Building RSSI-based Indoor Positioning Fingerprint Maps using Android-based Coordination.

1st Author Name, 2nd Author Name
School or Faculty
University or Affiliation Name
Email: email1@example.com, email2@example.com

***Abstract*—Indoor positioning systems (IPS) have emerged as a critical technology for location-based applications. Developing IPS system is challenging since technologies for outdoor positioning seem to be limited in indoor environment. Fingerprinting is a technique to build an offline map and compare the current location with it. While fingerprinting remains a popular technique for indoor positioning, its reliance on extensive manual data collection is a significant challenge. These data points can be the Received Signal Strength Indicator (RSSI) of the Wi-Fi signal or signals from the triangulation of Bluetooth/cellular beacons. However, the conventional grid-based fingerprint technique is facing challenges when the target area is being large. This research proposes an automated approach to gathering Wi-Fi RSSI data for building indoor positioning maps using the Android-based triangulated coordination. Our method demonstrates a substantial reduction in data collection time (79%) compared to traditional grid-based techniques. The resulting dataset effectively supports machine learning models for indoor positioning, achieving a Mean Distance Error (MDE) of less than 2 meters different.**

***Index Terms*—Indoor Positioning System (IPS), Received Signal Strength Indicator (RSSI), Triangulated Coordination, and Android Application.**

I. Introduction

Positioning Systems, both indoor and outdoor, are widely used for various use cases such as navigation, asset tracking, and recue robotics. Outdoor positioning systems like Global Positioning System (GPS) and Global Navigation Satellite System (GNSS) are highly effective in providing accurate location data in open environments, such as road, parks, and mountains. However, the outdoor positioning system are often unreliable or unavailable indoor due to the signal attenuation caused by walls and other obstacles.

To overcome this limitation and enable positioning capabilities within closed environment, such as inside buildings, Indoor Positioning Systems (IPS) have been developed to provide accurate location information [1]. These IPS are useful and necessary for complex buidlings and large infrastructure, such as hospital, school campuses, malls, and offices. IPS can enable extensive possiblilities to develop systems that require accurate indoor positions, such as service employee dispatching system or indoor delivery robots.

Since GPS or GNSS signals are inreliable in indoor, IPS require other wireless technologies or signals to replace them. However, there is no one-to-one comparable signals to GPS or GNSS for indoor localization. Potential wireless signals that are usually adopted for IPS are Wireless-Fidelity (Wi-Fi) networks, Bluetooth, Radio-Frequency IDentification (RFID), light-based systems [2], and Ultra-Wide Band (UWB) signals [3], [4]. However, most of the wireless technologies aforementioned require dedicated infrastructure and device installation, e.g., beacons or repeaters. Since almost every building or closed environment nowsday provides internet networks through the installation of Access Point (AP) devices, Wi-Fi signals seem to be the promising resource for IPS as it is cost-effective (i.e., no more hardware installation needed) [5].

Leveraging those wireless technologies, there are various techniques used for IPS, such as trilateration, triangulation, and fingerprinting. Based on the previous work [6]–[9], fingerprinting technique using Wi-Fi Received Signal Strength Indicator (RSSI) proved by researchers as a promising solution for IPS with acceptable accuracy and real-time response time. Conventionally, fingerprinting technique develops the indoor positioning capabilities based on two phases: *Offline* and *Online*. In the offline phase, system development team will separate indoor maps (i.e., floor plans) into grids. We can consider a grid as a small segment of the indoor area to be localized, e.g., every 1 square meter area is a grid. Then, the development team must collect the signal value (e.g., RSSI) at the exact location of each grid. When compiling all the collected data points of grids in the area, the team can build a offline database called a radio map. Then, in the online phase, the development team can either train a machine learning model to estimate or predict the current grid based on the collected data points, or directly compare the current signal value with the existing map.

While the fingerprinting technique reports efficient outcomes for IPS in term of accuracy, the development process of the IPS using this technique seems to be a major drawback [10]. In the offline phase, the buidling of radio maps is time-consuming and requires much effort during the data collection. The development team must collect the exact location of the grid and follow the process pattern to manually collect the signal value to the database. It is obvious that moving and checking the exact location of a grid cause a burden to the data collector and easy to be mistaken.

