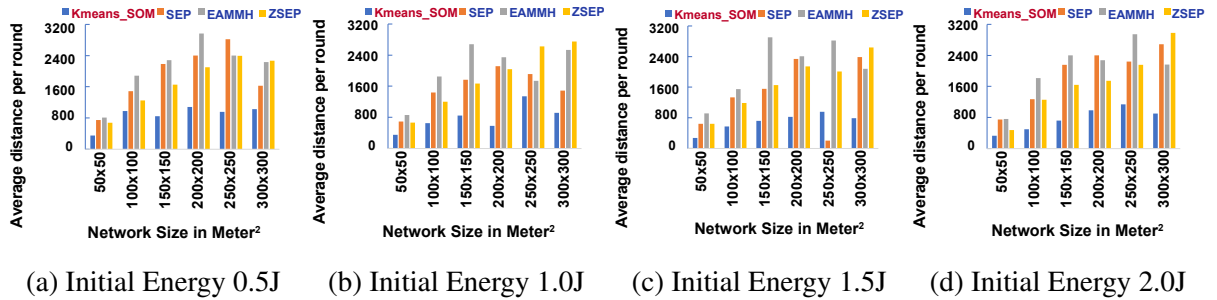


Figure 5.10: Heterogeneous Protocols: Average Distance per Round

### 5.3.2.3 number of data packets transmitted

This criterion evaluates the number of data packets transmitted throughout the transmission process. Each approach transmits data packets of 4000 bits. The number of transmitted

packets reflects the network's sensing capability—networks transmitting more packets will likely capture more information about the phenomenon. The total number of packets transmitted is calculated using Equation 5.7.

$$= \sum_{i=1}^n (PT_i(node, CH) || PT_i(node, BS)) \quad (5.7)$$

Here,  $PT(node, CH)$  demonstrates the number of data packets transmitted from a node to the respective cluster head, and  $PT(node, BS)$  exhibits the number of data packets transmitted from the node to the base station. Figures 5.11 and 5.12 show the count of data packets transmitted for homogeneous and heterogeneous protocols, respectively. Figures 5.11 (a)-

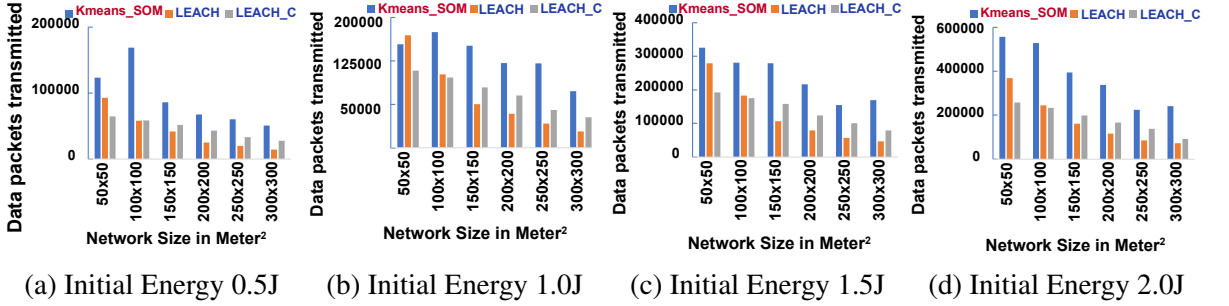


Figure 5.11: Homogeneous Protocols: Number of Data Packets sent

(d) present results for homogeneous protocols, while Figures 5.12 (a)-(d) show results for heterogeneous protocols, with initial sensor node energies of 0.5 J, 1.0 J, 1.5 J, and 2.0 J, respectively. The results demonstrate that the proposed KMeans\_SOM protocol outperforms existing methods like LEACH and LEACH-C regarding the number of data packets transmitted to the base station.

The proposed KMeans\_SOM approach performs well compared to SEP and EAMMH but falls slightly short in a few cases compared to Z-SEP. It is likely because Z-SEP's node distribution is not random—around 90% of nodes are concentrated near the base station, and transmission energy is proportional to the distance from the base station. These factors likely enable Z-SEP to transmit more packets in specific scenarios.

