SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

SCSE21-0145

# Bluetooth Low Energy (BLE) based asset tagging system

**SUBMITTED BY:** POH JUN RONG

MATRICULATION NUMBER: U1821563B

**SUPERVISOR:** MR OH HONG LYE

**EXAMINER:** A/P LAU CHIEW TONG

A final year project report presented to the Nanyang Technological University
in partial fulfilment of the requirements of the degree of
Bachelor of Engineering (Computer Engineering)

2021

**Abstract**

Asset Tracking is a valuable technology that most businesses want to leverage on, especially the well developed GPS-based outdoor asset tracking system. However, indoor localization is still not well developed as GPS is not accurate in indoor environment. One good option is to utilize BLE technology for indoor localization. However, by using the RSSI value itself is not accurate. Therefore, this project will develop an asset tracking system to collect RSSI fingerprinting data and increase localization accuracy by using various machine learning algorithms. The experimental results show a significant improvement over a previous BLE indoor localization study, but there is still opportunity for improvement, such as adopting different machine learning techniques as a comparison.

**Acknowledgements**

This report paper would not have been successfully completed without the assistance of the following people. The author would like to express his gratitude and appreciation to them.

Mr Oh Hong Lye, for his advice and encouragement throughout the course of this project. His knowledge and experience provided the author with insightful and valuable recommendations.

The author would also like to thank Dr Seanglidet Yean for providing the author

with helpful and informative ideas based on her expertise and guidance.
Finally, the author would like to express his gratitude to Singtel Cognitive and Artificial Intelligence Lab for providing the test site.

**Table of Contents**

## Contents

**List of Figures**

**List of Tables**



**Abbreviation**

| | |
|---|---|
| BLE | Bluetooth Low Energy |
| nRF52 | Adafruit Feather nRF52 Bluefruit LE |
| RPi | Raspberry Pi |
| RSSI | Received Signal Strength Indicator |
| GPS | Global Positioning System |
| RFID | Radio-frequency identification |
| MAC Address | Media Access Control Address |
| ATT | Attribute Protocol |
| GATT | Generic Attribute Profile |
| GAP | Generic Access Profile |
| CSV | Comma Separated Values |

**Terminology**

| Term | Description |
|---|---|
| Asset Tag | nRF52 that only perform broadcasting |
| Post | nRF52 that is connected to RPi and will scan for Asset Tag |
| Fingerprinting | An approach to collect RSSI value from a set of fixed beacons in a given area |



# 1 Introduction

## 1.1 Background

Internet of Things (IoT) have been incorporated into our daily life and is an especially crucial asset for businesses to leverage in Industry 4.0. In 2020, the asset tracking market was valued at USD 17.14 billion and is expected to reach USD 34.82 billion by 2026[1]. The mainstream asset tracking systems adopted by most of the organisations are namely, RFID, GPS, Wi-Fi, and BLE. However, as the affordability and efficiency of real-time asset tracking systems like GPS, Wi-Fi and BLE improve, many organisations are opting for it [2]. GPS is able to achieve great accuracy and reliability. However, GPS performance is not reliable for indoor environments as compared to outdoor environments [3]. On other hand, in a small indoor environment, there are limited Wi-Fi access points to perform Wi-Fi fingerprinting. As a result, BLE is one good alternative technology to adopt in a small indoor environment. However, just by utilizing the RSSI value, it is difficult to approximate the location of the Asset Tag[4]. Therefore, the proposed indoor asset tagging system will implement fingerprinting and a machine learning algorithm to achieve a proximate estimation of the Asset Tag location.

### 1.2 Project Objectives

The objective of this project is to develop and implement an indoor asset tagging system that will assist users to locate their assets indoors. The indoor asset tagging system will utilise nRF52 and RPi to achieve a proximate estimation of the Asset Tag location.

Multiple machine learning algorithms are applied to the BLE RSSI fingerprint data. Localization accuracy of these algorithm are evaluated and compared under the context of an Asset Tag location System.

### 1.3 Project Scope

The indoor fingerprinting localization system consists of three major components namely, BLE-based Asset Tracking Infrastructure, BLE RSSI fingerprint data collection and cleaning, and location prediction. The scope of each component are as follows:

BLE-based Asset Tracking Infrastructure

- Develop a system consisting of multiple nRF52 and an RPI.
- Extract RSSI value from nRF52 to RPi for data collection.

BLE RSSI fingerprint data collection and cleaning

- Research on the positioning of nRF52 for data collection.
- Research on how to implement and prepare the data for machine learning

Location prediction

- Research and implement different machine learning algorithms to the fingerprinting data
- Evaluate and compare the performance of the models.

### 1.4 Report Organization

The organization of this report are structured as follows. Section 2 discusses projects that are related. Section 3 introduces the hardware, software and development tools used in this project. Section 4 indicate the time schedule of this project. Section 5 describes the system design, technologies, and machine learning models utilized in this project. Section 6 describes the implementation of the technologies on the system. Section 7 discusses the techniques to prepare the dataset for machine learning. Section 8 evaluate and compare various techniques to improve the performance of different models. Finally, Section 9 discusses the limitations, future recommendations, and conclusion of the project.

## 2 Related Works

### 2.1 Indoor localisation using Bluetooth

In this research paper[5], there are 2 major components. Firstly, the author used two different fingerprinting approaches namely the zones method and the XY method. Secondly, the author used three machine learning algorithms namely Convolutional Neural Networks (CNN), Decision Jungle, and Decision Forest to predict the location of the beacon.

The author compared various machine learning models and evaluated the accuracy of each model. Hence, all three models can achieve accuracy of 65.78% for sub 2m, 88.31% for sub 3m and 97.73% for sub 4m with XY fingerprinting approach, compared to the Zone approach that can achieve accuracy of 98.75% for sub 2m.

### 2.2 Indoor localisation using Wi-Fi

In this research paper[6], the author applied three algorithms namely, Euclidean Distance, Support Vector Machine, and Neural Network to two different indoor Wi-Fi fingerprint data. The Neural Network approach has the smallest mean localisation error of 7.27m compared to the other two algorithms. The author then performs transfer learning by training data and applying data from different floors, resulting in a reduced error variance compared to a neural network that is being trained by a smaller training data sample. This proved that a well pre-trained model can get a reasonable performance.

