ID: s3826723

Student Name: Kristian James P. Guinto

I certify that this is all my own original work. If I took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in my submission. I will show I agree to this honor code by typing "Yes": *Yes*.

Comparison of KNN and Decision Tree Classification Algorithms for Bluetooth Fingerprinting using RSSI

Author:

Kristian James P. Guinto

RMIT University

s3826723@student.rmit.edu.au

10 June 2020

Table of Contents

Indoor positioning is a developing technology with a wide range of industrial and commercial applications. One of the methods being explored utilizes the power received from short range wireless transmitters such as Bluetooth. This method, often called fingerprinting, works by creating a database of power levels received from transmitters on different locations in a space, then comparing power levels received from an unknown location to this database to make inference. In this paper, K-Nearest Neighbours and Decision Tree algorithms were trained and tested using Bluetooth Received Signal Strength Indicator (RSSI) to perform fingerprinting. Moving average was used as substitute for filters to smoothen the RSSI readings from the dataset and it was found that KNN was able to outperform decision tree with around 10% difference in mean accuracy.

1. Introduction

Location detection is a widely researched and commercially applied field of technology. Companies make use of this technology to offer location-aware services for indoor and outdoor navigation, information retrieval as well as social applications. The accuracy and availability required by outdoor applications have been provided by Global Navigation Satellite Systems (GNSS) which uses satellites to triangulate a receiver's location. However, as discussed by Motte [4] the accuracy of this technology to determine indoor locations diminishes greatly due to multipath effects and the lack of line-of-sight signals. This limitation meant that alternative technologies have to be explored for location detection indoors.

Technologies	Indoors	Outdoors
Network of sensors	1–5 m	Not suitable
RF ID	<1 m	<1 m
WLAN	few meters	Not suitable
UWB	≈10 cm	Not suitable
Cell-Id	500 m to 10 km	100 m to 10 km
E-OTD (2G)/TDOA (3G)	≫200 m	<100 m
GNSS	Not available	≈5 m ↘
A-GNSS	10 m to not available	≈5 m ↘
Pseudolites	≈10 cm	≈5 m ↘
Repeaters	≈1–2 m	≈5 m ↘
Inertial	<1 m (time dependent)	<1 m (time dependent)
Image pattern recognition	≈1–2 m	A few meters
SLAM	<1 m	A few meters
…	…	…

Table 1 Technologies being used for indoor positioning along with indoor and outdoor accuracy [3 p. 25]

Samama [3] discusses the different technologies that are currently being used and studied for indoor positioning. As we can see in figure 1, different levels of accuracy can be achieved by various technologies. However, he [3 p. 18] argues that for indoor positioning 'the difficulty arises when one wants to combine technical requirements: accuracy and simplicity, terminal cost, infrastructure cost, autonomous mode, etc.' These combinations of different technical aspects are necessary due to variety of use cases for indoor positioning. For example, indoor navigation on shopping malls, airports and museums would require a wide coverage, e.g. covering the entire space, about less than 1m of accuracy and the knowledge of absolute orientation of the receiver for it to be useful to consumers. Indoor positioning combined with robotics are also used to improve efficiencies in manufacturing. In this setting, the indoor positioning system is required to be able to guide multiple units to avoid collision and find the most efficient pathways. Samama [3] also argues that there is currently no one solution that offers indoor positioning capabilities comparable to outdoor positioning achieved by GNSS.

In this paper our focus will be on WLAN-based approaches, specifically using Bluetooth technology often called Bluetooth fingerprinting.

WLAN Indoor Positioning

Wireless Local Area Network or WLAN refers to a computer network designed to wirelessly connect multiple computing devices that are close to each other. For indoor positioning systems (IPS) two WLAN standards are often used, WiFi and Bluetooth.

WiFi, described in IEEE 802.11 standards, is used to offer devices wireless connection to the internet. Access points (AP) are used to create local area networks that devices can connect to for sending and receiving packets. For IPS applications, data collected from AP's of a larger WiFi network is used to determine a device's location. IPS based on WiFi has the advantage of having existing infrastructure already deployed on commercial spaces.

On the other hand, Bluetooth is a short-range wireless network standard used to create Personal Area Networks (PANs) for data transfer between devices. As described by Mier [6], 'the main features of Bluetooth technology are Bluetooth devices with low cost, low power consumption, small range, robustness and global use.'. The low cost and low power consumption of Bluetooth is a critical advantage when deploying new infrastructure to create IPS.

WLAN positioning makes use of signal strength-based probability using Received Signal Strength Indicator (RSSI). Signal strength is used as an indicator of location because of its relationship with distance as described by the *Log-Distance pathloss model*:

$$RSSI = -10n\ log_{10}\frac{d}{d_o} + A_o$$

Here *d* represents the distance of a receiver from a beacon, A_o is the RSSI measurement at distance d_o and *n* is a constant dependent on the environment. This equation models loss of signal strength with respect to distance in an indoor environment using different values of *n.*

Signal strength-based probability approach for indoor positioning makes use of RSSI from beacons laid out in a space to predict a receiver's likely location. A necessary component of this method is creating some kind of map or database of RSSI levels received from different locations for each beacon in a space. Location inference can then be made by comparing the RSSI values received from an unknown location to the RSSI map or database.