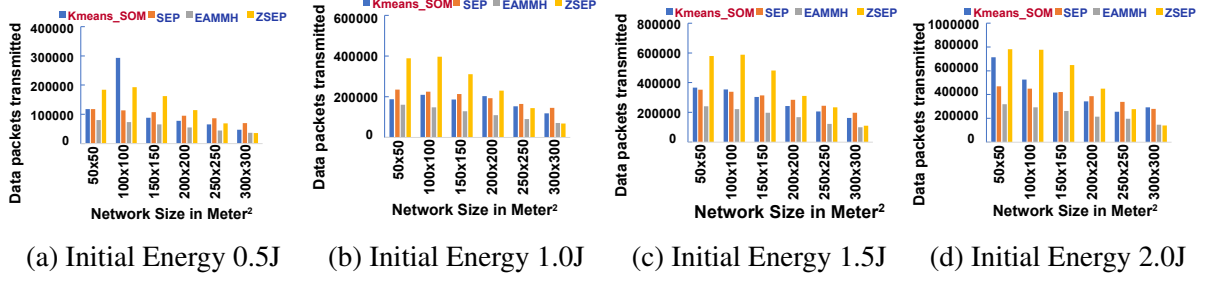


Figure 5.12: Heterogeneous Protocols: Number of Data Packets sent

### 5.3.2.4 Frequency of cluster formation

The frequency of clustering during transmission is an important factor. Frequent clustering is undesirable in real-world scenarios due to the high energy and computational costs involved in performing them. Figures 5.13 and 5.14 display the number of clusters formed for homogeneous and heterogeneous protocols, respectively. Homogeneous protocols (such as

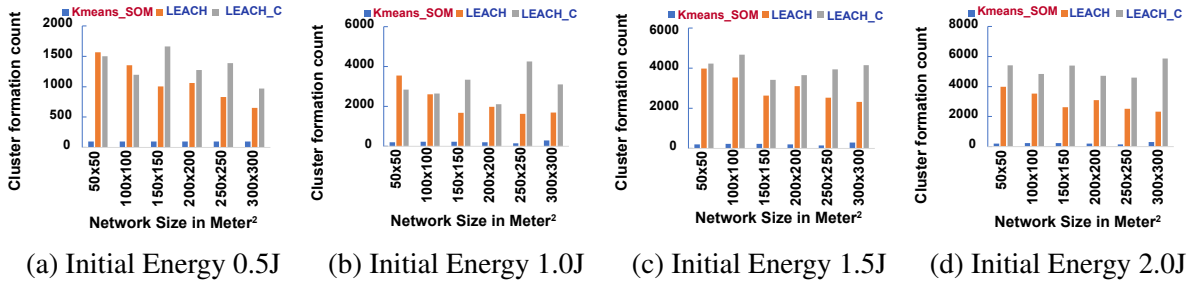
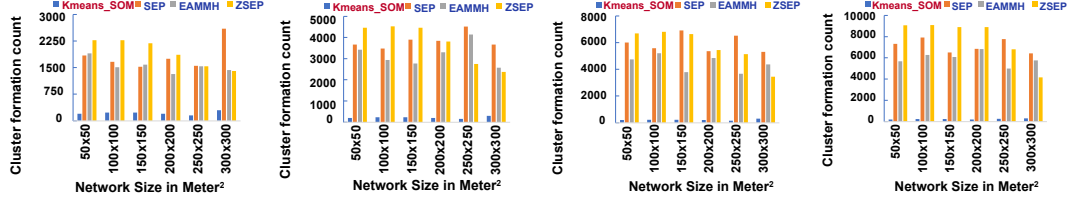


Figure 5.13: Homogeneous Protocols: Number of times clustering process initiated

LEACH and LEACH-C) and heterogeneous protocols (like SEP, EAMMH, and Z-SEP) require new clusters to form after every transmission round, which is impractical in real-world applications. These protocols select new Cluster Heads (CHs) based on certain parameters, and nodes join the nearest CH based on Euclidean Distance. In difference, the proposed approach only performs clustering when the residual energy of all nodes in a cluster falls below a specified threshold (RE threshold).



(a) Initial Energy 0.5J (b) Initial Energy 1.0J (c) Initial Energy 1.5J (d) Initial Energy 2.0J

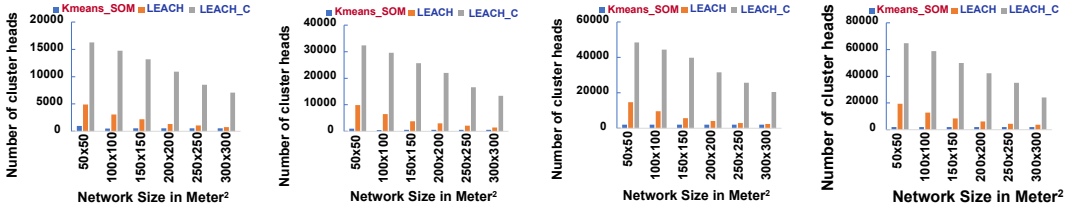
Figure 5.14: Heterogeneous Protocols: Number of times clustering process initiated