## 3 Resources

The following resource(s) were required to carry out this project:

### 3.1 Hardware

| # | Item | Quantity | Description | Cost (SGD) |
|---|---|---|---|---|
| 1 | Raspberry Pi 3 Model B | 1 | Communication with nRF52 and computer | $56 |
| 2 | Adafruit Feather nRF52840 Express | 5 | Act as Asset Tag | $125 |
| 3 | Adafruit Feather nRF52840 Express | 6 | Act as Post | $150 |
| 4 | Adafruit Feather nRF52832 | 2 | Act as Post | $50 |
| 5 | Lithium Ion Polymer Battery - 3.7V 400mAh | 13 | Power nRF52 | $90 |
| Total Cost | | | | $471 |

### 3.2 Software

- Raspberry Pi
  - Raspbian Stretch - OS for RPi
  - Python
    - Bluepy library
    - Socket library
- Arduino
  - Adafruit Circuit Playground

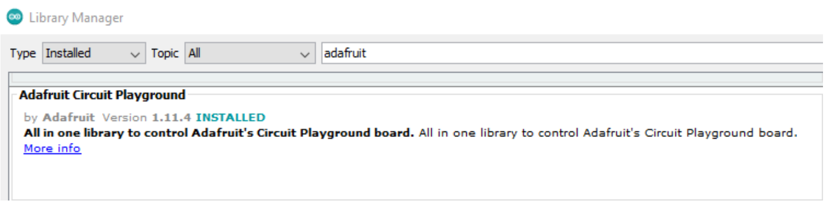



*Figure 1: nRF52 Arduino library*

- Computer
  - Python
    - Socket library
    - Sklearn
    - Matplotlib
    - Numpy
    - Pandas
    - Seaborn

### 3.3 Development and Implementation Tools

- Visual Studio Code
- Arduino

## 4 Project Schedule

| Deliverables | AY 2021/2022 Semester 1 | | | | | AY 2021/2022 Semester 2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May |
| Research on Adafruit nRF52 Series | | | | | | | | | | |
| Listening Post perform scanning for Asset Tag | | | | | | | | | | |
| Testing of GATT | | | | | | | | | | |

Protocol
Research RPi as BLE Central Device
Setup the whole system
Data collection for RSSI fingerprint data
Interim Report
Research on machine learning algorithm
Applied machine learning to captured RSSI Data
Hyperparameter Tuning on models
Evaluate the performance of models
Final Report
Amendments to Final Report
Final Presentation

## 5 System Design

This section will provide an overview of the project design, implementation, and testing.

### 5.1 Architecture

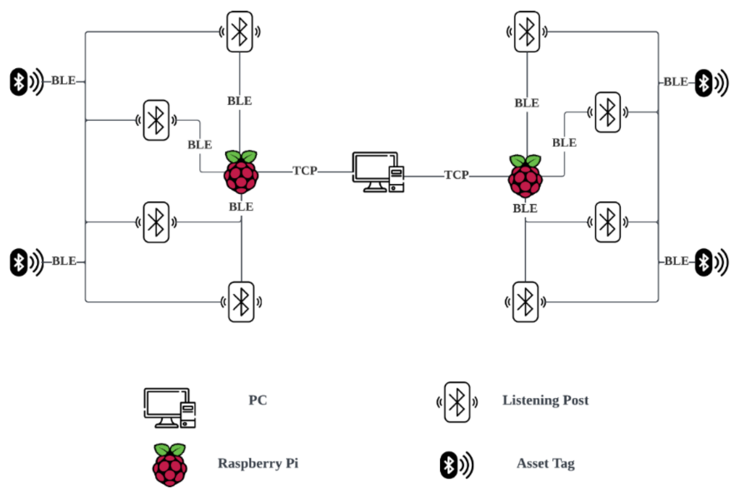



Figure 2 : Overall System Architecture

The overall system architecture consists of a Computer, RPi, Listening Posts, and Asset Tags as shown in Figure 2 and all data communication will be wireless either via Wi-Fi or BLE. The system is scalable where it just requires to deploy additional RPi and Listening Posts to expand the capability to scan for Asset Tags in another area as shown in Figure 2.

The hardware provided for the BLE section are Adafruit Feather nRF52840 and Adafruit Feather nRF52832. In order to reduce variance to the fingerprinting data, all Asset Tag will use Adafruit Feather nRF52840. Whereas, the reminding Adafruit Feather nRF52840 and nRF52832 will be allocated as Post.

### 5.2 Data Flow

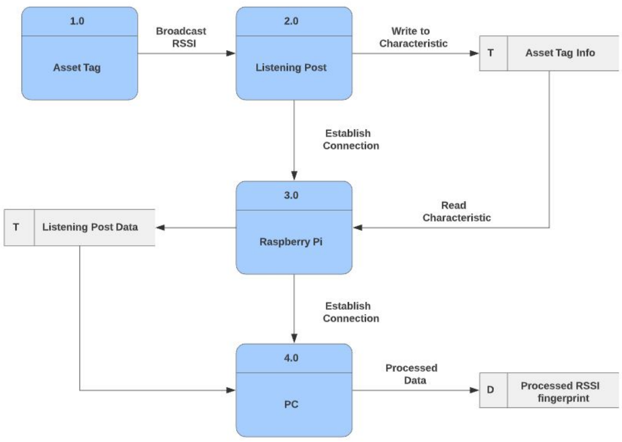



Figure 3: Data Flow Diagram

In Figure 3, it shows how the information flows in the system.

1. The Listening Post will scan for any RSSI being broadcasted by Asset Tag and write Asset Tag information to its characteristic such as:
   a. Asset Tag MAC Address
   b. Asset Tag RSSI Value
2. RPi will establish connection with Listening Posts to read the Asset Tag information.
3. RPi will establish connection with PC and send Listening Post Data such as:
   a. Listening Post MAC Address
   b. Asset Tag MAC Address
   c. Asset Tag RSSI Value
4. The PC will process and compile the data received and store into CSV file.

### 5.3 Adafruit Feather nRF52

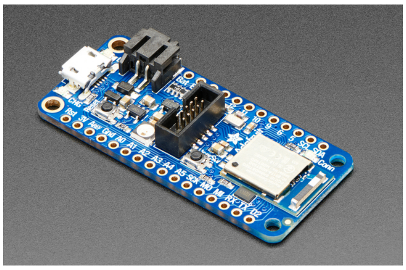


Figure 4 : nRF52840[7]

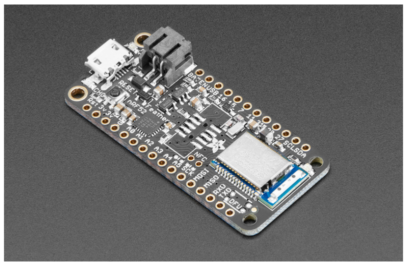


Figure 4. 1: nRF52832[8]

The Feather nRF52 series is a microcontroller equipped with ARM Cortex M4F and the BLE is compatible with 2.4GHz spectrum wireless protocol, it is only 51mm x 23mm in size and is as light as 6 grams[7]. In addition, it is also portable with the additional attachment of a Lithium Ion Polymer Battery that can last for hours[7]. This chip has 1MB flash, 256KB SRAM for nRF52840 and 512KB flash, 64KB SRAM for nRF52832 with the built-in micro USB port to enable USB to Serial programming[7]. The nRF52 series also supports both Arduino IDE and CircuitPython programming[7].

#### 5.3.1 GAP Protocol

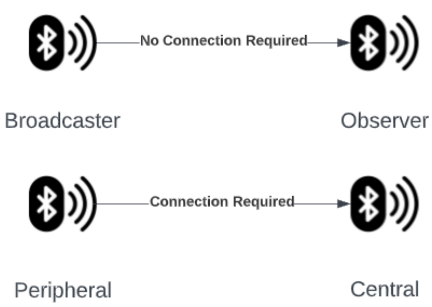



Figure 5: GAP Protocol Roles

GAP Protocol consist of 4 roles namely Broadcaster, Observer, Peripheral, and Central as shown in Figure 5.
The broadcaster can only advertise itself periodically by sending advertising data packets, in contrast to the observer that can only scan for advertising data packets advertised by the broadcaster. Both the broadcaster and observer does not require connection to be establish with each other. On the other hand, both the peripheral and central requires connection to be establish with each other.
The peripheral is also known as server while the central is client. The peripheral will be in the broadcaster role by advertising itself to allow the central to scan for the advertising data packets and initiate connection to perform bidirectional GATT data communication. In addition, the peripheral only allows single connection to central whereas central allows multiple connections to peripheral.

