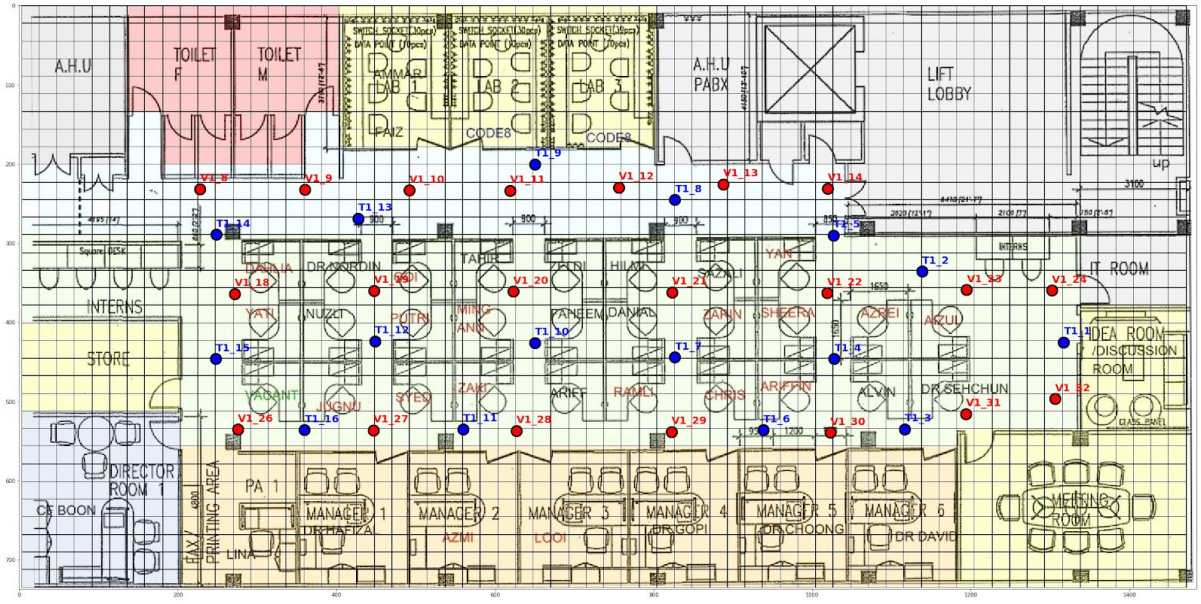
BLE Indoor Localization
Report 1

Prepared by: Tan Chin Huan

Prepared on: 16/12/2019

1 Data Collection

Training data is collected at 21 different locations where each location takes 2 minutes of transmitting bluetooth signal. The training data has 49663 records.
Test data is collected at 16 locations different from the training data collection where each location takes 30 seconds. The test data has 8458 records.



Training data fingerprints are marked red and test data collection are marked blue.