According to the aforementioned drawback, we consider this process impractical and ineffective in the real-world settings. As a result, this paper aims to propose an automated coordinate approach for data collection and radio map building

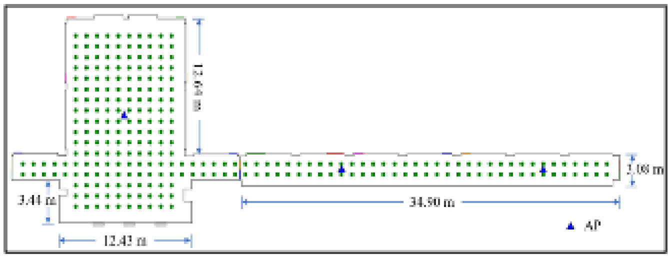

Fig. 1. An example of grid overlays on the floor plan using the grid method. Each green small dot represents the center of each grid on the floor plan.

for IPS using the fingerprinting technique. This work will employ the Fused Location Provider (FLP) module in Android operating system to gather the baseline coordinates (i.e., latitudes, and longitudes) instead of using grids. Then, the proposed approach binds the coordinates with the collected signal values, which is Wi-Fi RSSI values. Our main contributions of this paper is summarized as:

- This paper proposes a coordinate-based approach for building a radio map for the fingerprinting technique that can efficiently enhance the data collection processes during the offline phase.
- This paper proposes an implementation method of the data collection mobile application in Android operating system, following the proposed approach, that can reduce collection time while maintaining the acceptable performance for IPS.
- This paper reports the comparison of the data collection quality compared to the conventional grid approach for fingerprinting.

The remainder of this paper is organized as follows: Section II briefly explains the fundamental knowledge of IPS and the techniques used; Section III describes the proposed coordinate-based approach; Section IV guides through the experiment and evaluation, as well as the result analysis and discussion; and Section V concludes this paper and provide future directions.

II. Preliminaries

A. Data Collection using Grid Method for Fingerprinting IPS

Grid method is the traditional way to collect signal values as data points from the target area. It commences with the requirement of manual effort for measuring and designing the grid patterns of the target area. The grid size needs to be determined prior to the data collection, such as 1×1 square meter or 2×2 square meters. The grid size directly affects the performance and effort of the data collection process. In other words, with the same target area, the smaller grid size the longer it take to collect signal values of the entire target area, but it can be expected to have higher accuracy when it comes to the positioning task. An example of the floor plan that is overlayed with the grid pattern is shown in Figure 1.

Typically, each green point must be marked on the target area. The data collector is required to measure the area and place markers (e.g., stickers) on the floor to identify the green point on the actual location. Then, the data collector must walk to each green point and collect the signal value point-by-point. It can be summarized that the accuracy of the data collection is reversely proportioned by the grid size (e.g., 2 for the 2×2 grid), and the target area in square meters (m^2), defined as:

$$accuracy \propto \frac{1}{grid\_size \times area} \quad (1)$$

B. Fused Location Provider (FLP)

Fused Location Provider or FLP [12] is a location Application Programming Interface (API) in Google Play services. It estimates or predicts current location based on the integration/triangulation of multiple wireless signals, such as GPS, Wi-Fi, and cellular network because there are some situations that Android devices cannot access to some specific signals.

For GPS signals, the FLP API will collect them from at least four GPS satellites and the API will calculate the current distance based on the average Time of Arrival (ToA) [11]. The result of distance calculation is triangulated in order to give the coordinates. This approach of distance calculation is also applied to Wi-Fi and cellular signals as well. They will use the APs and carrier beacons as reference points. In summary, the core concept of the process behind the FLP is the triangulation of distance between available signal transmitters. Then, the FLP API will calculate the averaged coordinates based on different signal sources. It also gives weights to different types of signal transmitters, which is formally defined as:

$$L_{avg} = \frac{\sum_{i=1}^{n}(W_i \times L_i)}{\sum_{i=1}^{n} W_i} \quad (2)$$

where L_{avg} is the averaged location calculated from various sources, W_i is the weight of source i, L_i is the estimated location from source i, and $n \in \mathbb{N}^+$ is the total number of available sources. In most cases, we can set up the weight property as: $0 \leq W_i \leq 1$ and $\sum_{i=1}^{n} W_i = 1$.

The FLP API offers acceptable accuracy and it has been widely-used across various Android application. However, with the FLP API, the application is limited to the availability of signal transmitters and it cannot adapt to some critical situations, such as the unavailable of multiple sources.

