Real-Time Location System Case Study

James Vasquez, Javier Saldana, Sara Zaheri

Table of Contents

1 Introduction 2
2 Methods 2
2.1 Data Source 4
2.2 Explanatory Data Analysis (EDA) 5
2.3 KNN Analysis 8
2.4 Weighted KNN Analysis 8
3. Result and Discussion 9
4 Conclusion 10
5 References 11
6 Appendix 12

Table of Figures

Figure 1. Error Rate vs K Value plots for the three different approaches 3
Figure 2. X & Y Histogram of the position of the points 5
Figure 3. The degree distribution 6
Figure 4. Degree distribution with 8 bins 6
Figure 5. Signal corrolation with time, positios, and degree 6
Figure 6. Signal distribution 7
Figure 7. Position & signal strength Bins Box Plot 7
Figure 8. Position & signal strength relationship 7
Figure 9. KNN Output & Results 8
Figure 10. Weighted KNN Output & Results 9
Figure 11. Performance Comparison of all Models 9

Real-Time Location System Case Study

James Vasquez, Javier Saldana, Sara Zaheri

1 Introduction

Real-time location systems (RTLS) are a significant emerging technology that becomes a vital part of life (1). RTLS automatically track and identify the location of people or objects wirelessly. A classic RTLS example is location tracking of objects in a room. This RTLS application is called an indoor positioning system (6). Hardware such as Bluetooth tags or cell phones are employed as mobile beacons to communicate with strategically placed access points around the room. These beacons can be attached to people, other devices, and even robots (4). A beacon produces a signal that is read by the access points around the room (3). Each signal provides data that used to determine the location of the mobile device within the room. Software applications utilize triangulation, trilateration or a combination of location determination algorithms which actively translate this location data and produce usable interfaces for people to identify the location of a beacon (2).

Applications for real-time location systems are vast and extend beyond a simple enclosed space. Advances in wireless technologies and proliferation of tracking tags have made real-time location systems widespread in manufacturing, inventory management, and navigation (3). Identifying the current location of a package is a popular application of RTLS technology. Tracking previous location history and tracing locations to predict future locations are also common tasks (7). Additionally, active tags can provide critical data on the temperature of a package or the blood sugar level of a patient along with location information (9).

In the current study, we aimed to investigate the accuracy of the previous work performed by Nolan and Temple Lang (1) in which, K-Nearest Neighbors model utilizing the access point has been considered. Furthermore, the authors of the mentioned study dropped a duplicated access point from their training dataset and used six of the seven access points for their K-Nearest Neighbors model. Moreover, they tried to extend their mean-based k-nearest neighbors approach by implementing a weighted K-Nearest Neighbors model to predict the location of previously unseen signal strength data.

2 Methods

In this paper, we considered the ways to improve the analysis by Nolan and Temple Lang (1) using indoor wireless signal strength data provided by Mannheim University. This data is generated from a single mobile device at 166 locations on one floor of a multi-story building. Eight orientation angles are considered at each location and 110 readings are taken for each (x,y) location, angle combination. For each reading, seven access points provided signal strength data to the mobile device. In the analyses that carried out by Nolan and Temple Lang (1), they implemented an indoor positioning system using signal strength data and a mean-based K-Nearest Neighbors model. The model is applied to predict the location of a mobile device using offline signal strength data.

The first task was to identify MACs which are at the same location. In order to locate these MACs, the team implemented a correlation method to identify these MACs. The team aggregated records based on (x,y) position. Signal strengths from each access point for each position are mean-aggregated, forming a dataset where each record is unique based on (x,y) position. Each position is associated with a vector of six signal strength values. From this subset of data, the team was able to produce a correlation table which helped identify strong correlations in the dataset. To identify the same MACs, it was aimed to find correlations that were higher than 0.9.

For a complete analysis, four different models were built using K-Nearest Neighbors from SKLEARN. In this method, K is a parameter which must be chosen. To determine best K value, an elbow method is employed. This analysis enabled us to select the best value for K. Figure 1 illustrates our result through elbow method.

