Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 1 (2024)

Hybrid Localization and Navigation System for Dynamic Target Employing RSSI and A* Algorithm

Asjad Suhail Akhtar, Garima Srivastava, Sachin Kumar
Amity University Uttar Pradesh, Lucknow Campus, (U.P)

Abstract: - Indoor localization in recent years has gain a place of interest significantly in various application, whether it's in IOT or mobile applications to enhance user's experience, from navigation inside a mall to improving healthcare facilities by providing them an accurate location. In this proposed work we will be developing an innovative approach for Indoor localization system using of already existed hardware with bit of modification and integrating the following Received Signal Strength Indicator (RSSI), A* algorithm and trilateration method. The objective is to develop an accurate and robust system for indoor localization in real-time. One thing kept in mind while developing this approach is to use already existing hardware and software and to keep the cost of applying this approach minimal. In the modern era of technology and with the availability and ease of access to internet and mobile devices, one of the issues is yet to be solved. Indoor navigation is one of the Millennium Problem among computer science and IOT enthusiastic, although Global Positioning System (GPS) technology is already existed but as it is satellite-based radio navigation system which led to low accuracy for indoor localization and navigation.

Keywords *RSSI, GPS, A* Algorithm, trilateration, Access Points*.

1. Introduction

Indoor localization in recent years has gain a place of interest significantly in various application, whether it's in IOT or mobile applications to enhance user's experience, from navigation inside a mall to improving healthcare facilities by providing them an accurate location.

The goal is to make a system that can deliver highly accurate coordinates of user's position, with minimal hardware installation and at low cost. This was possible with using Received Signal Strength Indicator (RSSI) to determine the location of the user's by identifying the strength of the signal received to know wither the device is close or far from the reference point [1] which in this approach it will be Access Point (AP). The Access Point are reference points at known locations which will emit the signal once the device is connected to the reference point, after that using modified trilateration method by taking at least three reference point and using a certain calculation logic by calculating the RSSI value an estimated coordinates will be generated in terms of the x and y [2], by plotting the output generated from this calculation on two dimension map we will have the location of the user and now it is possible to navigate.

The navigation logic will be done as the location of the reference point are in known location so does the blueprint of the building required to be navigated, and also we have user's coordinates, now using modified A* algorithm which is pathfinder algorithm, we will be able to navigate the user through the two dimension map with the help of heuristic function to determine wither the user is close from the target or not.

2. Literature Review

Localization is a hot topic among the enthusiasts of IOT and Computer Science, during the period of studying and referencing old work we can classified the previous work into two categories:

I. Hardware Approach

II. Software Approach

Furthermore, many of the studied shared a common technology which is Wi-Fi for indoor localization combining it with received signal strength-based techniques or fingerprinting.

In [1], they considered for the navigation system a Wi-fi based approach, because of the fluctuations in RSS that lead to issues in term of accuracy in results. To overcome this issue, they introduced a fingerprint spatial gradient (FSG), they also noted that many machine learning algorithms in the last ten years, such as K-nearest neighbour (KNN), support vector machine (SVM), and neural networks, have been used in radio fingerprint-based indoor localization methods as pattern matching [3].

In recent years Visible Light Communication (VLC) become one of the important technologies in navigation and identification of objects in physical spaces, which can offer ab accurate indoor positioning system with an easy and simple configuration. A system for localization is made using visible light that uses modified light source in [4] by Zhao et al. the system named as “NaviLight”, working of the system was that it used the existing light source as transmitters and treated light intensity as “LightPrint” which was basically a fingerprint. Again, this system was heavily inspired by Wi-Fi based approach for indoor localization which use Received Signal Strength Indicator (RSSI) fingerprint to position the location of the user, furthermore, using the LightPrint was quite challenging with a VLC system since the intensity of light is different in physical space comparing to signal strength. Additionally, there was no null communication between the sender and the receiver, operating Wi-Fi based approach, because this approach uses existing light infrastructure it is considered easy and cheap approach with high accuracy.

One of the approaches used was hybrid positioning system, from its name it is designed to combine more than one system for improving the performance in each system [5]. Each one of the systems used have drawbacks, creating system with more than one technology into hybrid system could lead to promising results in indoor localization, however the systems proposed to use hybrid of RF technologies with VLC have been understudied.

Huang et al. in [6] The working of proposed theoretical system is by deriving the location of the receiver based on the Li-Fi signals identifier and then using coefficient calibration with the RSSI values of the access point (AP) it will obtain the distance between the access point (AP) and the early identified lamp. However, the model proposed was not applied yet in real-world, although it has a lot of potential to solve the complexity of fingerprinting approach.

Luo et al. [7] proposed using VLC and Bluetooth we can deploy an effective and low-cost indoor positioning, using fingerprinting the visible light positioning system calculate the initial position information, then to rectify any errors in positioning it uses spring model. There are two nodes in the spring model: Mobile Device (MD) and anchor points (AP). A communication pipeline will be stablished between those nodes from a Bluetooth network, and it will be able to communicate with each other, such that any unknown node (MD) will be able to communicate its position to a neighbouring MD and known nods (APs), which will result to obtaining the distance between any pair of MDs and APs. Here both VLC and Bluetooth combination is used to create hybrid positioning system resulting in better usage rather than using each of them separately.

Having to find multiple paths while navigating in indoor environment, and as they are continuously changing a wide range of algorithms has been used and developed to fit best use case of each environment. These changes while navigating indoor play a vital role in overall accuracy of the used algorithms as survey say’s in [8]. There are also the range-based methods such trilateration and the range-free e.g., fingerprint to establish localization, to estimate user’s position in both methods and also in two dimensions space we required at least three access points [9], and at least four access points in three dimensions space.

Ma et al. proposed in [10] a positioning algorithm to improve the result of Wi-Fi RTT ranging, they also have explained the fine time measurement (FTM) in Wi-Fi characteristics. The results of the proposed approach achieved a localization error of 1.20 m in static mode and 1.31 m for dynamic mode.

Zhou et al. in [11] proposed for indoor positioning system an algorithm with anchor selection and matrix completion. From the results of the proposed system and approach they managed to achieve a localization error estimated 1.52 m.

While talking about fingerprint algorithms, there are deterministic approaches like Kalman Filter, SVM, DT, PCA and neural networks and other approaches, and for the probabilistic approach like, Kernel method, hidden Markov model, Gaussian distribution, particle filter, and Naive Bayes, taking the example using the fingerprint approach and the Kalam filter approach, Giovanelli et al. proposed in [12] an indoor positioning system using RSSI and ToF data fusion. It showed that it is about 50% lower than just RSSI data are used in the mean RMS error of data fusion, 5.69 m and 2.78 respectively. The system which was proposed utilizing both ToF and RSSI as location-dependent characteristic. The ToF measurement fluctuated a lot because of jitter, the resolution limited to intervals time, combination of both. Therefore, it can be lowered with averaging [13, 14].