### 5.3.2 ATT Protocol
Attribute Protocol defines the data representation format in the server to allow the client to access which composed of four fields[9]:
- Attribute type, defined by UUID
- Attribute handle, unsigned number unique for the attribute
- Attribute permissions, read/write to the data
- Attribute value

### 5.3.3 GATT Protocol

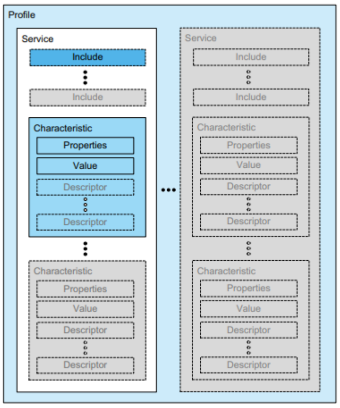


Figure 6: GATT-Based Profile hierarchy[9]

GATT is built on top of ATT to define the device role to be either server or client. In addition, GATT establish common operations and framework for the data stored by ATT[9]. The data is organized hierarchically as shown in Figure 6 that include a profile that contains one or more services required to fulfil a use case. In addition, service contains one or more characteristic or reference to other services. The characteristic composed of a value and optional description that describe the value or permit configuration with respect to the value[9].

### 5.4 Asset Tag
The hardware of Asset Tag consisted of Adafruit Feather nRF52840 and Lithium Ion Polymer Battery - 3.7V 400mAh. Asset Tag can be configured into either broadcasting or peripheral role. The peripheral role will allow the user to interact with the asset tag with a central device to either write or modify value(s) for the Listening Post to read, write or modify.

### 5.5 Listening Post
The Listening Post consisted of Adafruit Feather nRF52832, Adafruit Feather nRF52840, and Lithium Ion Polymer Battery - 3.7V 400mAh. On the nRF52 series, the Adafruit library allow the configuration of the device into dual roles, allowing the device to operate in both peripheral and central roles simultaneously. The central role enables the connection with the Asset Tag to utilize the value(s), whereas peripheral mode enables the connection with RPi via BLE to allow the utilizations of value(s) in Listening Post.

## 5.6 Raspberry Pi

The Raspberry Pi 3 Model B is a microprocessor that has a built-in BCM43438 wireless LAN and BLE, that allows the interconnection between Listening Post via BLE and computer via Wi-Fi [10]. In addition, RPi will enable the scalability of the indoor asset tagging system, simply just require a Wi-Fi access point to expand or deploy a new set of system in any indoor environment.

## 5.7 Fingerprinting

The dataset consisted of all deployed Posts MAC Addresses, Asset Tags MAC Addresses, X coordinate of Asset Tag, Y coordinate of Asset Tag, and timestamp. Each row of appended data will only have 1 unique Asset Tag MAC Address followed by all of the deployed Listening Post captured RSSI value of that specific Asset Tag that is shown in Figure 7 below.

| E0:34:D2:05:89:8B | C6:78:DD:81:C4:9D | E2:A5:AF:AC:F0:FD | D3:E1:D1:04:4E:33 | FA:11:88:10:A3:FB | C0:D1:95:3D:DA:8F | E9:35:4C:AA:F2:72 | FC:39:3F:11:CF:AD | Asset | X_Coord | Y_Coord | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| -65 | -61 | -61 | -66 | -64 | -63 | -69 | -75 | D3:76:4C:D3:76:9F | 0 | 0 | 1642527965 |
| -50 | -52 | -52 | -58 | -56 | -48 | -62 | -67 | FB:E3:89:62:ED:11 | 0 | 0 | 1642590498 |
| -50 | -62 | -61 | -62 | -63 | -46 | -67 | -71 | F1:9B:2D:6E:31:EF | 0 | 0 | 1642590508 |
| -58 | -59 | -57 | -65 | -61 | -53 | -70 | -69 | C4:E9:1D:C1:B3:B6 | 0 | 0 | 1642590519 |

Figure 7: Fingerprinting data format

## 5.8 Machine Learning Algorithm

The proposed supervised machine learning algorithms can be used to solve both regression and classification issues. In this case, where the labelled output is coordinates and the desired outcomes of the result should be in magnitude, the regression approach is preferred over classification.

### 5.8.1 Decision Tree

The decision tree will divide the dataset into smaller subsets and gradually develop the associated decision tree, eventually returning the most favourable prediction result. The final tree is composed of a root node, decision nodes and leaf nodes. The root node is the beginning of the decision tree that split into two decision nodes according whether the condition is true or false. The decision nodes will split until a leaf node is reached where further splitting is not possible. The leaf node can be reached earlier when the hyperparameter is fulfilled to prevent overfitting of the model.

### 5.8.2 Random Forest

The random forest constructs a multitude of decision trees and outputs the mean of the prediction of individual trees. In addition, random forest uses bootstrap aggregation (bagging), which involves selecting a random sample of data from the training set with replacement to reduce the variance and prevents overfitting. There is also no interaction between the trees during the construction of the tree to protect each other from their individual errors.

### 5.8.3 Support Vector Machine (SVM)

The support vector regression objective is to find a hyperplane in an n-dimensional space that has the maximum number of points. Support vectors are the closest data points on the either side of the hyperplane. The support vector regression will attempt to fit the optimal line within the hyperplane and boundary line. In addition, support vector regression's fit time complexity is more than quadratic with the number of samples, making it challenging to scale with huge datasets.

### 5.8.4 Hyperparameter Tuning

The value of hyperparameters has a significant impact on the performance of a model. In addition, there is no way to know the best values for hyperparameters without trying all possible values to know the optimal values. GridSearchCV is commonly used approach to automate the tuning of hyperparameters. GridSearchCV helps to loop through predefined hyperparameters and fitting them into the estimator with training dataset and returning the best parameters.

□

# 6 System Implementation

## 6.1 GATT Service and Characteristic UUID

The UUIDs used to identify the GATT service and characteristic is important in order to avoid misinterpretation of the values if the labels are incorrect. The GATT service UUID 0x1821 is specifically allocated for indoor positioning[11].

However, there are allocations for longitude and latitude that could lead to a misinterpretation for global positioning system (GPS) value. Thus, to avoid misinterpretation two distinct UUIDs are used for coordinate X and Y that are not assigned for any purpose.

### 6.1 Asset Tag

Asset Tag can be operate in either broadcast or peripheral role.

#### 6.1.1 Broadcast Role

The broadcasting role is preferable over the peripheral role for the setup to collect the RSSI fingerprint data to increase the rate of Asset Tag advertising package being scanned by Listening Post. The Asset Tag is set to broadcast advertisements in intervals of 20ms to 152.5ms which is 6.5 to 50 times per second to avoid the Listening Post inability to scan for all Asset Tags advertising packages. The manufacturer ID 0x0822 is set to represent as Adafruit device to allow central role to filter out other BLE device advertising package in the environment.

#### 6.1.2 Peripheral Role

The peripheral role for Asset Tag is more practical when deploy as indoor localization system. This will be a valuable feature to allow the user to interact with the Asset Tag using their central device to input relevant values related to the asset for the system to capture.