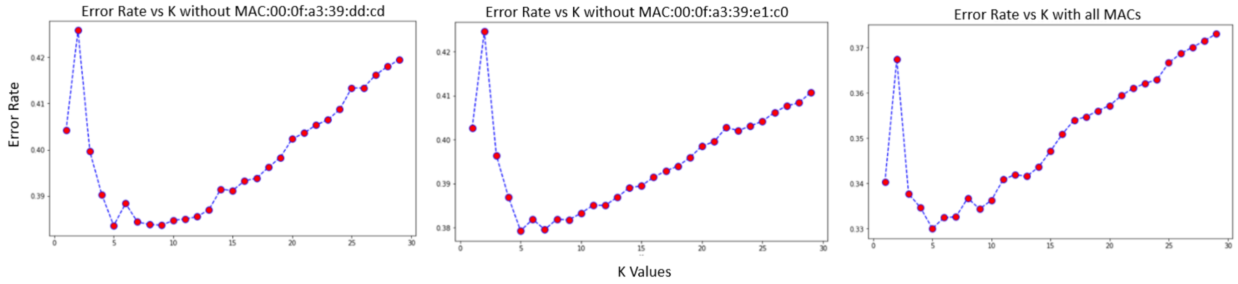


Figure 1. Error Rate vs K Value plots for the three different approaches. The team decided to use the K Value of five for the analysis

A function was written to optimize all models. The function provides modeling, plotting, and scoring of each model. The score for determining the best model, the Root Mean Square Error (RMSE) is used. RMSE allows an interpreter to determine how close real locations are to the model's predictive locations.

If y_i is the real location for a point and $\hat{y}_i$ is our prediction, $(\|y_i - \hat{y}_i\|)^2$ is the square error in which || is the Euclidian distance. Therefore, the root mean square error defines as following:

$$\sqrt{\frac{1}{n}\sum_{J=1}^{n}(\|y_i - \hat{y}_i\|)^2}$$

The six models that we applied to tackle this problem were as following:

- MAC analysis without 00:0f:a3:39:e1:c0
- MAC analysis without 00:0f:a3:39:dd:cd
- MAC analysis using both MACs
- MAC weighted analysis without 00:0f:a3:39:e1:c0
- MAC weighted analysis without 00:0f:a3:39:dd:cd
- MAC weighted analysis using both MACs

The first three analyses are designed using a uniform weight meaning that the models treat each point the same. This was done to determine whether using all MACs except one of them generates better results. The last three models are designed using the weighted mean. This approach allows the model to weight the points differently and compare against the uniform method.

2.1 Data Source

The dataset originates from:

http://rdatasciencecases.org/Data/offline.final.trace.txt

http://rdatasciencecases.org/Data/online.final.trace.txt

Each line of data represents a time in which the measurements were recorded. Multiple MACs are recorded at the same time which is indicated by multiple MACs within the same line. Lines with '#' are non-essential and will be skipped when reading the data (Python Function: process_raw_data).

The data is separated by ';' for the time, id, position, degree, MAC.

- Within the position feature this can be further broken down into position X / Y / Z components
  - example: pos=0.0,0.0,0.0
  - The values separated by ‘,' are in X, Y, Z in that order
- Feature MACs can be broken down into signal, frequency, and mode components
  - example: 00:14:bf:b1:97:90=-56,2427000000,3
  - For this particular MAC values are separated by ',' are in signal, frequency, and mode in that order
    - mode 3 = Access Point
    - mode 1 = Adhoc

For better use of the data each MAC will have 3 columns:

- 00:14:bf:b1:97:90_sig
- 00:14:bf:b1:97:90_freq
- 00:14:bf:b1:97:90_mode

The values within these columns represent the signal, frequency, and mode for that MAC ID. The position, time, and ID columns are then appended to the MAC columns.

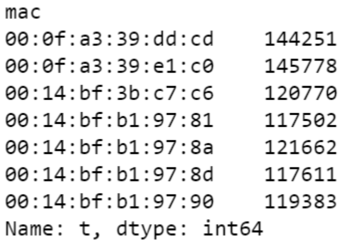
The end results of our data prep allow for a table which contains:

- Pos_x
- Pos_y
- Pos_z
- t
- id
- degree
- The MAC features for sig (signal)
- The MAC features for mode
- The MAC features for freq (frequency)

This format allows for a data frame used for modeling purposes.