Rizk et al. [15] showed an indoor positioning system with Wi-Fi RTT and RSSI. To overcome of the signal fluctuation problem, fingerprint interference, errors of multipath Propagation, and NLOS transmissions, they managed to achieve with the proposed system a localization error of 0.51 m and 0.59 m respectively, for both office and lab environments.

Singh et al. [1] showed an overview of machine learning indoor positioning system base and with RSSI value fingerprint. The overview provided a machine learning based Wi-Fi RSSI fingerprinting for localization and comparison of system's performance, Machine Learning predication modules the performance of them has been compared based on their accuracy classification, positioning error, robustness, scalability, complexity, and localization space.

The author in [16] evaluates that CNN has a high robust, low complexity and high scale ability. Then from the summarized view schemes table it can was concluded that KNN will increase to be more robust and decrease the error positioning, while it is possible to decrease the complexity using PCA to reduce the computation time.

Chin et al. [17] they proposed a positioning system based on MIMO with CSI data using artificial neural network. They compared the performance of FCNN, CNN, and GCNN. For the GCNN the error distance that is below 0.2m which is more than 90%, and for the proposed CNN the error distance that is below 0.2m is 75%, lastly error distance is all above 0.4m for the FCNN.

As mentioned above most of the proposed solutions use model based or fingerprint based approach, the performance of this approach depends on numbers of factors such as number of reference points, the number of packets collected, and the matching algorithm, RADAR is one of the best works that uses this approach [18], by combining both signal propagation modeling and empirical measurements to find user's location.

Wang et al. [19] in their approach used databases of three fingerprints which was collected at different distances for comparing 8 positing algorithms. The approach used in [19, 18] has two fatal drawbacks: first is the challenge to create such databases as the approach requires to have data of multiple locations, and the second is as we are aware that the environment is subject to changes so it needs to be updated frequently which is not only time consuming but also exhausting resources, these two issues makes it a difficult method to be adopted.

Talking about the cost of collecting singles to minimize it, Wu et al. [20] the approach they took was to use virtual reference points based on the long-distance model. Similarly, Zhang et al. [21] to create the database of fingerprint they used a path loss approach, their work has shown that if we decrease the date collocation, although it will be beneficial financially, it will lead to decline in location identification, specifically for mobile devices as their ability is limited to room level resolution.

To reduce the effort to implement IPSs signal propagation models can be used, Onofre et al. [22] to measure the distance between BLE tags they implemented an adjusted distance curve using a long-distance model. The curve approach used eliminates the requirement of large data. A note to be taken that the experiment was performed in small room (13 m) with no real-life variables such as obstacles, taking that note into consideration the derived distance curve may not be generalized to large-scale environments.

Moghtadaiee et al. [23] they proposed a path loss propagation model using a zone based approach, it was shown by their experiments that the average error was reduced by 26% using zone based when compared to other path loss models [21, 24, 20]. However, the optimal path loss exponent value is already known in the proposed methodology in [23].

Yang et al. [25] proposed a new algorithm using trilateration and RSSI, they used Gaussian filter to preprocess the data and to reduce the RSSI variation and exponent of path loss is estimated through the last squares curve fits (LSCF). Similarly, Cengiz [21] proposed to improve and betterment of location estimation and map RSSI to distance is to increase and use more of anchor nodes and use line fitting algorithms, in the previous works mentioned they adopted a technique to fix parameter values in the propagation model to describe the signals in all regions of the environment. However, it is not optimal in large case environments due to the behavior of the signal in the different regions of the environment.

Finally, Shi et al. [26] proposed a system that will adjust the path loss exponent depending on filters applied to the date (RSSI values), the drawback of the [26] is the need to be calibrated, which involve multiple factors such as model parameter adjusting which requires a large effort.

Our proposed solution is designed to minimize the modification of already existed structure of the site required to be installed and to depend on software solutions more than on hardware approaches, the system to be proposed will be using RSSI singles from Wi-Fi and trilateration method to determine the positioning of the user’s device, it is worth noting that the proposed system will be assuming that the site has at least three access points and that their locations are fixed and known which will help in calibrating process.

3. Objectives

3.1. Introduction

Based on the cite information above and to overcome the limitations in other implementations, our proposed solution is to develop an Indoor Positioning and Navigation System (IPNS) that combines advanced signal processing techniques and trilateration algorithms with inputs from RF and inertial sensors to estimate real-time location and navigation using A* algorithm for dynamic users inside buildings with submeter accuracy. Concept developed comes under hybrid category as the algorithm embedded will work with or without internet access [25], one of the key features of the solution is to have minimal modifications of site setup and to achieve localization and navigation without installing any hardware. e.g., beacons or Bluetooth devices.

3.2. Objectives

i. Develop a robust IPNS that can accurately locate and navigate users within unknown indoor environments.

ii. Estimate real-time location information for dynamic users inside buildings with submeter accuracy via inputs from RF and inertial sensors.

iii. Implement trilateration algorithms to estimate user positions based on signal strength measurements and inputs from RF and inertial sensors.

iv. Create a user-friendly interface for seamless navigation.

v. Integrate sensor fusion techniques to enhance positioning accuracy.

vi. Establish a calibration process to fine-tune signal strength to distance mapping.

4. Methods

The proposed methodology to create a robust system capable of locating user’s device and navigating it through a desired location keeping in mind the minimize modifications and maximize the result and accuracy.

The proposed system will operate on two levels:

i. Hardware Level (HL)

ii. Software Level (SL)

Each level will be dependent on the other level to guarantee a smooth experience.

4.1. **Hardware Level (HL)**

This level is very critical and important for the system to be achieve as the maximum input of Software Level will be the resultant output of this level (HL) [27], the challenge is to use hardware parts assuming that are already exist at the implementation site because of the widely used or more to be minimal stander in their category.

4.1.1. **Access Point (AP)**

The star of the show, the most important element in the whole system, it is assumed that in a large building where a navigation is needed to have multiple access points in each floor, the need of these access points to act as reference points at fixed locations it will play a vital role in trilateration method which will be discussed in the SL part [28].

When a device is connected it will obtain a measuring metric called Received Signal Strength Indicator (RSSI), this metric is used in wireless communication systems to quantify the strength of the received radio signal, it is possible to determine the location of the device using this indicator a higher RSSI value generally indicates a stronger signal and lower RSSI value indicates a weaker signal which means in other words, if the RSSI value received is high the location of the device is near to the access point, and if it's low the location of the device is far to the access point [29].

