BLE Indoor Positioning System: A Data-Driven Perspective

Azmyin Md Kamal
Graduate Student,
Department of Mechanical Engineering
University of Louisiana at Lafayette
Lafayette, USA
azmyin-md.kamal1@louisiana.edu

Allison Mixon
Graduate Student,
Department of Mechanical Engineering
University of Louisiana at Lafayette
Lafayette, USA
allison.mixon1@louisiana.edu

Abstract: BLE Indoor Positioning Systems that are implemented using propagation loss models have shown to estimate positions with large errors due to the uncertainty of RSSI signals caused by indoor obstacles and electromagnetic phenomena. To circumvent this issue, we implemented a "fingerprinting" Indoor Positioning System which utilizes Machine Learning models to create a RSSI-Location map of the environment for improved location estimation. We tested our BLE IPS in four test scenarios where two different hypotheses were evaluated. The first hypothesis investigates the question: does the accuracy of Bluetooth Low Energy (BLE) IPS scale linearly with the number of beacons; the second hypothesis evaluates the question: does the accuracy increase when the minimum distance between two "fingerprint" positions is 1m while the mapping area does not change. Experimental results showed that Hypothesis 1 is true where the best accuracy obtained was 71% using 4 nodes and a Random Forest Classifier. Hypothesis 2 was proven to be true with an average accuracy of 89% achieved by a 4-node configuration using the Random Forest Classifier. Our results clearly indicate to the scalability of "fingerprinting" IPS provided the receiving device can get data from all the nodes present in the system at all possible localization points.

***Index Terms*—Indoor Positioning, Supervised Learning, BLE Beacons, Fingerprinting, RSSI**

I. Introduction

The Internet of Things, more commonly referred to as IoT, is a complex system containing thousands of devices that connect to one another over the Internet [6]. Indoor positioning systems are one such IoT application which has the potential to change how people navigate indoor locations in a manner similar to GPS for outdoor locations [7]. However, GPS technology is not suitable for this task since it consumes a lot of power and can provide a maximum resolution of 5 meters whereas the required resolution of IPS systems can be as low as 1 meter [8].

The necessity for Indoor Positioning Systems (IPS) has increased greatly in the recent years due to its myriad applications such detecting the location of products stored in a warehouse, location detection of medical staff and personnel inside of a hospital, location of emergency first responders inside a building, and many more [2, 3 ,7]. There are three factors that must be taken into consideration for designing an IPS system viz. the arrangement of the receiver and transmitter, the type of RSSI analysis and the wireless technology that is used (WiFi, BLE, Zigbee, etc) [2,3].

One of the most common practice in IPS design is to estimate position using Received Signal Strength Indicator (RSSI) [3]. RSSI values can be obtained by capturing the periodic broadcast signals generated by a wireless device (WiFi, BLE Beacons). Comparing this data with a known transmitted signal strength, the propagation loss can be calculated which then can estimate position using a propagation loss model [2,3]. One of its primary drawbacks is the difficulty in choosing which propagation model (free space path loss model, Log-Distance Path Loss model, etc.) is most appropriate as their performance varies widely with indoor conditions [3,7].

Another common approach, often dubbed the "fingerprinting" technique, involves creating a RSSI-Location [3] map by taking repeated measurements of RSSI values at known locations. The values for each location is called a "fingerprint" [9]. A user's position is then estimated against this "fingerprint" map. Common models used in this approach are probabilistic model, support vector classifiers, random forest classifier, kNN and so on [2,9]. Note that, this "fingerprinting" method is an application of "Supervised Machine Learning" methods [2].

Although the first approach is mathematically simple and quick to implement, it is prone to very high error due to the unpredictable changes in RSSI [4] caused by signal attenuation due to obstacles, multipath fading, radio dead zones, relative motion between Tx and Rx antennas, and so on [3,5]. In contrast, the "fingerprinting" method offers greater accuracy than the propagation loss models since it assumes that the condition in which the RSSI-Location map was created holds true for predicting a human's movement [8]. However, "fingerprinting" methods are environment exclusive i.e. an exhaustive characterization of the environment is needed to be known before deploying the system.

