Journal of Informatics and Web Engineering

Vol. 3 No. 2 (June 2024) eISSN: 2821-370X

Real Time 3D Internal Building Directory Map

Zi Yang Chia[1], Pey Yun Goh[1] *

[1] Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia

**corresponding author: (pygoh@mmu.edu.my, ORCiD: 0000-0003-2060-3223)*

Abstract - Global Positioning System (GPS) is a famous technology around the world in identifying the real time precise location of any object with the assistance of satellites. The most common application of GPS is the use of outdoor maps. GPS offers efficient, scalable and cost-effective location services. However, this technology is not reliable when the position is in an indoor environment. The signal is very weak or totally lost due to signal attenuation and multipath effects. Among the indoor positioning technologies, WLAN is the most convenient and cost effective. In recent research, machine learning algorithms have become popular and utilized in wireless indoor positioning to achieve better performance. In this paper, different machine learning algorithms are employed to classify different positions in the real-world environment (e.g., Ixora Apartment - House and Multimedia University Malacca – FIST building). Received Signal Strength Indication (RSSI) is collected at each reference point. This data is then used to train the model with hyperparameter tuning. Based on the experiment result, Random Forest achieved 82% accuracy in Ixora Apartment and 84% accuracy in one of the buildings in Multimedia University Malacca. These results outperformed the other models, i.e., K-Nearest Neighbors (KNN) and Support Vector Machine (SVM).

Keywords— Indoor Positioning, Wi-Fi, KNN, SVM, Random Forest

Received: 13 June 2023; Accepted: 18 December 2023; Published: 16 June 2024

I. INTRODUCTION

With the widespread usage of wireless networks, location-based services have integrated into our daily lives. Nowadays, there are two main categories for navigation systems. These categories are indoor (inside a building) and outdoor (outside a building with open air commonly). Global Positioning System (GPS) is well known and one of the outdoor navigation techniques which use the satellite to give accurate position of objects. However, this technique only works effectively in open areas with a direct line of sight to the satellites. GPS signals will be interfered or blocked due to the building structure such as walls, roof, and others. Therefore, the indoor location cannot be determined and signal is lost inside the building. In other words, GPS is not applicable when indoor positioning is needed.

Although GPS is not applicable, the advancement of technology has made indoor positioning available. These technologies have been proposed to implement indoor positioning, i.e., Radio Frequency Identification (RFID), Wi-Fi, Bluetooth and ultra-wide band (UWB). Not just positioning, these technologies support indoor navigation as well. This allowed the development of indoor maps which ease the human life in airports, parking garages, alleys, underground locations or inside the multi-storey buildings. With the indoor positioning technology, passengers can receive a customized route with guided directions to the required destination by using a mobile phone. This reduces unnecessary travel time and enhances the passenger experience. Among all the technologies for indoor positioning (IPS), Wi-Fi is preferred because no additional change on the existing infrastructure and hardware requirements. Besides that, Wi-Fi is commonly available in public areas. However, capturing real-time indoor position signals is a

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2024.3.2.3

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

great challenge [1]. Thus, machine learning is embedded in the solution with the assumption that it improves the accuracy of real time position.

In this paper, the main aim is to develop an indoor virtual map where real-time positioning is embedded. Then, the performance of the real-time positioning with machine learning is studied. Few environments such as some specific areas of Faculty of Information Science and Technology (FIST) in Multimedia University (MMU) Melaka and rooms of the hostel are used to test the efficacy of the application.

II. LITERATURE REVIEW

Various technologies are used for indoor positioning. These technologies are explained and described in the following subsections and comparisons are done.

A. Bluetooth

Bluetooth is a short-range wireless communication technology that is used for indoor localization. Beacons are the major components during the implementation of Bluetooth-based localization to help in transmitting a continuous signal that other devices can see [2]. Besides, the compact size of the beacon device allows for effortless installation and wearability [3]. However, due to the short transmission distance, the distance between the devices must be within 10-30 meters and it can determine accurately the position up to 3 meters.

Zuo et al. [4] presented a method to estimate the indoor positioning using beacons and graph optimization. In the case of dense beacon situation and sparse beacon situation, the mean errors are 1.27m and 2.26m respectively. The proposed approach has showed that the beacons' positioning performance is considered good compared with many indoor positioning applications. Besides, this paper also stated that some important factors that will affect the distance estimation, including the effects of walking person in the environments, the upper bound error, the numbers of beacons needed and others.

With the development of the technology, Bluetooth has also come with the latest version which is called BLE. It is supported by the majority of today's smartphone and only requires low energy to function. However, BLE may be interfered when it was installed in a lot of Wi-Fi signal area because they share the spectrum and both use the 2.4 GHz frequency [5]. Researchers [6] published a paper that they estimate the distance based on the RSSI values and transmitting power value from BLE tags. Finally, the results indicates that the range of the distance estimation's accuracy is around 4 meters.

B. Radio-Frequency Identification

An RFID indoor positioning system comprises two components which are RFID tags as a receiver to contain the data and an RF based system as an interrogator to read the receiver's data. Since the RFID tags do not require an energy source, hence it can reduce the cost of maintenance. However, the limitation of the RFID is unstable Received Signal Strength (RSS). Hence, currently it is not mature enough to use RFID technique alone to develop an indoor positioning system [7].

Wang et al. [8] have designed a system for indoor real-time location combined with Kinect and active RFID. Domingo et al. [9] have also proved that the Kinect was incredibly useful for locating indoor positioning of public buildings. According to the experimental findings, the proposed system may be used to create an indoor positioning system in terms of accuracy and stability. For the hardware requirements, the proposed system only required one RFID Reader, one Kinect, and some Tags and Repeaters in the entire process. Thus, an indoor positioning system's expenses can be reduced effectively. However, signal loss tolerance and indoor layout must be tested before implementing the positioning. Therefore, it may be a challenge when the experimental environment is wide [8].

C. Wi-Fi

Wi-Fi-based indoor positioning system for determining indoor location information has become a prominent tool for various reasons. Nowadays, Wi-Fi access points are installed ubiquitously. By using existing wireless access points, it can lower the cost of adopting it on a bigger scale. In addition, Wi-Fi has a wider coverage area than other techniques and can simply penetrate through opaque things. However, it is challenging to apply in a three-dimensional environment and lower positioning accuracy especially for the moving objects.

In order to implement indoor positioning using Wi-Fi, it mainly includes several methods such as Fingerprinting, Triangulation, Multilateration and Trilateration. To start with, Fingerprinting methods usually require a database with information about the locations with which this data can be compared. Then, the current location will be estimated from the similarity between the scenes and environments. Lee et al. [10] have proposed an indoor location detection system based on Wi-Fi fingerprints and Random Forest for improved position accuracy. Based on the experimental result, it obtained the most accurate result when the learning time was 20 but did not achieve 100% accuracy despite the fact that this study was carried out in an enclosed environment with no walls. Therefore, the authors have mentioned that the Wi-Fi noise or signal attenuation is an important element that need to be considered to achieve a high position result.

Another widely used algorithm is Triangulation which is a technique for determining a position that employs the observed angles from two reference points and known distance between two measurement devices [11]. The angel-side-angle triangle congruency theorem is then used to locate their intersecting point, which is calculated as the localization necessary location.

On the contrary, Trilateration identifies a location by measuring the distance rather than angles. This method utilizes the intersection point generated by a minimum of three access points to determine the location of the object. The distances usually are estimated using a variety of signal measurement techniques such as Time Difference of Arrival (TDoA), Time of Arrival (ToA), Received Signal Strength (RSS) and etc. Rusli et al. [12] have proposed an enhanced Wi-Fi Trilateration-based indoor positioning system method. The reference point (RP) was utilised by the authors to increase the accuracy of the estimated position in an environment where obstacles obstruct the signal between device and access point. It is possible to lower the error rate caused by signal interference by utilising the distances from the reference point to each AP. For the testing, two different experiments have been conducted to evaluate the positioning results.