C. Wi-Fi Manager for Android Applications

In mobile devices, their operating system is required to have a module to connect with wireless technologies, such as Bluetooth or Wi-Fi internet network. It is essential that this module can work as a connection interface and adaptor to signal transmitters.

For Wi-Fi internet signal, Android operating system along with Google Play services introduces WifiManager [13], which is an API module for faciliate all aspects of Wi-Fi connectivity. This module allows Android applications to retrieve information about nearby Wi-Fi network, including the Basic Service Set IDentifier (BSSID), Service Set IDentifier (SSID), and RSSI values. For instance, an available Wi-Fi access point has an SSID as "John's House" and a BSSID of

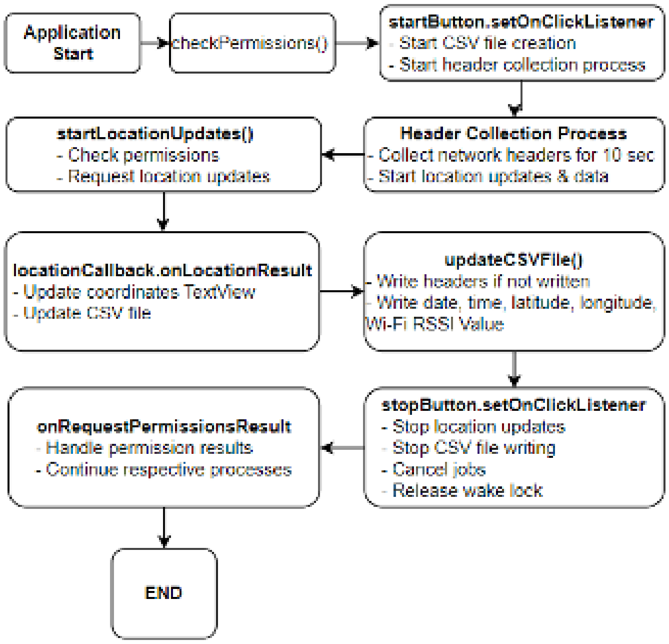

Fig. 2. An overview of the Android application workflow, implementing the proposed approach for automated Wi-Fi RSSI value collection.

"00:1A:2B:3C:4D:5E". Then, the RSSI value is the power strength of the Wi-Fi signal in dBm (decibels milliwatts), which is commonly calculated by the logarithm of the power ration expressed in watts, as follows:

$$RSSI = 10 * \log_{10}(P_{received}) \tag{3}$$

where $P_{received}$ is the power ratio of the received Wi-Fi signals. With this calculation, the WifiManager module can facilitate the collection of RSSI value of each BSSID/SSID on each location. The range of RSSI values is typically varied by the Wi-Fi modem manufacturer. In this paper, an agreed-upon-community RSSI value scheme is used and defined RSSI values in the range between -100 dBm to 0 dBm. The closer to 0 dBm meaning the stronger signal strength received.

III. Automated Coordinate-based Wi-Fi RSSI Value Collecting Approach for IPS

While maintaining the quality of collected data, this paper aims to fasten the collecting time and lessen the effort used during the data collection process for building IPS maps. This paper propose an automated coordinate-based approach for collecting Wi-Fi RSSI values to IPS. The proposed approach consists of three phases: Network Scanning, Data Collecting, and Data Cleansing. An overview of the proposed approach is illustrated in Figure 2. It can be seen that the proposed approach faciliate the development of an Android application to work with FLP and WifiManager modules.

A. Network Scanning Phase

Before the data collection is commenced on different coordinates in the target area, the data collector (in this case, an Android application) must scan for all current available networks based on their unique BSSID, and prepare the overview of the available wireless network in the area. This phase is mandate because it has to be a reference point when RSSI values are collected.

The separation of the network scanning phase is designated to reduce any error occurred during the Input/Output (I/O) processes. This paper recognized from trials of the developed application that bottlenecks can be occurred if the data collector scans for the available network every time the data is collected. Based on the experiment, this paper realized a significant delay, around 5 to 10 seconds, between the network scanning phase and data collecting phase is suggested in order to ensure the readiness of the data collector. There are potentials that the list of available Wi-Fi networks has not been completely prepared before the

B. Data Collecting Phase

Once the available networks are scanned and collected, the data collecing process will be ready for the operation. In this phase, the proposed approach suggests that the data collector must leverage the use of FLP to calculate the current position in terms of latitude and longitude value. However, in the closed environment, the data collector should not completely rely on the coordinates calculated from the FLP because there are possible cases that the WifiManager module in the device has not truly accurate in all cases. Sometimes, it has been found that the module is mulfunctioned and mistaken due to the lack of wireless signal availability.