A major design challenge for IoT based IPS system is the resource constraint receiving device which often must allocate its limited resources to multiple process. Thus, BLE Beacon based Indoor Positioning System has become quite popular since BLE Beacons are low cost, low power consuming device which can broadcast a data byte without requiring the receiving device to pair with itself [8]. Three common protocols for BLE Beacons are Estimote, Apple's iBeacon and Google's Eddystone [10].

Two major questions remain largely unsolved in BLE IPS designed with "fingerprinting" method [2,3,7]. The first is *how many BLE beacons* can adequately characterize the environment and the second is *how close can we predict a person's position* between successive "fingerprints"?

II. Problem Statement

The aim of our project is to develop a "fingerprint" Indoor Positioning System to evaluate the following hypothesis

i. *Accuracy of BLE IPS scales linearly with the number of beacons.*
ii. *Minimum distance between two "fingerprint" positions is 1m, given BLE beacons are spaced >=1.5m apart and, the mapping area does not change.*

III. Assumptions

1. BLE beacons will not move from their position during training and testing phase
2. 200 samples per "fingerprint" (from hereon referred to as *grid point*) is sufficient to capture the RSSI dynamics.
3. Interference of WiFi and other BLE signals are negligible.
4. Test scenarios made using 2 nodes, 3 nodes and 4 nodes are sufficient to answer hypothesis 1.
5. For all scenarios, the receiving device shall be able to get "sense" all the BLE beacons present.

IV. Experimental Setup

Figures 1 shows the experimental environment for this project. Figure 2-5 depicts our four test scenarios with additional details.