Table 1. Summary of the Positioning Result [12]

Test	Exact Location. (x,y) in meters	Observed location (x,y) in meters	Error in positioning (x,y) in (meters)
Test 1 (Trilateration Technique)	(1, 2)	(-1.35, 3.21)	(2.35, 1.21)
Test 2 (Proposed Model)	(1, 2)	(-0.17, 2.6)	(1.17, 0.6)

Based on the Table 1 above, it can be inferred that the proposed model is beneficial in reducing the positioning errors compared to merely the Trilateration algorithm. Besides, the proposed model may be improved by performing further experiments and testing [12].

Multilateration is also a type of localization technique that involves calculating the time difference of arrival (TDOA) of a signal transmitted by the object to at least three receivers [13]. In 2021, a combination of Kalman filter and Multilateration indoor localization system was presented. By combining two methods, the results can reach an average error of 2.33m as presented in Table 2 [14].

The authors have mentioned that the main purpose of using Multilateration technique is because it does not need external hardware, power consumption and reduces costs. For future improvement, it will involve using other techniques such as Fingerprinting to implement a hybrid indoor positioning system to get a better accuracy [14].

Besides, Abishek et al. [15] have presented a study by using the machine learning algorithms to analyse the obtained receive signal strength intensity (RSSI). By changing the different machine learning algorithms in the experiments, the result shows that ANN and KNN works well compared with the SVM which can obtain the mean accuracy up to 74.72% and 74.52%. Moreover, the authors have pointed out that the precision can be enhanced by adding more hidden layers and adjusting the parameters.

Table 2. System Position Average Error (in Meters) [14]

	POS1	POS2	POS3	POS4	POS5	Mean
MLT	2.83	3.27	1.90	1.95	2.92	**2.57**
MLT+KF	2.50	2.97	1.85	1.61	2.72	**2.33**

D. UWB

Recently, Ultra-wideband technology (UWB) has been adopted and taken into consideration to develop an indoor positioning system. Similar to Bluetooth and Wi-Fi, Ultra-wideband is a radio-wave-based wireless communication protocol [16]. The advantages of UWB in indoor positioning system is that signal bandwidth is related directly to the estimation of the distance and UWB only uses low power and high bandwidth (>500MHz) [17]. Therefore, it can be proved that this technology is applicable for precise indoor positioning especially for environments where multipath is high.

Tian et al. [18] have proposed an IPS using Ultra-wideband (UWB) technology which can obtain centimetre-level positioning accuracy in complex multipath environments. First, the tag's distance from each base station will be measured by using UWB. The TDOA position algorithm will assist to get the original position of the tag. Then, a sliding window filtering is applied to filter the data in order to obtain more accurate positioning results. Experiment has been conducted on a stroller in a shopping mall as shown in Figure 1. Finally, the result shows that the positioning error can be reduced within a range of 20 cm. However, the limitations of the UWB are the requirement for dedicated infrastructure with all related problems including coverage areas and costs [19].

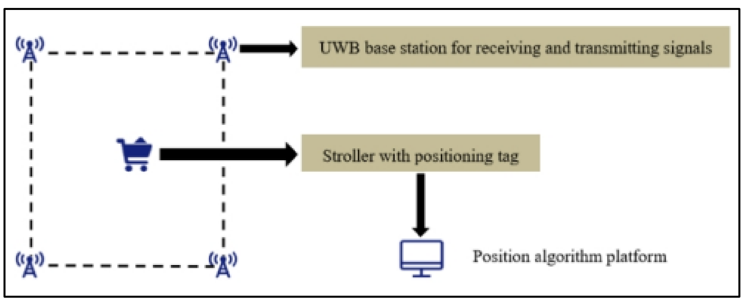


Figure 1. Design of UWB Positioning System [19]

E. Summary

The Table 3 above presents a summary of different technologies and their characteristics. Based on the table above, GPS does not require any infrastructure to implement and offers an accuracy of less than 20 meters. However, GPS is primarily used for outdoor positioning and needs a high amount of energy. On the contrary, Bluetooth (BLE Beacons) supports both outdoor and indoor positioning with an accuracy of less than 15 meters. The consumption of energy is moderate but requires some infrastructure and hardware to implement such as BLE beacons and smartphone. Similarly with the Bluetooth (BLE Beacons), RFID and UWB support both outdoor and indoor positioning. Although less energy consumption is needed, both of them require specific infrastructure and hardware. Compared with other techniques, Wi-Fi enables the system use of existing Wi-Fi network infrastructure, high environmental adaptability and low cost [20]. It provides an accuracy of less than 15 meters and less energy consumption is needed. Apart from that, Wi-Fi indoor positioning system works well with other algorithm to improve the accuracy and flexibility and each smartphone is having a Wi-Fi module [21]. Using this pre-existing research as a stepping stone, Wi-Fi was selected as the ideal technique to build an indoor positioning system. Although Wi-Fi is user convenient and adaptable to the environment, a lot of challenges such as complex algorithm is needed, prone to noise still remain unsolved [22]. Therefore, machine learning algorithms will be applied to address the issues and improve the accuracy of the positioning system.

Table 3. Comparison with Different Technologies [23]

	GPS	Bluetooth (BLE Beacons)	Wi-Fi	RFID	UWB
Outdoor & Indoor Positioning	Outdoor only	Outdoor & Indoor	Outdoor & Indoor	Outdoor & Indoor	Outdoor & Indoor
Consumption of Energy	High	Medium	Low	Low	Low
Accuracy of Positioning	< 20m	< 15m	< 15m	<2m	<1m
Infrastructure	No required	Required	Required (existing infrastructure)	Required	Required
Hardware	Smartphone	BLE beacons and smartphone	Wi-Fi access point and smartphone	RFID tags and RFID tag readers	UWB tags and UWB tag reader

III. SYSTEM DESIGN

A. Interface Design

The proposed system is developed utilizing the Unity platform with the aims of assisting new visitors or students to find the location in an indoor environment. In the proposed system, it contains 3 scenes which are Homepage (Figure 2), Submit Location (Figure 3) and Main (Figure 4).

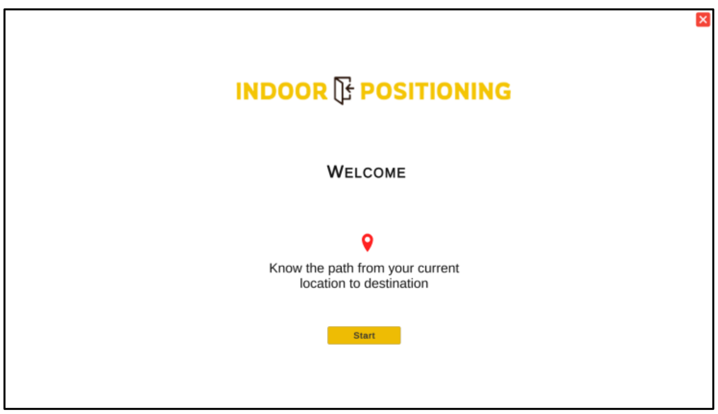


Figure 2. Homepage

Figure 3. Submit Location Scene

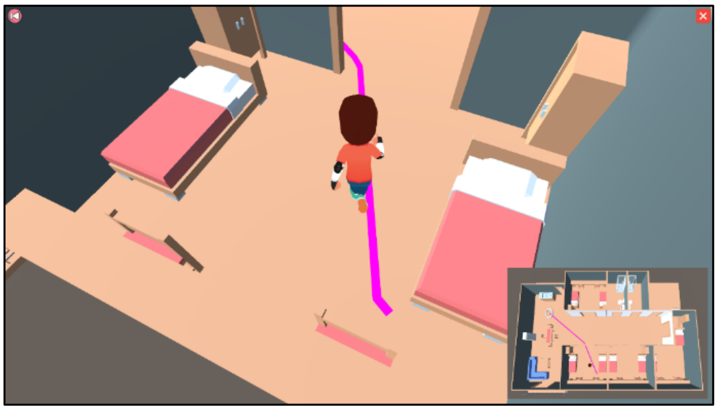


Figure 4. Main Scene

B. Flow Chart / Activity Diagram

Based on the Flow Chart in Figure 5, the proposed system will consist of two phases which are the offline phase and online phase. For the offline phase, it is required to define the indoor environments then collect the signal strength data on a device. Then, the data will be sent and saved into the Fingerprint data store. Next, the machine learning algorithms such as Random Forests, Support Vector Machines, and K-Nearest Neighbors will be used to learn and train a model based on the collected fingerprint data. Since the Wi-Fi fingerprints are equivalent across all the devices, it is more convenient and efficient to perform the learning process.