Samama [3 p. 43] argues that the principal challenge of using RSSI positioning is that it requires multiple transmitters to create a unique set of RSSI values that can identify all possible locations. Additionally, there are also challenges in processing RSSI values as signal propagation is often affected by diffraction and absorption introduced by the obstacles in an indoor environment. These make RSSI noisy and erratic. As a solution to this, researches often use filters to pre-process the data before being used for positioning algorithms. An often-used filter to solve this problem is known as Kalman filters. Zhou et al. [1] shows an implementation of positioning that uses RSSI and Kalman filters. They presented how Kalman filters can make unstable readings stable which makes location predictions more accurate.

In this paper, readings from Bluetooth *ibeacons* were explored for indoor positioning. Although Kalman filters are the widely used pre-processing method, a much simpler moving average was used since the objective of this paper is only to compare the performance of two algorithms for Bluetooth fingerprinting. These two algorithms are K Nearest Neighbours and Decision Tree and will be discussed on the next sections.

K Nearest Neighbours

K Nearest Neighbours (KNN) is a simple but effective algorithm used in classifying observations into different classes [8]. A simple description of how this algorithm works is it tries to classify a new observation based on the classification of its *k* nearest neighbours. Nearness is determined by treating a set of numerical values representing the observation as a vector and computing its distance (e.g. Euclidean distance, Manhattan distance) to all available vectors in the model and finding the *k* vectors that have the least distance to it. The new observation will be classified the same as the majority class from the *k* nearest neighbours.

The crucial parameter of this algorithm is selecting the value of *k* or the number of nearest neighbours to consider in classifying an observation. Aside from that, another parameter that can be adjusted is the weights that are given to each *k* nearest neighbour. A uniform weight means that each neighbour is weighted equally and will only rely on frequency. A different weighting method is using distance which makes closer neighbours have more influence on the classification.

Decision Tree

Decision Tree algorithm works by breaking the dataset to form a tree structure composed of decision nodes and terminal nodes. The algorithm makes multiple splits to create terminal nodes representing the different possible classes. The tree structure will then be used to make classification predictions of unobserved data. Decision splits are made by forming the most homogenous groups that can be created on each decision node.

The parameters for a decision tree are used to control the size of the tree. This is an important consideration since if the tree is allowed to grow without bounds, it will create terminal nodes for each possible case that can be extracted from the data, resulting to overfitting. The parameters can control the maximum vertical and horizontal size of a tree as well as introduce constrains on decision points. Another parameter is how homogeneity is calculated. Possible criteria are *gini* and entropy, both of which are computations of homogeneity of one group.

2. Problem Statement

The objective of this paper is to compare the performance of KNN and decision tree when used as the algorithm to perform Bluetooth fingerprinting using RSSI data.

3. Methodology

This section will discuss the characteristics of the dataset that was used and how it was prepared for KNN and decision tree classification. Furthermore, the steps that were taken to explore and analyse the data will also be laid out. Lastly, the training, optimization and validation of the methods will also be discussed.

3.1 Bluetooth Low-Energy (BLE) RSSI Dataset

This dataset contains RSSI readings gathered from thirteen (13) *ibeacons* deployed in a library setup. The RSSI have negative values where a higher value represents shorter distances between the receiver and a beacon. Additionally, a -200 value was assigned to beacons that were out-of-range of the receiver. An iPhone 6s was used to collect RSSI readings and the measurements were done on six (6) different dates. A total 1420 data points were gathered with each point collected two seconds apart.

Aside from the RSSI readings, the dataset also contains location labels and timestamp information. The labels were given in the form of a letter between A and W followed by a 2-digit number between 1 and 18 to specify grid locations. A photo of how the grid was laid-out in the library and the locations of the 13 beacons is shown in figure 1.