### 5.3.2.5 Frequency of change of cluster heads

Most of the cluster-based approaches route data through the cluster heads to the base station and often require changing the cluster heads after each transmission round. In contrast, the proposed approach changes the cluster heads only when the residual energy of the current cluster head falls below a threshold. Equation 5.8 outlines the parameter used to calculate the number of cluster heads.

$$= \sum_{i=1}^n (CH_i \setminus CH_{i-1}) \quad (5.8)$$

Where,  $CH_i$  and  $CH_{i-1}$  represent the set of cluster heads in the current and previous rounds, respectively. The frequency of CH changes is calculated as the sum of the differences in the current and previous rounds. Requiring a change in cluster heads after every transmission round, as many existing approaches do, is impractical and energy-intensive. Figures 5.15 and 5.16 illustrate the number of CH changes for homogeneous and heterogeneous protocols, respectively. In Figure 5.15 (a)-(d), we compare the number of cluster head changes



(a) Initial Energy 0.5J (b) Initial Energy 1.0J (c) Initial Energy 1.5J (d) Initial Energy 2.0J

Figure 5.15: Homogeneous Protocols: Number of Cluster Heads Changed

for homogeneous protocols throughout the transmission process with initial energy values

of 0.5 J, 1.0 J, 1.5 J, and 2.0 J. While LEACH and LEACH-C change cluster heads after every transmission round, the proposed KMEANS\_SOM approach only changes them based on residual energy. This results in a significant difference between KMEANS\_SOM and the existing protocols. Figures 5.16 (a)-(d) compare the proposed approach with existing ones

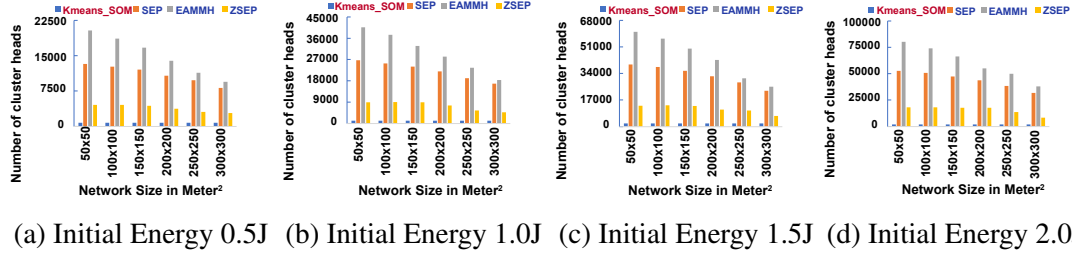


Figure 5.16: Heterogeneous Protocols: Number of Cluster Heads Changed

regarding cluster head changes in heterogeneous networks with initial node energies of 0.5 J, 1.0 J, 1.5 J, and 2.0 J. While protocols like SEP, EAMMH, and Z-SEP change cluster heads after every transmission round, the proposed KMEANS\_SOM changes cluster heads based on residual energy, not after each round.

### 5.3.2.6 Number of Hops

The number of hops is another essential parameter for the comparison of the proposed approach with the existing approaches. The energy used to transmit packets generally increases with the distance covered. The distance covered by data packets may or may not be in exact conformance with hop counts, but hop counts do give a fair idea of the same and, consequently, energy expended. For example, consider two nodes: one near the base station and one farther away. If both nodes transmit data over two hops, the distance covered in each hop will differ. Although hop counts alone may not fully reflect the distance in complex scenarios, they still offer valuable insights in simpler networks. The number of hops can be calculated as:

$$= \sum_{i=1}^n \sum_{j=1}^k R_i \left( node_j \begin{cases} count = 1, & \text{if direct communication} \\ count = c + 1, & \text{communication via CH} \end{cases} \right) \quad (5.9)$$

Here,  $R_i$  represents the  $i^{th}$  round. In each round  $i$ , the number of hops is calculated for all  $k$  nodes that are still alive. The parameter counts one hop for nodes that communicate directly and  $c + 1$  hops (where  $c$  is the number of intermediate CHs) for nodes that communicate through the Cluster Head (CH).

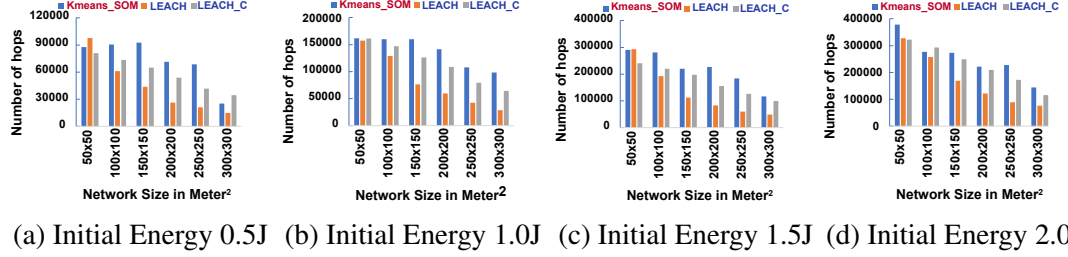


Figure 5.17: Homogeneous Protocols: Number of Hops

Figures 5.17 (a)-(d) show the number of hops by homogeneous protocols during the complete transmission process for an initial energy of 0.5 J, 1.0 J, 1.5 J, and 2.0 J, respectively.