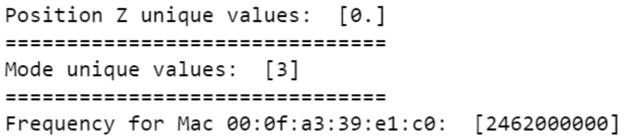
2.2 Explanatory Data Analysis (EDA)

The dataset includes data for over 20 MAC addresses. However, most of the MAC addresses have a considerably small number of instances. As a result, we trim the dataset and only focus on the 7 MAC addresses for this assignment:



```
mac
00:0f:a3:39:dd:cd    144251
00:0f:a3:39:e1:c0    145778
00:14:bf:3b:c7:c6    120770
00:14:bf:b1:97:81    117502
00:14:bf:b1:97:8a    121662
00:14:bf:b1:97:8d    117611
00:14:bf:b1:97:90    119383
Name: t, dtype: int64
```

Once we stack the MAC addresses and drop any instances where there are values missing, we find ourselves with a data frame that has 886,957 instances. As we begin to explore the attributes, we look into Position Z, Frequency, and Mode. A quick insight into these attributes reveals that there is only 1 unique value in the columns. The column Position Z is filled with 0 and the column Mode is filled with 3. In addition, the Frequency is MAC-specific, which makes it a categorical variable equivalent to that of the actual MAC address. As a result, we can proceed to drop these columns since they fail to bring any value to the data set.



```
Position Z unique values:  [0.]
===============================
Mode unique values:  [3]
===============================
Frequency for Mac 00:0f:a3:39:e1:c0:  [2462000000]
```

Position X & Y

With Position Z now removed, we are left with Position X and Position Y. Figure 2 clearly illustrates the position of the coordinates once plotted against each other in a grid.

Based on the distribution histogram of each position, we can now see the majority of the points are located in the hallways and entrance corridor. Considering the building's design, it would

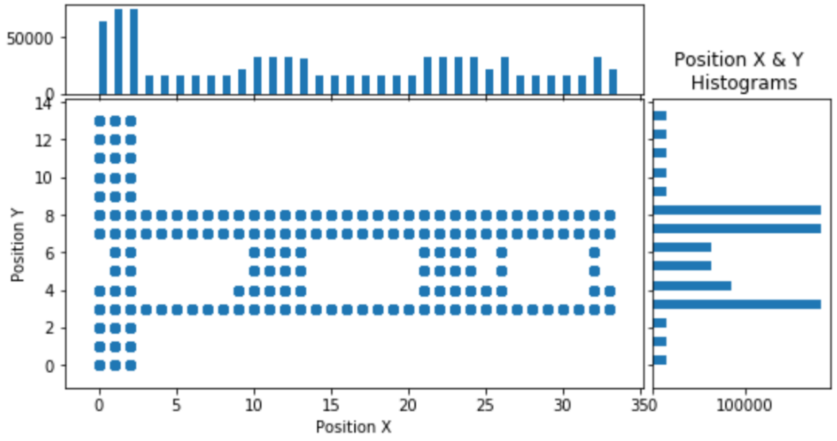


Figure 2. Plot of Position X & Y along with respective distribution histograms. Plotting the position not only illustrates the location of the devices but also the concentration points, which are clearly the main corridors of the building.

make sense for this to be the case since the device would have to travel through these corridors in order to get to its destination.

Degrees

The degrees in the data set pertain to the orientation of the device on a 360 scale. In this data set, 0 and 360 are equal to each other, which is why the data set only goes up to 359.9. The distribution of the degree attribute illustrates that there are what appear to be clear common degrees.