### 6.2 Listening Post

#### 6.2.1 Dual Roles

Dual Roles composed of both peripheral and central roles.

##### 6.2.1.1 Peripheral Role

GATT Service UUID 0x1821 is added to label as Indoor Positioning and composed of Characteristic UUID listed in the table below:

Table 1: Assigned Characteristic UUID

| UUID | Description |
|---|---|
| 0xABCD | Asset Tag MAC address |
| 0xABCE | Asset Tag RSSI value |
| 0xABCF | Listening Post MAC address |

This will allow RPi to connect to the Listening Post and read Asset Tag MAC Address, Asset Tag RSSI captured by the Listening Post and the connected Listening Post MAC Address.

##### 6.2.1.2 Central Role

There are a lot of BLE devices in the environment and creates a lot of noise. Manufacturer ID 0x0822 filter is used to solve this issue, this will allow the Listening Post to only scan for peripheral devices with manufacturer ID 0x822. The Listening Post will only capture the advertising package and not establish connection with the Asset Tag if the Asset Tag is not configured to peripheral role.

### 6.3 Raspberry Pi

The RPi requires Bluetooth connectivity and bluepy library installed to enable the connection with Listening Post via BLE. As mentioned in Listening Post central role section, there are lots of BLE devices in our environment and creates a lot of noise, filter is required to allow RPi to quickly establish connection with Listening Post. Hence, a whitelist of Listening Post MAC addresses is introduced to allow the ease of connection between RPi and Listening Post. This method also acts as a form of security as MAC address is unique which only allow RPi to connect to peripheral devices that are whitelisted. The RPi is also configured into AP mode to allow communication with the computer via socket programming. The data will be read in real-time from Listening Post to the computer.

### 6.4 Test Site

The test site is located at Singtel Cognitive and Artificial Intelligence Lab

(SCALE) Lab in the School of Computer Science and Engineering (SCSE). The XY method was used instead of the Zone method for fingerprinting because hanging the Asset Tags was challenging in SCALE Lab as shown Figure 8.1. The following Figure 8.2 will illustrate the placement of both Listening Posts and Asset Tags at the test site.


Figure 8. 1: Test Site


Figure 8. 2: Placement of Devices

### 6.4.1 Asset Tag Placement

The indoor test site is roughly 8m by 6m, Asset Tags will be placed within a 6m by 4m area spreading 1m apart from each other to minimise crosstalk. The Asset Tags will also be placed on a table with a height of 46cm to minimise any attenuation that may affect the RSSI value when placed on the ground itself as shown in Figure 8.3.

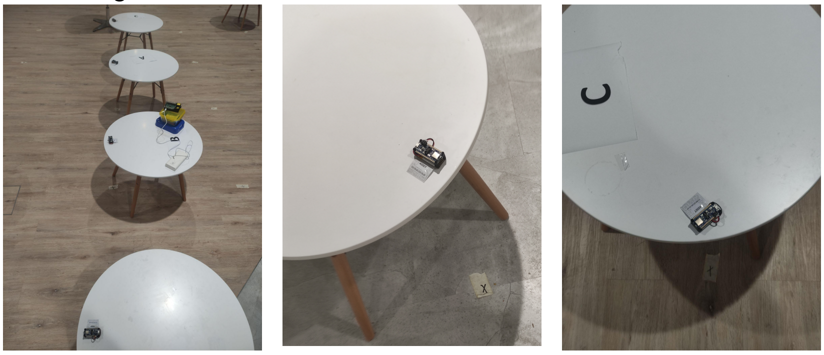


Figure 8. 3: Asset Tags placement

### 6.4.2 Listening Post Placement

The 6 nRF52840 Listening Posts are placed 4m apart from each other surrounding the Asset Tags, the reminding 2 nRF52832 Listening Posts are placed on both sides of the wall 2m apart from the nRF52840 shown in Figure 8.4.

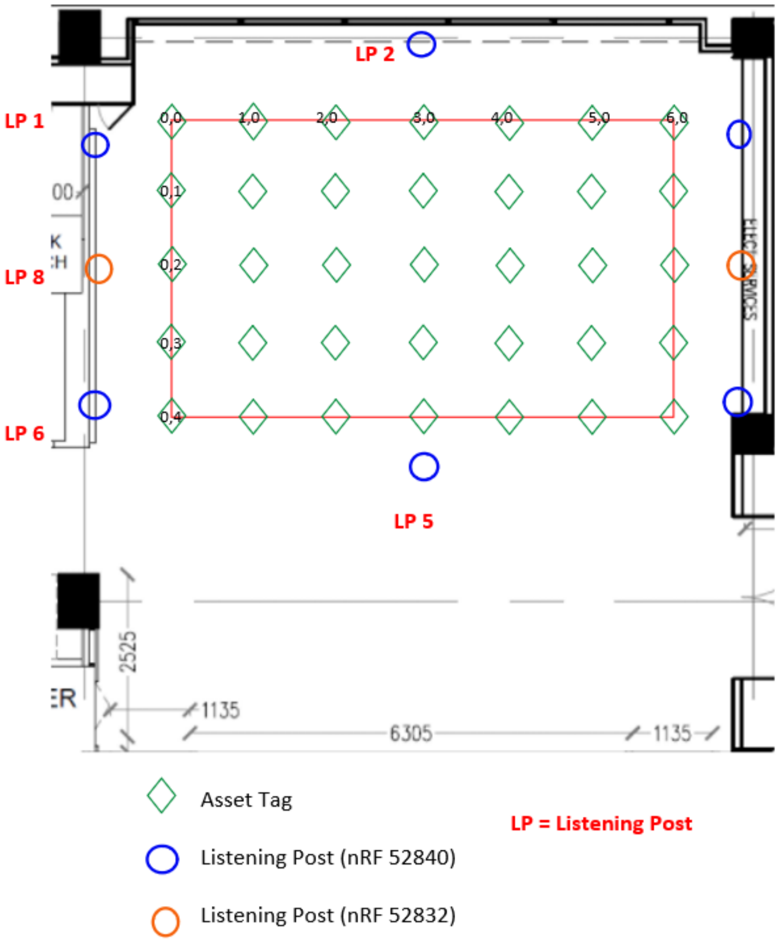



Figure 8. 4: Test Site Layout

**6.5 RSSI Fingerprinting**

Real-Time data is collected from Listening Post via RPi and stored in a CSV file on the computer. The X and Y coordinates are labelled accordingly as shown the Figure 3.7. In addition, the data will be formatted as shown in the table 2.