The obtained values are typically expressed in dBm (decibels referenced to 1 milliwatt) or in dB (decibels), In the context of Wi-Fi, for example, RSSI values may range from around -30 dBm (very strong signal) to -100 dBm (very weak signal).

An important note to be kept in mind that the access point must be working on at least 802.11n protocols where both 2.4GHz and 5GHz is supported with bandwidth 20 or 40 MHZ and above [28], and to have at least 3 in numbers in the required area to be navigated for accuracy.

4.1.2.**User's Device**

User's device will act as mediator between the user and the access point, it will be key responsible to navigate the user to the required point, the device will be also responsible in estimating the position by obtaining the RSSI values from the AP [27].

The role of the device we will like mentioned above to determine the position of the user and the to navigate him the required position, this will be possible when the device connects to the nearest AP; a handshake operation will occur, in that operation an exchange of information will be done among the exchanged information are is obtaining RSSI value of the AP to determine how near or far is the reference point [30], all this will happen in matter of milliseconds when the first connection is established [26], then through an app on user's device it will ask the user to enter the required location to be navigated to, using pathfinding A* algorithm which will be discussed further in SL part a path will be generated for the user to follow, and the navigation will be done using Inertial Sensors which will be explained further in 4.1.3.

In the recent years with the technology booming especially the mobile industry e.g., smartphones and tablets, most to the widespread devices are already equipped with latest required technology to achieve this system [31], so any device bought from recent years will be sufficient to work with the setup environment.

4.1.3.**Inertial Sensor (IS)**

On of the hardest question to answer while designing and implementing the system is after determining the location of the user how it to tell the user at which direction to take? and how to know if the user is moving?

To answer this question the system is heavily dependent on two inertial sensors for navigation point of view:

i. Gyroscope Sensor.

ii. Accelerometer Sensor.

4.1.3.1. Gyroscope Sensor

To answer the first question the Gyroscope Sensor is used, although it is used in aircraft, spacecraft, and drones, it's widely used in user consumer electronics e.g., smartphones and tablets.

Gyroscope sensors rely on the conservation of angular momentum, which is a fundamental concept in physics. Angular momentum is the rotational analogy of linear momentum and is preserved in the absence of external torques [32].

Using this sensor, it will be possible to know the directions of the device and that will ease the guiding.

4.1.3.2. Accelerometer Sensor

The second part of the question is "how to know if the user is moving?" by using this sensor not only it will be possible to know the movement of the user's device but also the acceleration speed also, which will help in determining Time of Arrival (TOA or ToA) [33].

Accelerometers work by detecting changes in velocity and, therefore, acceleration they work operate based on the principles of motion and inertia, it can measure acceleration in multiple axes (e.g., x, y, and z) and provide information about both static (e.g., gravity) and dynamic (e.g., motion) accelerations [32].

4.1.4. Server

The backbone for the whole proposed system will be the sever, where all the calculation operations happen, and it will act as the coordinator between user's device the reference points (Access Points) [15].

As it is mentioned this server will play vital role in the whole system, yet the hardware specifications aren't demanding, the server used in this architecture can be any kind, wither its industry grade server or a home server lab.

4.2. Software Level (SL)

This level as it was discussed in previous point it will heavily dependent on the HL for its functionality, the challenge was in the Hardware Level to keep the environment setup as it is with minimal hardware modifications, so although it will be dependent on HL but yet the setup to be assumed it will be at it is in the area to be localized, so the challenge in this level is to create an robots system using the software available and utilize the hardware at its peak.

Achieving this challenge and goal, the following technologies has been used.

4.2.1. Python

The implementation of this project needs to have some sort of programming/scripting language, the one to be chosen in this implementation is python.

Python got chosen for its support of cross platform and large-scale libraries supported for all kinds of operations, also as it is a scripting language which can easily be deployed on the server for automation, few libraries that been used are as the following.

4.2.1.1. Matplotlib

Python external library used to plot graphs and diagram, in this project it will be used for plotting the 2D maps and diagrams like walls and objects for simulation.

4.2.1.2. **NumPy**

it is used to perform a wide variety of mathematical operations on arrays, in this project will be used for processing and analysing signal strength measurements.

4.2.2.**MongoDB**

In the 4.1.1. it was mentioned that in this approach the location of the access points will be known so does the mapping of the area, and also that this approach will have a multiple reference, to contain all this we need to used DBMS which is MongoDB, it is classified as NOSQL database program, in this project it will store and keep information about the user and the location of reference points and access points.

4.2.3.**NodeJS**

To hold and fetch everything in the backend NodeJS is used, it is back-end JavaScript runtime environment, here in this project will be used to handle the back-end part such as constructing the APIs and the server with their routes.

4.3. **Mathematical Formulas**

To asset SL and achieve the required objective, multiple mathematical formulas have been used, these Mathematical formulas are key sole to make the proposed system work [34], the whole manipulation of the obtained RSSI values and with the coordinates of known location of the access point will be of no use if it's not utilized efficiently [35], here are brief list of the used formulas.

4.3.1.RSSI-to-Distance Mapping:

- Formula: $d = 10\ \frac{RSSI-A}{-10.n}$
- Explanation: This formula is used to estimate the distance (d) between a reference point and a device based on the Received Signal Strength Indicator (RSSI) value. It utilizes two parameters:
  - $\boldsymbol{RSSI}$: $the\ measured\ RSSI\ values$
  - $\boldsymbol{A}$: $the\ signal\ strength\ at\ a\ refrence\ distance\ (usally\ 1\ meter)$
  - $\boldsymbol{n}$: $the\ path\ loss\ exponent\ that\ characterizes\ signal\ propagation\ in\ the\ environment.$
- Usage: By measuring the RSSI value and knowing the reference point's signal strength parameters (A and n), you can estimate the distance from the reference point [15]. This distance information is crucial for later trilateration calculations.

4.3.2.**Two-D Trilateration:**

- **Formula:** $x = \frac{r_1^2+r_2^2+d^2}{2d}, y = \pm\sqrt{r_1^2 - x^2}$
- **Explanation:** In 2D trilateration, where *x* and *y* represent the estimated coordinates of the target device, this formula calculates the device's position based on the distances (r_1, r_2, etc.) from the device to at least three reference points with known positions.
- **Usage:** Given the distances from at least three reference points and their coordinates, you can use this formula to estimate the 2D position (x, y) of the target device [25].

4.3.3. **Three-D Trilateration:**

- Formula: This extends the 2D trilateration to three dimensions, where you calculate the x, y, and z coordinates of the target device based on the distances from reference points.
- Usage: Like 2D trilateration, but in 3D space, this formula is used when you have reference points with known 3D coordinates and distances to the device [25].