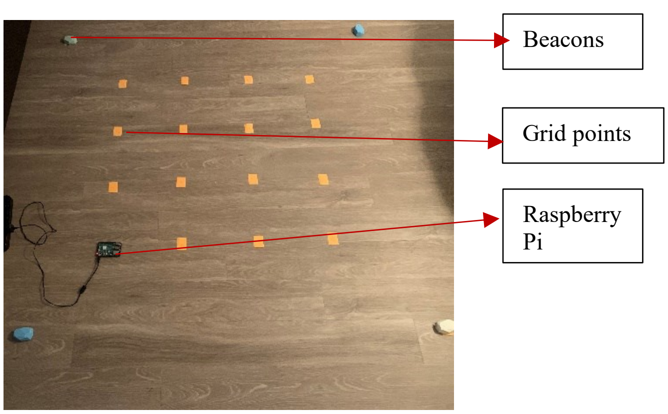

Fig 1. Experimental Setup

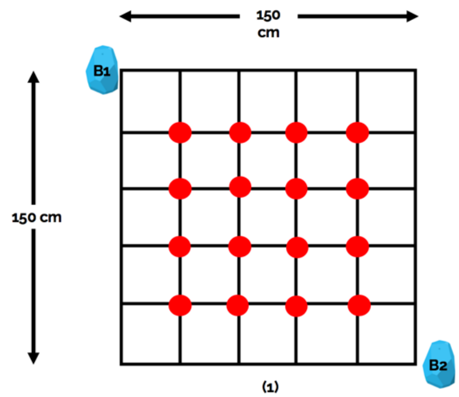

Fig 2. Experimental setup for Scenario 1

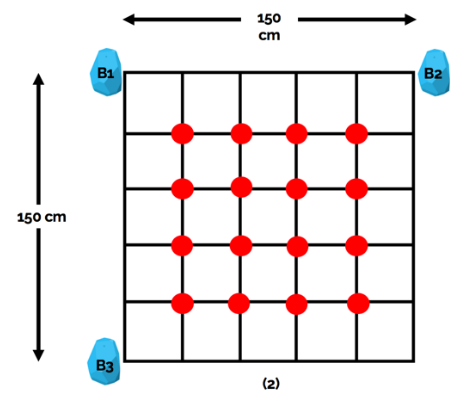

Fig 3. Layout for Scenario 2

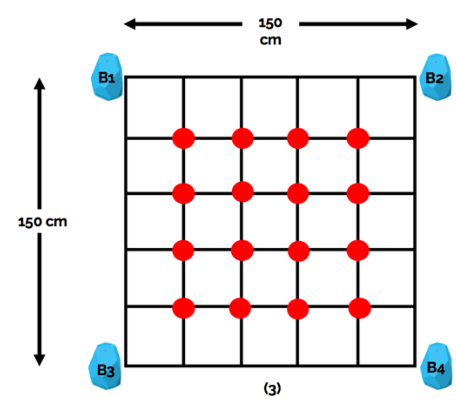

Fig 4. Layout for Scenario 3

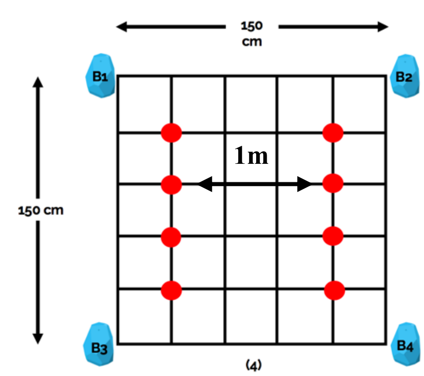

Fig 5. Layout for Scenario 4

Our “test room” is taken as a 1.5m by 1.5m square, marked on a wooden floor. B1 ~ B4 represent Estimote Proximity beacons. We chose this device due to its support for iBeacon protocol and over-the-air reprogrammable features [10]. Each BLE Beacon is powered by a coin cell which can last approximately 1.5 ~ 2 years on iBeacon mode[10]. The receiving device is a Raspberry Pi model 4.0 running Debian Buster 2019, Kernel 9.19. Each of the yellow points marked on the floor represents one grid point labelled through 1 to 16 (See Figure 4). Note that, each of the ordinal number marked here is a “class label” that the model has to predict after “training” on the collected data. The raspberry pi communicates with a Windows computer over SSH communication. Software to parse iBeacon advertisement is adopted from [11]. Python modules like Numpy, Scikit learn, Pandas, Matplotlib were used for data analysis.

V. Experiment Methodology

Using the Estimote app [10], all the BLE Beacons were switched to iBeacon mode and assigned a unique minor number. For example, B1 has a minor number of 1000, for B2 it is 2000 and so on. The BLE Beacons will keep on sending out an “advertisement” packet as long as its power lasts in the data format shown in Figure 6.

Table 1 iBeacon Scan Deconstructed

iBeacon MAC Address	iBeacon UDID	iBeacon Major Number	iBeacon Minor Number	TX Power at 1m	RSSI
cf:68:cc:c7:33:10	b9407f30f5f8466eaff925556b57fe6d	13072	52423	-74	-78

Fig 6. Beacon Data Packet

As introduced in Section I, an iBeacon BLE Beacon functions only to broadcast a data packet within its operational area. To understand how this is used to estimate position, we present the analogy of a “lighthouse”. A lighthouse shines a light of fixed intensity into the ocean. A ship can “estimate” its relative position with respect to the light house by tracking how intense the light beam gets as it draws near to the lighthouse. The same principle is applicable for the RSSI values. If the receiver is close to the beacon, the value is less and progressively increases as the receiver moves farther away.

After configuring the BLE nodes and setting up the experiment area, we placed the Raspberry Pi in each of the yellow grid points (Figure 1) and collected 200 samples per point. Table 1 shows a sample dataset for Scenario 1.

Table I
Sample Dataset for Scenario 1

Index	B0	B1	Label
s1	-85	-70	1
s2	-88	-69	1
s3	-75	-84	1

Here “B0” and “B1” corresponds to the position of the BLE nodes shown in Figure 2. In terms of Machine Learning lingo, our “feature vectors” are the columns that corresponds to the RSSI value from a node whilst the “Label” column is the expected outcome.