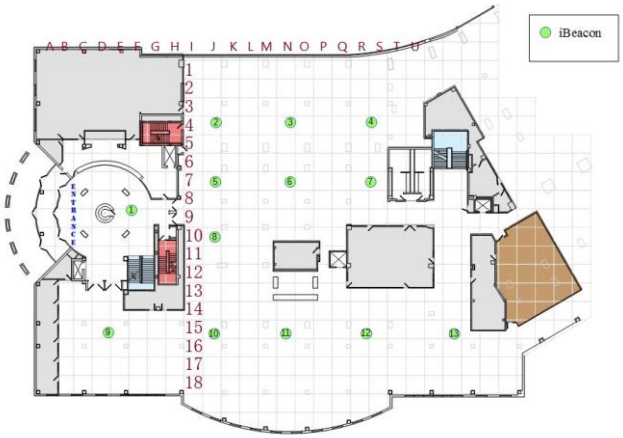

Figure 1 Grid layout of the library showing the positions of the 13 beacons

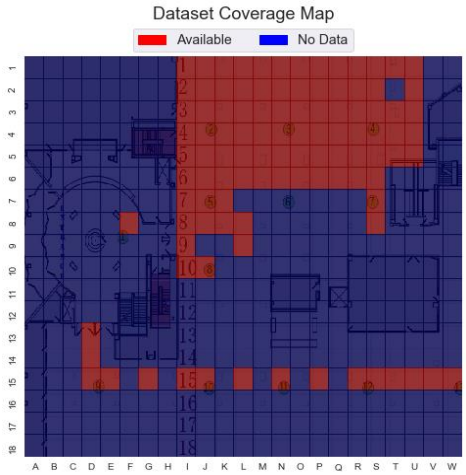

Figure 2 Coverage of location labels in the dataset

There were almost 400 different possible labels on the grid setup shown in figure 1. However, the dataset only contains 105 unique location labels which only covers approximately a quarter of the entire area as shown in figure 2. This puts a limit to the prediction capability of any classifier. It can also be seen in figure 3 the unbalanced distribution on the number of data points for each label. Furthermore, the valid RSSI readings, defined as a negative value greater than -200, only takes a small proportion of the total dataset. Table 2 shows a breakdown of the frequency and proportion of valid readings per beacon.

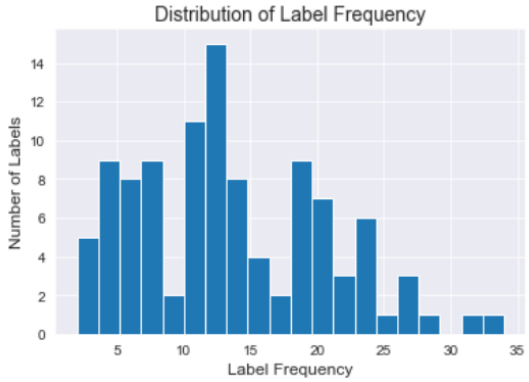

Figure 3

BEACON	FREQ	PROP
b3001	25	0.02
b3002	497	0.35
b3003	280	0.2
b3004	402	0.28
b3005	247	0.17
b3006	287	0.2
b3007	50	0.04
b3008	91	0.06
b3009	31	0.02
b3010	29	0.02
b3011	25	0.02
b3012	35	0.02
b3013	44	0.03

Table 2 Frequency & Proportions of Valid RSSI reading per Beacon

Possible implications of the scarcity of valid data points is a lower classification accuracy for the model and difficulty to generalize on unseen data. The unbalance on the data could also lead to high variance on the classification accuracy when validated multiple times.

This dataset was made by Mehdi Mohammadi and Ala Al-Fuqaha both from the computer science department of Western Michigan University and was downloaded from UCI machine learning repository. [10]

3.2 Data Preparation

The dataset was checked for both missing values (NA) and out of bounds readings, i.e. values less than -200 or greater than zero, and all of the beacons had no instances of both. The readings from each beacon were also plotted in a histogram to see the distribution of the data and to check for any outliers. This showed that the RSSI readings for all beacons except beacons 2 and 12 had readings ranging from around -90 to -55. Beacons 2 and 12 had outlier datapoints as shown in figure 4 and these were interpreted as erroneous readings and were removed from the analysis.

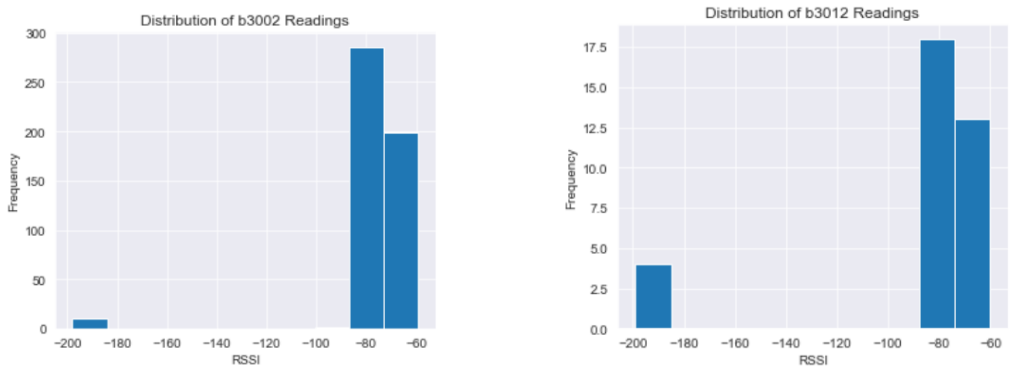

Figure 4 Outlier readings from beacons 2 and 12

3.3 Data Exploration

This section will discuss the exploration that were done focusing mainly on three different dimensions, 1.) the propagation of signal in space, 2.) Valid RSSI readings in a specific location and 3.) RSSI readings vs time. The relationship between readings from each beacon were also explored using scatter plots. However, there were no directly observable relationships between these variables that were not shown by the other explorations considered.

3.3.1 Propagation of Signal in Space

Given that RSSI has an inverse relationship with the logarithm of the distance as described by the *Log-Distance pathloss model,* the hypothesis for this exploration was that RSSI will decrease uniformly as it moves away from the source beacon. However, it is also noted that the RSSI could be affected by the various signal obstructions present in the area.