4.3.4.**A Heuristic Function: ***

Formula: $h(n) = Manhattan\ Distance(n, end)$

Explanation: In the A* pathfinding algorithm, the heuristic function (h(n)) estimate the cost from a given node (n) to the destination node (end). The Manhattan distance is often used as the heuristic, which calculates the sum of the absolute differences in x and y coordinates.

Usage: A* uses this heuristic to guide its search for the shortest path. It Favors nodes that are closer to the destination, improving the efficiency of the search.

4.3.5.**A Cost Function: ***

i. **Formula:** $f(n) = g(n) + h(n)$
ii. **Explanation:** In A*, the cost function $(f(n))$combines two components:
   - $\boldsymbol{g(n)}$: The cost of reaching the current node (n) from the start node. It accumulates as you move through the graph.
   - $\boldsymbol{h(n)}$: The heuristic cost estimate from the current node to the destination node.
iii. **Usage**: A* uses this cost function to evaluate and prioritize nodes during the search. Nodes with lower $f(n)$ values are explored first.

4.4. **Algorithm**

Clubbing the *HL and SL* and topping it with the formulas mentioned in *4.3.* we will have our environment setup ready form both hardware and software side, the resulted proposed system will be something as shown in Figure 1. The setup will have two nodes Sender and Receiver, the actor of sender in this case will user's device e.g., smartphone or tablet, and the receiver node will act by both access point and server [36].

The core algorithm will involve trilateration based on RSSI measurements from reference points, coupled with sensor fusion techniques to improve accuracy [12].

4.4.1. **Detailed Steps:**

Step 1: Define the Indoor Environment

i. Define the layout of the indoor environment, including walls, obstacles, and reference point positions.
ii. Create a map representation where each cell indicates the layout and accessibility of the environment.

Step 2: Initialize Reference Points and Fingerprint Data

i. Define a set of reference points within the indoor environment.
ii. Record the signal strength patterns (RSSI values) at each reference point. This is the fingerprint data.
iii. Store the reference point positions and their associated fingerprint data.

Step 3: Define A* Algorithm

i. Implement the A* algorithm to find the shortest path between two points in the indoor environment.

Step 4: Implement Fingerprint-Based Position Estimation

ii. Create a function, estimate_position, that takes signal strengths (RSSI values) as input.
iii. Calculate the Euclidean distance between the signal strengths and the fingerprint data of each reference point.
iv. Identify the reference point with the smallest distance, indicating the best match for the current signal strengths.
v. Return the position of the matched reference point as the estimated position.

Step 5: Test the System

i. Define test cases with known true positions and corresponding RSSI values.
ii. For each test case, calculate the true position and obtain RSSI values.
iii. Use the estimate_position function to estimate the position based on the RSSI values.
iv. Apply the A* algorithm to find a path from the estimated position to the true position.
v. Calculate the error distance between the true position and the estimated position.

Step 6: Visualization

i. Create a visualization function, plot_map, to display the indoor environment.
ii. Plot the true position, estimated position, walls, obstacles, and the A* path on the map.

Step 7: Error Analysis

i. Calculate error distances for all test cases.
ii. Analyze and report the error percentages and patterns in accuracy.

4.4.2.**Pseudocode**

Define the indoor environment, reference points, and fingerprint data

Function estimate_position(signal_strengths):

i. Create an empty list weighted_positions
ii. For each reference_point in reference_points:
    a. Calculate the Euclidean distance between signal_strengths and reference_point's fingerprint data
    b. Calculate weight = 1 / (distance + 1)
    c. Append (reference_point, weight) to weighted_positions
iii. total_weight = sum of weights in weighted_positions
iv. Initialize estimated_position = [0, 0]

For each (position, weight) in weighted_positions:

    a. estimated_position[0] += (position[0] * (weight / total_weight))
    b. estimated_position[1] += (position[1] * (weight / total_weight))

Return estimated_position

Function calculate_error (true_position, estimated_position):

i. Calculate and return the Euclidean distance between true_position and estimated_position

For each test_case in test_cases:

ii. Get true_position and RSSI values from the test_case
iii. Use estimate_position to estimate the position from RSSI values
iv. Use A* algorithm to find a path from the estimated position to the true_position
v. Calculate the error_distance as calculate_error(true_position, estimated_position)
vi. Visualize the environment and the path
vii. Print true_position, estimated_position, and error_distance

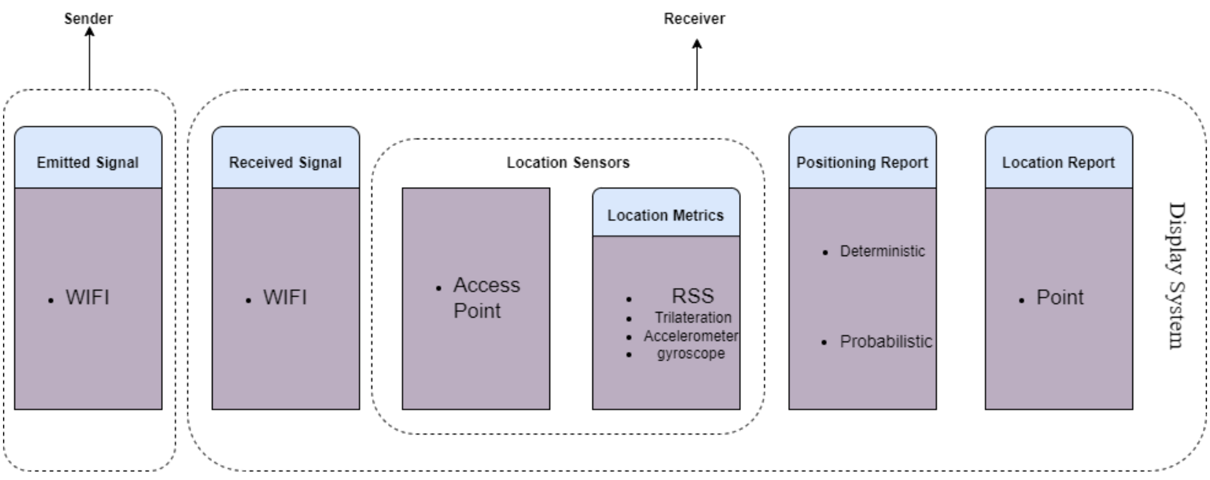

Figure 1. Generalized block diagram.

5. **Experimental Setup**

To check the credibility and accuracy of the system an experiment is needed, the final setup of approach will be as it is shown if Figure 2, where taking a scenario of a three story building, and each floor have n numbers of rooms and at least three access points as denoted by FiAP#j in Figure 2, where i is the floor number and j is the number of the access point/reference point, each of them will be connected to server remotely.

