BLE Indoor Localization
Report 3

Prepared by: Tan Chin Huan

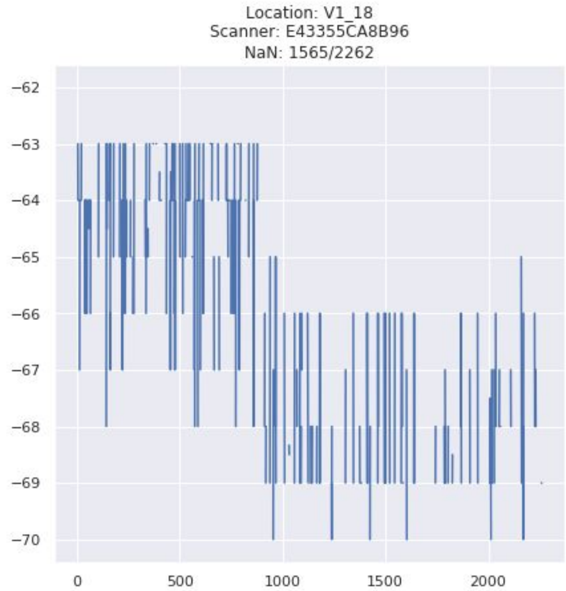
Prepared on: 04/02/2020

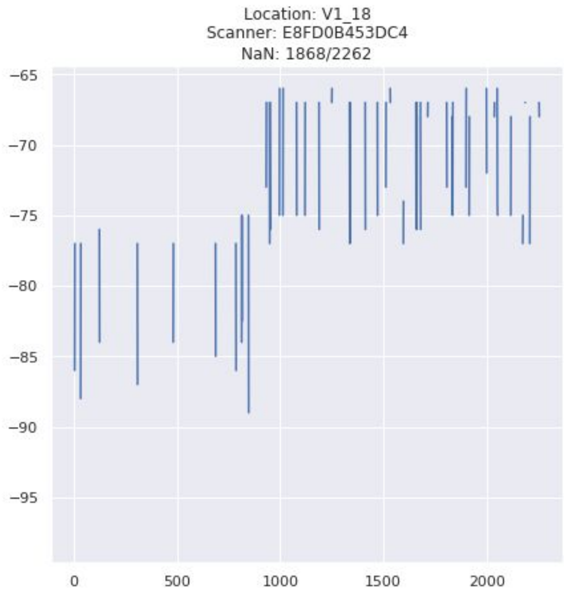
Table of Contents

1 Preliminary Analysis

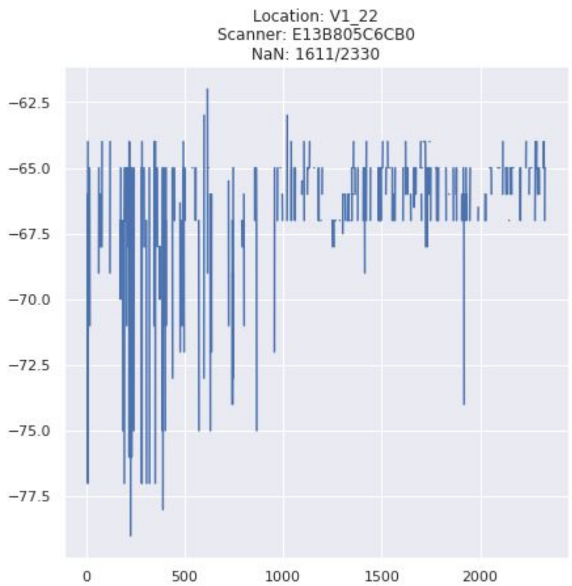
1.1 Battery Level

Battery level of the bluetooth beacon affects the signal strength received. However, it is still possible that the signal strength difference is caused by a change in the environment. The signal strength is different in terms of its mean and standard deviation. The diagrams below show the three examples where signal strength is collected at two different battery levels. The change in battery used occurs at x-axis *800.*





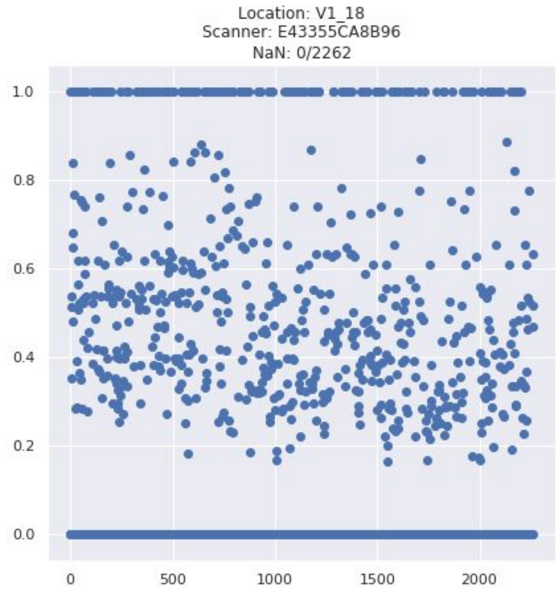
The signal strength could possibly go up or down.