To observe this relationship an RSSI map plotting the mean RSSI from a single beacon was created. Additionally, a dot plot of the RSSI values with respect to distance from the beacon was also generated. Straight-line distances from the beacon were used as estimates and calculated via Euclidean metric using the square grid slices as units. The location of the beacon was set to the grid location with the most overlap, e.g. beacon 2 was located at J04 and beacon 12 at R15. The mean RSSI per distance calculated was also overlaid on the dot plot. Mean values were used to show the trend the RSSI readings had which was difficult to observe from the raw readings alone.

The RSSI maps generated for each beacon showed that there was an uneven propagation of the signals in space. This was evident from figure 5 where the signal from beacon 3 appears to propagate mostly towards north of the beacon. Potential reasons for this are the obstructions present during data gathering or signal interference from other beacons.

The dot plots on the other hand showed how the mean RSSI followed the expected trend of decreasing RSSI moving away from the beacon. The trend can be described as having a sharp decrease for distances close to the

beacon and a flatter but more erratic trend as it moves further. This trend somehow resembles the theoretical inverse log plot expected as described by the *Log-Distance pathloss model.*

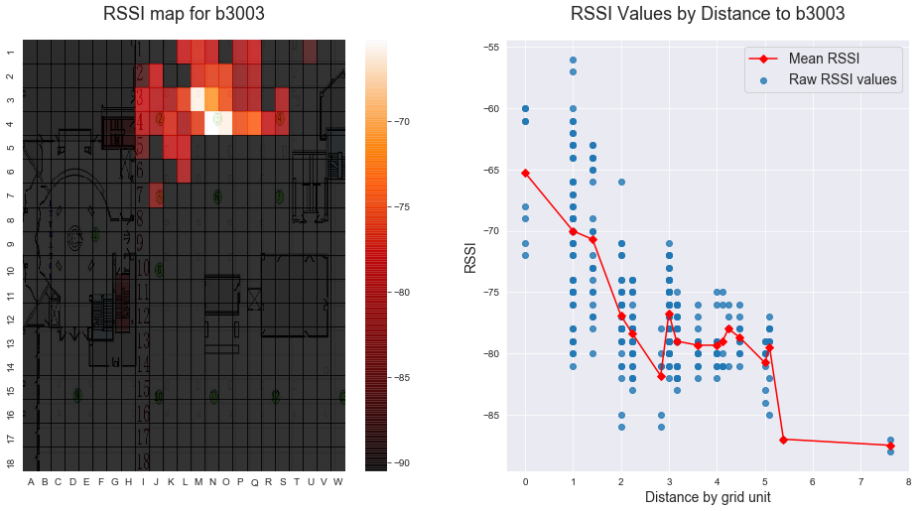

Figure 5 Propagation of signals from beacon 3 based on RSSI values

The dot plot also visualized one of the main challenges of using RSSI data for any analysis, its unstable nature. Raw RSSI measured from the same distances had wide variations with some readings one unit away having the same value as those for six or seven units away. This is a major problem for probability-based algorithms since these algorithms depend on uniformity of measurements on specific positions.

3.3.2 Valid RSSI Readings on Grid Locations

The spatial visualizations revealed that the trend of RSSI readings somehow followed what was theoretically expected. This section is an extension of the spatial exploration and made use of dot plots that shows the distribution of the valid RSSI readings from a specific location on the grid. The hypothesis was that RSSI should be higher for locations near a beacon and lower for those that are farther. An additional point of interest for this exploration is which beacons had valid readings on each location.

The locations were selected from high reading density areas i.e. areas with more valid readings. Three different specifications were explored, readings from a 1.) location of a beacon, 2.) location between two beacons, and 3.) location between more than 2 beacons.

The readings on the locations of beacons 2, 3 and 4 are shown in figure 6. As expected, the beacons that were closest to the location had the highest mean RSSI readings. However, similar to the readings from a specific distance, discussed on the previous section, there were also wide variations in the raw RSSI values. The readings from beacon 2 on its grid location (J04) were spread from -80 up to more than -60. The readings from beacon 4 had very similar range while those from beacon 3 had a narrower range, around -73 to more than -60. A final observation from these graphs was that aside from the RSSI values, the quantity of valid readings also looked to be an indicator of how close an unknown location was from a beacon.

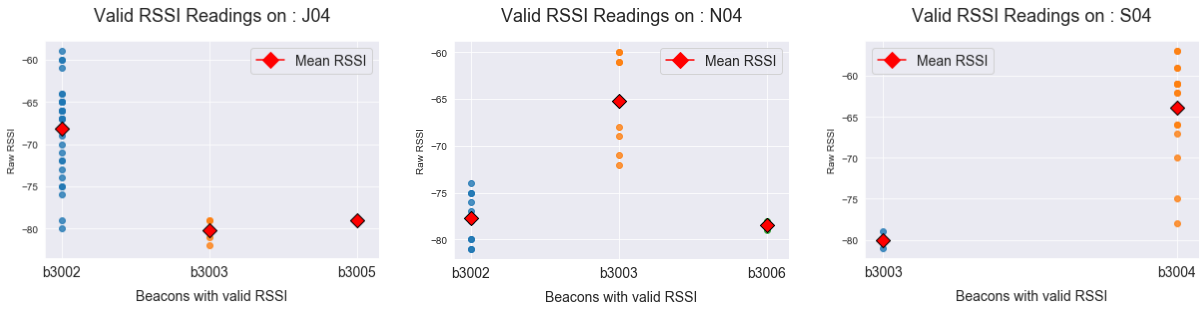

Figure 6 Valid RSSI readings from location of beacon 2 (J04), beacon 3 (N04), beacon 4 (S04)