With the use of the FLP and WifiManager, an Android application can be developed and used to collect data automatically in the target area with a specific interval. To illustrate the process of the "Start location updates" task in the fourth box of Figure 2, execution steps can be made as follows:

1) The data collector's device starts to record timestamp and device ID, and calculate the current cooridnate using the FLP API.
2) The data collector's device starts to collect RSSI values using the WifiManager module.
3) The data collector's device saves the collected data into their own data storage (e.g., writing on a file or pushing to a database).
4) The data collector's device starts a delay timer to the specific interval, e.g., 5 or 10 seconds and the data collector moves to the next location.
5) Initiate a new round of data collection and repeat step 1) to step 4).

After conducting the data collection process following the above-mentioned steps, the data collector's device will gather a set of data points where each of them contains device ID, timestamp, current coordinate (latitude, and longitude), and RSSI values for all available BSSIDs. If we plot a location map presenting the collected data, it will be identical to Figure 3.

C. Data Filtering Phase

As it can be seen in Figure 3, the collected data points are located within the target area. However, some outliners are

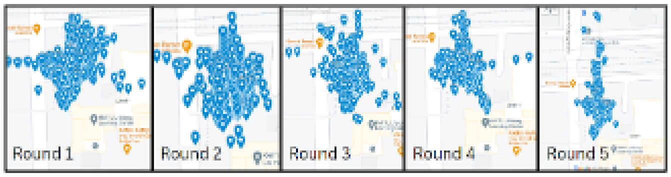

Fig. 3. An example of collected data points using the proposed approach, plotted in the map using the current coordinate from the FLP API.

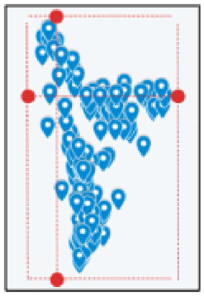

Fig. 4. An overview of the data filtering method proposed in this paper. It shows that the four points, i.e., top-most, bottom-most, left-most, and right-most, are identified and the square area is overlayed the data points on map.

recognized in the plot. For example, multiple data points are located in areas outside the target building in rounds 3 to 5. This situation shows limitations of the data collector's device, affecting the completeness of the collected data.

This paper recognized that the proportion of outliners in the collected data is insignificant (less than 5 percent) and they could be removed from the dataset without any effect to the overall performance of the collected data. This paper proposes a data filtering method in this phase, where outliners is defined as data points that are sparsed outside the predefined area. In most cases, the predefined area will be the target area, i.e., the building-shaped area from the floor plan. The proposed method for data filtering is:

1) The data collector must identify the top-most and bottom-most coordinates that align with the border of the target area. These coordinates will be referred to as Y_{top}, and Y_{bottom}, respectively.
2) The data collector must identify the leftmost and right-most coordinates that aling with the border of the target area. These coordinates will be referred to as X_{left}, and X_{right}, respectively.
3) The data collecter must filter out all data points that beyond the area encapsulated by (X_{left}, Y_{top}, X_{right}, Y_{bottom}) points.

With this data filtering method, outliners are eliminated, remaining only data points that are expected to be in the target area. The data filtering pane following the proposed method can be shown as in Figure 4 and the end result of the collected data plots from different rounds after data filtering is depicted in Figure 5.

With the proposed approach, coordinates are successfully collected with RSSI values across the target area. With this

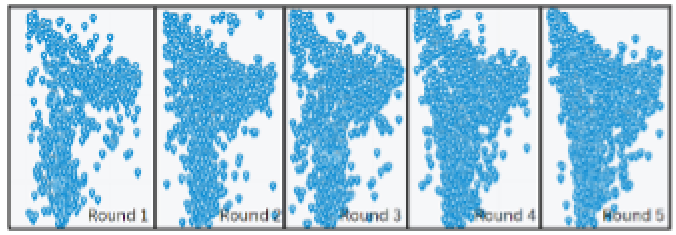

Fig. 5. An example of collected and *filtered* data points using the proposed approach from five different rounds, plotted in the map using the current cooridnate from the FLP API. This shows the raw data points after the data filtering method is applied.

approach, it can be implemented as an Android application, which data collectors can install on their mobile phone. The application enable the automatic data collection capability where data collectors can run the operation and work around the target area, without any concerns on the grid location.