For the online phase, users are initially required to select their desired destination from the system. Once the user clicks the "Start" button, the system will collect the real-time RSSI data of the user's current position. During the positioning process, the system will use real-time RSSI as input data then make the classification by using the trained model to obtain the positioning result. Once the location result is obtained, the destination will be set and an animated tool guide will start to demo how to walk from current position to reach the destination. Lastly, the user can either continue to navigate another location or stop the program.

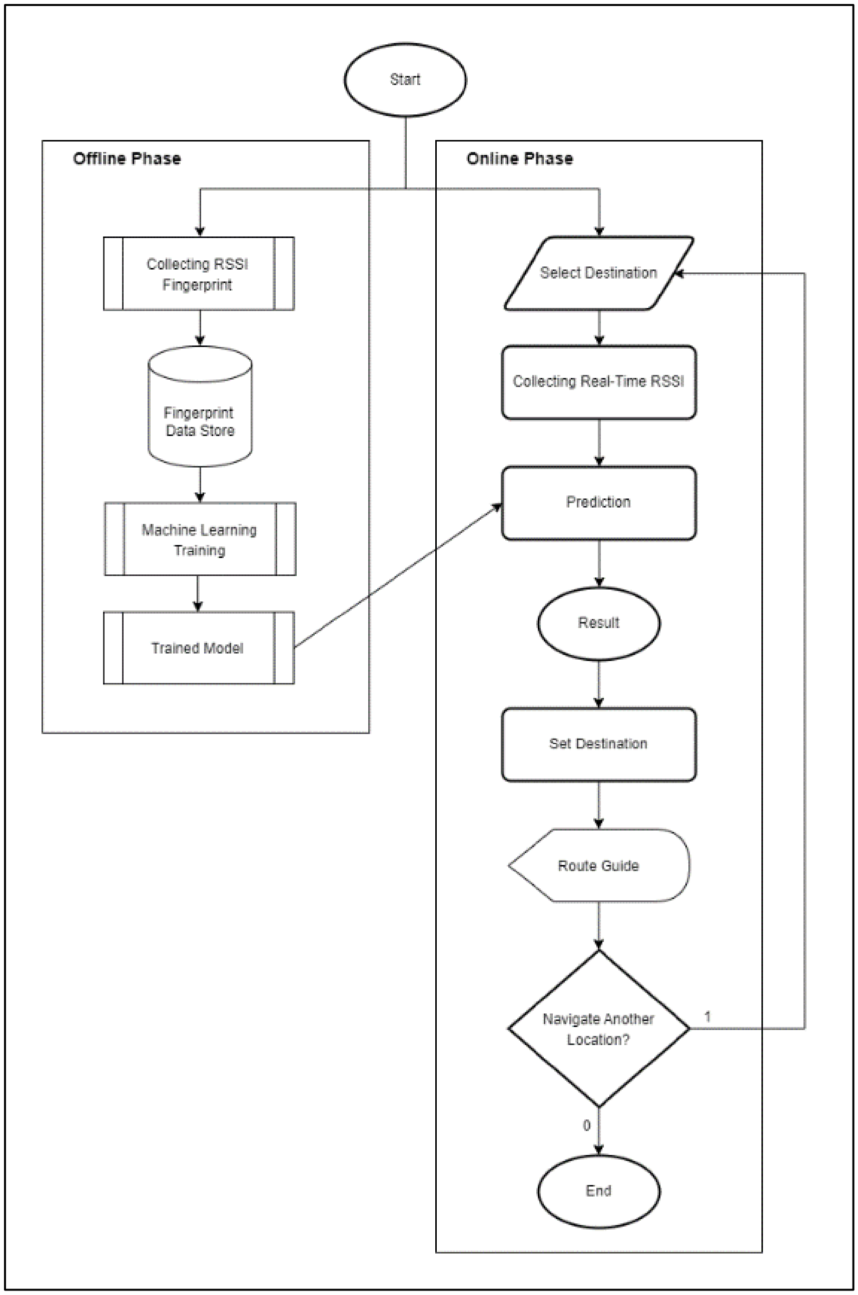


Figure 5. Flow Chart for the System

IV. IMPLEMENTATION

A. Experimental Environment

In this paper, two experiments will be conducted in a house in Ixora Apartment, Malacca and a building in Multimedia University Malacca (MMU). These two scenarios are typical indoor situations. In the first experiment scenario, the house consisted of 1 living room, 4 bedrooms, 2 toilets, 1 balcony and 1 yard. Figure 6 showed the experimental area in the first scenario. The blue stars represent the reference points (RP) and the router icons represent the position of the AP. Besides, this Figure 6 also will include the other elements of the indoor environment which can be understood according to the legend.

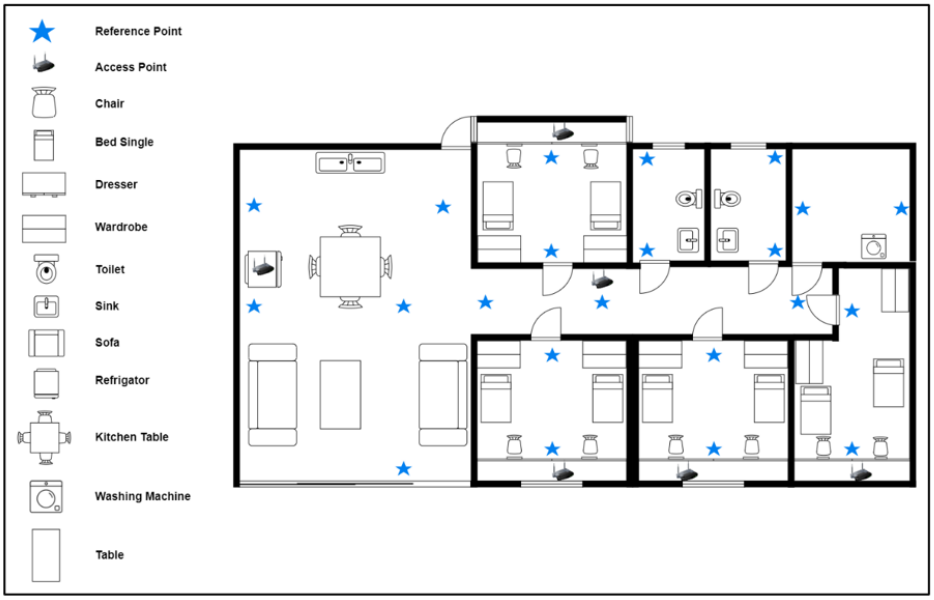


Figure 6. The Layout of the Experiment Area for Scenario 1

For the second experiment scenario, FIST building has been selected to evaluate the performance of the proposed method. The FIST building is a three-story building. For the testing purposes, only one floor will be selected as the experimental environment. 6 access points are distributed in the second experiment scenario. The layout of scenario 2 has been presented in Figure 7.