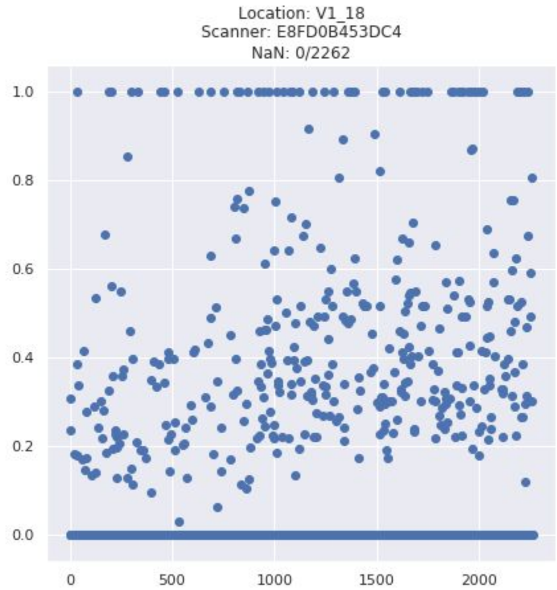


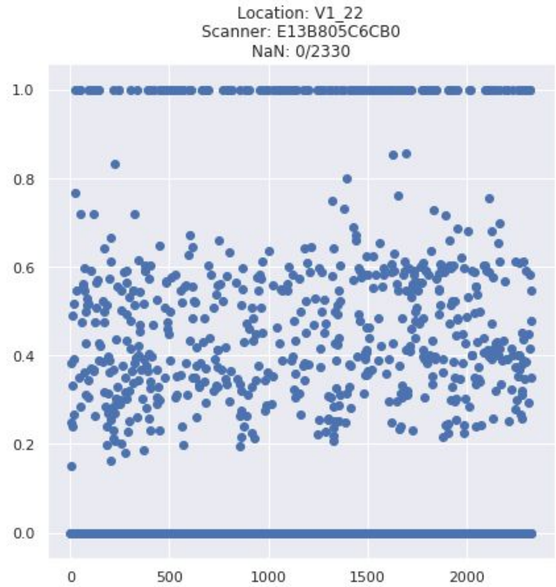
The standard deviation might change as well.

1.2 Normalization

By doing *L1 normalization,* the difference in signal strength could possibly be brought closer.