Table 2: Fingerprint Data Header

| Data Column Header | Description |
|---|---|
| E0:34:D2:05:89:8B | Listening Post 1 MAC Address |
| C6:78:DD:81:C4:9D | Listening Post 2 MAC Address |
| E2:A5:AF:AC:F0:FD | Listening Post 3 MAC Address |
| D3:E1:D1:04:4E:33 | Listening Post 4 MAC Address |
| FA:11:88:10:A3:FB | Listening Post 5 MAC Address |
| C0:D1:95:3D:DA:8F | Listening Post 6 MAC Address |
| E9:35:4C:AA:F2:72 | Listening Post 7 MAC Address |
| FC:39:3F:11:CF:AD | Listening Post 8 MAC Address |
| Asset | Asset Tag MAC Address |
| X_Coord | Coordinate X |
| Y_Coord | Coordinate Y |
| Time | Timestamp of data being stored |

The Listening Post can only write the RSSI value into its characteristic when a new advertising package from the Asset Tag is being scanned. As a result, iteration and check conditions are required to ensure that the fingerprint data is complete, with no multiple missing RSSI values of a specific Asset Tag across all deployed Listening Post. RSSI value of 100 is assigned to ensure that the iteration can terminate in circumstances where the Listening Post is out of range to obtain the Asset Tag advertising package. Threading is implemented, it will temporarily save the RSSI value of a specific Asset Tag from all the Listening Posts, and if no value is recorded, it will assign an RSSI value of 100 at the end of the iteration. This will decrease the time required by N times where N is the amount of Asset Tags being deployed to complete the fingerprinting data. There are 5 Asset Tags being deployed during the collection of RSSI fingerprint data. Hence, every minute can log up to 1.5 samples of a specific Asset Tag. We will sample for 60 minutes at each position, resulting in around 90 samples per position.

# 7 Machine Learning

## 7.1 Data Preparation

The RSSI values that is 100 will all be replaced with -100 to indicate that the Listening Post is too far away from the Asset Tag. The dataset is then split into input and output datasets, with the input dataset containing all listening posts values and the output dataset consists of X and Y coordinates.

## 7.2 Model

### 7.2.1 Training and Test Dataset

K-Fold is used to split the data into multiple folds instead of the traditional train test split which is just 2 datasets to prevent overfitting. The model will be trained N iterations using various folds to train and test, with N-1 folds being used for model training and 1 fold being used for testing as shown in Figure 9.

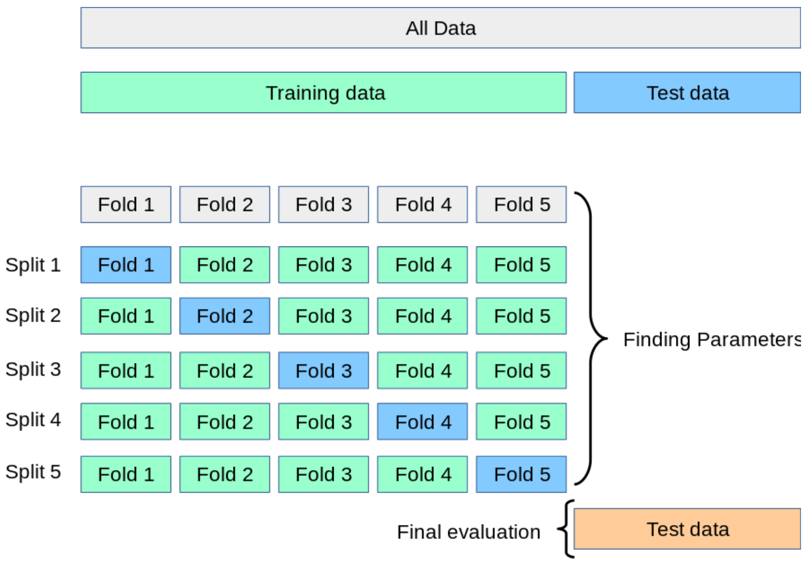



Figure 8: K-Fold[12]

# 8 Result

## 8.1 Evaluation Metric

The labelled value of the dataset is the Asset Tag coordinates, which are a two-dimensional array made up of X and Y. Assessing the error of the predicted coordinates is equivalent to measuring the magnitude of a vector as shown in Figure 10. Vector Norm is defined as a square root of the sum of squares of each component of a vector as shown in Equation 1.

$$||\vec{x}|| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

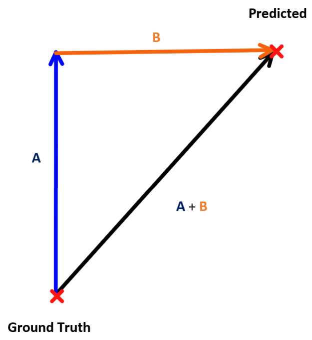



Figure 9: Vector Addition        Equation 1: Vector Norm

The vector norm will return a result of absolute error, and we can calculate the mean absolute error by taking the sum of absolute errors and divides by the numbers of errors.

**8.2 Hyperparameter Tuning**
This section will compare the performance of all models before and after hyperparameter tuning. Appendix 1 contains more detailed information on hyperparameter adjustment for all models.

**8.2.1 Decision Tree**
The performance of post-tuning model performed better with lower mean absolute error as shown in Table 3. The model's performance is depicted in greater detail in Figure 11.

Table 3: Decision Tree Hyperparameters Tuned (Meters)

| | Default | Post-tuning |
|---|---|---|
| Mean | 1.511 | 1.387 |
| Median | 1 | 1.114 |
| Maximum | 6.4 | 6.918 |
| Minimum | 0 | 0 |
| Third quartile | 2.236 | 1.993 |
| Outliers | >= 5.59 | >= 4.125 |

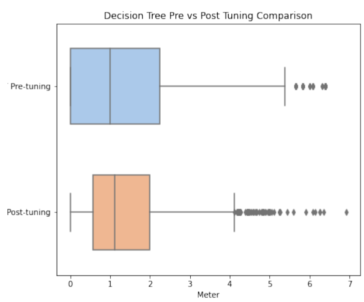



Figure 10: Decision Tree Hyperparameter Tuning

**8.2.2 Random Forest**
The performance was relatively similar for both models with post-tuning slightly better as shown in Table 4 and Figure 12.

Table 4: Random Forest Hyperparameters Tuned (Meters)

| | Default | Post-tuning |
|---|---|---|
| Mean | 1.207 | 1.207 |

| | | |
|---|---|---|
| Median | 0.992 | 0.973 |
| Maximum | 5.744 | 5.703 |
| Minimum | 0 | 0.001 |
| Third quartile | 1.738 | 1.716 |
| Outliers | >= 3.605 | >= 3.551 |

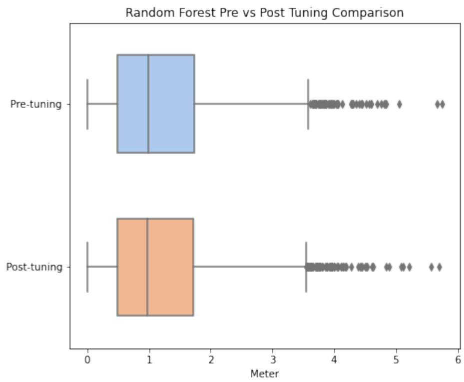



Figure 11: Random Forest Hyperparameter Tuning

**8.2.3 SVM**

RBF kernel performed better compared to other models with lower mean absolute error, Sigmoid had the worst performance of all models as shown in Table 5 and Figure 13.1. RBF kernel is then further tuned with parameter C and the performance improved slightly as shown in Figure 13.2.