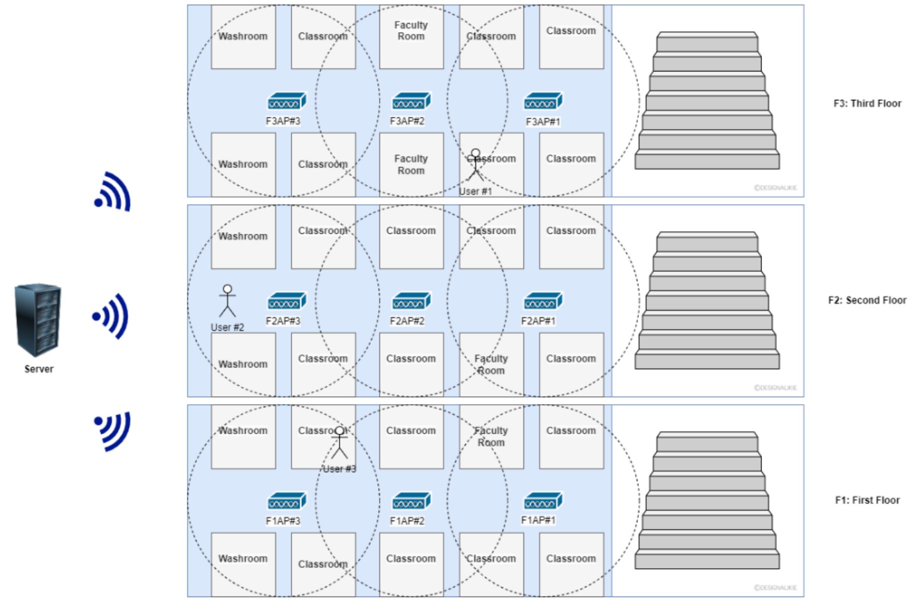

Figure 2. Block Diagram showing the approach of the idea.

Although the above taken scenario must look very realistic and well structed but it´s hard to check for the accuracy of the experiment with all the interference, so a simulation was done to isolate the setup and to get the maximus accuracy of the experiment.

5.1. **Two-Dimensional Map**

To keep the experiment simple focusing on the result and the accuracy of the localization and predication the setup was done on 2D map of 5x4 array that is represented in the following format:

$$\begin{array}{ccccc} S & 1 & 2 & W & 3 \\ 4 & 5 & 6 & 7 & 8 \\ W & 9 & E & 10 & W \\ 11 & 12 & 13 & 14 & 15 \end{array}$$

iv. $Where: S: Start$
v. $E: End$
vi. $W: Wall$
vii. $Numbers: Reference\ point\ signal\ strength\ values$

5.2. **Reference Points (RP)**

As per discussed in the proposed system that the reference points will be at known location, so dose in the experiment we took the following coordinates as fixed reference points with the RSSI values at that point:

- $(0, 3) = signal\_strengths: [80, 70, 60]$
- $(0, 2) = signal\_strengths: [70, 65, 55]$
- $(1, 1) = signal\_strengths: [75, 68, 58]$
- $(1, 2) = signal\_strengths: [75, 68, 58]$

5.3. **Walls and Obstacles:**

To simulate a real-life experiment to place some virtual obstacles and walls between the coordinates, which will take effect in applying A* algorithm for find the best path and so dose on the RSSI values, these are the coordinates of walls and obstacles taken:

- $walls = [(0,3),(2,0),(2,4),(3,0),(3,4)]$
- $obstacles = [(1,3)]$

5.4. **Plotting the Path**

for the plotting the path in the experiment an external Python library is used while writing the source code of the experiment called Matplotlib which was discussed and mentioned in 4.2.1.1.

5.5. **Final Setup**

Compiling previous points for the experiment the two-dimensional map shown in Figure 3 is the final setup for the experiment to be conducted.

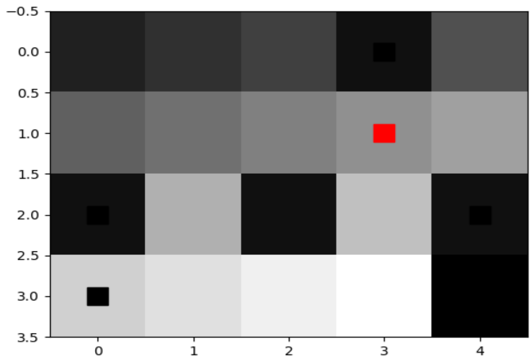

Figure 3. Two-Dimensional map for the setup.

The balck boxes represent walls in real life and the red box represent an obstecal, during exction the walls and obstecal will pervent the user to move on the highlighed blocks as it is in real life, and they will act to weakn the radio reviced or emitted signals (*RSSI*).

6. **Results and Discussion**

Taking previously mentioned setup to check the credibility and accuracy of the approach, in this section of the paper it will continue the result, or the outputs generated from the test and the test cases taken and also to discuss the metrics and their justification.

6.1. **Test Cases**

To add to the acuracy and creadability of the system an unified test cases has been taken across every test to genrate output, and easily compare and dettect the margin of error of each test and metric used.

The following are the coordinates and RSSI values used in each test case:

- $True\ Position: (3,3), RSSIs: [80,70,60]$
- $True\ Position: (1,1), RSSIs: [80,70,40]$
- $True\ Position: (2,3), RSSIs: [80,65,75]$
- $True\ Position: (0,3), RSSIs: [55,60,75]$
- $True\ Position: (1,0), RSSIs: [65,62,55]$
- $True\ Position: (3,2), RSSIs: [70,68,62]$

Where *True Position* refers to the actul coordination of the user on the two-dimensional map in form of *x and y* coordinates, and *RSSIs* refer to the RSSIs values given while proforming the test cases.

A note to be taken for the reason of having three RSSI values in each test cases, in the simulated scenario, trilateration is used to estimate the position of the user's device based on RSSI (Received Signal Strength Indicator) values received from three reference points (beacons or access points). These RSSI values are a way to estimate the distance between the user's device and the reference points. Trilateration typically requires at least three reference points to estimate the position in a 2D space or four reference points for a 3D space.

6.2. **Matrics Used**

Last point before moving forward to the result and it's discussion, the matrics used to genrate the results.

The matric to be disscued will be number of access points, in total fiften expermintas have been done and categories into three parts:

6.2.1. **Normal:** where the approch is terdional with no optimaztion is carried and the number of refrence points to be taken are three in total and their coordinates has been mention above in *5.2* the reason to take three refrence points is for the proposed system to work it needs at least three refrence points to perform trilateration as mention in *4.3.2.*

6.2.2. **Weighted Average:** where for each reference point, the function calculates a weight based on the similarity (or distance) between the reference point's signal strengths and the current signal strengths. The weight is inversely proportional to the distance. Reference points that are closer to the current signal strengths receive higher weights, while those farther away receive lower weights, The sum of all weights is computed. This total weight is used to normalize the influence of each reference point on the estimated position, the estimated position is computed as a weighted average of the reference point positions. The weight of each reference point is divided by the total weight to ensure that the weights sum to 1, making it a proper weighted average.The reason to chose Weighted average-based estimation approch beacus it is a standard technique in positioning systems and can be easily implemented and understood.