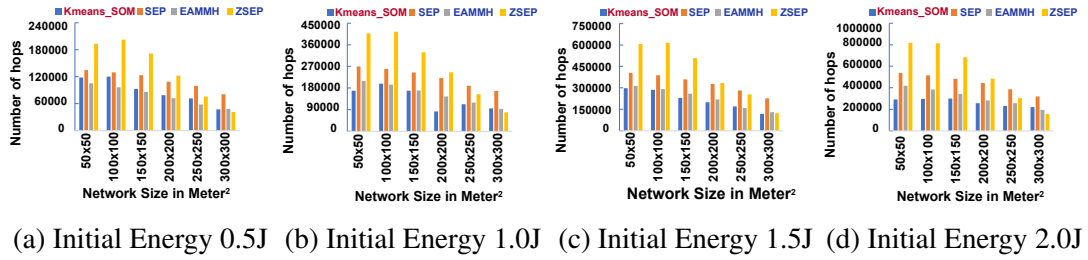


Figure 5.18: Heterogeneous Protocols: Number of Hops

Figures 5.18 (a)-(d) show the number of hops by heterogeneous protocols during the complete transmission process for an initial energy of 0.5 J, 1.0 J, 1.5 J, and 2.0 J, respectively.

### 5.3.2.7 Performance Gain

This section evaluates the performance gain of the proposed method compared to existing methods based on above mentioned parameters for both homogeneous and heterogeneous

methods. The performance gain obtained from different sizes of the region of interest is normalised and summarised in Table 5.3. The gain of the proposed approach is normalised to one and compared with the existing methods as: Table 5.3 compares the performance

Table 5.3: Performance Gain

| Parameter          | Homogeneous |         |                              | Heterogeneous |        |        |                              |
|--------------------|-------------|---------|------------------------------|---------------|--------|--------|------------------------------|
|                    | LEACH       | LEACH-C | <i>Proposed</i> <sup>*</sup> | SEP           | EAMMH  | Z-SEP  | <i>Proposed</i> <sup>*</sup> |
| $R^{\S}$           | 0.3313      | 0.4091  | 1                            | 0.5388        | 0.4577 | 0.5691 | 1                            |
| $ADpR^{\dagger}$   | 0.3951      | 0.5717  | 1                            | 0.9839        | 0.5338 | 0.8403 | 1                            |
| $DPT^{\dagger}$    | 0.3584      | 0.3901  | 1                            | 0.8490        | 0.7988 | 0.7988 | 1                            |
| $Hops^{\P}$        | 0.3932      | 0.6783  | 1                            | 0.9520        | 0.5909 | 0.7718 | 1                            |
| $FoCF^{\dagger}$   | 0.1038      | 0.0147  | 1                            | 0.0106        | 0.0084 | 0.0087 | 1                            |
| $FoCCH^{\ddagger}$ | 0.0107      | 0.0002  | 1                            | 0.0067        | 0.0079 | 0.0063 | 1                            |

**Note:**  $R^{\S}$  denotes the number of rounds completed until all the nodes die;  $ADpR^{\dagger}$  denotes the average distance covered per round;  $DPT^{\dagger}$  denotes the number of data packets transmitted;  $FoCF^*$  denotes the frequency of cluster formation;  $FoCCH^{\ddagger}$  denotes the frequency of change of cluster heads; and  $Hops^{\P}$  denotes the number of hops.

of algorithms across different factors. For  $R^{\S}$ ,  $ADpR^{\dagger}$ ,  $DPT^{\dagger}$ , and  $Hops^{\P}$ , higher values indicate better performance. Conversely, for  $FoCF^*$  and  $FoCCH^{\ddagger}$ , lower values are preferable (with performance measured as the reciprocal of the actual values). The normalization process has been structured to set the proposed method's parameter value to 1, with corresponding values for other methods adjusted proportionally.