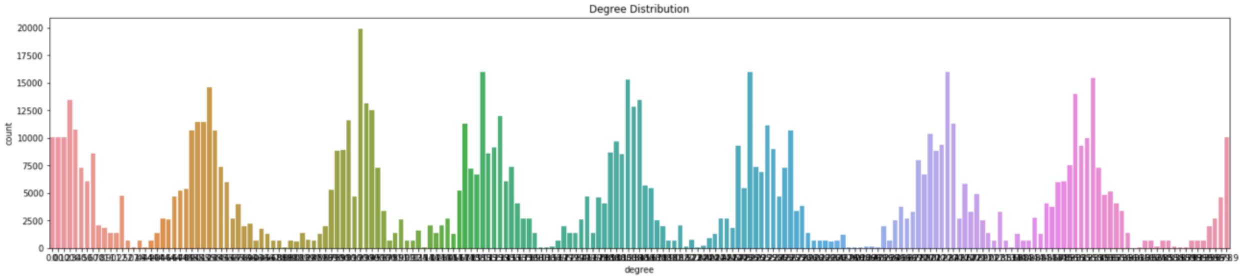


Figure 3. The degree distribution illustrates 8 peaks and valleys, which is the justification used to create the 8 bins to segment the degrees in further analysis.

This demonstrates we can see that there are clear peaks and valleys. In total, we can see 8 peaks throughout the dataset. When we categorize the values into 8 bins of 45 degrees each, we can see the data is now more evenly distributed.

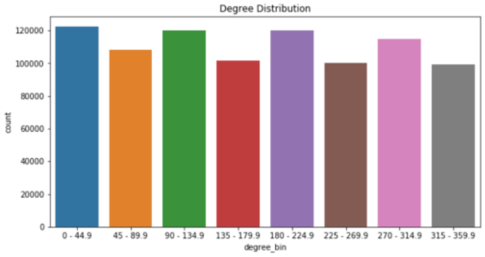


Figure 4. The degree distribution after the degrees have been placed in bins shows a much cleaner view into the direction of the device. We can see the hard points (N, S, E, W) have greater instances than most, which are expected considering the layout of the building.

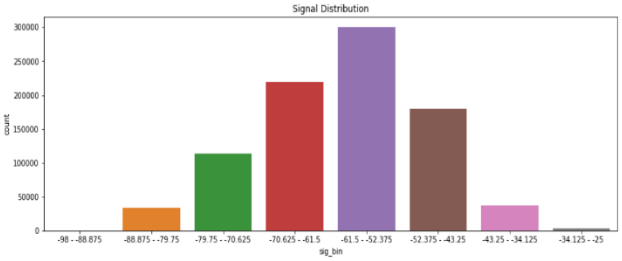


Figure 5. The signal distribution once segments into bins clearly preserves its gaussian distribution. Moreover, it illustrates the concentration of devices in the center than those in the edges of the signal bands.

Signal

When we apply the same analysis to signal, we can see there is a concentration in the center of the spectrum. As a result, this generates what appears to be a Gaussian distribution of signal. However, signal spectrum is too large so we can reduce it by creating bins. As a result, we create bins and can see that the Gaussian distribution was preserved by the creation of the bins as well.

Areas of Interest

In order to see the relationships of the attributes and their respective MAC addresses, we generated a heat map showing the correlation between the attributes by MAC address. In this heat map we can see there is a moderate correlation between Position X and Position Y and that is to be expected. As a device moves across the building, it is anticipated that the movement will not be vertical or lateral only. Therefore, we can dismiss this correlation with confidence. However, there is an area of interest in every MAC address. There appears to be a considerable correlation between Position X and Signal and this correlation rings true at every MAC address. Figure 7 clearly illustrates the standard deviation of the position over the signals. In the weakest signal, we can see a concentration at position above 30 with outliers starting around 25. We can see there is a narrow concentration of signal at higher position X but, and the mean shifts to the lower Position X values as the signal progresses.

In fact, when we look at the relationship between the two variables, we can see there is a sigmoid relationship. As signal increases, position X also increases slightly until about the middle where there is an exponential growth that is then followed by a slow in growth. This relationship could be the culprit of the high correlation between both variables that we see in the heat map. One of the ways we could address this issue could be by considering a log transformation to see if we can reduce the correlation. The only concern regarding the log transformation would be the negative values that pertain to the signal. As a result, it would be best to normalize all of the values and then review under a log transformation to see if this correlation is addressed.