IV. Experiments and Evaluation

In order to prove that the proposed approach is efficient to provide quality datasets, and require less effort and time compared to other approach, this paper sets up two experiments to ensure both the proposed approach and its outcomes systematically.

A. Experimental Setup

For both experiments, this paper will use the same experimental setup for data collection, but the proposed approach will be evaluated from different aspects. In this section, the experimental setup we used will be explained in three parts: target area, data collection procedures, and device specification.

1) Target Area: To set the same test site, this paper select a floor of a campus building in a university as a target area. The floor area is around 120 square meters and contains various rooms and partitions. However, this paper scoped down the data collection to only the hallway to prove the concept first. An example of the floor plan of the target area for the experiments is shown in Figure 6.

2) Device Specification: Since the FLP API and WifiManager modules are featured in Androind operating system on mobile phone, the device specification should be factors that affects the capability to extract the current coordinate. As mentioned in Section III-C, the FLP API module in the device might be mulfunctioned and cause incorrect coordinates (i.e., outliners). To ensure that the experiments control variations of devices implementing the proposed approach, this paper prepare 5 devices with three groups of specification. The devices used in the experiments are as follows:

- *Group 1:* Three Android phone with Mediatek MT6769Z Helio G85 processor.
- *Group 2:* One Android phone with Qualcomm SM6225 Snapdragon 685 processor.
- *Group 3:* One Android phone with Qualcomm SM7150 Snapdragon 732G processor.

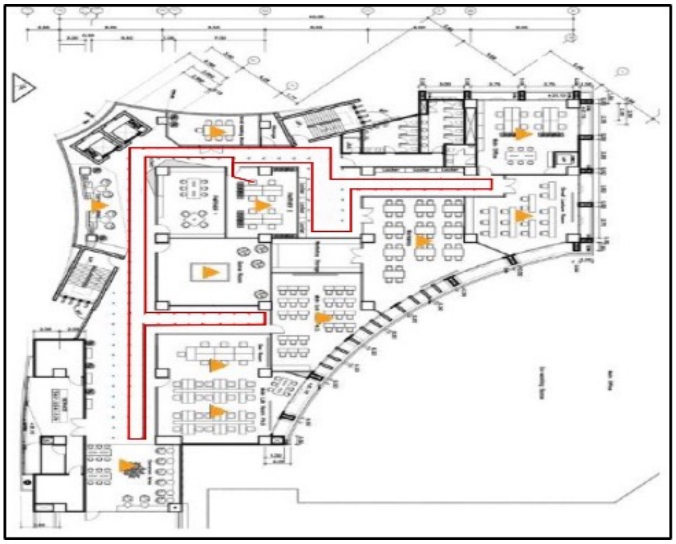

Fig. 6. A floor plan of the target area for testing the proposed approach, overlayed with a red-line route that indicates the data collection procedure.

We believe that these differences in device specification can cover the device landscape and overcome the hardware and module limitations.

3) Data Collection Procedures: There are two data collection procedures defined for the two experiements.

1) This experiment aims to compare the data collecting time between the conventional grid method and the proposed coordinate method. To make a fair comparison, we segmented the target area into a small $3times3$ square area. This area segment is separated into 9 of 1×1 square meter grid. Firstly, for the grid method, the application is implemented to collect timestamp and RSSI values of the exact grid. The data collector will walk from the top-left grid towards the right and ends with the bottom-right grid. Secondly, for the coordinate method, we set a round to slowly walk in the same path for the grid method.
2) This experiment aims for evaluating the quality of collected data in the task of IPS. Since the proposed approach does not require any control to collect data on the exact location of a grid, the data collection procedure can be set as a walking route to collect the data in the target area. To make it clear, this paper illustrates the walking route for data collection and overlays it on the floor plan, as shown in Figure 6. We conduct the experiment in 5 rounds and walk through the same route in every round with all devices used.

B. Experimental Results

To report the experimental results and evaluate the efficiency of the proposed method, this paper conducted two experiments as procedures mentioned earlier. The result of each experiment is elaborated below.