## 5.4 Limitations

Routing through cluster heads is currently the most effective approach, as packets can be directed along the best paths via these heads. However, after several transmission rounds, the energy of cluster heads decreases, and they can no longer serve in this role. This necessitates a new cluster head selection, which is time-consuming and energy-intensive. Without

software-defined networking (SDN) for updates, manual node updates become an unmanageable task. Additionally, the grid-search method used for optimisation is thorough and time-consuming but ultimately identifies the most optimal values.

## **5.5 Conclusion and Future Work**

This chapter introduces a novel way to save the wireless sensor nodes (WSNs) energy using routing via cluster heads. The proposed technique works effectively with homogeneous and heterogeneous nodes, improving the lifespan of the network. It involves forming clusters and routing data through the cluster head. An exhaustive search was conducted to select the best clustering algorithms and hyper-parameters for choosing cluster heads. The approach was validated to be more effective than existing methods for extending network life and was successfully tested through a real-world prototype implementation. In the future, machine learning will replace grid-search for hyper-parameter tuning.



## Chapter 6

# Conclusions and Future Works

Wireless sensor network nodes are mostly run with batteries which are finite sources of energy; thus preserving energy is the primary requirement of such networks. This is especially true when the network is deployed in an outdoor and inaccessible terrain where battery replenishment is challenging.

In this thesis, we explore the exercise of localisation and energy-efficient routing in WSN which are widely believed to be the most energy-intensive tasks. Localisation comprises establishing the location of unknown nodes “*UN*” using a minimal number of beacon nodes “*BN*” (whose locations are known in advance). Localisation is performed using two approaches both of which are easily practicable in the real world: the first performs localisation utilising the RSSI values (for distance measurements) and the AoA (for angle measurements) between the *BN* and *UN*. The method is effective in finding the location of UNs in an iterative manner and works well for large outdoor locations. The method was compared with a large number of settings and also with other methods, and was found to outperform state-of-the-art methods on every comparison parameter. The primary goal was to localise the network efficiently using a small number of beacon nodes, and our proposed method was able to achieve this.

Similarly, the other localisation approach comprises the use of a machine learning-based algorithm, Random Forest (*RF*), in combination with a multi-iteration localisation approach. The RSSI dataset to support this endeavor was first created within our Institute premises and was used to train the model. Not only is the proposed approach easy to implement and flexible, it also outperforms state-of-the-art methods and is able to localise the nodes more efficiently. The effectiveness of the proposed work was also tested in a real-world environment thus validating its efficacy. The primary motive of this approach is to perform the localisation in large outdoor locations where energy becomes the primary constraint.

Routing is the process of transmitting data from source to destination through an optimal path. There are quite a few methods for routing, but clustering-based routing is widely considered the most efficient of these. In this thesis, a dynamic cluster head-based routing approach was proposed for energy-efficient data transfer between nodes to the base station via the cluster heads. The method follows two types of cluster formation wherein the primary clusters are formed based on the location of the nodes, and secondary clustering is done based on both the location and the remaining energy of the nodes. After forming clusters, the cluster heads are selected using parameters like energy remaining, distances, and connectivity within the network. The weights of these parameters and the clustering algorithm are selected through an exhaustive grid search over various potential weights and algorithms. The main objective of the algorithm is to form a routing algorithm that can increase the lifespan of the network. The proposed method was found to successfully outperform state-of-the-art methods in simulation as well as in real-world prototypical deployments across settings.

An innovative set-up that we put together in our localisation efforts comprises a stepper motor bound with a laser for Angle of Arrival (AoA) calculations. The set-up was quite effective in 2-dimensional environments that did not have physical interferences.

Some future directions of the research that we contemplate comprise machine learning-based localisation; due to the high cost and variable size of the regions, there are a few datasets available for outdoor localisation. In the future, some publicly available datasets will be required so the machine learning models can be better trained on them, increasing the prediction accuracy and making it more flexible to work in extensive regions. The new datasets can be made using the existing WiFi infrastructure and can be easily collected in any publicly available place. The localisation accuracy can be further improved by collecting the temporal dataset, which helps in analysing the signal behaviour in different times. New range measurement algorithms can be incorporated for localisation and routing algorithms because previous methods consume comparatively more energy. In the proposed cluster-based routing approach, optimal weights are selected with the help of an exhaustive search method (i.e., Grid search), which is an extensive search method that searches for the optimal values after running the algorithm for every combination. If the nodes themselves do the whole process, then it consumes lots of energy. Instead of the Grid search method, in future, another heuristic method (like Simulated Annealing (SA) or Tabu Search (TS)) can be used, which can save the searching time and consume fewer resources for optimal parameter selection. The cluster formation process can also be improved by using other parameters than distance measures (like distance to base station and centroid), which change rapidly when we try to perform it with nodes that change their positions dynamically.