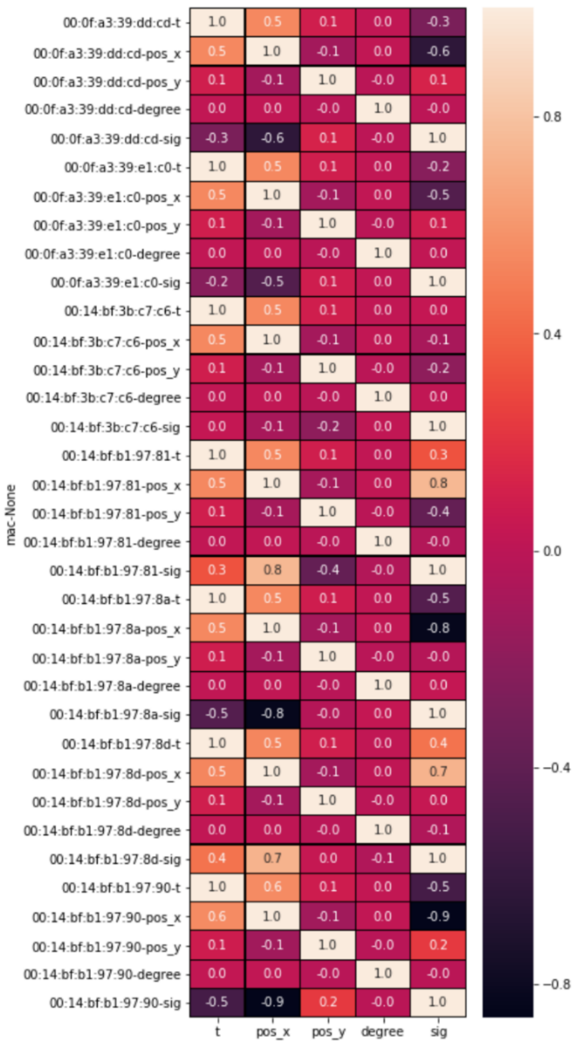


Figure 6. Heat index of correlation between attributes by MAC address. Points of interest lie in the relationship between Position X and Signal

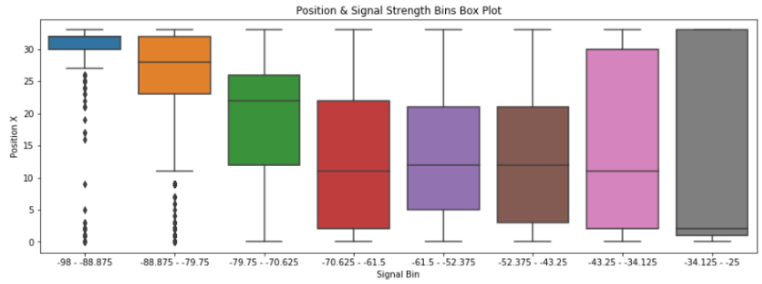


Figure 7. Position & signal strength Bins Box Plot. Illustrates the sigmoid relationship discussed where the mean is clearly shifting with each signal bin.

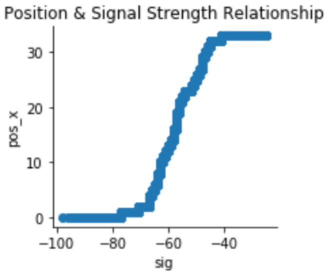


Figure 8. Position & signal strength relationship shows a clear sigmoid relationship between the two variables. A log transformation may help address this correlation.

2.3 KNN Analysis

K Nearest Neighbors allows the data to assume similar items exist close to each other. To make this a more meaningful analysis a K value needs to be assigned. For a more detailed explanation please see Section 2 Methods. To ensure a systematic approach for modeling the three different scenarios was taken a function was built in Python to fit, predict and plot the data (Function: analyze).

The results show that the RMSE score are as followed:

- RMSE without MAC cd: 2.29
- RMSE without MAC c0: 2.21
- RMSE with all MACs: 2.14

The image below plots a random sample of 100 points within the model. It is found that using all MACs produce a more accurate prediction.