Table 5: SVM Hyperparameters Tuned (Meters)

| | Linear | RBF | Polynomial | Sigmoid |
|---|---|---|---|---|
| Mean | 1.848 | 1.455 | 1.588 | 2.288 |
| Median | 1.75 | 1.29 | 1.423 | 2.285 |
| Maximum | 6.542 | 5.15 | 8.11 | 3.744 |
| Minimum | 0.107 | 0.013 | 0.03 | 0.141 |
| Third quartile | 2.37 | 1.98 | 2.154 | 3.036 |
| Outliers | >= 4.2 | >= 3.773 | >= 4.064 | <= 0.2 |

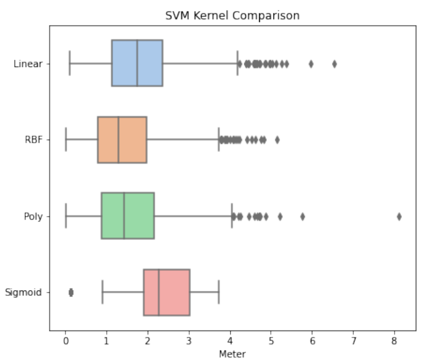



Figure 13. 1: SVM Kernel Comparison

Table 6: RBF Hyperparameters Tuned (Meters)

| | Default | Post-tuning |
|---|---|---|
| Mean | 1.414 | 1.344 |
| Median | 1.215 | 1.12 |
| Maximum | 5.308 | 5.696 |
| Minimum | 0.019 | 0.017 |
| Third quartile | 1.924 | 1.837 |
| Outliers | >= 3.712 | >= 3.64 |

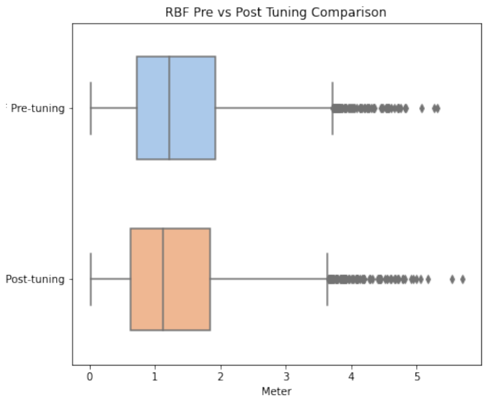



Figure 13. 2: RBF Hyperparameter Tuning

**8.3 Performance Comparison of all Models**

Based on the optimal result from each model, we can see that random forest performed the best with the lowest mean absolute error out of all the models as shown in Table 7 and Figure 14. In addition, our random forest model performance is much better with mean absolute error of 1.206 m compared with past research of 1.94 m[5].

Table 7: Models Comparison (Meters)

| Model | Mean | Median | Minimum | Maximum | Upper Extreme |
|---|---|---|---|---|---|
| Decision Tree | 1.388 | 1.114 | 0 | 6.918 | 4.125 |
| Random Forest | 1.206 | 0.973 | 0.001 | 5.703 | 3.551 |
| SVM (RBF) | 1.344 | 1.12 | 0.017 | 5.696 | 3.64 |

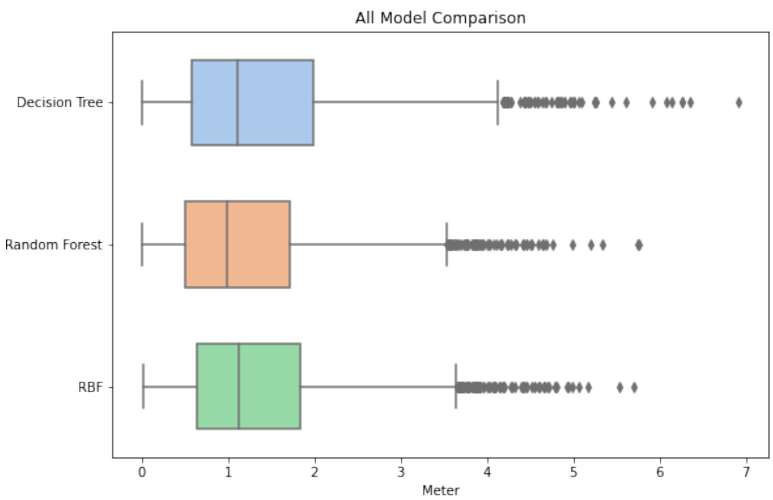



Figure 12: Models Comparison

Appendix 2 contains the plots of ground truth vs predicted correlation to visualize the prediction point of all models. By cross referencing the Figures in Appendix 2, Figure 8.1, and Figure 8.4 you can see the correlation where the model's predictions are better for the coordinates that are near to the Listening Post without obstacles. In addition, SVM model tends to predict out of the boundary while both decision tree and random forest prediction is always within the boundary.

# 9 Conclusion

## 9.1 Limitations

### 9.1.1 Test Site & Project Scale

Due to the limitation of the hardware provided, the test site was not fully utilized to perform RSSI fingerprint data collection. In addition, to prioritise the number of Listening Posts being deployed the limited Asset Tags deployed greatly contributed to the time required to collect the dataset.
The project has been scaled down, but it can be expanded into a larger project in the future when additional hardware is available.

## 9.2 Recommendation for Future Work

### 9.2.1 Exploration on more Machine Learning Algorithms

The initial half of this project was spent on the development of the architecture that enabled RSSI fingerprinting, this reduced the amount of time available to investigate further machine learning techniques that was being implemented in the related project [5][6].

### 9.2.2 Different Fingerprinting Approach

The research paper[5] stated that the Zone approach performance is much better compared to XY approach that is difficult to be performed on the test site provided. If there is another test site that permit the Zone approach, we might be able to get even better results with those RSSI fingerprints.

### 9.2.3 Conclusion

As mentioned earlier, this project has 3 major milestones that is to develop the indoor asset tagging system, RSSI fingerprint data collection, and to evaluate various machine learning models performance on the fingerprinting data.
The performance of models utilized in this project can achieve an average prediction of 1.3 m from the ground truth position. Moreover, random forest models were found to perform the best compared to other models. This achievement is even better than the past related project random forest models.
One limitation was the lack of hardware components to effectively exploit the size of test site, which might improve the performance of the models with additional Listening Post and larger dataset.
The Zone approach should be performed if the test site permits in the future studies. In addition, neural networks and other machine learning algorithms can be used evaluate and compare both fingerprinting approaches.

# 10 Reference


1. "Asset tracking market: 2022 - 27: Industry share, size, growth - mordor intelligence," *Asset Tracking Market | 2022 - 27 | Industry Share, Size, Growth - Mordor Intelligence*. [Online]. Available: https://www.mordorintelligence.com/industry-reports/asset-tracking-market. [Accessed: 21-Jan-2022].
2. "10 best IOT Asset Tracking Systems," *Operator by Hologram*. [Online]. Available: https://www.hologram.io/blog/10-best-iot-asset-tracking-systems. [Accessed: 21-Jan-2022].
3. "Types of asset tracking technologies: Energy, Resources & Industrials," *Deloitte Netherlands*, 17-Jun-2021. [Online]. Available: https://www2.deloitte.com/nl/nl/pages/energy-resources-industrials/articles/industry40-types-of-asset-tracking-technologies.html. [Accessed: 21-Jan-2022].
4. Z. Ma, S. Poslad, J. Bigham, X. Zhang, and L. Men, "A BLE RSSI ranking based indoor positioning system for generic smartphones," *2017 Wireless Telecommunications Symposium (WTS)*, 2017.