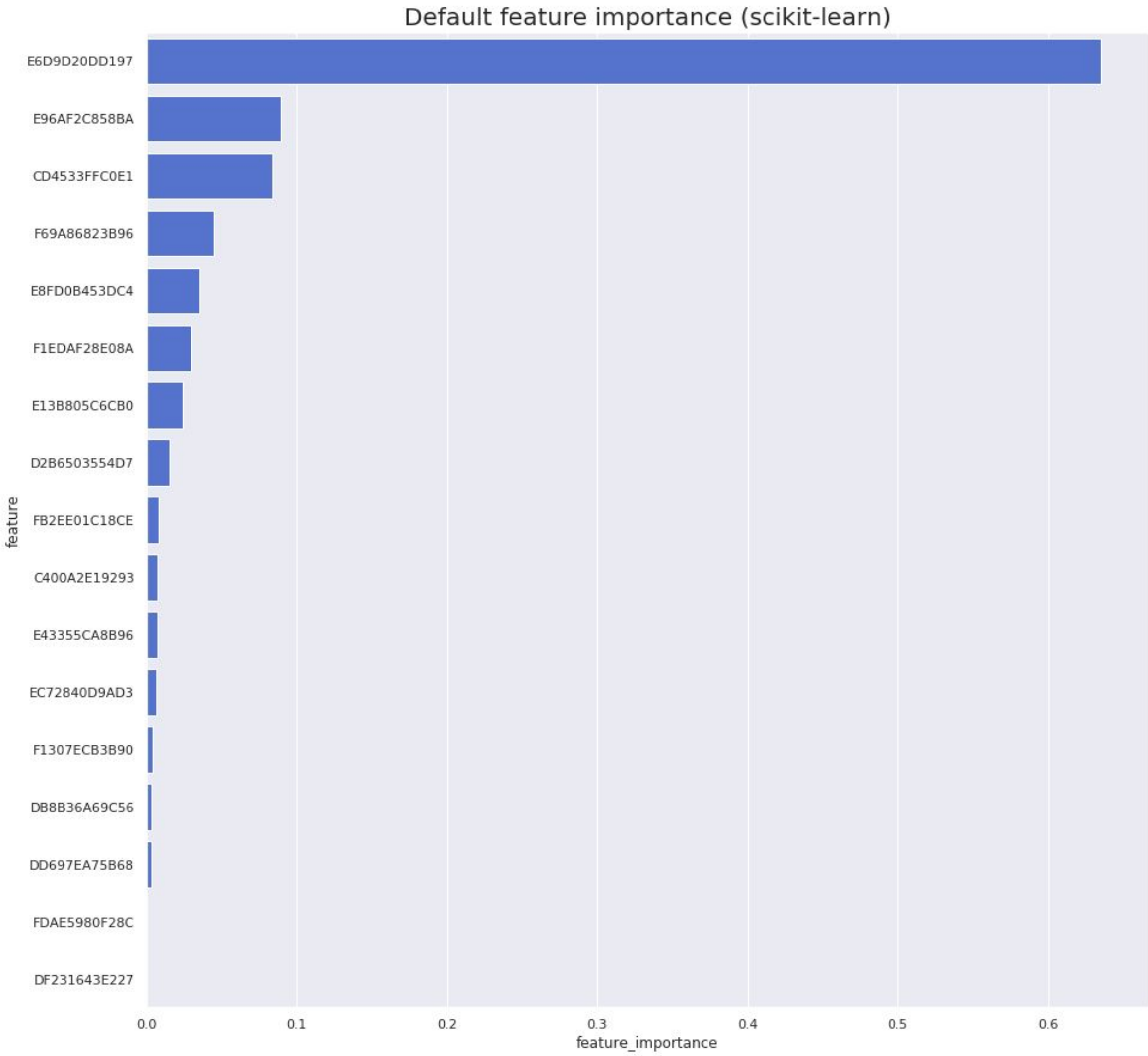




1.3 Feature Importance

1.3.1 Default Scikit-Learn Feature Importance

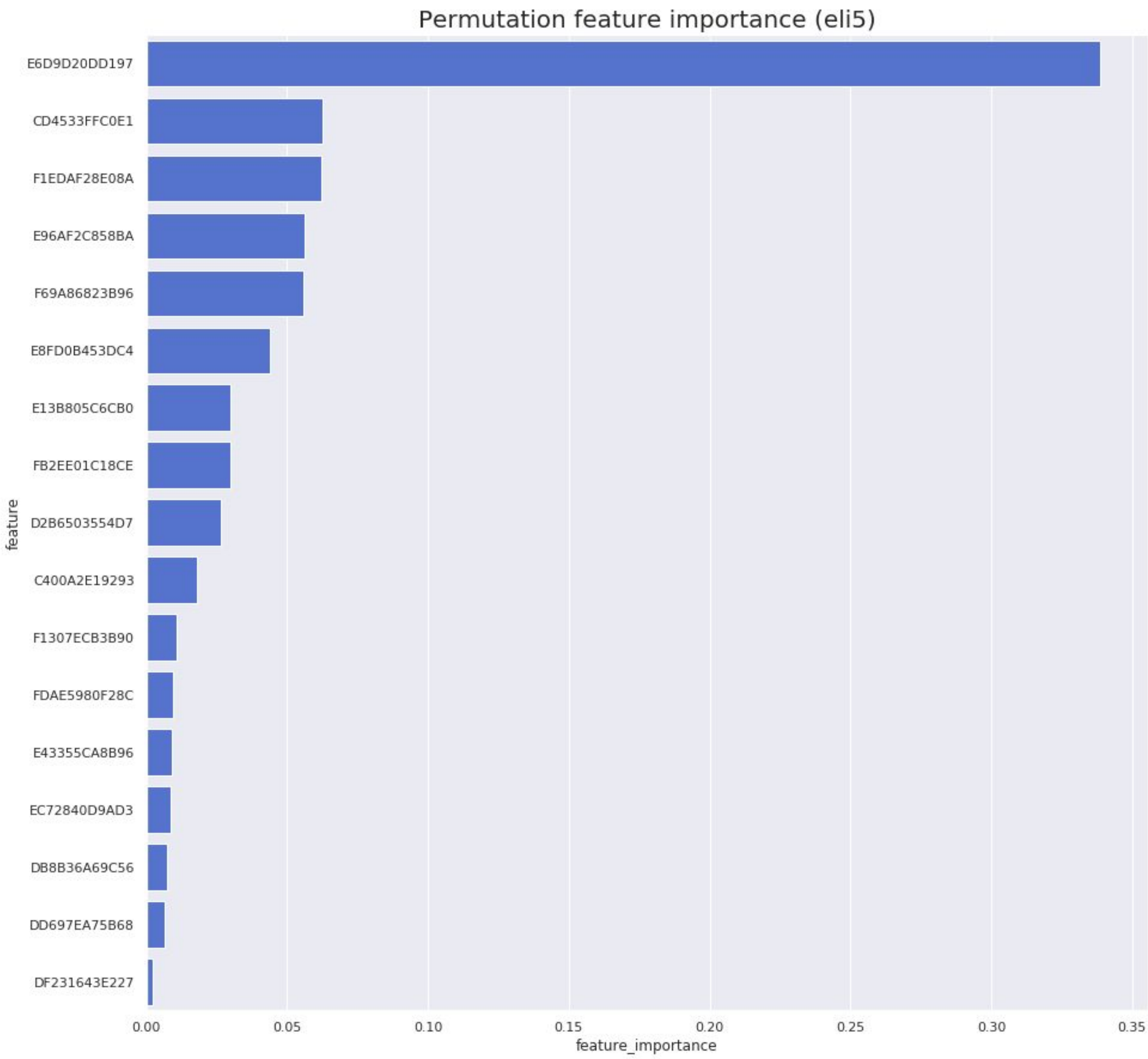
The importance of each feature is examined using *scikit-learn RandomForestRegressor* because it is computational intensive to use a neural network.
The following shows the default *scikit-learn RandomForestRegressor* feature importance. It seems that a bluetooth receiver dominates the result of localization. The receiver is *E6D9D20DD197* which is located at one end of the office.