Experiment 1: Number of Data Points Gathered and Average Time Taken – We commenced with the collection of data points within the same time range. We controlled the time used by the grid and coordinate methods. Then, we controlled the data collection route to the same and determine the average time taken for the two methods. The summary of the experimental results is reported in Table I. It can be seen that, with the same collecting time, the proposed coordinate method could gather a double of data points compared to the grid method. On the other hand, when the collecting area and route are controlled, the time taken for the proposed coordinate method is around 79% less than the grid method.

Experiment 2: Accuracy when data is used in training machine learning models – We conducted this experiment by training five conventional machine learning models for IPS, including random forest, decision tree, linear regression, gradient boosting, and K-nearest neighbors, with two different datasets: One is collected by the grid method, while another by the proposed coordinate method. The grid identifier and the coordinate are used as an output label and the RSSI values are the input value to the model. The performance in term of IPS accurcy is measured by the Mean Distance Error (MDE) through 10-fold cross validation. The experimental results is visualized as a collection of bar graphs in Figure 7. It can be seen that the data collected by both method works best with the K-Nearest Neighbors (KNN) model. However, the proposed coordinate method should a bit lower in accuracy on all models. The MDE of the data collected by the proposed method is mostly around 2 meters more than the one collected by the grid method.

TABLE I

A SUMMARY OF EXPERIMENTAL RESULTS ON THE NUMBER OF DATA POINTS GATHERED AND AVERAGE TIME TAKEN, COMPARING THE GRID AND COORDINATE METHODS

Variable	Grid	Coordinate
Number of RSSI value data points gathered (when collecting time is controlled)	4,895 points	10,161 points
Time taken (when collecting area and route are controlled)	∼144 minutes	∼32 minutes

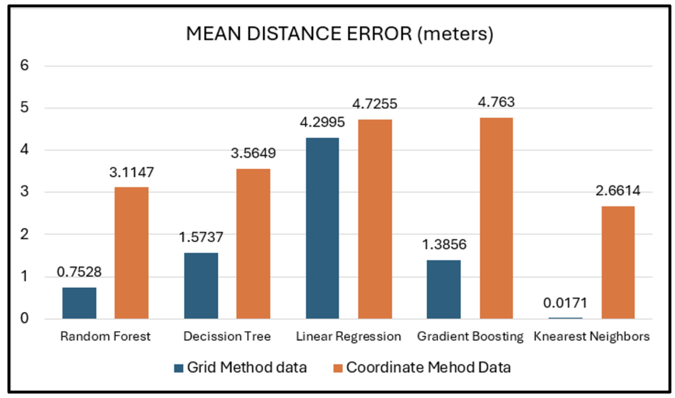

Fig. 7. A summary of experimental results on the data quality in term of accuracy of IPS, measured by the mean distance error.

C. Discussion

From the experiments and evaluation, it clearly shows both benefits and limitations of the proposed approach for collecting data for IPS. It can be realized that, with the use of the triangulated model of the FLP API module, it greatly facilitate the automatic data collection of RSSI values in different location. Instead of putting a strict concern on grid alignment to data collectors, the proposed approach lowers the restriction and allows data collection to access larger amount of RSSI value data points and reduce data collection time taken. The double in the number of data points gathered and the approximately 79% less collecting time highlights the great benefit to the development team. This benefit does not just mean to lessen the overall time taken for IPS development, but it opens an opportunity to the development team to more data, which can help developing more comprehensive and accurate IPS. To be specific, the use of deep learning model that works well with large datasets might be possible for IPS. Another advantages of the proposed approach is the less effort in preparation activities. With the grid method, the IPS development team and data collector are mandate to put much effort in analysis of the target area, such as analyzing the available area to collect data or analyzing the optimal grid size that maximize the IPS accuracy. The proposed approach offers zero time required to the above-mentioned tasks.

Contrastingly, the proposed approach also comes with some drawbacks and limitations. It can be found from the experiments that the data collected by the propaoed approach has larger MDE compared to the data collected by the grid method. Overall, while the grid method provides better accuracy as reflected in the lower MDE, its practical limitations in terms of mobility and the high demands it places on the data collector must be carefully considered. The proposed coordinate method, although less accurate, offers a more efficient and adaptable approach, particularly for applications involving large or complex environments. The choice between these methods ultimately depends on the specific requirements of the project, including the balance between accuracy and practicality. Another drawback is the reliance on the FLP API module on the device. It shows from the map plots that the mulfunction of this module generates some outliners. However, this drawback can be easily overcame by the control of the device specification.

With the above discussion, we found that benefits of the proposed approach could be significant advantages for IPS development while limitations are acceptable and resolvable.