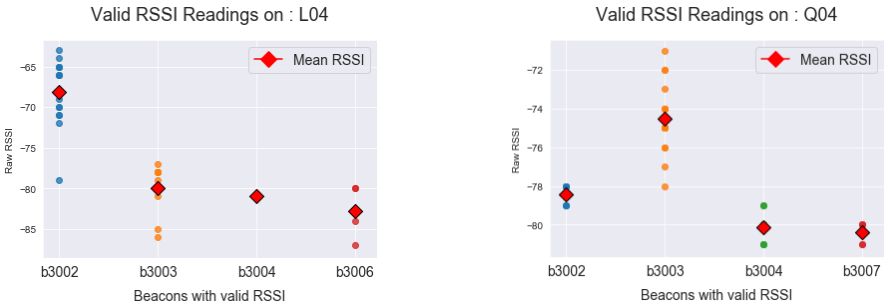

Figure 7 Valid RSSI readings from a location between two beacons; L04 (2 & 3), Q04 (3 & 4)

The second specification are locations in the middle of a line connecting two beacons. The left graph in figure 7 shows valid readings in L04 which was two units away from both beacons 2 and 3. Here it was observed that despite L04 having approximately equal distances from beacons 2 and 3, readings from beacon 2 were consistently greater than those from beacon 3. Readings from Q04 also contradicted the theoretical assumption. Q04 was estimated to be two units away from beacon 4 and three away from beacon 3. However, the readings from beacon 3 had much higher mean RSSI than those from beacon 4. Interestingly, the readings from beacon 2 also had a slightly higher mean value as those from beacon 4. These inconsistencies could come from the differences on the environment when the readings were made or the beacons themselves could have variations in transmission characteristics.

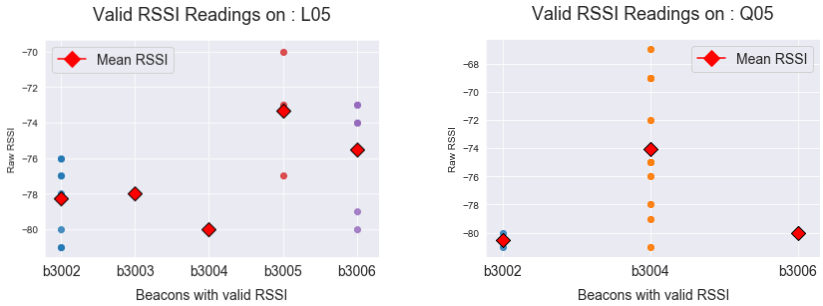

Figure 8 Valid RSSI readings for location between multiple beacons

The final specification are those locations that were between multiple beacons. Two grid locations were inspected, L05 and Q05. The readings from L05 show weaker signals received from beacons 2, 3 and 4 while significantly stronger signals from beacons 5 and 6. The situation for Q05 was very different with most readings coming from beacon 4 only.

To conclude, this exploration showed that valid RSSI readings were only received from beacons that were close to the location. However, shorter distances did not always translate to higher RSSI readings. This exploration also supported that the raw RSSI measured from a constant location can have wide variations in value. These variation in RSSI could be as wide as -80 to -55. Furthermore, it was also observed that aside from signal strength, the quantity of valid signals received from a beacon also acts as an indicator of location.

Spatial explorations revealed that RSSI approximated the theoretical inverse log trend between RSSI and distance. However, the raw RSSI readings had very wide variations which could greatly diminish the accuracy of probability-based positioning. These wide variations can be attributed to the random noise that a signal suffers coming from multipath effects and interference.

3.3.1 RSSI and Distance in Time

Aside from spatial location and RSSI readings, the dataset also has temporal information via timestamps for each row of the dataset. Analysing the timestamps reveal that the readings were done on six (6) different dates and measurements were made every two seconds. The hypothesis for this exploration was readings taken close to each other were related.

This section will discuss how RSSI readings and distance to beacon, derived the same way as explained in section 3.3.1, changed over time and a line plot was used to visualize these relationships. Some issues with this

technique was the visualizations covering the entire time span available were too wide and resulted to very crowded graphs. Another was the presence of out of range readings represented in the data as -200. These values were converted to nulls to keep them from showing up on the graphs, however the jumps on the data also made the graph harder to interpret. These issues were resolved by visualizing only significant slices from the data, i.e. time duration with valid signal readings.