However, it is known that the default feature importance has some bias in it. Therefore, we use other methods to examine the feature importance as well.

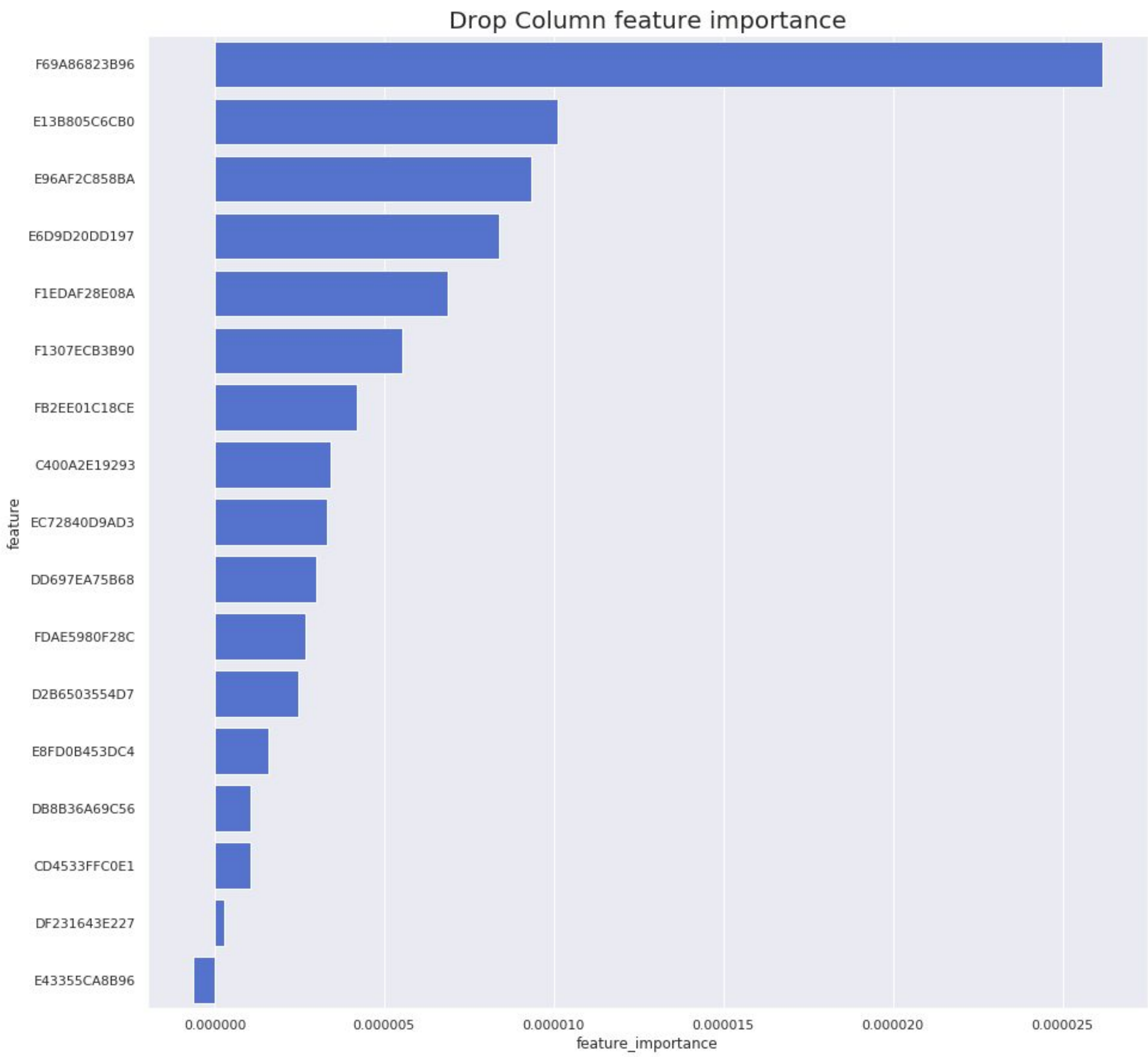
1.3.2 Permutation Feature Importance

The following shows the permutation feature importance using a library *eli5*. In permutation feature importance, the features are permuted and the model is run to give the result. Using the result, we examine how greatly the feature contributes to the accuracy before they are permuted. The result is similar to the default feature importance with some features in different order.



1.3.3 Drop Column Feature Importance

Next is *Drop Column Feature Importance*. As the name implies, this method drops a column first and then the model is run to obtain the result. The result is again used to measure how greatly the feature contributes to the accuracy.



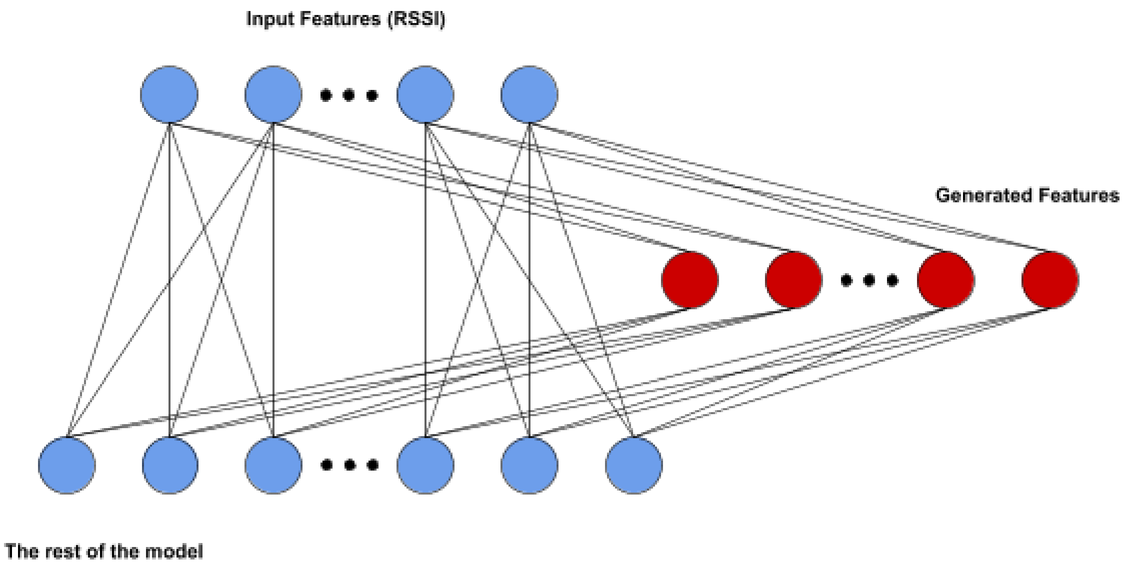
There is an interesting importance here. *E43355CA8B96* has a negative importance which means that if the feature is removed, we might get a better result.

2 Data Modelling

2.1 Discrete Fingerprints

The fingerprints are collected from 21 locations and the test data are collected at 16 different locations. The data is then used to create a model for indoor localization. Three models are created *MLP-Small*, *MLP*, *MLP (Features Generation)*.

MLP-Small has the smallest architecture which has only 1 hidden layer with 9 nodes. It is used to serve as a benchmark to compare the other models.
MLP has a slightly larger architecture which is tuned to have a decent accuracy. The model predicts for x and y coordinates separately.
When proposing a tentative methodology, the statistics (such as power, exponential) among the RSSI values received might contribute to the localization accuracy.
MLP (Features Generation) has a few features created from fully-connected layers before passing into a full *MLP* model. The architecture is shown in the figure below.



The reason is because we find a good *joint contribution* by different features through *RandomForestRegressor* using a library *treeinterpreter*.

