# COMP4336/9336

# Mobile Data Networking 2020 Term 2

## Project

**Distance Estimation using Wireless Signal Strength**

**Stage – 1**

**Name: Rui Li**

**zID: z5202952**

# Data collection

## How many data:

I have collected both indoor and outdoor data from 1~10m. For indoor, I have collect data for at least 1 hour per distance; For outdoor, I have collect data for at least 10 min per distance. After apply the filter, there are 74717 valid data.

| Data Type | Data Count |
|---|---|
| Outdoor | 14549 |
| Indoor | 60168 |
| Grand Total | 74717 |

*Table 1.1 Number of valid data*

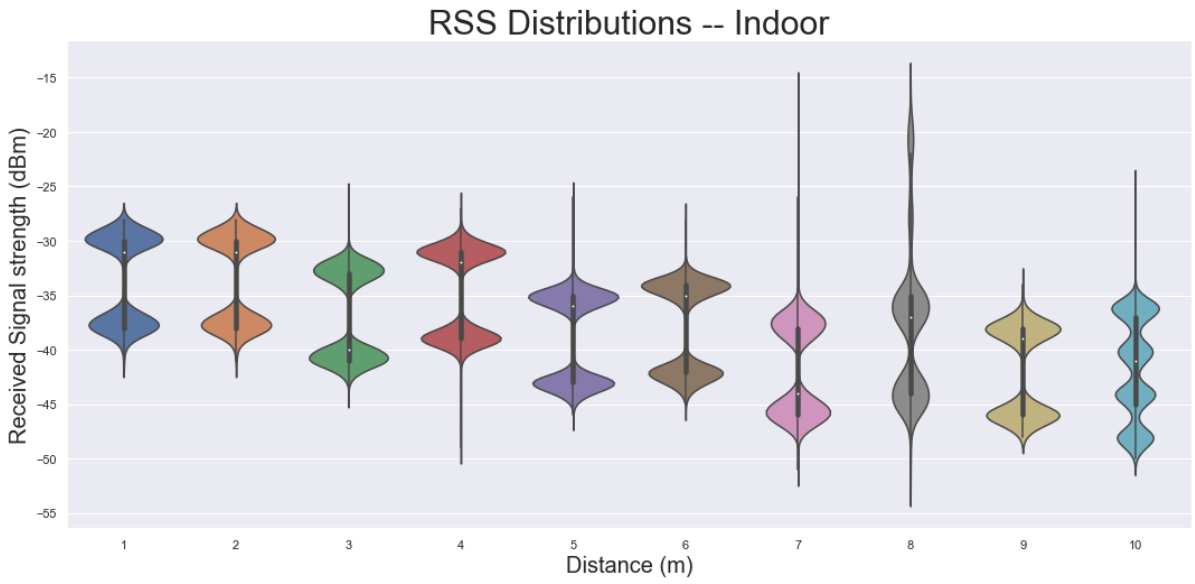
## Describe the data

### Average RSS:

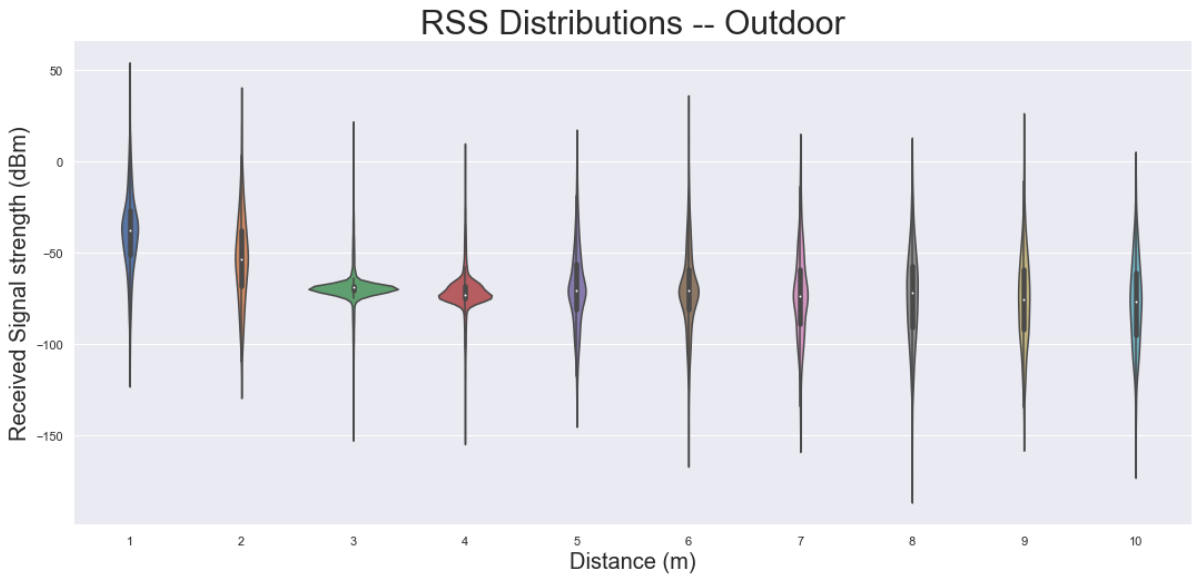
| | Distance (m) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **RSS (dBm)** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **Average** |
| Outdoor | -38.93 | -52.70 | -69.13 | -71.12 | -68.84 | -70.16 | -72.73 | -73.55 | -75.46 | -77.39 | -67.00 |
| Indoor | -33.63 | -33.63 | -36.89 | -34.76 | -38.85 | -37.93 | -41.87 | -38.03 | -41.93 | -41.76 | -37.93 |
| Average | -36.28 | -43.17 | -53.01 | -52.94 | -53.84 | -54.04 | -57.30 | -55.79 | -58.70 | -59.58 | -52.47 |