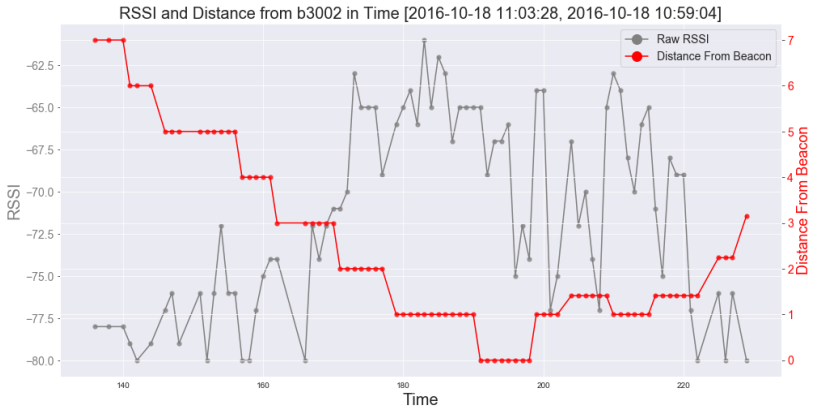

Figure 9

Figure 9 shows an end-to-end sequence of valid readings from beacon 2, i.e. start of valid reading to end of valid reading. It can be seen that the distance from the beacon (red line) decreases gradually from left to right until it reaches an inflection point then the distance gradually increases until the receiver becomes out-of-range. This U-shaped pattern was observed from readings of most beacons. The gradual changes show that the datapoints were sequential and represents the speed and direction of the movement of a receiver in reference to a beacon. The step-like pattern seen from the distance graph can be attributed to the way distances were calculated. However, it also suggests that consecutive readings were likely made from the same location or at least locations very close to each other. Finally, the inverse relationship between distance (red) and RSSI (grey) was also observed and the erratic nature of raw RSSI values were highlighted.

To add from the insights of spatial explorations, this section showed the sequential nature of the data with respect to time and how readings taken close to each other were likely taken on close proximity.

3.4 Feature Extraction & Model Selection

The goal of this project is to test the performance of two classification algorithms for fingerprinting using the BLE RSSI dataset. The models used, which were K Nearest Neighbours (KNN) and Decision Tree, were pre-determined and were briefly discussed in section 1. Both of these are supervised-learning algorithms that require labels to train. Obviously, to accomplish the project's goals the location variable from the dataset were used as labels and the RSSI values as features. However, as seen from the explorations discussed in section 3.3, the RSSI readings suffer from random noise resulting to erratic readings for similar locations. A common solution to this noise issue is to use filters to suppress the noise. Zhou [1] researched an indoor positioning method based on RSSI data that uses Kalman Filter which was described by Bulten [9] as a 'state estimator that makes an estimate of some unobserved variable based on noisy measurements.' Making use of these filters however are out of the scope of this project so as an alternative, a much simpler moving average will be used to simulate filtering the RSSI. This works by replacing the raw RSSI readings with the average from a set of consecutive values in a specific window.

The blue line in figure 10 shows how the moving average still followed the trend of raw RSSI values but with smaller variations for consecutive readings. As highlighted by the green box, it was effective in removing spikes on consecutive data points that represent the same location. However, it was not as effective in representing quick transition points, shown in the red box, where the raw RSSI readings were more uniform than its moving average counterpart. The decision to use this method was derived from the sequential nature of the RSSI readings observed and discussed in section 3.3.1. Furthermore, the mean RSSI was also observed to better follow theoretical trends than the raw RSSI values. The downside of using this method is it requires multiple data points to perform classification. For this project, the window size was limited to five consecutive data points or 10 seconds of readings which was determined to be a reasonable number of data points that can be

collected for a single prediction. Also, the data included in the window were determined in time since we know that the measurements were done on different dates and we only want consecutive data to be considered.

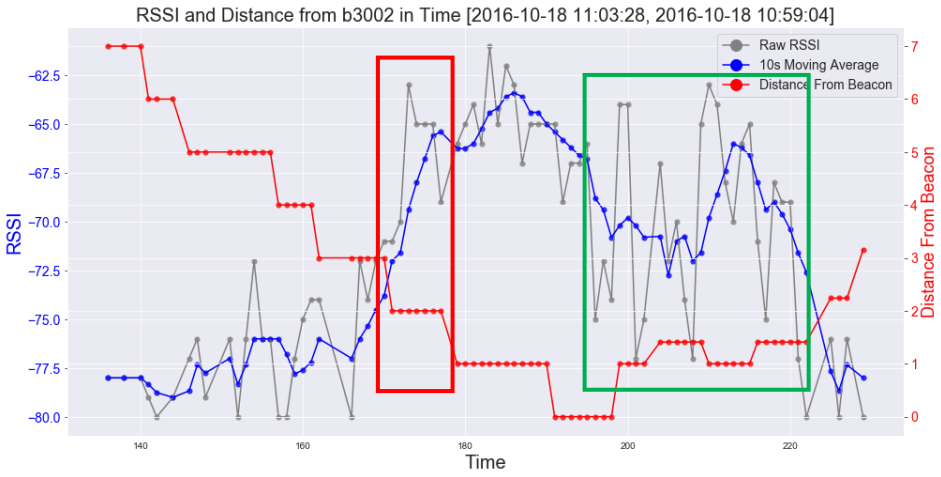

Figure 10 Moving average approximation of raw RSSI readings

Another consideration in applying moving average to this dataset was the presence of out-of-range measurements. To avoid these readings from affecting the average, they were replaced with null (np.nan) before calculating the moving averages and then replaced back to -200 afterwards.