There are a few applications in which our proposed methods of localisation and routing can be used. The localisation can be used for localising the fire or flood in large outdoor environments like a forest, and it can also be used for monitoring the movements of troops on the battlefield. The routing can be widely used in mobile ad-hoc networks.



# Bibliography

- [1] S. Tomic, M. Beko, and M. Tuba, “A linear estimator for network localization using integrated rss and aoa measurements,” *IEEE Signal Processing Letters*, vol. 26, no. 3, pp. 405–409, 2019.
- [2] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, “A novel approach to indoor rssi localization by automatic calibration of the wireless propagation model,” in *VTC Spring 2009-IEEE 69th Vehicular Technology Conference*. IEEE, 2009, pp. 1–5.
- [3] D. K. Goldenberg, A. Krishnamurthy, W. C. Maness, Y. R. Yang, A. Young, A. S. Morse, and A. Savvides, “Network localization in partially localizable networks,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1. Miami, FL, USA: IEEE, 2005, pp. 313–326.
- [4] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in *Proceedings of the 7th annual international conference on Mobile computing and networking*. Rome, Italy: ACM New York, NY, USA, 2001, pp. 166–179.
- [5] S. Hara and D. Anzai, “Experimental performance comparison of rssi-and tdoa-based location estimation methods,” in *VTC Spring 2008-IEEE Vehicular Technology Conference*. IEEE, 2008, pp. 2651–2655.

- [6] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. Philadelphia, Pennsylvania, USA: ACM New York, NY, USA, 2004, pp. 45–57.
- [7] Z. Yang, Y. Liu, and X.-Y. Li, "Beyond trilateration: On the localizability of wireless ad-hoc networks," in *IEEE INFOCOM 2009*. Rio de Janeiro, Brazil: IEEE, 2009, pp. 2392–2400.
- [8] X. Liu, J. Yin, S. Zhang, B. Ding, S. Guo, and K. Wang, "Range-based localization for sparse 3-d sensor networks," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 753–764, 2018.
- [9] Z. Yang and Y. Liu, "Understanding node localizability of wireless ad hoc and sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1249–1260, 2011.
- [10] Z. Yang, L. Jian, C. Wu, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, pp. 1–20, 2013.
- [11] G. Bhatti, Y. Javed, M. Naveed, and S. Asif, "Out-door localization in large-scale wireless sensor networks by using virtual nodes," *2020 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–7, 2020.
- [12] J. Luomala and I. Hakala, "Adaptive range-based localization algorithm based on trilateration and reference node selection for outdoor wireless sensor networks," *Computer Networks*, vol. 210, p. 108865, 2022.

- [13] J.-R. Jiang, C.-M. Lin, F.-Y. Lin, and S.-T. Huang, "Aldr: Aoa localization with rssi differences of directional antennas for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, p. 529489, 2013.
- [14] A. T. Le, L. C. Tran, X. Huang, C. Ritz, E. Dutkiewicz, S. L. Phung, A. Bouzerdoun, and D. Franklin, "Unbalanced hybrid aoa/rssi localization for simplified wireless sensor networks," *Sensors*, vol. 20, no. 14, p. 3838, 2020.
- [15] D. Zou, S. Chen, S. Han, W. Meng, D. An, J. Li, and W. Zhao, "Design of a practical wsn based fingerprint localization system," *Mobile Networks and Applications*, vol. 25, pp. 806–818, 2020.
- [16] N. Saeed, H. Nam, T. Y. Al-Naffouri, and M.-S. Alouini, "A state-of-the-art survey on multidimensional scaling-based localization techniques," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3565–3583, 2019.
- [17] M. Singh, S. K. Bhoi, and S. K. Panda, "Geometric least square curve fitting method for localization of wireless sensor network," *Ad Hoc Networks*, vol. 116, p. 102456, 2021.
- [18] S. Barai, D. Biswas, and B. Sau, "Sensors positioning for reliable rssi-based outdoor localization using cft," *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pp. 1–5, 2020.
- [19] M. S. Mozamir, R. B. A. Bakar, W. I. S. W. Din, and Z. Musa, "Gbln-pso algorithm for indoor localization in wireless sensor network," *IOP Conference Series: Materials Science and Engineering*, vol. 769, no. 1, p. 012033, 2020.