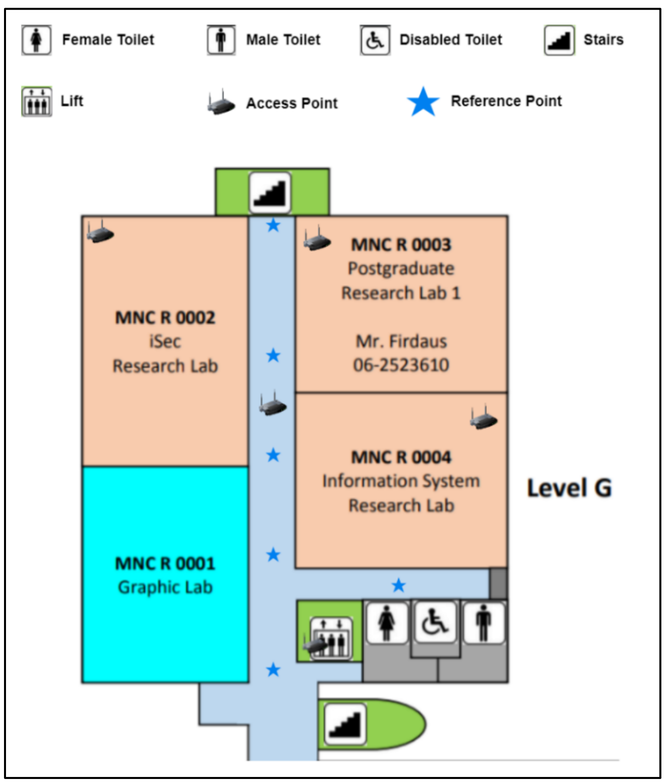
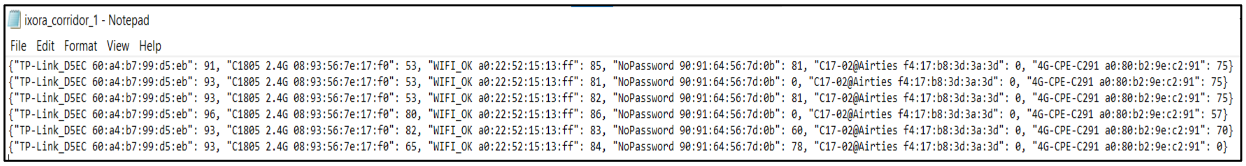


Figure 7. The Layout of the Experiment Area for Scenario 2

B. Data Collection

To collect the data, a Python library developed by kootenpv called “whereami” is used and installed on an Acer Nitro 5. This library has been specifically designed to gather fingerprint data based on the device's location. During a Wi-Fi-enabled device scans nearby access points, it will receive a unique identifier of the access point as well as a signal strength that correlates with distance to the access point. Therefore, the collected inertial data will comprise the SSID, MAC address and corresponding signal strength. Then, these data are systematically saved into individual text files with each text file representing a corresponding location. The Figure 8 and 9 show examples of data collection on different locations.



ixora_corridor_1 - Notepad

File Edit Format View Help

```
{"TP-Link_D5EC 60:a4:b7:99:d5:eb": 91, "C1805 2.4G 08:93:56:7e:17:f0": 53, "WIFI_OK a0:22:52:15:13:ff": 85, "NoPassword 90:91:64:56:7d:0b": 81, "C17-02@Airties f4:17:b8:3d:3a:3d": 0, "4G-CPE-C291 a0:80:b2:9e:c2:91": 75}
{"TP-Link_D5EC 60:a4:b7:99:d5:eb": 93, "C1805 2.4G 08:93:56:7e:17:f0": 53, "WIFI_OK a0:22:52:15:13:ff": 81, "NoPassword 90:91:64:56:7d:0b": 0, "C17-02@Airties f4:17:b8:3d:3a:3d": 0, "4G-CPE-C291 a0:80:b2:9e:c2:91": 75}
{"TP-Link_D5EC 60:a4:b7:99:d5:eb": 93, "C1805 2.4G 08:93:56:7e:17:f0": 53, "WIFI_OK a0:22:52:15:13:ff": 82, "NoPassword 90:91:64:56:7d:0b": 81, "C17-02@Airties f4:17:b8:3d:3a:3d": 0, "4G-CPE-C291 a0:80:b2:9e:c2:91": 75}
{"TP-Link_D5EC 60:a4:b7:99:d5:eb": 96, "C1805 2.4G 08:93:56:7e:17:f0": 80, "WIFI_OK a0:22:52:15:13:ff": 86, "NoPassword 90:91:64:56:7d:0b": 0, "C17-02@Airties f4:17:b8:3d:3a:3d": 0, "4G-CPE-C291 a0:80:b2:9e:c2:91": 57}
{"TP-Link_D5EC 60:a4:b7:99:d5:eb": 93, "C1805 2.4G 08:93:56:7e:17:f0": 82, "WIFI_OK a0:22:52:15:13:ff": 83, "NoPassword 90:91:64:56:7d:0b": 60, "C17-02@Airties f4:17:b8:3d:3a:3d": 0, "4G-CPE-C291 a0:80:b2:9e:c2:91": 70}
{"TP-Link_D5EC 60:a4:b7:99:d5:eb": 93, "C1805 2.4G 08:93:56:7e:17:f0": 65, "WIFI_OK a0:22:52:15:13:ff": 84, "NoPassword 90:91:64:56:7d:0b": 78, "C17-02@Airties f4:17:b8:3d:3a:3d": 0, "4G-CPE-C291 a0:80:b2:9e:c2:91": 0}
```

Figure 8. Data of RSSI Values from Different Wi-Fi Access Point (Ixora)

Name	Date modified	Type	Size
ixora_corridor_1	18/5/2023 1:47 AM	Text Document	2 KB
ixora_corridor_2	18/5/2023 1:47 AM	Text Document	2 KB
ixora_corridor_3	18/5/2023 4:08 AM	Text Document	3 KB
ixora_kitchen_1	18/5/2023 1:47 AM	Text Document	3 KB
ixora_kitchen_2	18/5/2023 1:47 AM	Text Document	4 KB
ixora_livingroom_1	18/5/2023 4:08 AM	Text Document	5 KB
ixora_livingroom_2	21/5/2023 3:35 AM	Text Document	5 KB
ixora_livingroom_3	21/5/2023 3:35 AM	Text Document	5 KB
ixora_room1_1	14/5/2023 12:09 PM	Text Document	2 KB
ixora_room1_3	14/5/2023 12:09 PM	Text Document	2 KB
ixora_room2_1	15/5/2023 3:15 AM	Text Document	2 KB
ixora_room2_3	15/5/2023 3:01 AM	Text Document	4 KB
ixora_room3_1	15/5/2023 3:24 AM	Text Document	2 KB
ixora_room3_3	15/5/2023 3:24 AM	Text Document	2 KB
ixora_room4_1	15/5/2023 3:01 AM	Text Document	4 KB
ixora_room4_3	15/5/2023 3:01 AM	Text Document	4 KB
ixora_toilet1_1	15/5/2023 3:02 AM	Text Document	4 KB
ixora_toilet1_2	15/5/2023 3:02 AM	Text Document	3 KB
ixora_toilet2_1	15/5/2023 3:02 AM	Text Document	4 KB
ixora_toilet2_2	15/5/2023 3:02 AM	Text Document	4 KB
ixora_yard_1	21/5/2023 3:31 AM	Text Document	3 KB
ixora_yard_2	21/5/2023 3:31 AM	Text Document	4 KB

Figure 9. Different Locations and Corresponding Inertial Data (Ixora)

C. Data Cleaning

After performing the data collection, removing duplicates is essential steps to minimize the bias and inaccuracies during the analysis and modelling process. To address this requirement, the following python code snippet is used to remove duplicate data from a text file.

```
import ast
import json

# read the contents ixora_corridor_3 the text file and convert to list of dictionaries
with open('ixora_corridor_1.txt', 'r') as f:
    contents = f.read().splitlines()
    dict_list = [ast.literal_eval(line) for line in contents]

# remove duplicate dictionaries
cleaned_dict_list = [dict(t) for t in set([tuple(d.items()) for d in dict_list])]

# write the cleaned list of dictionaries back to the text file
with open('ixora_corridor_1.txt', 'w') as f:
    for d in cleaned_dict_list:
        f.write(json.dumps(d) + '\n')
```

By removing duplicates during the data cleaning process, likelihood of overfitting can be reduced and promoted a more balanced dataset [24]. This helps ensure that the model learns the underlying patterns and relationships rather than being influenced by duplicated instances.