3.5 Model Validation and Parameter Tuning

Another issue in this dataset was the unbalance in the number of datapoints available for each label. To minimize the effect of the unbalance, the data was randomly shuffled and divided into ten equal parts. The training-testing sequence was then performed ten times taking one part as test data and the remaining nine as training data. This method is called *K-fold cross validation*. Another reason for shuffling the data is because of its sequential nature with respect to time. Without shuffling, readings from the same locations will be grouped together introducing bias to the validation method.

A brute-force approach was taken to find the optimal hyperparameters. This was implemented using *GridSearchCV* library available from *sklearn* and the parameters considered for both models are given below.

Algorithm	Parameters Considered	Optimal Parameters
K Nearest Neighbours	Weights - Uniform and distance n_neighbors - From 1 to 10 p – [1,2]	Weights = Uniform or distance n_neighbors = 1 p = 1
Decision Tree	Criterion - Gini or entropy Min_samples_split - From 2 to 10 Mins_samples_leaf - From 2 to 10	Criterion = Gini Min_samples_split = 2 Mins_samples_leaf = 4

Table 3 Parameters considered for GridSearch and optimal parameters

For KNN optimization the most crucial parameter is the value of *n_neighbors*. Using a range of 1 to 10 showed that higher accuracies were observed with smaller *n_neighbors.* For decision tree, the optimization focused on the parameters that did not directly limit the size of the tree. This decision was made due to the large number of possible classes that were being considered. Similar to the case for *n_neighbors*, it was also found that smaller values for *min_samples_split* and *min_samples_leaf* can achieve higher accuracies so the range was just limited from 2 to 10.

10-fold cross validation was performed on two different sets of data using the optimal hyperparameters shown in the third column of table 3. The first set uses the raw RSSI measurements as features to train both KNN and decision tree and the second set uses the moving average to perform the same thing.

4. Results

Statistic	KNN using Raw RSSI	KNN using moving average	Decision Tree using raw RSSI	Decision Tree using moving average
Min, Max Accuracy	29.79%, 40.71%	41.13%, 47.52%	18.57%, 26.43%	29.29%, 39.00%
Mean Accuracy	33.75%	44.27%	22.49%	33.95%
Standard Dev. Acc	3.78%	2.12%	2.58%	3.64%
Mean Distance Error	1.72 units	1.20 units	2.14 units	1.69 units

Table 4 Summary statistics of the model validation

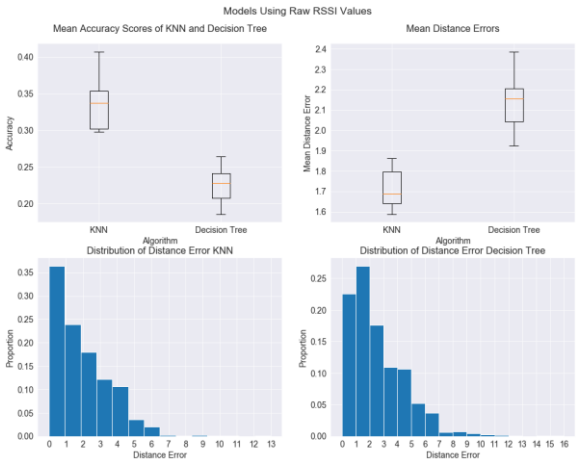

Figure 11 Validation results of models using raw RSSI

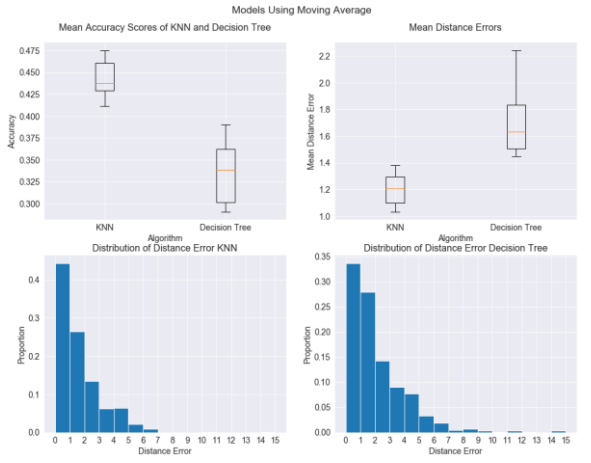

Figure 13 Validation results of models using moving average

KNN was able to outperform decision tree on both feature sets by around 10% greater mean accuracy. The best performing model was the KNN trained using the moving average RSSI having a mean accuracy of 44.27% (SD = 2.12%). This accuracy was around 10% better than the decision tree algorithm trained using the same data which had mean accuracy of 33.95% (SD = 3.64%). Additionally, filtering the noise, simulated by converting raw RSSI to moving averages, was also observed to increase the mean accuracy of both models by around 10%.

Distance error, defined as the distance between the predicted location and the actual location, was also used to validate the performance of the model. Distances were calculated the same way as discussed in section 3.3.1. The histograms show the distribution of the rounded distance errors from the 10-fold cross validation. From this, it was observed that all of the models except the decision tree with raw RSSI as feature set had a purely decreasing trend with zero distance error, i.e. correct classification, having the highest proportion. Furthermore, the mean distance errors were also calculated. For the model with the highest accuracy, the average distance error was 1.20 units away, which means that on average, the prediction of this model will be 1.20 units away from the correct location. The worst performing model, decision tree with raw RSSI as feature sets, only had an average distance error of 2.14 units away.