- [20] S. J. Bhat and S. K. Venkata, "An optimization based localization with area minimization for heterogeneous wireless sensor networks in anisotropic fields," *Computer Networks*, vol. 179, p. 107371, 2020.
- [21] Shilpi and A. Kumar, "A localization algorithm using reliable anchor pair selection and jaya algorithm for wireless sensor networks," *Telecommunication Systems*, vol. 82, no. 2, pp. 277–289, 2023.
- [22] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on computer communications. Nineteenth annual joint conference of the IEEE computer and communications societies (Cat. No. 00CH37064)*, vol. 2. Ieee, 2000, pp. 775–784.
- [23] X. Nguyen, M. I. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 1, pp. 134–152, 2005.
- [24] K. Shi, Z. Ma, R. Zhang, W. Hu, and H. Chen, "Support vector regression based indoor location in ieee 802.11 environments," *Mobile Information Systems*, vol. 2015, no. 1, p. 295652, 2015.
- [25] A. Payal, C. Rai, and B. Reddy, "Artificial neural networks for developing localization framework in wireless sensor networks," in *2014 international conference on data mining and intelligent computing (ICDMIC)*. IEEE, 2014, pp. 1–6.
- [26] S. Yun, J. Lee, W. Chung, E. Kim, and S. Kim, "A soft computing approach to localization in wireless sensor networks," *Expert Systems with Applications*, vol. 36, no. 4, pp. 7552–7561, 2009.

- [27] W. Kim, J. Park, J. Yoo, H. J. Kim, and C. G. Park, "Target localization using ensemble support vector regression in wireless sensor networks," *IEEE transactions on cybernetics*, vol. 43, no. 4, pp. 1189–1198, 2012.
- [28] E. Cayirci, H. Tezcan, Y. Dogan, and V. Coskun, "Wireless sensor networks for underwater surveillance systems," *Ad hoc networks*, vol. 4, no. 4, pp. 431–446, 2006.
- [29] H. Q. Tran and C. Ha, "Improved visible light-based indoor positioning system using machine learning classification and regression," *Applied Sciences*, vol. 9, no. 6, p. 1048, 2019.
- [30] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, H. K. Qureshi, and M. Gidlund, "Rssi fingerprinting-based localization using machine learning in lora networks," *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 53–59, 2020.
- [31] N. Xu, S. Li, C. S. Charollais, A. Burg, and A. Schumacher, "Machine learning based outdoor localization using the rssi of multibeam antennas," in *2020 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2020, pp. 1–5.
- [32] S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon leach protocol for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 623–645, 2013.
- [33] S. D. Muruganathan, D. C. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. S8–13, 2005.
- [34] G. Smaragdakis, I. Matta, and A. Bestavros, "Sep: A stable election protocol for clustered heterogeneous wireless sensor networks," Boston University Computer Science Department, Tech. Rep., 2004.

- [35] H. Chaudhary and V. Kumar, "Performance analysis of leach-c and eammh protocols in wsn in matlab," in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*. IEEE, 2021, pp. 1367–1370.
- [36] S. Faisal, N. Javaid, A. Javaid, M. A. Khan, S. H. Bouk, and Z. A. Khan, "Z-sep: Zonal-stable election protocol for wireless sensor networks," *arXiv preprint*, pp. 1–9, 2013.
- [37] A. Pathak, "A proficient bee colony-clustering protocol to prolong lifetime of wireless sensor networks," *Journal of Computer Networks and Communications*, vol. 1, no. 1, pp. 1–9, 2020.
- [38] R. Thiagarajan and Moorthi, "Energy consumption and network connectivity based on novel-leach-pos protocol networks," *Computer Communications*, vol. 149, no. 1, pp. 90–98, 2020.
- [39] S. A. Sert, A. Alchihabi, and A. Yazici, "A two-tier distributed fuzzy logic based protocol for efficient data aggregation in multihop wireless sensor networks," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3615–3629, 2018.
- [40] M. Abdolkarimi, S. Adabi, and A. Sharifi, "A new multi-objective distributed fuzzy clustering algorithm for wireless sensor networks with mobile gateways," *AEU-International Journal of Electronics and Communications*, vol. 89, no. 1, pp. 92–104, 2018.
- [41] S. A. Sert and A. Yazici, "Increasing energy efficiency of rule-based fuzzy clustering algorithms using clonalg-m for wireless sensor networks," *Applied Soft Computing*, vol. 109, no. 1, p. 107510, 2021.