2.2 Continuous Trajectory

The fingerprints are also collected in the form of trajectory. The data collection relies on a fully-built Ultra-Wide Band (UWB) indoor localization system. The bluetooth signal strengths are collected while the locations are labelled through UWB positioning. A few models are created to model the temporal sequence, *MLP (Timesteps), Con1D-MLP, LSTM and Conv1D-LSTM.*

MLP (Timesteps) has a temporal sequence of RSSI values as input. The input features are then flattened before feed forward to a MLP. The output is a single *(x, y)* coordinate.
Conv1D-MLP has additional 1D convolution and pooling layers before passing into fully-connected layers.
LSTM is also used for modelling. The output of LSTM is passed to fully-connected layers.
Conv1D-LSTM has convolution and pooling layers added before the *LSTM*.

3 Results

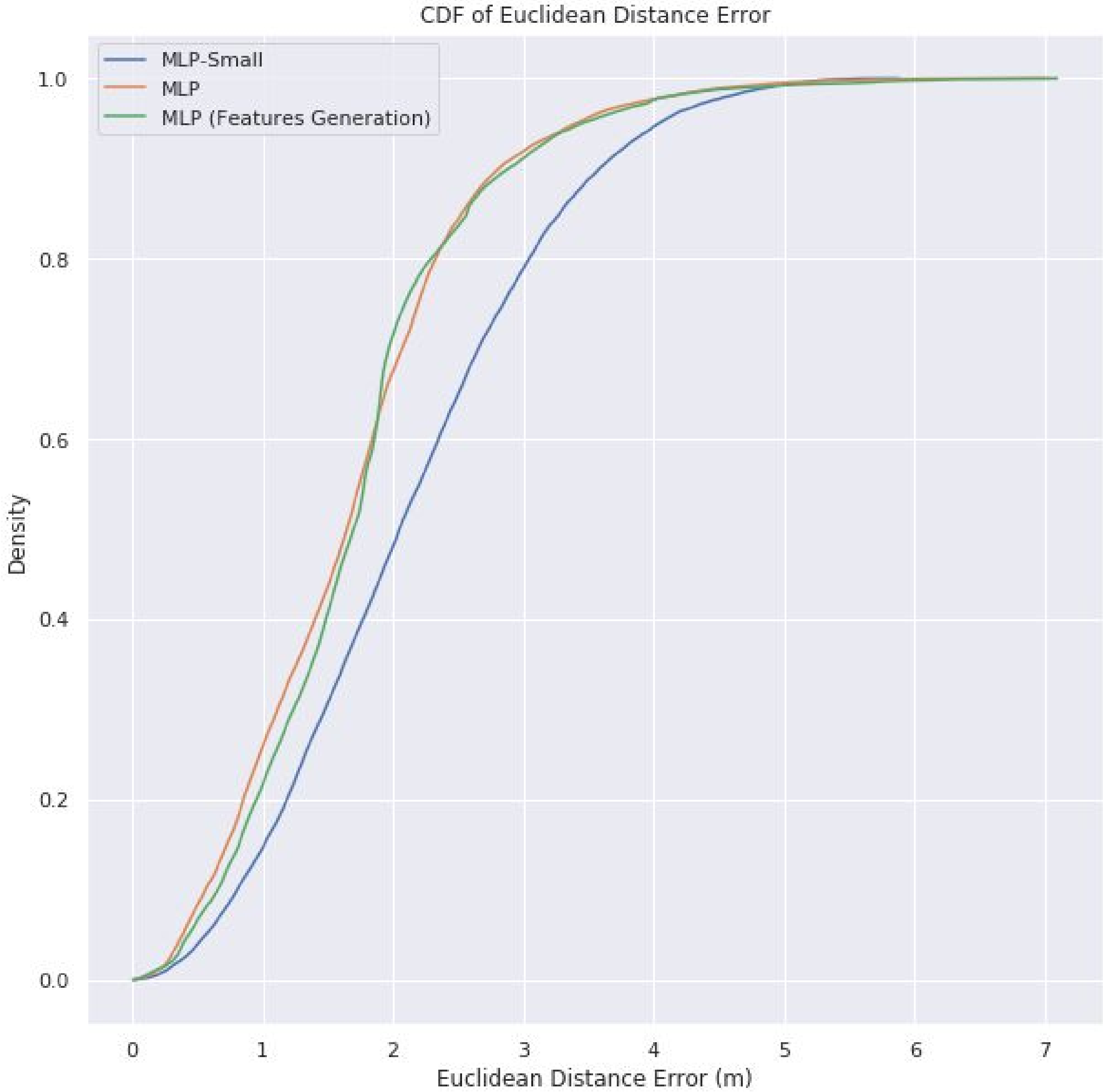
The results are separated into two parts where the test data in the first part has only static fingerprints whereas the test data in the second part has both static fingerprints and moving trajectories.