5. Discussion

KNN algorithm was able to outperform decision tree with close to 10% improvement in mean accuracy for Bluetooth fingerprinting using RSSI. Both models however had subpar performances as none of the combinations of model and feature sets were able to reach high-levels of accuracy. This could be attributed to the combination of how the models make inference and the nature of RSSI.

As we've seen from the various explorations, the RSSI values were inherently erratic with very wide variation of possible values which means that similar labels can possibly have very different feature values. For KNN, this means that the nearest neighbours of an unknown location could have different labels. These different labels however are likely to refer to locations that are close to each other. While for Decision Tree, this means that finding homogenous splits will be much more difficult and this could be the reason why decision tree had the lower accuracy. The huge number of possible classes could also be another reason why decision tree had lower accuracies.

Moving average was used to simulate a filter that suppresses the effects of noise on the data and smoothen the RSSI readings. This resulted to an increase of around 10% in the mean accuracy for both models. These findings again highlight the significant effect of noisy RSSI values on the predicting power of the classifiers. The increase in accuracy however comes at the cost of needing multiple data points to make predictions as the filter must also be applied to the features of the unknown data point. For this project, the window for the moving average was limited to only 5 consecutive datapoints which for this dataset spans 10 seconds. This was determined to be a reasonable amount of datapoints to balance the increase in accuracy with the constraint imposed by this method. It was also observed that increasing the window size also increases the accuracy of the classifier reaching mean accuracies as high as 70% for a window size of 120 seconds or 60 datapoints. However, implementing this method distorts the feature values effectively creating new data points instead of simulating a filter.

Distance errors were also considered to have a broader perspective on the performance of the models. The best performing model was able to make predictions that were on average only 1.20 units away from the correct location while the worst model had an average distance error of more than 2.00 units away.

Finally, another consideration for the models that were created is the limited coverage of location labels from the dataset. As shown in section 2.1, the unique labels in this dataset only covers around a quarter of the entire available space meaning this model will never be able to correctly classify locations coming the other 75%.

6. Conclusion

The objective of this project is to compare K-Nearest Neighbours and Decision Tree as algorithm in performing Bluetooth fingerprinting. The results of the analysis showed that KNN was the better suited algorithm for Bluetooth RSSI fingerprinting. KNN had consistently higher mean accuracy scores and lower distance errors when modelled using RSSI data.

Using the filtered RSSI to train a KNN model with optimized parameters, we were able to create a model that on average correctly predicts the location label 43.77% of the time. Additionally, predictions made by this model were on average 1.20 units away from the correct label. On the other hand, the decision tree was only able to achieve a maximum mean accuracy of 33.81% and mean error distance of 1.69 units.

7. References

[1] Zhou, C., Yuan, J., Liu, H. et al. "Bluetooth Indoor Positioning Based on RSSI and Kalman Filter." *Wireless Pers Commun 96*, 4115–4130 (2017). https://doi-org.ezproxy.lib.rmit.edu.au/10.1007/s11277-017-4371-4

[2] Y. Wang, Xu Yang, Yutian Zhao, Yue Liu and L. Cuthbert, "Bluetooth positioning using RSSI and triangulation methods," *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 2013, pp. 837-842, doi: 10.1109/CCNC.2013.6488558.

[3] N. Samama, *Indoor positioning : Technologies and performance.* Hoboken, New Jersey: John Wiley & Sons, 2019.

[4] Motte, Henk & Wyffels, Jeroen & De Strycker, Lieven & Goemaere, Jp. (2011). *Evaluating GPS Data in Indoor Environments. Advances in Electrical and Computer Engineering.* 11. 25-28. 10.4316/aece.2011.03004.

[6] Mier J., Jaramillo-Alcázar A., Freire J.J. (2019) *At a Glance: Indoor Positioning Systems Technologies and Their Applications Areas*. In: Rocha Á., Ferrás C., Paredes M. (eds) Information Technology and Systems. ICITS 2019. Advances in Intelligent Systems and Computing, vol 918. Springer, Cham

[7] Wikipedia n.d., *Wi-Fi*, Wikipedia The Free Encyclopedia, viewed on June 8, 2020 <https://en.wikipedia.org/wiki/Wi-Fi>

[8] Dr. Ren, Y. 2020, '*Practical Data Science: Classification*, lecture notes, COSC2670, RMIT University, Melbourne

[9] W. Bulten, "Kalman filters explained: Removing noise from RSSI signals." wouterbulten.nl https://www.wouterbulten.nl/blog/tech/kalman-filters-explained-removing-noise-from-rssi-signals/ (accessed June 9 2020.)

[10] M. Mohammadi and A. Al-Fuqaha and M. Guizani and J. S. Oh, *BLE RSSI Dataset for Indoor localization and Navigation Data Set*, UCI Machine Learning Repository, 2017, [online]. Available: https://archive.ics.uci.edu/ml/datasets/BLE+RSSI+Dataset+for+Indoor+localization+and+Navigation