V. Conclusion and Future Work

This paper proposes an automated approach for collecting RSSI value data for IPS development task. The proposed approach leverages the FLP API and WifiManager modules to facilitate the data collection task easily, which could help the IPS development team to develop an Android application for data collection. The proposed approach shows significantly less effort and time taken than the traditional grid method. However, the proposed approach reported higher MDE, but it rathers acceptable and need to be carefully considered based on the specific project requirements.

For the future work, we planned to improve the quality of the data collected by the proposed approach by conducting deeply data analysis and fine-tuning the positioning model to fit with the collected data. Also, the proposed approach could be studied with different context and aspects of machine learning to leverage the full potential of the collected data.

References

[1] F. Y. M. Adiyatma, S. Sunimit, T. Chokporntaveesuk, K. Lualum, N. Chaisang, and P. Cherntanomwong, "Hierarchical KNN for Smartphone-based 3D Indoor Positioning," in *Proc. 39th Int. Technical Conf. on Circuits/Systems, Computers, and Communications (ITC-CSCC 2024)*, 2-5 July 2024, Okinawa, Japan.

[2] M. Maheepala, A. Z. Kouzani and M. A. Joordens, "Light-Based Indoor Positioning Systems: A Review," IEEE SENSORS JOURNAL, vol. 20, No. 8, April 15, 2020. DOI: 10.1109/JSEN.2020.2964380.

[3] P. Dabove, V. D. Pietra, M. Piras, A. A. Jabbar and S. A. Kazim, "Indoor positioning using Ultra-Wide Band (UWB) technologies: positioning accuracies and sensors' performances," in *Proc. 2018 IEEE/ION Position, Location and Navigation Symp. (PLANS)*, 23-26 April 2018, DOI: 10.1109/PLANS.2018.8373379.

[4] N. Tyagi, J. Singh and S. Singh, "A Review of Indoor Positioning System: Technologies and Applications," in *Proc. 2022 Int. Conf. on Data Analytics for Business and Industry (ICDABI)*, 25-26 October 2022, DOI: 10.1109/ICDABI56818.2022.10041504.

[5] W. Joanne, S. Taking, N. Isa and K. Chao, "Indoor Navigation and Localisation Application System," in *Proc. 2016 3rd International Conference on Electronic Design (ICED)*, 11-12 August 2016, DOI: 10.1109/ICED.2016.7804661.

[6] N. Agah, B. Evans, X. Meng and H. Xu, "A Local Machine Learning Approach for Fingerprint-based Indoor Localization," in *Proc. IEEE SOUTHEASTCON 2023*,01-16 April 2023, DOI: 10.1109/SoutheastCon51012.2023.10115169.

[7] L. Mainetti, L. Patrono and I. Sergi, "A Survey on Indoor Positioning Systems," in *Proc. 2014 22nd Int. Conf. on Software, Telecom. and Computer Networks (SoftCOM)*, 17-19 September 2014, DOI: 10.1109/SOFTCOM.2014.7039067.

[8] L. Bastistic and M. Tomic, "Overview Of Indoor Positioning System Technologies," in *Proc. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 21-25 May 2018, DOI: 10.23919/MIPRO.2018.8400090.

[9] Y. Tao, L. Zhao, Q. Zhang and Z. Chen, "Wi-Fi Fingerprint Database Refinement Method and Performance Analysis," in *Proc. 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 22-23 March 2018, DOI: 10.1109/UPINLBS.2018.8559772.

[10] J. Sangthong, "The Indoor Navigation Using Mapping Technique Based On Signal Strength Difference," in *Proc. 2018 21st Int. Symp. on Wireless Personal Multimedia Communications (WPMC)*, Chiang Rai, Thailand, 2018, pp. 284-287, doi: 10.1109/WPMC.2018.8713065.

[11] B. Gurram and N. Giri, "Improving localization accuracy of android's Fused Location Provider API using Kalman Filter," in *Proc. 2016 Int. Conf. on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2016, pp. 1-4, doi: 10.1109/ICCCI.2016.7479935.

[12] Google Play, "Simple, battery-efficient location API for Android," Accessed on Aug 19, 2024. [Online] Available: https://developers.google.com/location-context/fused-location-provider.

[13] Google Play, "Google WifiManager," Accessed on Aug 19, 2024. [Online]. Available: https://developer.android.com/reference/android/net/wifi/WifiManager.