6.2.3. **Weighted Average & Increasing in Reference Points:** this approch is semiral to the previous approch the only difference that it increased the number of access points/refrence point, in pervious tests the number of refrence points were three, but in this approch the number increased to be come seven in total, and the reason for increasing that although it is nessary to have at least three refrence points for the system to work but the avrage number of access points in large scale bulding are seven, and the extra refrence points added are:

- $(3,0) = signal\ strengths: [70,60,50]$
- $(3,3) = signal\ strengths: [85,75,65]$
- $(2,4) = signal\ strengths: [60,55,45]$

6.3. **Results**

In Total 15 experiment been conducted in the setup simulation and by applying the algorithm mentioned above and the calculations used, the result of these experiments can be divided in three categories as mentioned in 6.2., show the results of them one by one.

6.3.1. **Normal Setup:**

The total number of experiments conducted in this category are 5 in total, the input of each experiment was mentioned in test cases mentioned in 6.1, the result shown here in Figure 5 are only for the first test case.

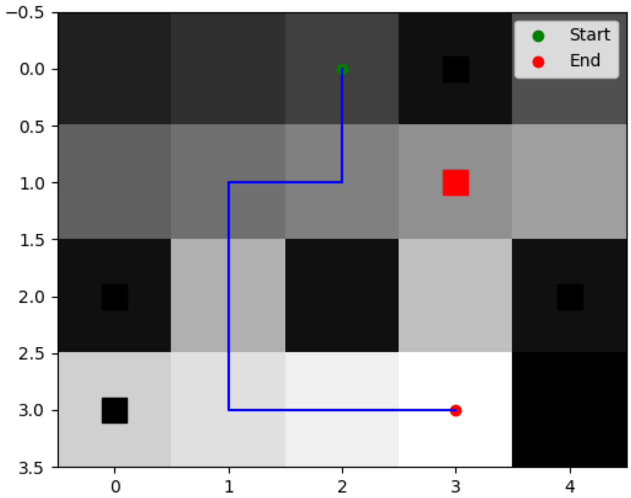

Figure 4. Output of first test case in the normal setup.

True Posistion of the device in *(3,3)* in *x and y* coordinates, and the estimated position was *(0,2)* in *x and y* coordinates, the result in error distance is *3.162%*.

Here are is a table showing the result of all 5 experiment using the normal setup:

Table 1: Normal setup results.

Ture Position *(x,y)*	**Estimated Position *(x,y)***	**Error Distance %**
(3, 3)	(0, 2)	3.16227766016837
(1, 1)	(0, 2)	1.414213562
(2, 3)	(0, 2)	2.236067977
(1, 0)	(0, 2)	2.236067977
(3, 2)	(0, 2)	3

6.3.2.**Weighted Average Setup**

The total number of experiments conducted in this category are 5 in total, the input of each experiment was mentioned in test cases mentioned in 6.1, the result shown here are in Figure 5 only for the first test case.

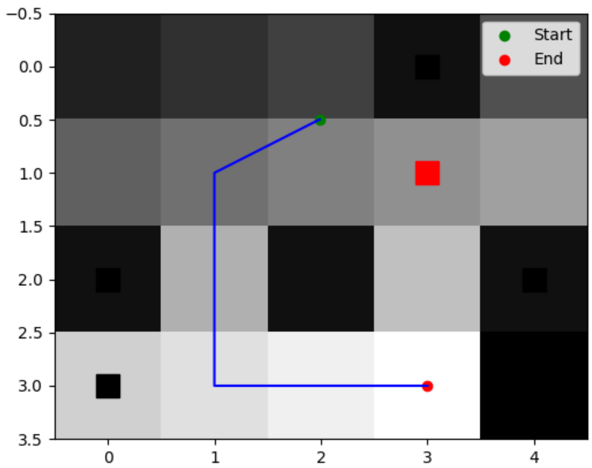

Figure 5. Output of first test case in the weighted average setup.

True Posistion of the device in *(3,3)* in *x and y* coordinates, and the estimated position was *(0.497949745853588825, 1.9878972121454896)* in *x and y* coordinates, the result in error distance is *2.69900120923592%.*

Here are is a table showing the result of all 5 experiment using the normal setup:

Table 2. weighted average setup results.

Ture Position *(x,y)*	**Estimated Position *(x,y)***	**Error Distance %**
(3, 3)	(0.49794974585358825, 1.9878972121454896)	2.699001209
(1, 1)	(0.497969834126953, 1.9880335215201081)	1.10826194
(2, 3)	(0.49794339850287783, 1.9878410773032356)	1.811253632
(1, 0)	(0.497996667071947, 1.988056896244629)	2.050457893
(3, 2)	(0.4979713964312587, 1.9879409195763722)	2.502057664

6.3.3.**Weighted Average & Increasing in Reference Points**

The total number of experiments conducted in this category are 5 in total, the input of each experiment was mentioned in test cases mentioned in 6.1, the result shown here in Figure 6 are only for the first test case.

True Posistion of the device in *(3,3)* in *x and y* coordinates, and the estimated position was *(1.449836665206648, 2.158740821631737)* in *x and y* coordinates, the result in error distance is *1.76372428960041%.*

Here are is a table showing the result of all 5 experiment using the normal setup:

Table 3. Weighted Average & Increasing in Reference Points Setup Results.

Ture Position *(x,y)*	Estimated Position *(x,y)*	Error Distance %
(3, 3)	(1.449836665206648, 2.158740821631737)	1.76372429
(1, 1)	(1.4496686393714597, 2.1583885492891106)	1.242604489
(2, 3)	(1.4498762198010477, 2.158867420804982)	1.005057306
(1, 0)	(1.4492845598610296, 2.1582624635224845)	2.204530217
(3, 2)	(1.4495511266812637, 2.158587844695286)	1.558538358

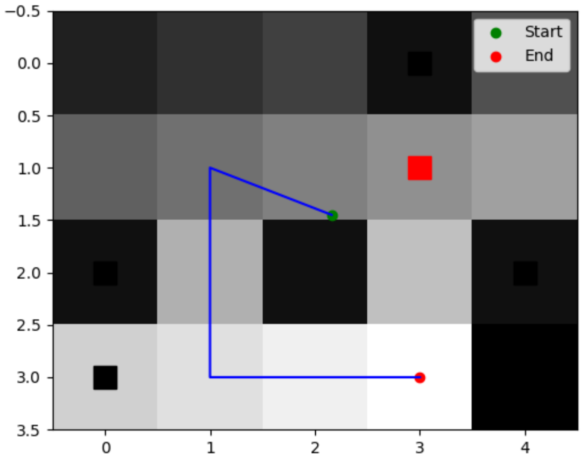

Figure 6. Output of first test case in the Weighted Average & Increasing in Reference Points setup.