2 Data Characteristics

2.1 Data distribution

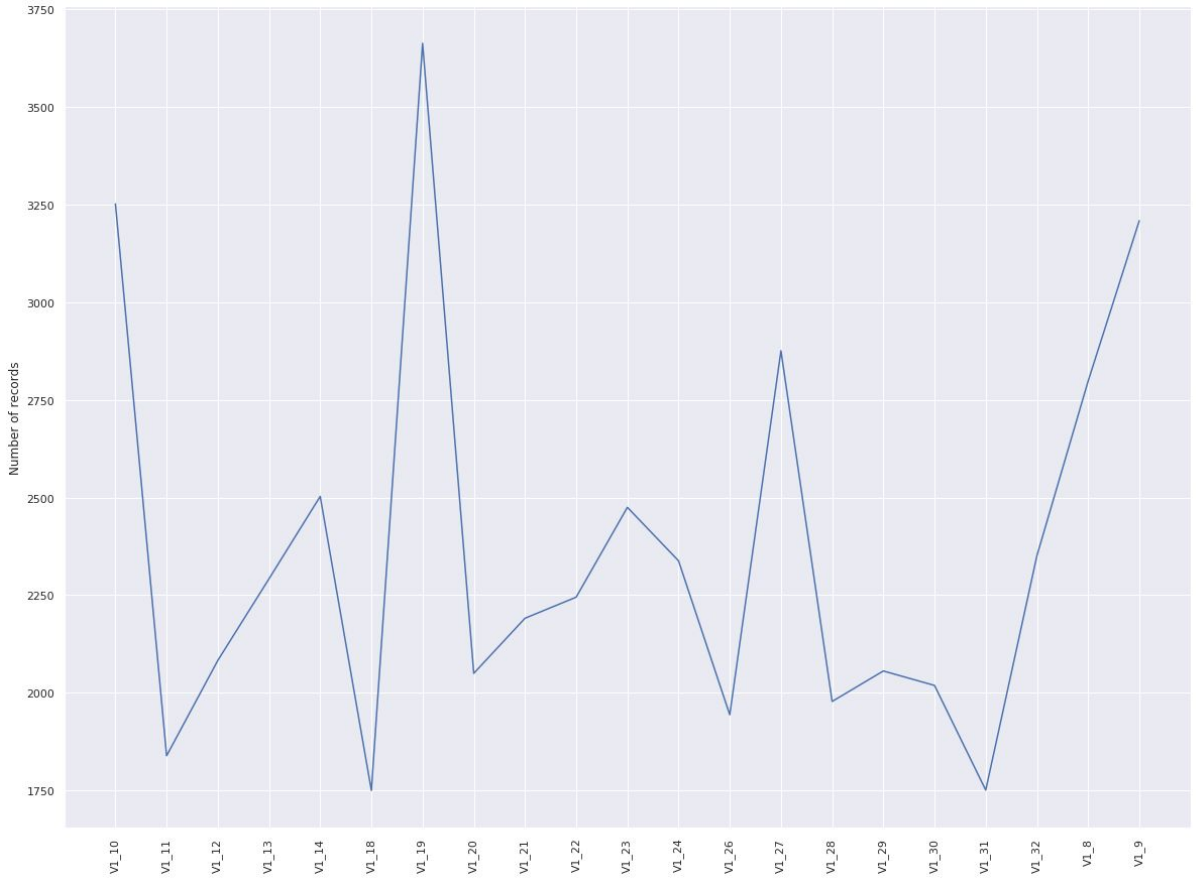
Although the data at each location is collected for the same amount of time, 2 minutes, there are still a different number of records received by the receivers. In other words, some locations are more strategic in transmitting in bluetooth signal where more receivers are able to detect the signal. (The reasoning still requires further investigations).

Location	V1_8	V1_9	V1_10	V1_11	V1_12	V1_13	V1_14	V1_18	V1_19
Number of records	2798	3209	3252	1839	2083	2292	2503	1750	3663

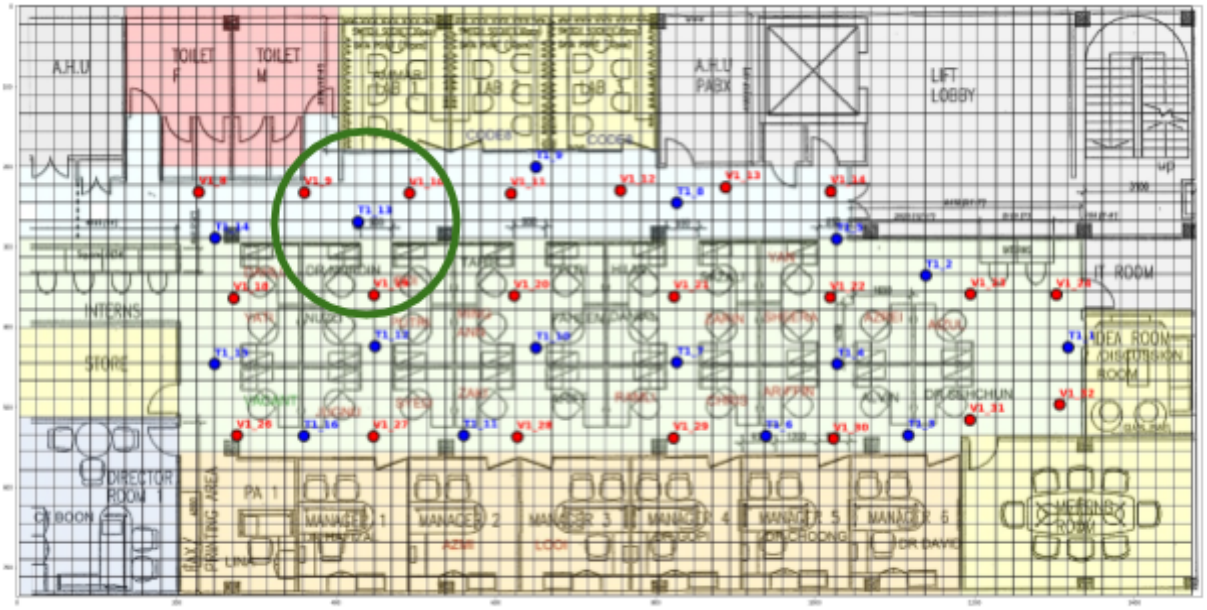
Location	V1_20	V1_21	V1_22	V1_23	V1_24	V1_26	V1_27	V1_28	V1_29
Number of records	2050	2192	2245	2475	2338	1944	2876	1978	2056