D. Data Transformation

Next, data transformation is necessary to convert the data from text file to CSV dataset. The main purposes of this process are intended to ensure data is in a structured and organized format for model training and evaluation. The python code snippet below is used to convert the text file to the CSV dataset.

```python
import csv
import json
import csv

with open('ixora_corridor_1.txt') as f:
    data = []
    for line in f:
        obj = json.loads(line.strip())
        data.append(obj)

# Define the column names for the CSV file
fieldnames = list(data[0].keys()) + ['label']
print(fieldnames)

# Write the data to a CSV file
with open('ixora_corridor_1.csv', 'w', newline='') as f:
    writer = csv.DictWriter(f, fieldnames=fieldnames)
    writer.writeheader()

    for d in data:
        writer.writerow({**d,'label': 'ixora_corridor_1'})
        # **d is used for unpack the dictionary, it will find the matches values
```

After the data transformation, it produced multiple CSV datasets that represented different locations. In this case, it may be necessary to combine these individual CSV datasets into a single dataset for further analysis or modelling. Hence, the following code snippet is applied to combine multiple CSV files using the Pandas library. The final output is shown as Figure 10.

```python
import pandas as pd

# Read the first CSV file
combined_df = pd.read_csv('ixora_room1_1.csv')


csv_files = ['ixora_room1_3.csv',
             'ixora_room2_1.csv', 'ixora_room2_3.csv',
             'ixora_room3_1.csv', 'ixora_room3_3.csv',
             'ixora_room4_1.csv', 'ixora_room4_3.csv',
             'ixora_toilet1_1.csv', 'ixora_toilet1_2.csv',
             'ixora_toilet2_1.csv', 'ixora_toilet2_2.csv',
             'ixora_livingroom_1.csv','ixora_livingroom_2.csv', 'ixora_livingroom_3.csv',
             'ixora_yard_1.csv', 'ixora_yard_2.csv',
             'ixora_corridor_1.csv', 'ixora_corridor_2.csv', 'ixora_corridor_3.csv',
             'ixora_kitchen_1.csv', 'ixora_kitchen_2.csv']


for file in csv_files:
    df = pd.read_csv(file)
    combined_df = combined_df.append(df, ignore_index=True)

# Save the combined dataset as a new CSV file
combined_df.to_csv('ixora_db.csv', index=False)
```

TP-Link_D5EC 60:a4:b7:99:d5:eb	C1805 2.4G 08:93:56:7e:17:f0	WIFI_OK a0:22:52:15:13:ff	NoPassword 90:91:64:56:7d:0b	C17-02@Airties f4:17:b8:3d:3a:3d	4G-CPE-C291 a0:80:b2:9e:c2:91	label
91	35	92	38	40	0	ixora_room1_1
93	0	88	31	0	86	ixora_room1_1
93	72	93	0	40	65	ixora_room1_1
93	75	92	0	0	65	ixora_room1_1
89	0	88	0	0	0	ixora_room1_1
89	75	88	0	0	65	ixora_room1_1
93	75	91	40	40	0	ixora_room1_1
96	43	85	31	0	78	ixora_room1_3
97	75	84	31	0	81	ixora_room1_3
97	62	86	0	0	81	ixora_room1_3
97	78	83	0	0	81	ixora_room1_3
96	67	87	0	0	80	ixora_room1_3
97	78	82	0	0	62	ixora_room1_3
87	57	67	0	0	82	ixora_room2_1
88	40	81	0	0	81	ixora_room2_1
86	40	81	0	0	81	ixora_room2_1
92	57	82	0	0	80	ixora_room2_1
85	57	75	0	0	83	ixora_room2_1
91	67	82	38	0	0	ixora_room2_1
92	72	81	38	0	78	ixora_room2_1
85	62	0	62	0	82	ixora_room2_1
85	35	80	0	0	84	ixora_room2_1
95	38	81	0	0	88	ixora_room2_3
90	65	87	43	0	88	ixora_room2_3
96	82	85	0	0	87	ixora_room2_3
90	75	83	0	0	85	ixora_room2_3

Figure 10. Final CSV Training File

Now, it becomes feasible to make the integration with model training pipelines, enabling efficient experimentation and evaluation of different machine learning models and techniques.

E. Data Characteristics

Following the data cleaning and data transformation stages, the dataset is now ready for model training. For the Ixora environment, there are 7 features have been selected which are **"TP-Link_D5EC", "C1805 2.4G", "WIFI_OK", "NoPassword", "C17-02@Airties", "4G-CPE-C291"** and **label**. Similarly, For the MMU environment, there are 5 features, i.e. **"eBfi@MMU Open", "eBfi@MMU CERT", "eBfi@MMU", "MMU 2"** and **label**. Overall, one sample consists of different signal strengths from each of the access points and finally label by the corresponding reference points name. Below Figure 11 and 12 show the features selection of Ixora and MMU.

TP-Link_D5EC	C1805 2.4G	WIFI_OK	NoPassword	C17-02@Airties	4G-CPE-C291	label
91	35	92	38	40	0	ixora_room1_1
93	0	88	31	0	86	ixora_room1_1

Figure 11. Features Selection of Ixora

eBfi@MMU Open	eBfi@MMU CERT	eBfi@MMU	MMU 2	label
46	40	82	88	mmu_1
40	46	81	90	mmu_1

Figure 12. Features Selection of MMU

For the Ixora environment, there is a total number of 22 reference points and each of them represents a specific location. By collecting samples from these reference points, a total number of 284 samples is obtained for training the model. Similarly, for the MMU environment, a total number of 6 reference points are involved and served as a specific location. After the data collection process from these reference points, the training dataset consists of 91 samples for model training.

To evaluate the performance and accuracy of the system, the testing dataset also has been prepared. For the Ixora environment, 39 validation samples have been collected to conduct the testing. For the MMU environment, 30 validation samples are obtained to evaluate the system performance. These validation samples were gathered from each of the reference points to ensure the system can perform well in the real-world situation. Below code snippets are used for calculating the total number of training samples and testing samples for different environments. The results are shown as Figure 13.

```
import pandas as pd

ixora_train_data = pd.read_csv('Ixora_Dataset.csv')
ixora_test_data = pd.read_csv('Ixora_ValidationSet.csv')
mmu_train_data = pd.read_csv('MMU_Dataset.csv')
mmu_test_data = pd.read_csv('MMU_ValidationSet.csv')

ixora_train_data = len(ixora_train_data)
ixora_test_data = len(ixora_test_data)
mmu_train_data = len(mmu_train_data)
mmu_test_data = len(mmu_test_data)

print("Total Number of Training Data For Ixora: ", ixora_train_data)
print("Total Number of Training Data For MMU: ", mmu_train_data)
print("Total Number of Testing Data For Ixora: ", ixora_test_data)
print("Total Number of Testing Data For MMU: ", mmu_test_data)
```

Total Number of Training Data For Ixora: 284

Total Number of Training Data For MMU: 91

Total Number of Testing Data For Ixora: 39

Total Number of Testing Data For MMU: 30

Figure 13. Total Number of Training Samples and Testing Samples for Different Environments

F. Model Training

The selection of a suitable machine learning model plays an important role in enhancing the accuracy and effectiveness of predictions in real time position. In this proposed solution, 3 machine learning algorithms have been chosen for training: Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbours (KNN). These algorithms were trained on the self-collected dataset to learn the relationships between the Wi-Fi signals and the corresponding locations. To train an effective and accurate model, parameter setting also plays a crucial part during the training. Hence, different hyperparameter tuning was applied to obtain the best configuration of each model. The tuning results will be shown in the following tables.