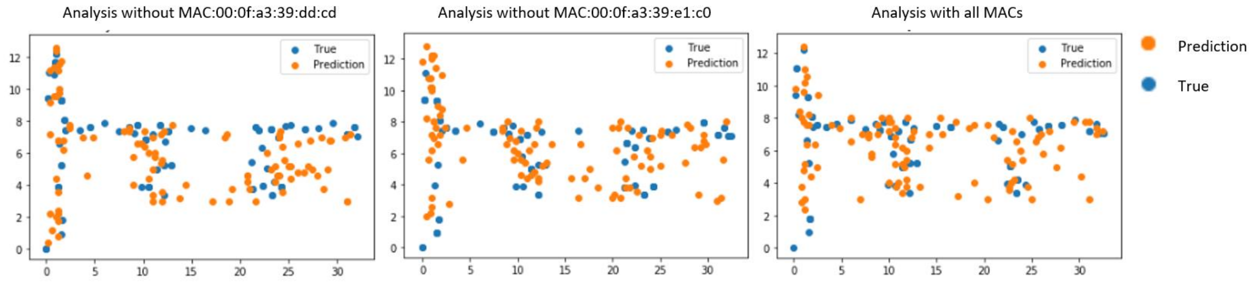


Figure 9. Figure shows the model prediction per scenario. Model using all MACs with KNN analysis provide the best results by use of the RMSE calculation.

2.4 Weighted KNN Analysis

The K Nearest Neighbors was completed again but in this effort the team decided to apply a “weighted” method to the analysis. Unlike the default method in KNN, the weighted method treats each point differently. It allows the point to have weight the priority depending on how close it is to the bins. The default KNN treats all points equally. This analysis was done to determine if the uniform (default) method is better than the weighted method. To model the data a systematic approach was taken, and a Python function was built to fit, predict, plot and score the models (Function: knn_predict).

The results show that the RMSE score are as followed:

- RMSE without MAC cd: 2.28
- RMSE without MAC c0: 2.21
- RMSE with all MACs: 2.13

The image below plots a random sample of 100 points within the model. It is found that using all MACs produce a more accurate prediction.

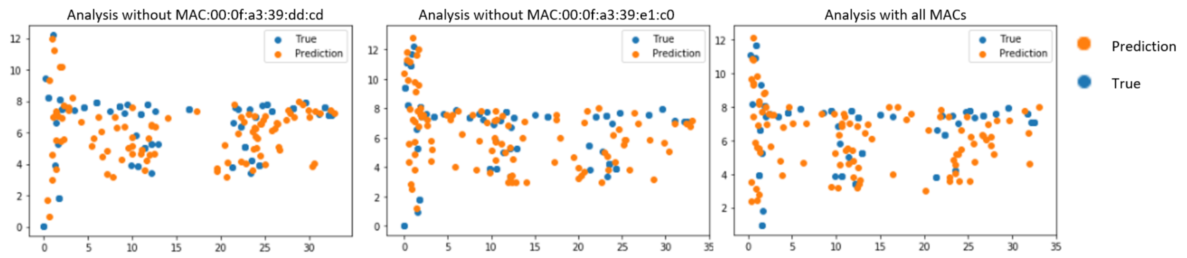


Figure 10. Figure shows the model prediction per scenario. Model using all MACs with Weighted KNN analysis provide the best results by use of the RMSE calculation.

3. Result and Discussion

As mentioned earlier, we employed 6 methods in order to determine the most useful model and our measurement value was the RMSE. A lower RMSE would indicate the regressor is close to the prediction, which is why we felt it would be the criterion of choice for performance of the models. The results for the models are as follows:

Model	*RMSE*	*RMSE (weighted)*
KNN w/o 00:0f:a3:39:e1:c0	2.21	2.21
KNN w/o 00:0f:a3:39:dd:cd	2.29	2.28
KNN w/ both MAC addresses	2.14	2.13

Figure 11 Output comparison of all KNN models. Table demonstrates there was no significant difference between the performance of the weighted approach versus the standard KNN model. In addition, removing either MAC only worsened the performance of the model.

Based on the results from the KNN and weighted KNN models, we find there is no significant difference in the RMSE of the models. When the MAC address 00:0f:a3:39:e1:c0 was excluded from the analysis, the RMSE was 2.21. Excluding the other MAC address, 00:0f:a3:39:dd:cd, only worsened the performance with an RMSE of 2.29. However, we find that when we include both, we improve the performance considerably by achieving an RMSE of 2.14. When we added weights to the model, we find that the performance appears to have an insignificant improvement of 0.01 at most.