Location	V1_30	V1_31	V1_32
Number of records	2019	1751	2351

As a summary in a plot as follows.



The top three most strategic locations are V1_9, V1_10 and V1_19 which are located within the green circle as follows.



2.2 Sample rate

The sample rates at different locations are tabulated in terms of mean and standard deviation. A few examples of sample rate over time at each location is plotted below. For example, at location V1_10, the sample rate is fluctuating around with mean around 1.21 and standard deviation of 0.53. The sample rate is shown to be stationary at a certain location where the mean and standard deviation does not change much throughout the data collection. Most of the locations have received two readings at the same time.

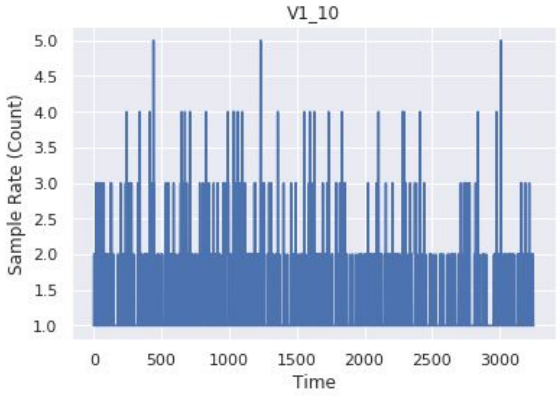
Note that some locations have slightly higher standard deviation. The reason is because they have a quite a number of samples where three or more readings are received at the same time.

Location	V1_8	V1_9	V1_10	V1_11	V1_12	V1_13	V1_14	V1_18	V1_19
Mean	1.15	1.17	1.21	1.17	1.10	1.11	1.09	1.09	1.18
Standard deviation	0.42	0.48	0.53	0.51	0.34	0.39	0.32	0.31	0.48

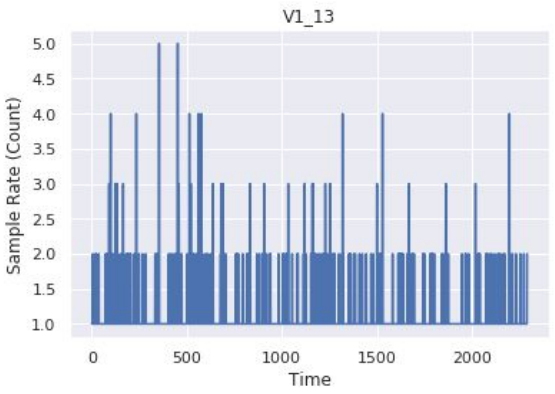
Location	V1_20	V1_21	V1_22	V1_23	V1_24	V1_26	V1_27	V1_28	V1_29
Mean	1.11	1.09	1.08	1.09	1.09	1.10	1.14	1.11	1.10
Standard deviation	0.39	0.32	0.30	0.32	0.35	0.36	0.41	0.38	0.34

Location	V1_30	V1_31	V1_32
Mean	1.09	1.07	1.12
Standard deviation	0.31	0.29	0.40

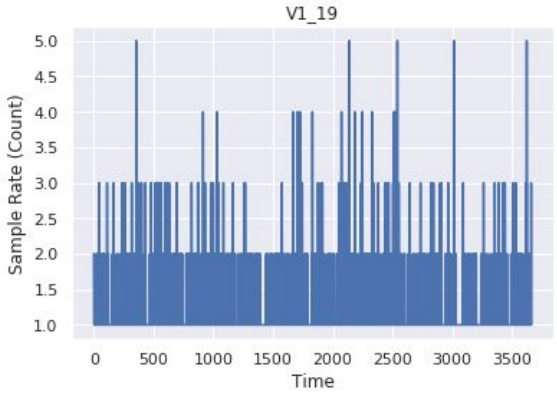
V1_10
Mean: 1.21, Standard deviation: 0.53