The Random Forest algorithm is an ensemble learning method that combines multiple decision trees to make predictions. During training, the algorithm will build a number of decision trees and each of them is using a random subset of features and data samples. Then, the algorithm combines the predictions from these individual trees to make a final prediction for a given Wi-Fi signal input. This ensemble approach helps to reduce overfitting and improve the accuracy of the indoor positioning predictions. Below Table 4 and 5 present the training result of different experimental environments using Random Forest.

K-Nearest Neighbors (KNN) is a machine-learning algorithm that uses supervised learning. KNN is also referred to as a “lazy learning” algorithm due to its learning characteristic method. During training, this algorithm will not be learning from the training set but instead stores the dataset and operates on it when classifying.

Consequently, KNN will begin with storing the datasets and then forming a reference set of labelled Wi-Fi signal samples and corresponding locations. When a new Wi-Fi signal input is provided, it will make the classification based on the majority voting by the K closest neighbors to determine predicted location. The calculation of K closest neighbors is usually determined based on a distance measure such as Euclidean distance, Hamming distance, and

Minkowski distance. Below Table 6 and 7 present the training result of different experimental environments using KNN.

Table 4. Experimental Results of Random Forest for Ixora

Hyperparameters		**n_estimators**			
max_depth	**class_weight**	200	400	600	800
20	balanced	**0.82**	0.79	0.79	0.81
40		0.77	0.79	0.79	0.79
60		0.79	0.81	0.79	0.81
80		0.75	0.75	0.77	0.81

Table 5. Experimental Results of Random Forest for MMU

Hyperparameters		**n_estimators**			
max_depth	**class_weight**	200	400	600	800
20	balanced	**0.84**	0.79	0.79	0.79
40		0.79	0.79	0.79	0.79
60		0.79	0.79	0.79	0.79
80		0.79	0.79	0.79	0.79

Table 6. Experimental Results of KNN for Ixora

Hyperparameters		**n_neighbors**			
weights	**leaf_size**	5	10	15	20
uniform	30	0.46	0.39	0.37	0.30
	50	0.46	0.39	0.37	0.30
	70	0.46	0.39	0.37	0.30
distance	30	**0.61**	0.51	0.46	0.47
	50	**0.61**	0.51	0.46	0.47
	70	**0.61**	0.51	0.46	0.47

Table 7. Experimental Results of KNN for MMU

Hyperparameters		**n_neighbors**			
weights	**leaf_size**	5	10	15	20
uniform	30	**0.79**	0.74	0.63	0.74
	50	**0.79**	0.74	0.63	0.74
	70	**0.79**	0.74	0.63	0.74
distance	30	**0.79**	0.74	0.74	0.74
	50	**0.79**	0.74	0.74	0.74
	70	**0.79**	0.74	0.74	0.74

Support Vector Machine (SVM) is a machine learning algorithm that can be implemented to solve classification or regression tasks. During the classification, finding the hyperplane that best split the different classes is the main objective of SVM. This can help to maximize the margin between the hyperplane and the closest data points from each class, known as support vectors. Below equation shows the mathematical formulas involved during finding the optimal hyperplane:

$$w * x + b = 0 \tag{1}$$

where w is a vector normal to hyperplane and b is an offset. Based on this formula, SVM can obtain the most appropriate values for w and b to make the classification and predict the position in an indoor environment.

Besides, SVM also relies on mathematical functions which are called kernel functions. Some kernel functions are commonly used such as linear kernel, Gaussian (RBF) kernel, sigmoid kernel, and polynomial kernel. These kernel functions can be used to transform the input data into a higher-dimensional feature space and enable the model to handle non-linear data. Therefore, the choice of kernel functions plays an important role to affect the performance during the indoor positioning classification. Below Table 8 and 9 show the training results of different experimental environments using SVM.

Table 8. Experimental Results of SVM for Ixora

Hyperparameter	**Regularization parameter, C**			
Kernel Function	2	4	6	8
linear	0.60	0.60	0.58	0.58
rbf	0.39	0.40	0.44	0.46
sigmoid	0.02	0.02	0.04	0.02
poly	0.56	0.60	0.60	**0.63**

Table 9. Experimental Results of SVM for MMU

Hyperparameter	**Regularization parameter, C**			
Kernel Function	2	4	6	8
linear	**0.84**	**0.84**	**0.84**	**0.84**
rbf	**0.84**	0.79	0.79	0.79
sigmoid	0.11	0.11	0.11	0.11
poly	**0.84**	**0.84**	**0.84**	**0.84**

V. DISCUSSION AND ANALYSIS

Based on the Table 10 and 11 below, the experimental results show that the Random Forest is the most accurate model in terms of precision, recall and F1-score. Despite the variations in the experimental environment, Random Forest model consistently achieves higher accuracy compared to the other algorithms. By tuning the hyperparameters, specifically setting the maximum depth to 20, utilizing class weight “balanced”, and increasing the number of estimators to 200, the highest accuracy of around 82% is obtained in the Ixora environment and 84% in the MMU environment. The outcome indicates an improvement of the model’s performance under these selected parameter configurations.

Table 10. Experimental Results Between Different Classification for Ixora

Methods	**Precision**	**Recall**	**F1-score**	**Accuracy**
Random Forest	0.88	0.82	0.83	**0.82**
KNN	0.68	0.61	0.61	0.61
SVM	0.74	0.63	0.62	0.63

Table 11. Experimental Results Between Different Classification for MMU

Methods	**Precision**	**Recall**	**F1-score**	**Accuracy**
Random Forest	0.85	0.84	0.84	**0.84**
KNN	0.81	0.79	0.79	0.79
SVM	0.85	0.84	0.84	**0.84**

From the results tables above, it has also proven that the Random Forest is considered more well-suited for the indoor positioning tasks. During the indoor positioning process, it often involves dealing with noisy and complex data (e.g., variations in signal strength, interference, and multipath effects). In this case, Random Forest model’s exceptional robustness to noise and its ability to handle high-dimensional and noisy datasets are key attributes to achieve higher accuracy and determine a reliable prediction. Below Figure 14 presents the confusion matrices for Ixora and MMU using Random Forest.

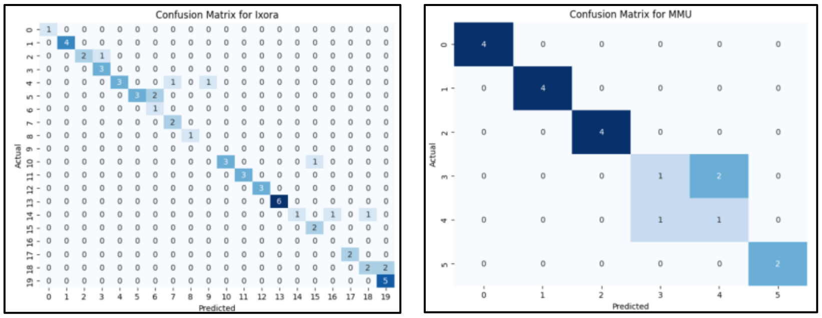


Figure 14. Confusion Matrices of Random Forest for Ixora (Left) and MMU (Right)

In addition to Random Forest, the results tables have shown that the SVM can obtain the accuracy up to 63% in the Ixora environment and 84% in the MMU environment. Although the accuracy obtained is relatively lower compared to Random Forest in the Ixora environment, it exhibits a stable and high level of performance in the MMU environment. This difference can be attributed to the different indoor positioning situation and data characteristics. Although SVM is well-known for its adaptability and ability to handle difficult classification tasks, it is sensitive with the noisy data and linear separability assumptions pose limitations [25]. In this case, it is possible that the data collection in the MMU environment contains less noise, clear signal strength and more linear separability. Therefore, SVM can perform higher accuracy during the classification. Below Figure 15 presents the confusion matrices for Ixora and MMU using SVM.