- [42] L. Nagarajan and S. Thangavelu, “Hybrid grey wolf sunflower optimisation algorithm for energy-efficient cluster head selection in wireless sensor networks for lifetime enhancement,” *Iet Communications*, vol. 15, no. 3, pp. 384–396, 2021.
- [43] S. E. Pour and R. Javidan, “A new energy aware cluster head selection for leach in wireless sensor networks,” *IET Wireless Sensor Systems*, vol. 11, no. 1, pp. 45–53, 2021.
- [44] N. Choudhury, R. Matam, M. Mukherjee, J. Lloret, and E. Kalaimannan, “Nchr: A nonthreshold-based cluster-head rotation scheme for ieee 802.15. 4 cluster-tree networks,” *IEEE internet of things journal*, vol. 8, no. 1, pp. 168–178, 2020.
- [45] P. Kułakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, “Angle-of-arrival localization based on antenna arrays for wireless sensor networks,” *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181–1186, 2010.
- [46] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.
- [47] S. Monfared, T.-H. Nguyen, L. Petrillo, P. De Doncker, and F. Horlin, “Experimental demonstration of ble transmitter positioning based on aoa estimation,” in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2018, pp. 856–859.
- [48] S. Azzouzi, M. Cremer, U. Dettmar, T. Knie, and R. Kronberger, “Improved aoa based localization of uhf rfid tags using spatial diversity,” in *2011 IEEE International Conference on RFID-Technologies and Applications*. Sitges, Spain: IEEE, 2011, pp. 174–180.

- [49] R. P. S. Hada, U. Aggarwal, and A. Srivastava, "A study and analysis of a new hybrid approach for localization in wireless sensor networks," *Journal of Web Engineering*, vol. 22, no. 2, pp. 279–302, 2023.
- [50] Y. Liu and Z. Yang, "Localizability," in *Location, Localization, and Localizability: Location-awareness Technology for Wireless Networks*. Springer, 2024, pp. 111–130.
- [51] R. Stoleru, T. He, and J. A. Stankovic, "Range-free localization," in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*. Springer, 2007, pp. 3–31.
- [52] B. Dil, S. Dulman, and P. Havinga, "Range-based localization in mobile sensor networks," in *European Workshop on Wireless Sensor Networks*. Springer, 2006, pp. 164–179.
- [53] N. Bulusu, J. Heidemann, and D. Estrin, "Gps-less low-cost outdoor localization for very small devices," *IEEE personal communications*, vol. 7, no. 5, pp. 28–34, 2000.
- [54] S. Kumar and D. Lobiyal, "An advanced dv-hop localization algorithm for wireless sensor networks," *Wireless personal communications*, vol. 71, pp. 1365–1385, 2013.
- [55] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 81–95, 2003.
- [56] D. Niculescu and B. Nath, "Ad hoc positioning system (aps) using aoa," in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 3. Ieee, 2003, pp. 1734–1743.

- [57] Y. Zhang and J. Zhao, "Indoor localization using time difference of arrival and time-hopping impulse radio," in *IEEE International Symposium on Communications and Information Technology, 2005. ISCIT 2005.*, vol. 2. IEEE, 2005, pp. 964–967.
- [58] T. Yang and X. Wu, "Accurate location estimation of sensor node using received signal strength measurements," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 4, pp. 765–770, 2015.
- [59] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.
- [60] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [61] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.
- [62] Y. Liu, Y. Wang, and J. Zhang, "New machine learning algorithm: Random forest," in *Information Computing and Applications*, B. Liu, M. Ma, and J. Chang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 246–252.
- [63] M. Shchekotov and N. Shilov, "Semi-automatic self-calibrating indoor localization using ble beacon multilateration," in *2018 23rd Conference of Open Innovations Association (FRUCT)*. IEEE, 2018, pp. 346–355.
- [64] C. Jo and C. Lee, "Multilateration method based on the variance of estimated distance in range-free localisation," *Electronics Letters*, vol. 52, no. 12, pp. 1078–1080, 2016.
- [65] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

- [66] J. Grover, Shikha, and M. Sharma, “Location based protocols in wireless sensor network—a review,” in *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*. IEEE, 2014, pp. 1–5.
- [67] P. Bholowalia and A. Kumar, “Ebk-means: A clustering technique based on elbow method and k-means in wsn,” *International Journal of Computer Applications*, vol. 105, no. 9, pp. 17–24, 2014.
- [68] D. M. Saputra, D. Saputra, and L. D. Oswari, “Effect of distance metrics in determining k-value in k-means clustering using elbow and silhouette method,” in *Sriwijaya international conference on information technology and its applications (SICONIAN 2019)*. Atlantis Press, 2020, pp. 341–346.
- [69] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [70] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm,” *Pattern recognition*, vol. 36, no. 2, pp. 451–461, 2003.
- [71] J. C. Bezdek, R. Ehrlich, and W. Full, “Fcm: The fuzzy c-means clustering algorithm,” *Computers & geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [72] D. Müllner, “Modern hierarchical, agglomerative clustering algorithms,” *arXiv preprint*, pp. 1–29, 2011.
- [73] J. Vesanto and E. Alhoniemi, “Clustering of the self-organizing map,” *IEEE Transactions on neural networks*, vol. 11, no. 3, pp. 586–600, 2000.