After completion of data collection, we checked the data for any missing values. This was found to show up as “0” in certain samples. In these cases, we simply copied the RSSI value from the previous sample. The preprocessed data was then split into 90 -10 train-test split. After that, we applied four machine learning models to the data and compared the results to answer our two stated hypothesis. Note that, Scenario 1~3 was used to answer Hypothesis 1 and, Scenario 4 was used to answer Hypothesis 2. Additionally, an automatic hyperparameter tuning technique called GridSearchCV[12] was used to search over the hyperparameter space of each of the four models to obtain the best parameters that fit the data well without overfitting. Test criterion was a Mean Squared Error function.

VI. Machine Learning Models

In this section we briefly touch upon the three family of Machine Learning models that constituted the four models introduced in Section V.

A. Support Vector Classifiers

Support Vector Machines (SVMs) belong to a set of supervised learning methods that can be used for multilabel classification, regression, and the detection of outliers [13]. The algorithm uses labeled training data to output a hyperplane which categorizes new data [13]. The accuracy of SVC can vary depending on multiple tuning parameters: kernel functions, regularization (C) and gamma (g). Kernel functions are a mathematical function which transforms input data to a form required by the support vector classifier. For our project we chose two kernel functions: Linear and Radial Basis [14].

B. Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. [15]. The advantages of using a decision tree model include they are easy to understand, they are able to handle both numerical and categorical data, and they are resistant to outliers. Detail discussion on Decision Tree can be found [15].

C. Random Forest Classifier

Random forest classifier (RFC) is an ensemble classifier which fits multiple decision-trees and estimates a class label based on “majority of voting” [16]. Each tree in the forest will make a classification prediction, the random forest classifier will aggregate the prediction from the tress and select the classification with the most votes as the final prediction [10]. Advantages of utilizing the random forest classifier include high accuracy, efficient use on large data sets, and the ability to handle thousands of input variables. However, random forest classifier models are also prone to overfitting in the same manner as the decision tree models

[16]. Detail discussion on Random Forest Classifier can be found on [16].

VII. Results and Discussion

Table II shows the outcome of the machine learning models for Scenario 1~3 respectively. DTREE and RF stands for Decision Tree and Random Forest Classifiers respectively.

Table II
Results for Scenarios 1-3

# nodes	SVC, Linear	SVC, RBF	DTREE	RF
2	40%	52%	47%	49%
3	41%	60%	55%	61%
4	55%	62%	64%	71%

From Table II, it is evident that, the accuracy for all the four ML models increases as the number of BLE nodes increases. Thus, we conclude Hypothesis I is true for our experimental environment. Additionally, the Random Forest Classifier performed significantly well in contrast to the other three classifiers. However, even with an ensemble method, our best accuracy is only 71%. We attributed this low accuracy number to the fact that, each of the grid points are only 30cm apart whereas the "common" rule of thumb was to separate them grid points by 1.5 ~ 2 meters [8]. This caused adjacent grid point pairs to have overlapping RSSI values as shown in Figure 8.

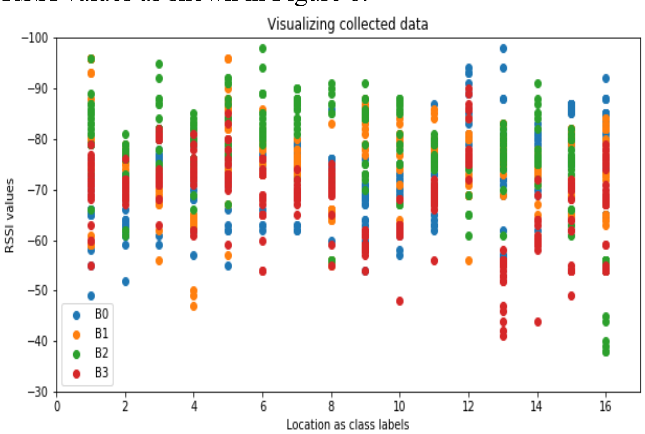

Fig 8. A subsample of data from Scenario 4

Thus, we formulated Hypothesis II to investigate the affect of grid point separation to the accuracy of the classifiers. Table III shows our findings from Scenario 4.