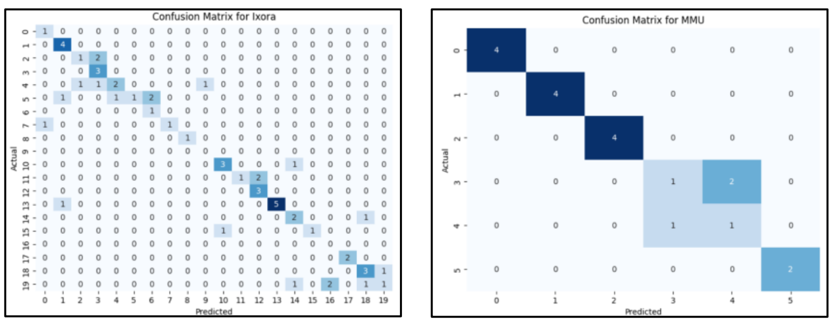


Figure 15. Confusion Matrices of SVM for Ixora (Left) and MMU (Right)

Lastly, KNN demonstrates lowest accuracy compared with the other algorithms which obtained accuracy around 61% in the Ixora environment and 79% in the MMU environment. Considering these factors such as high-dimensional and complex data, noise and interference, KNN struggles to effectively handle these factors and leading to lower performance during the training. Hence, KNN may not be a suitable choice for indoor positioning tasks. Below Figure 16 presents the confusion matrices for Ixora and MMU using KNN.

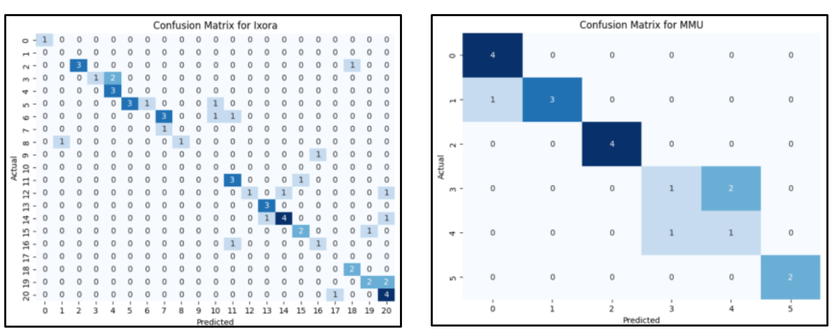


Figure 16. Confusion Matrices of KNN for Ixora (Left) and MMU (Right)

To evaluate the performance of the model, the best trained model (Ixora – Random Forest model, MMU – Random Forest model) from two different environments has applied to make the prediction on the prepared testing set. According to the testing result presented in Table 12, the trained model developed by MMU can obtain an accuracy of up to 87%. In comparison, the Ixora model exhibits a lower accuracy with an 82% accuracy. To obtain a comprehensive summary of the model's prediction (e.g., true positive, true negative, false positive, and false negative), confusion matrices has been generated to analyse the predicted labels against the actual labels. Based on the Figure 17 below, the result indicated that the trained model of MMU can classify 26 out of 30 data and only misclassifies 4 out of 30 data from the testing set. For the Ixora trained model, it can classify 32 out of 39 data and only misclassified 7 out of 39 data from the testing set.

Table 12. Testing Results Between Different Trained Models

Trained Models	Precision	Recall	F1-score	Accuracy
Ixora	0.91	0.82	0.84	**0.82**
MMU	0.90	0.87	0.87	**0.87**

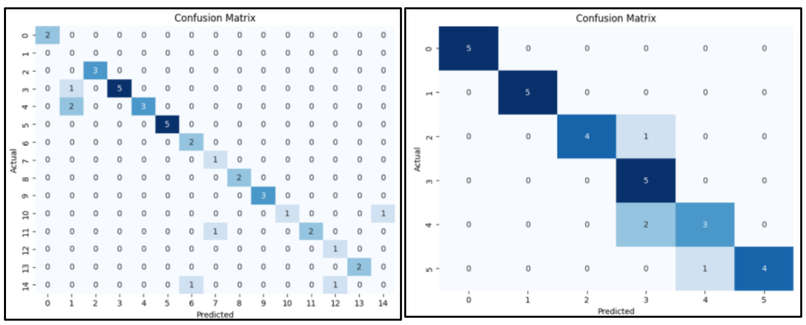


Figure 17. Confusion Matrices of Testing Results for Ixora (Left) and MMU (Right)

Based on the observation from the confusion matrices, it is evidence to show that the MMU model is significantly outperformed compared to the Ixora model. To further substantiate the claim, measurement metrics play an important role to prove the results. However, in the context of indoor positioning using Wi-Fi, obtaining the actual position data is difficult and not feasible due to a lack of tools or resources. To deal with these issues, the map matching method is used as an alternative way to evaluate the accuracy of the results. This method will compare the estimated position derived from the model with the collected references points. Then, Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are used to measure the average deviation between the predicted position and the reference points. Below equations 2 and 3 are used to compute the RMSE and MAE, where $Predicted_i$ is the predicted values of the target and $Actual_i$ is the actual values of the target.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(Predicted_i - Actual_i)^2}{n}} \tag{2}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | Actual_i - Predicted_i | \tag{3}$$

To implement these formulas, the scikit-learn library's **"mean_squared_error"** and **"mean_absolute_error"** functions have been used which provide a convenient way to make the calculation and get the final results. Below Table 13 presented the performance assessment using RMSE and MAE.

Table 13. Evaluation Metrics – RMSE and MAE

Environments	RMSE	MAE
Ixora	1.11	0.34
MMU	0.51	0.15

Table 13 shows that the RMSE and MAE values of Ixora model are 1.11 and 0.34. For the MMU model, it demonstrates superior performance with an RMSE value of 0.51 and an MAE value of 0.15. Overall, it can be concluded that both models can achieve good performance for indoor positioning using Wi-Fi.

In general, the MMU model is more effective and reliable compared to the Ixora model. In fact, there are several potential factors and challenges that are required to take into the consideration such as signal stability, complex indoor environments, limited data, method of data collection, and references point requirements. These factors have significantly impacted the results and should not be overlooked.

First of all, one of the potential factors to consider is the stability of the Wi-Fi signal. When indoor positioning is using Wi-Fi, signal stability can directly impact the positioning estimation and performance of the trained models. Unstable signals such as intermittent connectivity or signal dropouts during the data collection may lead to errors later in the prediction process. Additionally, complex environments with different layout and structures including wall, furniture, door can also cause the inconsistencies of Wi-Fi signal strength.

Due to the instability of Wi-Fi signals, the data collection process becomes more challenging and time-consuming. Despite collecting a seemingly large amount of data, a significant portion of it may not be acceptable for training and testing machine learning models. Therefore, the data pre-processing steps are necessary to ensure the data quality including examination and identification. However, it required an amount of time and effort to deal with data issues such as noise, outliers and inconsistencies. Since the limited time available to implement the entire system and features, this will be a challenge to develop an accurate real-time positioning system.

Another potential factor has been found in which the model will encounter difficulties during classifying the position when multiple reference points are located in close proximity to each other. Due to the limited spatial resolution of the Wi-Fi signals and the interference between nearby reference points, the model may have difficulties to compute an accurate position in this case. Hence, the placement and distribution of reference points in indoor positioning is important to help reduce signal interference and improve the system's ability during classifying the position.

VI. CONCLUSION

In conclusion, this study has developed a Wi-Fi based real-time indoor positioning system using different machine learning algorithms (e.g., Random Forest, KNN and SVM). Received Signal Strength Indication (RSSI) is self-collected at each reference point in order to proceed to the model training. During the training, different parameter tuning has been applied to find the best combination of parameter values that maximize the model in terms of accuracy, precision and recall. The experimental results indicate that Random Forest outperforms other classification models and achieve an accuracy of 82% in the Ixora environment and 84% in the MMU environment. Finally, the best performance model has been integrated into the Unity platform to provide a user-friendly interface for users to make the navigation and real-time position.