5. E. R. Sykes and A. Mustafa, “Enhancing micro-location accuracy for Asset Tracking: An evaluation of two fingerprinting approaches using three machine learning algorithms,” *Lecture Notes in Networks and Systems*, pp. 84–104, 2021.
6. L. Xiao, A. Behboodi and R. Mathar, "A deep learning approach to fingerprinting indoor localization solutions," 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), 2017, pp. 1-7, doi: 10.1109/ATNAC.2017.8215428.
7. A. Industries, “Adafruit Feather NRF52840 Express,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/4062. [Accessed: 21-Jan-2022].
8. A. Industries, “Adafruit Feather NRF52 Bluefruit Le,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/3406. [Accessed: 21-Jan-2022].
9. “Specifications,” *Bluetooth® Technology Website*. [Online]. Available: https://www.bluetooth.com/specifications/specs/core-specification-5-3/. [Accessed: 01-Mar-2022].
10. Raspberry Pi, “Buy A raspberry pi 3 model B,” *Raspberry Pi*. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/. [Accessed: 01-Mar-2022].
11. “16-bit UUID numbers document - .NET framework.” [Online]. Available: https://btprodspecificationrefs.blob.core.windows.net/assigned-values/16-bit%20UUID%20Numbers%20Document.pdf. [Accessed: 01-Mar-2022].
12. “3.1. cross-validation: Evaluating estimator performance,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Accessed: 11-Mar-2022].

**Appendix 1**

Table 8: Decision Tree Hyperparameters Tuned

| Parameters | Parameter Tuned |
|---|---|
| Maximum Depth of the Tree | 9 |
| Minimum number of samples required to be leaf node | 7 |

Table 9: Random Forest Hyperparameters Tuned

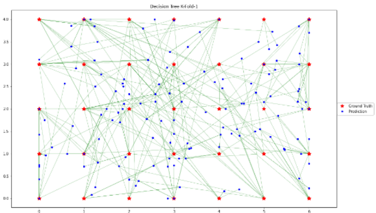
| Parameters | Parameter Tuned |
|---|---|
| Maximum depth of the Tree | 15 |
| Minimum number of samples required to be leaf node | 2 |
| Number of trees in the forest | 300 |
| Whether bootstrap samples are used when building trees | True |

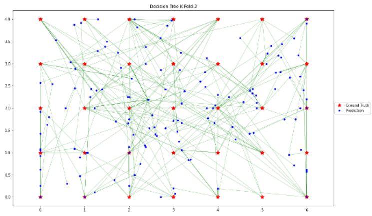
Table 10: SVM Hyperparameters Tuned

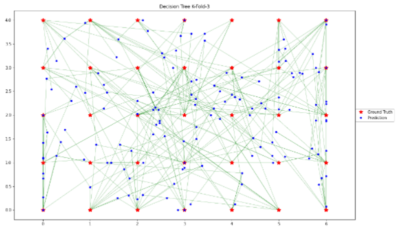
| Parameters | Parameter Tuned |
|---|---|
| Kernel | Radial basis function (RBF) |
| C, regularization parameter | 9 |

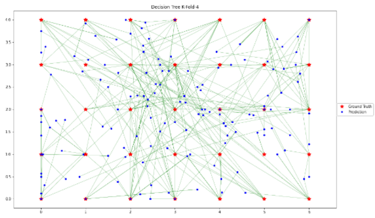
# Appendix 2

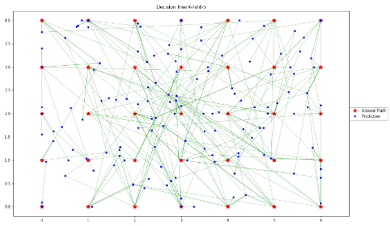
Decision Tree

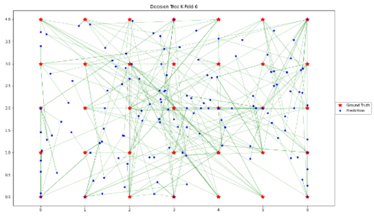


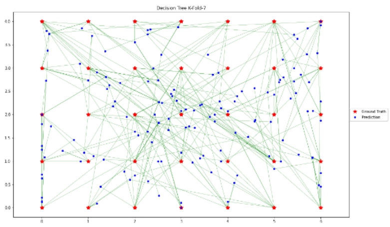



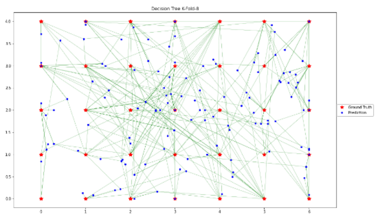



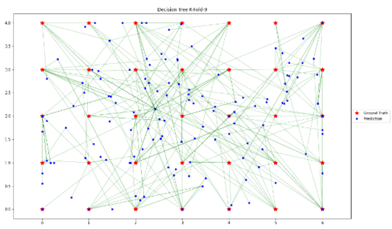



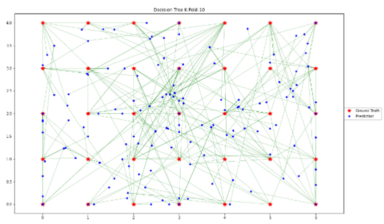



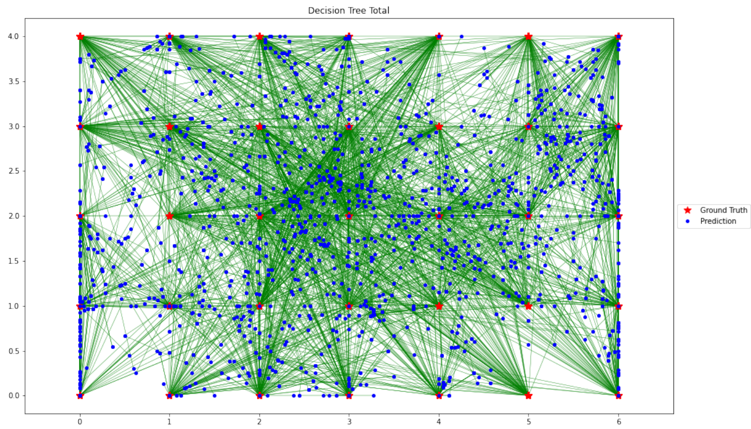
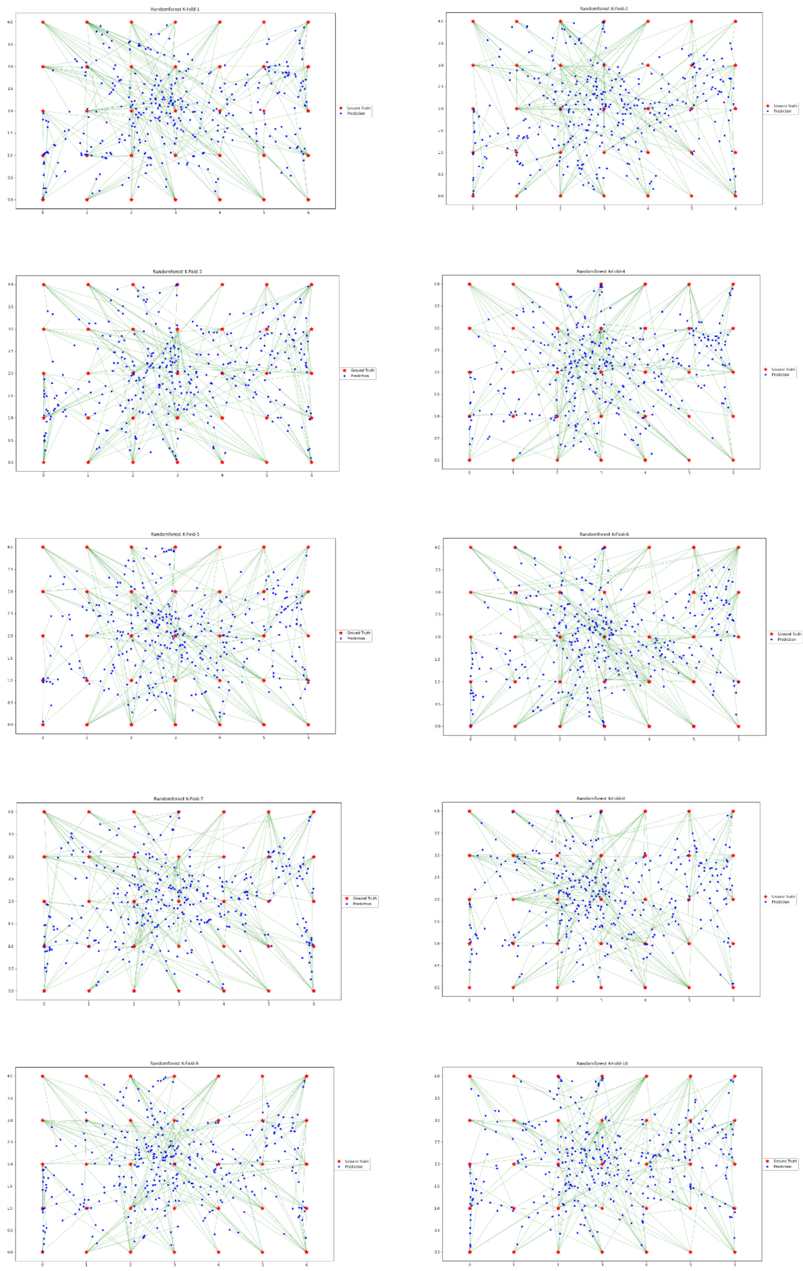