6.4. **Comparison Chart**

View the results of each setup one by one, a small picture can be painted stating which approach is more satiable in point of view of accuracy.

To compile all the result, a comparison chart has been created, the comparison chart focuses on the error distance as it is shown in each table.

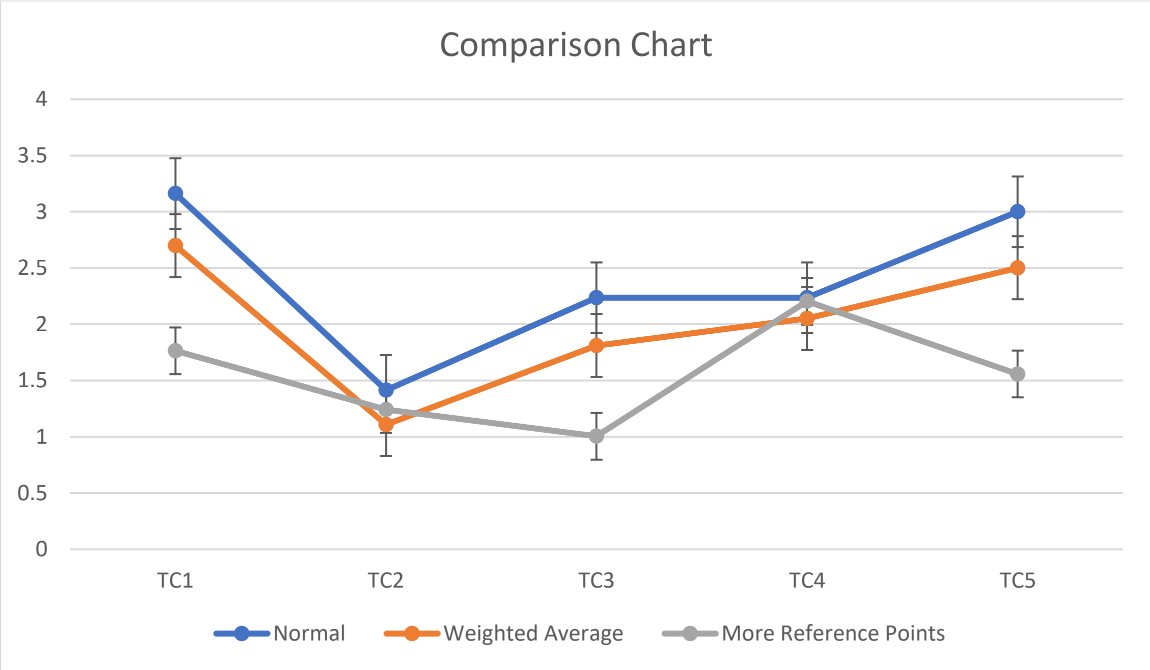

Figure 7. Comparison Chart between all setups.

6.5. **Discussion**

As per the number of the tests conducted using multiple vectors and methods and trying to simulate the experiments as much as close to a real-world simulation using the information mentioned above.

In the Figure 7 where the comparison chart is shown the more graph tend to Zero in the x axis the more accurate the result is, keeping this in the mind we see that the result of normal result are less accurate comparing to the other setups, and the reason is all reference points are treated equally, and their positions are averaged without considering the quality or similarity of their signal strengths to the current measurements. This means that even reference points with signal strengths that are quite different from the current measurements have the same influence as reference points with similar signal strengths. The normal averaging approach is sensitive to outliers. If there are reference points with noisy or inconsistent signal strength measurements, they can significantly affect the estimated position because their values are included in the average.

And the reason for the weighted average approach to be more accurate because it addresses these limitations by assigning higher weights to reference points that closely match the current signal strengths and lower weights to reference points that are less similar, it adapts to signal variability and interference by giving preference to reference points with signal strengths that are closer to the current measurements. And reduces the impact of outliers and noisy measurements because they typically result in higher distances, leading to lower weights. By focusing on reference points with signal strengths that closely match the current situation, the weighted average approach tends to provide more accurate estimates of the device's position.

7. **Conclusion**

Paper presents a novel approach for indoor localization and navigation using Received Signal Strength Indicator (RSSI), A* algorithm, and trilateration. The objective is to develop an accurate and robust system for indoor positioning and navigation while minimizing hardware modifications and costs. The proposed methodology operates on two levels, the Hardware Level (HL) and the Software Level (SL), with a strong interdependence between them. The experiments conducted using different approaches reveal important insights into the system's performance. The normal setup, which treats all reference points equally, yields less accurate results due to the sensitivity to outliers and signal strength variations. On the other hand, the weighted average approach, which assigns higher weights to reference points with similar signal strengths, offers improved accuracy and robustness, making it a preferred method for indoor positioning. Furthermore, the Weighted Average & Increasing in Reference Points approach, which introduces more reference points into the system, demonstrates even better accuracy and resilience to noise. This approach aligns more closely with real-world scenarios where multiple access points are available. The results from this approach showcase its effectiveness in providing accurate indoor localization and navigation. Overall, the proposed system shows promise in addressing the challenges of indoor positioning and navigation, with the weighted average approach being the most suitable for achieving accurate and reliable results. Further research and real-world testing may be needed to fine-tune and optimize the system for various indoor environments and applications.

References

[1] S. C. a. R. P. N. Singh, "Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview," *IEEE Access*, vol. 9, pp. 127150-127174, 2021.

[2] I. K. Pande S, "Robust trilateration based algorithm for indoor positioning systems," *Tanzania Journal of Science*, p. 1195–1210, 2021.

[3] L. F. a. H. L. J. Luo, "Indoor Positioning Systems Based on Visible Light Communication: State of the Art," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2871-2893, 2017.

[4] J. W. X. Z. C. P. Q. G. a. B. W. Z. Zhao, "NaviLight: Indoor localization and navigation under arbitrary lights," in *IEEE Conference on Computer Communications*, Atlanta, GA, USA, 2017.

[5] M. B. A. M. a. J. V. A. Correa, "A Review of Pedestrian Indoor Positioning Systems for Mass Market Applications," *Sensors*, vol. 17, no. 8, p. 1927, 2017.

[6] Q. Y. Z. Z. G. a. C. L. Huang, "Refining Wi-Fi based indoor localization with Li-Fi assisted model calibration in smart buildings," *arXiv preprint*, p. 1602, 2016.