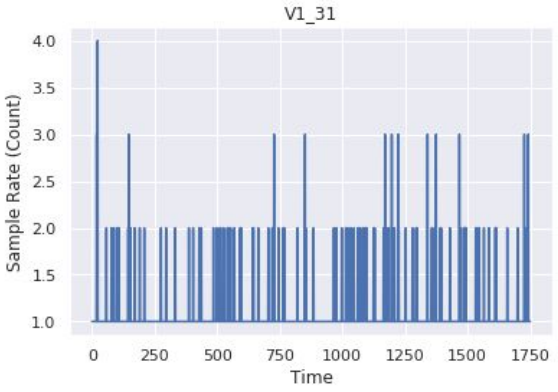
V1_13
Mean: 1.11, Standard deviation: 0.39



V1_19
Mean: 1.18, Standard deviation: 0.48



V1_31
Mean: 1.07, Standard deviation: 0.29



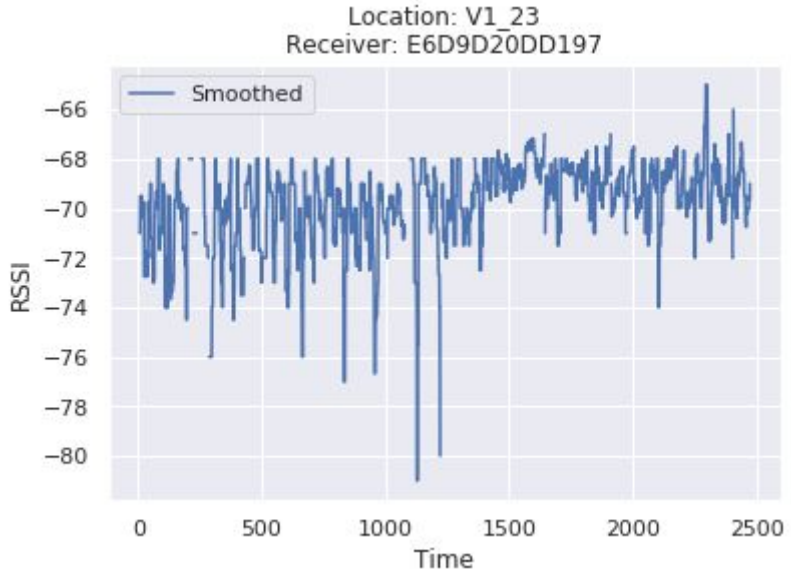
The standard deviation mainly lies around 0.3 - 0.5. There are some locations having a sample rate out of this range, e.g. V1_10, V1_11, V1_31. The standard deviation might be affected by the sample where a higher sample rate has a higher standard deviation. The sample rate at different locations has a similar distribution over time.

3 Methodology

3.1 Group By or Rolling

Two methods of smoothing the data are used. Group By and Rolling (or Moving Average). Group By will aggregate a few samples into one sample which will reduce the total training data set in total. However, by using the moving average, we can preserve the total number of training samples by aggregate the samples using a sliding window.
Moving average generally shows a better result in overall online testing compared to using Group By.

For example, the signal strength received by the receiver E6D9D20DD197 at location V1_23 is smoothed using moving average as follows.



3.2 Forward Fill

After applying the moving average, we might still have null values in the data because it is still possible that there is no reading within a single sliding window. We apply forward full after aggregating the data by moving average. Applying forward fill means that we are assuming that the current missing signal strength has the same value as the previously received signal strength. After that, the rest of the null values are imputed by -100.

3.3 MinMax scaling

MinMax scaling is chosen where minimum and maximum are fixed at -100 and -40 respectively. By doing so we are able to ensure the data distribution is maintained since it is not a normal distribution. -100 will be scaled to 0 and -40 will be scaled to 1 linearly.

3.4 Multi-Layer Perceptron

The neural networks contain two hidden layers and a single output layer as follows.