3.1 Static Fingerprints Test Data

The table below shows the results of the models accepting discrete fingerprints.

Models	**Average Euclidean Distance Error**	**Mean Squared Error**	**Mean Absolute Error**
MLP-Small	2.141	2.852	1.360
MLP	1.688	1.854	1.046
MLP (Features Generation)	1.733	1.927	1.053

Below shows the cumulative distribution of the prediction errors by different models.

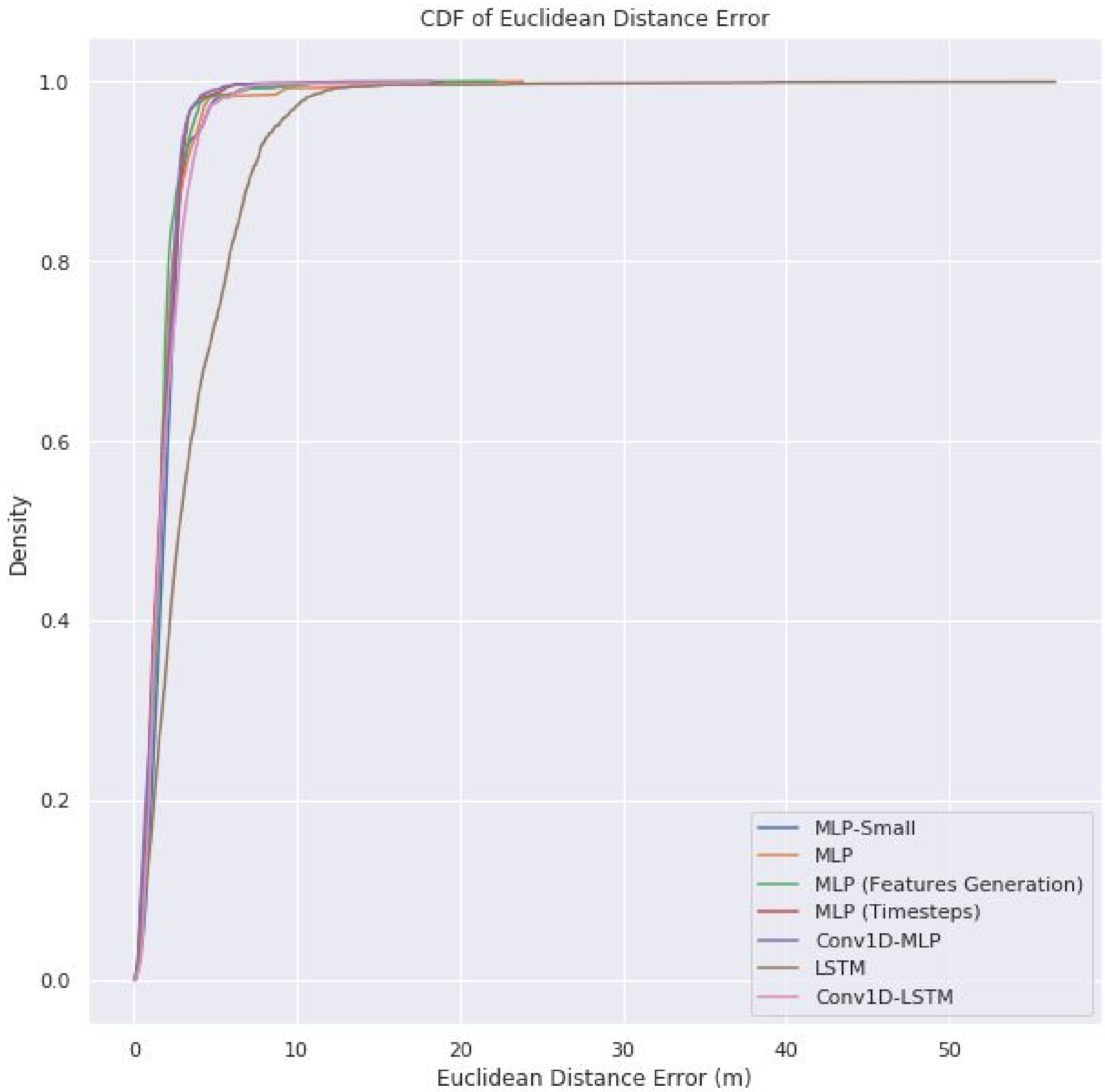


3.2 Moving Fingerprints Test Data

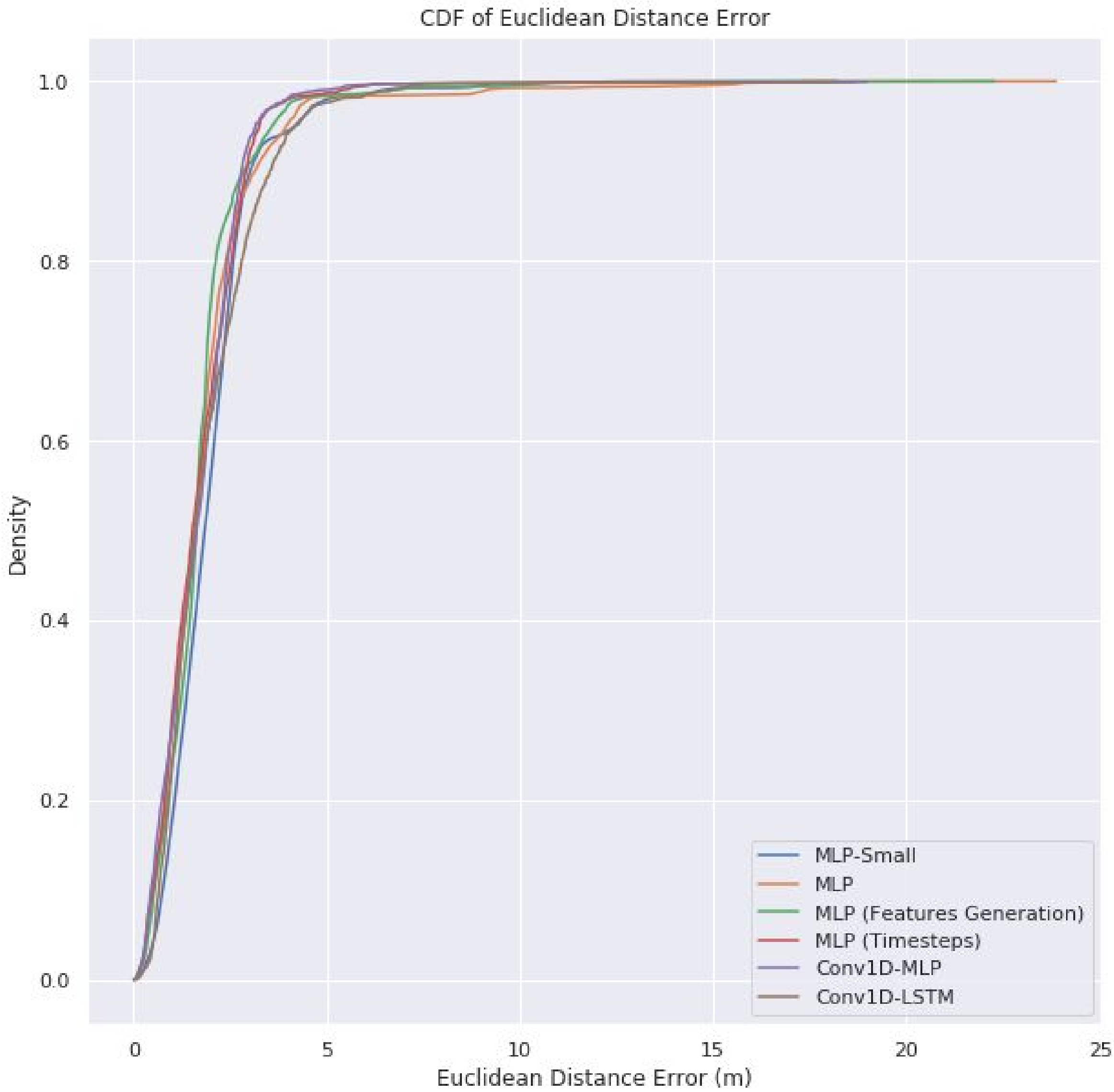
In this test, we combine all the static fingerprints test data and moving fingerprints test data. Notice that the error of the static models, *MLP* has become larger when a moving fingerprints are included.

Models	**Average Distance Error**	**Mean Squared Error**	**Mean Absolute Error**
MLP-Small	1.984	2.734	1.243
MLP	1.823	3.014	1.107
MLP (Features	1.730	2.253	1.052

Generation)			
MLP (Timesteps)	1.698	2.086	1.045
Conv1D-MLP	1.698	2.066	1.036
LSTM	1.781	2.314	1.092
Conv1D-LSTM	1.931	2.764	1.216



Due to a small dataset, we most probably overfit the LSTM model. Not all parameters in LSTM are trained properly. In this case, the LSTM does not give a good result. Besides, there are a lot more static fingerprints in the dataset compared to moving fingerprints. LSTM performs worse when the fingerprints are static.
In the diagram below, we remove LSTM to show the other results more clearly.



The models, *MLP (Timesteps)* and *Conv1D-MLP* are very slightly better than the other models.