Figure 15. 1: Decision Tree Ground Truth vs Prediction

## Random Forest



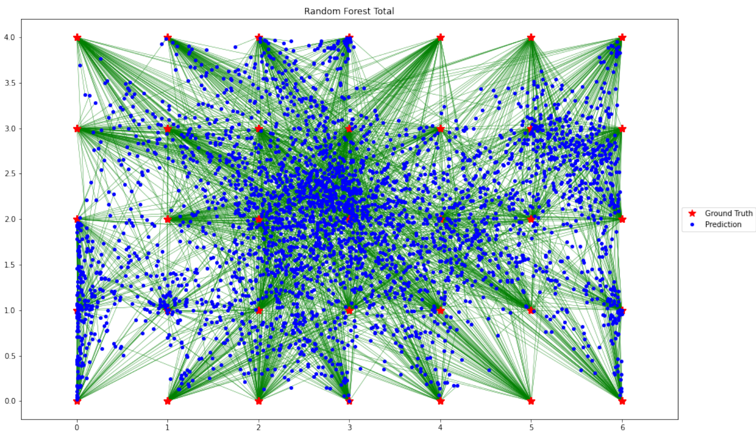
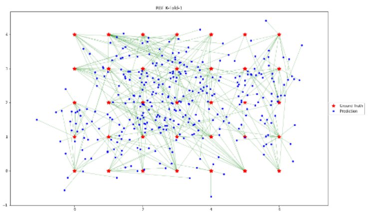
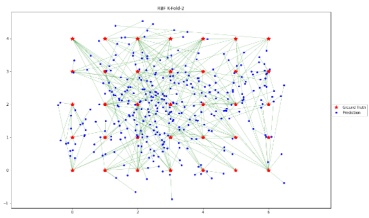
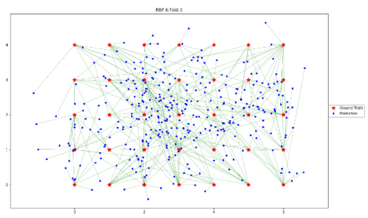


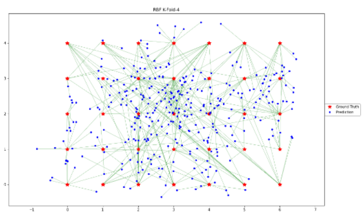

Figure 15. 2: Random Forest Ground Truth vs Prediction

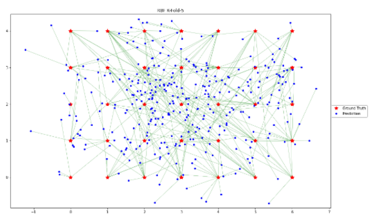
SVM (RBF)

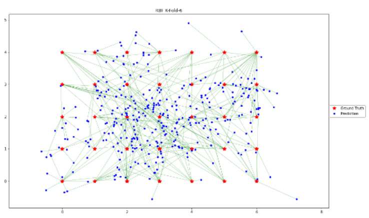


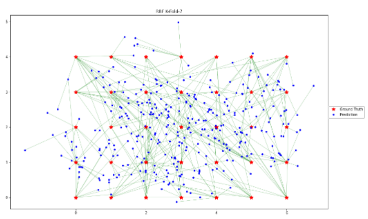


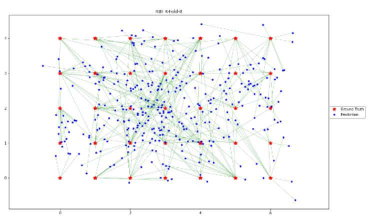


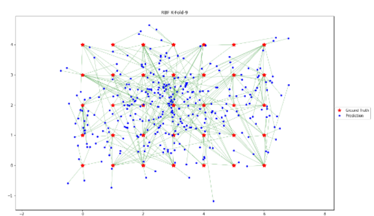


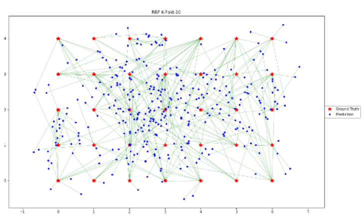












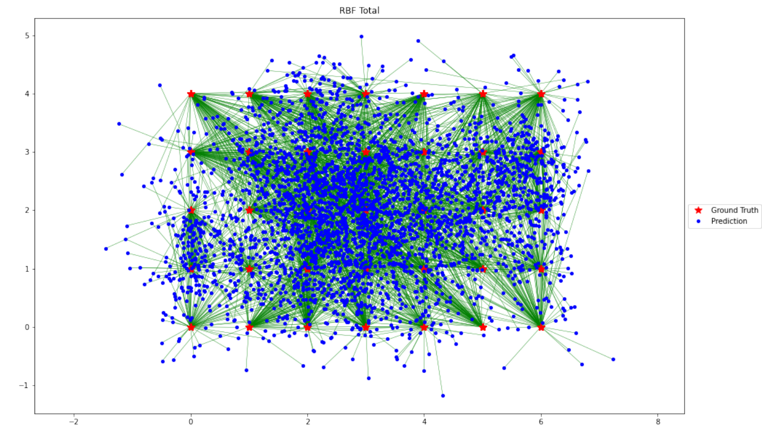



Figure 15. 3: RBF Ground Truth vs Prediction