In future work, an emphasis will be placed on collecting more data to improve the performance of the model. The data should be more refined and involve other pre-processing techniques such as removing noise and outliers to enhance the system's reliability. Comparison with benchmark datasets will be included to further validate the performance of the model. Besides, the indoor positioning currently only conducts within a single floor. In the future, multi-floor (occlusion) positioning should be considered and places with frequencies interferences or less detection points can be studied.

ACKNOWLEDGEMENT

A version of this paper was presented at the third International Conference on Computer, Information Technology and Intelligent Computing, CITIC 2023, held in Malaysia on 26th-28th July 2023.

The work was generously supported by internal fund with project ID MMUI/210025

AUTHOR CONTRIBUTIONS

Zi Yang Chia: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Validation, Visualization, Writing – Original Draft Preparation;
Pey Yun Goh: Conceptualization, Project Administration, Resources, Supervision, Writing – Review & Editing;

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

REFERENCES

[1] M. Nabati and S. A. Ghorashi, "A real-time fingerprint-based indoor positioning using deep learning and preceding states," *Expert Syst Appl*, vol. 213, Mar. 2023, doi: 10.1016/j.eswa.2022.118889.

[2] J. F. Raquet, "Navigation Using Pseudolites, Beacons, and Signals of Opportunity," *NATO STO lecture series SET-197, navigation sensors and systems in GNSS degraded and denied environments*, pp. 1–18, 2013.

[3] P. Spachos and K. N. Plataniotis, "BLE Beacons for Indoor Positioning at an Interactive IoT-Based Smart Museum," *IEEE Syst J*, vol. 14, no. 3, pp. 3483–3493, Sep. 2020, doi: 10.1109/JSYST.2020.2969088.

[4] Z. Zuo, L. Liu, L. Zhang, and Y. Fang, "Indoor positioning based on bluetooth low-energy beacons adopting graph optimization," *Sensors (Switzerland)*, vol. 18, no. 11, Nov. 2018, doi: 10.3390/s18113736.

[5] L. Pei, J. Liu, Y. Chen, R. Chen, and L. Chen, "Evaluation of fingerprinting-based WiFi indoor localization coexisted with Bluetooth," *The Journal of Global Positioning Systems*, vol. 15, no. 1, Dec. 2017, doi: 10.1186/s41445-017-0008-x.

[6] A. A. Kalbandhe and S. C. Patil, "Indoor Positioning System using Bluetooth Low Energy," in *International Conference on Computing, Analytics and Security Trends, CAST 2016*, Institute of Electrical and Electronics Engineers Inc., Apr. 2017, pp. 451–455. doi: 10.1109/CAST.2016.7915011.

[7] Y. B. Bai, S. Wu, H. Wu, and K. Zhang, "Overview of RFID-Based Indoor Positioning Technology," *GSR*, vol. 2012, 2012.

[8] C.-S. Wang, C.-L. Chen, and Y.-M. Guo, *A real-time indoor positioning system based on RFID and Kinect*. Springer, 2013.

[9] J. D. Domingo, C. Cerrada, E. Valero, and J. A. Cerrada, "An improved indoor positioning system using RGB-D cameras and wireless networks for use in complex environments," *Sensors (Switzerland)*, vol. 17, no. 10, Oct. 2017, doi: 10.3390/s17102391.

[10] S. Lee, J. Kim, and N. Moon, "Random forest and WiFi fingerprint-based indoor location recognition system using smart watch," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, Dec. 2019, doi: 10.1186/s13673-019-0168-7.

[11] A. Nightingale, "Triangulation," *International Encyclopedia of Human Geography*, pp. 489–492, Jan. 2009, doi: 10.1016/B978-008044910-4.00552-6.

[12] M. E. Rusli, M. Ali, N. Jamil, and M. M. Din, "An Improved Indoor Positioning Algorithm Based on RSSI-Trilateration Technique for Internet of Things (IOT)," in *Proceedings - 6th International Conference on Computer and Communication Engineering: Innovative Technologies to Serve Humanity, ICCCE 2016*, Institute of Electrical and Electronics Engineers Inc., Dec. 2016, pp. 12–77. doi: 10.1109/ICCCE.2016.28.

[13] L. Nomdedeu, J. Sales, R. Mar{\'\i}n, E. Cervera, and J. Aez, *Sensing capabilities for mobile robotics*. Elsevier, 2011.

[14] L. S. De Oliveira, O. K. Rayel, and P. Leitao, "Low-Cost Indoor Localization System Combining Multilateration and Kalman Filter," in *IEEE International Symposium on Industrial Electronics*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021. doi: 10.1109/ISIE45552.2021.9576353.

[15] R. Abishek, K. Abishek, N. Hariharan, M. R. Vaideeswaran, and C. S. Paripooranan, *Analysis Of Machine Learning Algorithms For Wi-Fi-based Indoor Positioning System*. IEEE, 2019.

[16] J. Chóliz, Á. Hernández, and A. Valdovinos, "A framework for UWB-based communication and location tracking systems for wireless sensor networks," *Sensors*, vol. 11, no. 9, pp. 9045–9068, Sep. 2011, doi: 10.3390/s110909045.

[17] A. Alarifi *et al.*, “Ultra wideband indoor positioning technologies: Analysis and recent advances,” *Sensors (Switzerland)*, vol. 16, no. 5. MDPI AG, May 01, 2016. doi: 10.3390/s16050707.

[18] D. Tian and Q. Xiang, “Research on Indoor Positioning System Based on UWB Technology,” pp. 662–665, 2020.

[19] P. Dabove, V. Di Pietra, M. Piras, A. A. Jabbar, and S. A. Kazim, “Indoor positioning using Ultra-wide band (UWB) technologies: Positioning accuracies and sensors’ performances,” in *2018 IEEE/ION Position, Location and Navigation Symposium, PLANS 2018 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jun. 2018, pp. 175–184. doi: 10.1109/PLANS.2018.8373379.

[20] Q. Yang, S. Zheng, M. Liu, and Y. Zhang, “Research on Wi-Fi indoor positioning in a smart exhibition hall based on received signal strength indication,” *EURASIP J Wirel Commun Netw*, vol. 2019, no. 1, Dec. 2019, doi: 10.1186/s13638-019-1601-3.

[21] Y. Wu, R. Chen, W. Li, Y. Yu, H. Zhou, and K. Yan, “Indoor Positioning Based on Walking-Surveyed Wi-Fi Fingerprint and Corner Reference Trajectory-Geomagnetic Database,” *IEEE Sens J*, vol. 21, pp. 18964–18977, 2021.

[22] G. Huang, Z. Hu, J. Wu, H. Xiao, and F. Zhang, “WiFi and vision-integrated fingerprint for smartphone-based self-localization in public indoor scenes,” *IEEE Internet Things J*, vol. 7, pp. 6748–6761, 2020.

[23] L. Lau *et al.*, “An autonomous ultra-wide band-based attitude and position determination technique for indoor mobile laser scanning,” *ISPRS Int J Geoinf*, vol. 7, no. 4, Apr. 2018, doi: 10.3390/ijgi7040155.

[24] P.-W. Chin, K.-W. Ng, and N. Palanichamy, “Plant Disease Detection and Classification Using Deep Learning Methods: A Comparison Study,” *Journal of Informatics and Web Engineering*, vol. 3, no. 1, pp. 155–168, Feb. 2024, doi: 10.33093/jiwe.2024.3.1.10.

[25] T. Ahmed Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. Mohd Su’ud, “Sentiment Analysis using Support Vector Machine and Random Forest,” *Journal of Informatics and Web Engineering*, vol. 3, no. 1, pp. 67–75, Feb. 2024, doi: 10.33093/jiwe.2024.3.1.5.