*Table 1.2 Average RSS for Indoor and outdoor*

### RSS Distribution:




*Graph 1.3 RSS Distributions for indoor*




*Graph 1.4 RSS Distributions for outdoor*

For the RSS distribution, I have drawn the violin plot to observe the distribution for both indoor and outdoor. I found there is totally different shape of the violin between indoor and outdoor, which means the standard deviation of indoor RSS Distributions is significantly higher than outdoor RSS Distributions. The reason of that may be multipath or fluctuations in transmit power.

# My naïve approach & Error reports:

## What is my approach?

I use the algorithm in scikit-learn to train a model and make the prediction. After comparing different algorithm in scikit-learn, I choose the LinearRegression as my algorithm.

## How do I perform this approach?

- Use Microsoft Network Monitor 3.4 to collect the data.
- Use Wireshark to apply the filter and output as CSV files.
- Use Pandas to merge/manipulate/visualise the dataset.
- Use Scikit-learn to train the model and predict.

## How do I choose the algorithm?

I perform a cross validation (cv=5) to compare the accuracies and mean square error of the algorithms:

| Algorithms | ACCURACY | | MSE | |
|---|---|---|---|---|
| | indoor | outdoor | indoor | outdoor |
| SVC | **0.557572** | **0.432155** | 7.264182 | 7.457828 |
| DecisionTreeClassifier | 0.557556 | 0.430916 | 7.264199 | 7.377803 |
| KNeighborsClassifier | 0.4952 | 0.38686 | 7.518103 | 7.862156 |
| GaussianNB | 0.224106 | 0.334433 | 12.79446 | 7.35948 |
| LinearDiscriminantAnalysis | 0.213901 | 0.341746 | 12.94205 | 7.55768 |
| LogisticRegression | 0.180013 | 0.335556 | 12.99915 | 7.474387 |
| **LinearRegression** | - | - | **6.234443** | **5.942667** |

*Table 2.1 Scikit-learn Algorithm performance comparison*

*PS: There is no accuracy score for LinearRegression due to the prediction output of the model is float but actual distance is int. (e.g. LinearRegression(-20dBm) → 3.15m, but actual distance is 3.00m).*

For accuracy, the SVC (C-Support Vector Classification) model is a good choice. But for MSE, the LinearRegression is the best choice.

I think the RSS-Distance model is much like a linear model as the data distribution. Therefore, the LinearRegression algorithm can always get a closer prediction result than other algorithms in this case (while it always cannot get the correct result).

## What is the performance:

| Performance | Indoor | Outdoor |
|---|---|---|
| Mean Squared Error | 6.314319 | 4.541636 |
| Explained Variance Score | 0.222491 | 0.091461 |
| Coefficiency | 4.292946 | -0.03222 |
| Coefficient of determination(R2) | 0.222369 | 0.091202 |

*Table 2.1 LinearRegression model performance*

## Further improvement:

- Try to add more data (especially for outdoor)
- More metadata/features for training model (e.g. weather, temperature, location)
- Explore other LinearRegression algorithm
- Apply more filter to clean the data (noise)

(end)