Model: "sequential"

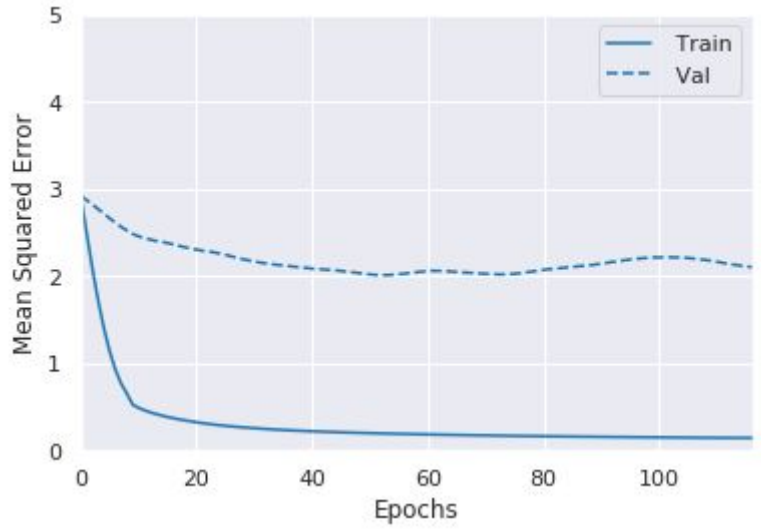
Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	1152
dense_1 (Dense)	(None, 64)	4160
dense_2 (Dense)	(None, 2)	130

Total params: 5,442
Trainable params: 5,442
Non-trainable params: 0

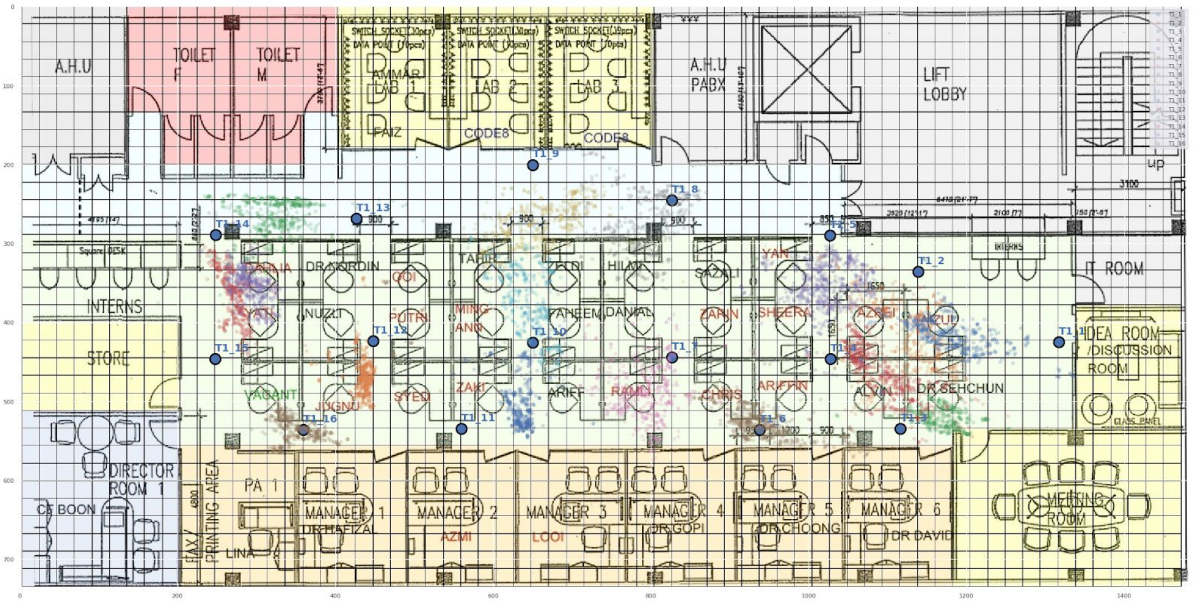
The training uses Early Stopping with patience of 50 epochs and monitoring validation loss. The optimizer used is RMSprop with learning rate of 0.001.

4 Results

The mean squared errors of the training data and validation data over the training epochs are shown in the graph below.



The results localization on the test data is as shown below.



The mean squared error is approximately 2.05 m.
The average distance error is about 1.77 m.

The locations which the model performs badly are T1_1, T1_2, T1_14, T1_15, which are located at the two ends of the fingerprints collection locations.

5 Future Work

5.1 Data Augmentation

Between two locations where fingerprints are collected, we may approximate the fingerprints by using the fingerprints from the two neighbours.

5.2 Convolutional Neural Network (CNN)

The image of the whole office is constructed where the location of the receivers having a high RSSI received will have a high pixel value. Using the image we infer the location of the beacon using CNN.

5.3 Long Short Term Memory (LSTM)

The current location is highly dependent on the previous predicted location. LSTM may be useful in predicting the location in the near future (seconds) since the localization currently developed has a lag in time due to moving average implementation.