[7] Z. Z. W. &. Z. G. Luo, "Improved spring model-based collaborative indoor visible light positioning," *Opt Rev* , p. 479–486 , 2016.

[8] F. L. J. Y. Y. W. W. H. D. C. P. a. N. Q. Liu, "Survey on WiFi-based indoor positioning techniques.," *IET Communications*, pp. 1372-1383, 2020.

[9] K. N. P. a. A. N. V. A. Kushki, "Kernel-Based Positioning in Wireless Local Area Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689-705, 2007.

[10] B. W. S. P. a. D. R. S. C. Ma, "Wi-Fi RTT Ranging Performance Characterization and Positioning System Design," *IEEE Transactions on Mobile Computing*, vol. 21, no. 2, pp. 740-756, 2022.

[11] Y. L. Y. W. Q. P. X. Y. a. W. N. M. Zhou, "Device-to-Device Cooperative Positioning via Matrix Completion and Anchor Selection," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5461-5473, 2022.

[12] E. F. D. F. a. D. M. D. Giovanelli, "Bluetooth-Based Indoor Positioning Through ToF and RSSI Data Fusion," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, France, 2018.

[13] A. C. P. P. a. D. F. D. Macii, "A Data Fusion Technique for Wireless Ranging Performance Improvement," *IEEE Transactions on Instrumentation and Measurement,* vol. 62, no. 1, pp. 27-37, 2013.

[14] B. M. a. E. F. D. Giovanelli, "Bluetooth Low Energy for data streaming: Application-level analysis and recommendation," in *6th International Workshop on Advances in Sensors and Interfaces (IWASI)*, Gallipoli, Italy, 2015.

[15] A. E. a. H. Y. H. Rizk, "A Robust and Accurate Indoor Localization Using Learning-Based Fusion of Wi-Fi RTT and RSSI," *Sensors,* vol. 22, no. 7, p. 2700, 2022.

[16] I. A. R. Z. M. T. a. R. B. W. Njima, "Deep CNN for indoor localization in IoT-sensor systems," *Sensors,* vol. 19, no. 14, p. 3127, 2019.

[17] C. -C. H. D. S. a. T. J. W. -L. Chin, "Intelligent Indoor Positioning Based on Artificial Neural Networks," *IEEE Network,* vol. 34, no. 6, pp. 164-170, 2020.

[18] P. B. a. V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, Tel Aviv, Israel, 2000.

[19] B. W. e. al., "A Novel Weighted KNN Algorithm Based on RSS Similarity and Position Distance for Wi-Fi Fingerprint Positioning," *IEEE Access,* vol. 8, pp. 30591-30602, 2020.

[20] Y. -L. C. a. S. -T. S. Y. -H. Wu, "Indoor location estimation using virtual fingerprint construction and zone-based remedy algorithm," in *International Conference On Communication Problem-Solving (ICCP)*, Taipei, Taiwan, 2016.

[21] G. H. N. S. a. L. S. J. Zhang, "Path-Loss-Based Fingerprint Localization Approach for Location-Based Services in Indoor Environments," *IEEE Access,* vol. 5, pp. 13756-13769, 2017.

[22] P. M. S. J. P. P. a. P. S. S. Onofre, "Surpassing bluetooth low energy limitations on distance determination," in *IEEE International Power Electronics and Motion Control Conference (PEMC)*, Varna, Bulgaria, 2016.

[23] S. A. G. a. M. G. V. Moghtadaiee, "New Reconstructed Database for Cost Reduction in Indoor Fingerprinting Localization," *IEEE Access,* vol. 7, pp. 104462-104477, 2019.

[24] A. C. A. J. P. A. a. M. P. B. V. Cantón Paterna, "A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering," *Sensors,* vol. 17, no. 12, p. 2927, 2017.

[25] L. G. R. G. M. Z. a. T. Z. B. Yang, "A Novel Trilateration Algorithm for RSSI-Based Indoor Localization," *IEEE Sensors Journal,* vol. 20, no. 14, pp. 8164-8172, 2020.

[26] W. S. X. L. a. X. X. Y. Shi, "An RSSI Classification and Tracing Algorithm to Improve Trilateration-Based Positioning," *Sensors,* vol. 20, no. 15, p. 4244, 2020.

[27] V. G. E. &. A. R. İlçi, "Performance Comparison of 2.4 and 5 GHz WiFi Signals and Proposing a New Method for Mobile Indoor Positioning," *Wireless Personal Communications*, vol. 110, no. 3, p. 1493–1511, 2020.

[28] N. A. M. M. a. W. Zaw, "Comparative Study of RSS-based Indoor Positioning Techniques on Two Different Wi-Fi Frequency Bands," in *17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand, 2020.

[29] J. &. T. J. &. P. S. Sangthong, "Indoor Wireless Sensor Network Localization Using RSSI Based Weighting Algorithm Method.," in *6th International conference on engineering, applied sciences and technology*, Chiang Mai, Thailand, 2020.

[30] S. M. R. L. J. Jondhale, "Trilateration-Based Target L&T Using RSSI. In: Received Signal Strength Based Target Localization and Tracking Using Wireless Sensor Networks.," in *EAI/Springer Innovations in Communication and Computing*, Springer, Cham, 2022, p. 65–96..

[31] C. Z. Q. Y. B. G. SHI Y, "PATTERN-BASED TRILATERATION POSITIONING ALGORITHM WITH LOW COMPUTING COST," *U.P.B. Sci. Bull., Series C*, vol. 85, no. 3, pp. 2286-3540 , 2023.

[32] L. Z. Z. S. Chen P, "Indoor PDR Trajectory Matching by Gyroscope and Accelerometer Signal Sequence without Initial Motion State," *IEEE Sensors Journal*, vol. 23, no. 13, pp. 15128-15139, 2023.

[33] J. K. A. A.-M. S. e. a. Kunhoth, "Indoor positioning and wayfinding systems: a survey," *Human-centric Computing and Information Sciences*, vol. 18, no. 1, 2020.

[34] Y. W. X. W. a. F. W. S. Xu, "Indoor High Precision Positioning System Based on Visible Light Communication and Location Fingerprinting," *Journal of Lightwave Technology*, vol. 41, no. 17, pp. 5564-5576, 2023.

[35] L. A. a. S. Alawad, "A Hybrid Indoor Positioning System Based on Visible Light Communication and Bluetooth RSS Trilateration," *Sensors*, vol. 23, no. 16, p. 7199, 2023.

[36] O. A. A. R. A. R. M. H. M. A. S. Asim Abdullah, "Robust and fast algorithm design for efficient Wi-Fi fingerprinting based indoor positioning systems," *Journal of King Saud University - Computer and Information Sciences,*, vol. 35, no. 8, pp. 1319-1578, 2023.