Table III
Results from Scenario 4

# nodes	SVC, Linear	SVC, RBF	DTREE	RF
2	74%	81%	82%	89%

Each of the ML model has shown to gain a significant improvement in accuracy against Scenario 3. Thus, we conclude Hypothesis 2 is true. We justify this improvement by pointing out that, with a "sparse" location grid, there was less overlap of RSSI data between adjacent positions and, thus confirms the observation made in [8].

VIII. Conclusions and Future Work

In this semester project, we successfully demonstrated the application of an Indoor Positioning System using BLE Beacons and Machine Learning models. We tested two hypothesis and showed them to be true provided certain assumptions were taken into consideration. However, there are multiple shortcomings in our experiment. The first being the experimental set-up is not reflective of a real-world Indoor Positioning System. A true system would be implemented on a much larger scale, and not in a 1.5m by 1.5m grid space. Secondly, during data collection, we did not move the Raspberry Pi which is not optimal as in real-life a person would be moving on and between "fingerprint" points. Finally, we did not consider the presence of WiFi and other BLE devices which may had contaminated the RSSI values. In the future, we plan to experiment on a larger lab space and use an Autonomous Ground Vehicle (AGV) that will move through the "fingerpoint" locations in during "training" phase. Additionally, we will test the "fingerprinting" method with more than 4 Beacons to check if Hypothesis I is still true and also incorporate the kNN unsupervised learning model.

IX. References

1. Amjad, Ayesha. "What Is the Importance of Adding Location in IoT?" Digital Myth Solutions, Digital Myth Solutions, 13 Mar. 2019, blog.dms-ksa.com/2019/03/13/importance-adding-location-iot/.
2. L. Hui, H. Darabi, P. Banerjee, and L. Jing, "Survey of Wireless Indoor Positioning Techniques and Systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, IEEE Trans. Syst., Man, Cybern. C, vol. 37, no. 6, pp. 1067-1080, 11/01/, 2007.
3. V. Cantón Paterna , A. Calveras Augé , J. Paradells Aspas , and M.A. Pérez Bullones , "A Bluetooth Low Energy Indoor Positioning System with Channel Diversity, Weighted Trilateration and Kalman Filtering," Sensors vol. 17, no. 12, pp. 2927, 2017.
4. True Range Multilateration." Wikipedia, Wikimedia Foundation, 3 Apr. 2020, en.wikipedia.org/wiki/True_range_multilateration.
5. "BLE Beacon Technology Made Simple
6. Ashton, Kevin. "That 'internet of things' thing." RFID journal 22.7 (2009): 97-114
7. Lymberopoulos, Dimitrios, et al. "A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned." Proceedings of the 14th international conference on information processing in sensor networks. 2015.
8. RSSI-Based Indoor Localization With the Internet of Things

9. Yanagaimoto el at. Indoor Positioning Estimation Using BLE Beacons
10. Technical specifications of Estimote BLE beacons, Available at: https://forums.estimote.com/t/technical-specification-of-estimote-beacons-and-stickers/7297
11. https://www.switchdoc.com/beaconair-ibeacons/
12. Scikit Learn, “GridSearchCV”, Available at: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
13. Scikit Learn, “Support Vector Machine” Available at: https://scikit-learn.org/stable/modules/svm.html
14. Data Flair, “Kernel Functions”, Available at: https://data-flair.training/blogs/svm-kernel-functions/
15. Scikit Learn, “Decision Tree”, Available at: https://scikit-learn.org/stable/modules/tree.html
16. Scikit Learn, “Random Forest Classifier”