The results appear to indicate that including and/or excluding either of the mac addresses fails to have a significant impact on the RMSE of the model. A potential explanation for this would be the substantial sample size that was utilized to train the model. Even when the mac addresses are excluded, there remains a substantial sample size which is able to cover the position grid of the data. A MAC address is essentially the physical address of a device. By removing the MAC addresses from the train data set, we were essentially removing two devices from the data set. Fortunately, it appears the remaining devices were able to maintain the integrity of the data and allow for an efficient prediction model using KNN and weighted KNN. Interestingly enough, we also found that utilizing the weighted method did not improve the RMSE of the model.

Like most models, KNN has its limitations. KNN is commonly referred to as a 'lazy learner' in the sense that it doesn't actually learn from historical data but instead uses it to find the 'nearest neighbor' for real-time decisions. This can become time consuming as the data sets grow since it must process all of the historical data in order to make a prediction. In addition, the simplicity of the algorithm also requires all features be normalized into an equal scale. In other words, if feature_a increases by 1 unit, then feature_b must also be able to increase by 1 unit which must be equal to the increase in feature_a. Failure to normalize the data would complicate the results produced. This creates even more problems as the dimensionality of the data set increases. With additional features comes an increase in noise. Being unable to filter out the noise, the performance of the model then begins to suffer as a result.

4 Conclusion

In this study we aimed to applied classification method for localization problem. Our problem was specifically an indoor localization. We grouped our training data set called offline data by their location. We applied explanatory data analysis to find the most important features. This analysis first confirmed the MACs that the researchers used for dropping are highly correlated. In addition, tells us correlation between different features and distribution of location.

Moreover, we proposed two different K-Nearest Neighbor approaches for classification. Using different weight was the only difference between these two methods. The result indicated quite similar performance as they are both based on K-Nearest Neighbor. Having both similar MACs does not really decrease the RMSE. The error with the better MAC is 2.21 and having both MACs decreases it just 0.07. That would be another evidence for having two MACs at the same location.

One problem in this study was having an unorganized dataset. The difference in location between online and offline situation will cause errors in our study. To address this issue, one can work with a compatible data set between train and test which could be an approach for future study.

5 References

1. Nolan, D., Temple Lang, D. DATA SCIENCE IN R: a Case Studies Approach to Computational Reasoning and Problem Solving. CRC PRESS, 2017.
2. Ojo-Osagie, O. “Dells Just In Time Inventory Management System.” Academia.edu, www.academia.edu/23256794/Dells_Just_In_Time_Inventory_Management_system.
3. Winick, E. “Amazon's Investment in Robots Is Eliminating Human Jobs.” MIT Technology Review, 4 Dec. 2017, www.technologyreview.com/the-download/609672/amazons-investment-in-robots-is-eliminating-human-jobs/.
4. ISO Committee. “Arameters for Air Interface Communications at 860 MHz to 960 MHz General.” International Organization for Standardization, 15 Jan. 2018, www.iso.org/standard/59644.html.
5. Barker, P., et al. "Performance modelling of the IrDA infrared wireless communications protocol stack." International Journal of Communication Systems, 2000. https://aetos.it.teithe.gr/~vitsas/publications/IntJCommSys_Peter.pdf
6. Boulos, Maged N Kamel, and Geoff Berry. “Real-Time Locating Systems (RTLS) in Healthcare: a Condensed Primer.” International Journal of Health Geographics, BioMed Central, 28 June 2012, www.ncbi.nlm.nih.gov/pmc/articles/PMC3408320/.
7. Madigan, D. et al. "Location Estimation in Wireless Networks: A Bayesian Approach." Rutgers University and Avaya Labs, 2006. http://dimacs.rutgers.edu/Research/MMS/PAPERS/wireless.ps.
8. Tarrío, P., et al. "Weighted Least Squares Techniques for Improved Received Signal Strength Based Localization." Sensors, 2011. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3231493/
9. Swedberg, C. “Toronto General Hospital Uses RTLS to Reduce Infection Transmission.” RFID Journal, 28 Feb. 2012, www.rfidjournal.com/articles/view?9266.

6 Appendix

The code is implemented in Jupyter notebook. We leave this part blank and submit the .ipython file together with the project.