Floor Localization

CS 435: Deep Learning
Course Project Milestone 2 Update

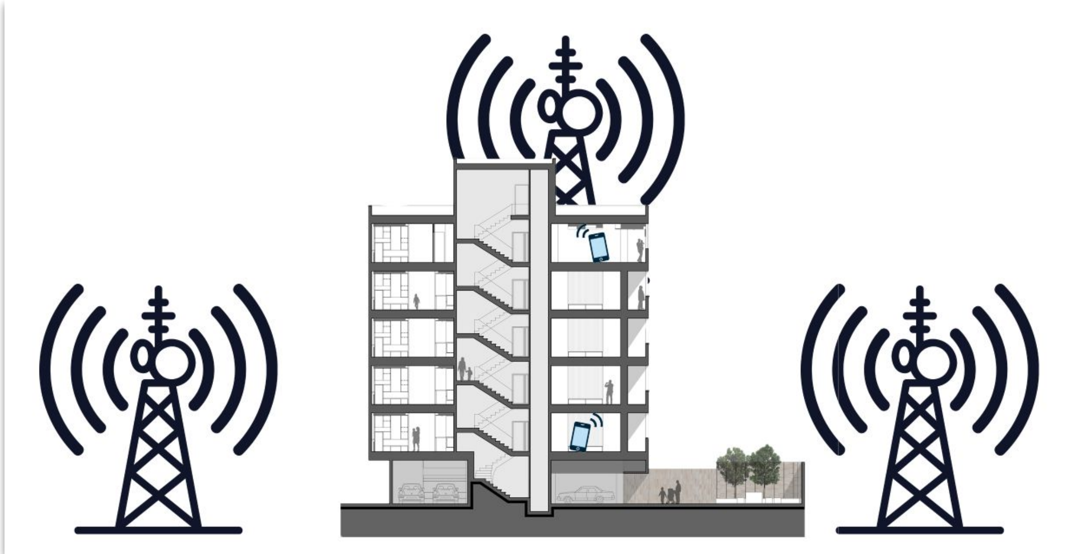
Aya Ashraf 02
Khaled Barie 21

Outline

- Problem Statement
- Progress Summary
- Base Models
  - SkyLoc Baseline
  - Fully Connected Architecture
- Dataset Setup
  - Using Homogeneous Datasets
  - Using Heterogeneous Datasets
- Time Series Classification Architectures
- Investigating Heterogeneity
- Feature Selection & Augmentation
- Next Steps

Problem Statement

- Create a floor localization system that can predict a user's floor depending only on cellular signals



Problem Statement

- Fingerprints are organized as a vector of the received signal strength heard from different cell towers in the area of collection. Each sample is paired with the floor it is collected from
- Below are examples for such fingerprints

		Cell tower #1	Cell tower #2	Cell tower #3	Cell tower #4	Cell tower #5		Floor #
Sample 1:	[	21	0	14	3	0	],	Floor 1
Sample 2:	[	18	5	0	0	2	],	Floor 2

Data Summary (Vodafone)

- This table summarizes the number of samples collected on each floor from 3 phones on vodafone
- 41 unique cell towers were detected across all samples from vodafone phones

Floor #	HTC X9	Moto G5	Oneplus 6	Total
0	5553	5133	2776	13462
1	5945	9955	6102	22002
2	6694	8011	4555	19260
3	6297	7510	4254	18061
4	6146	7201	5786	19133
5	6015	7297	2738	16050
Total	36650	45107	26211	107968

Data Summary (Orange)

- This table summarizes the number of samples collected on each floor from 2 phones on orange
- 18 unique cell towers were detected across all samples from orange phones

Floor #	Galaxy S5	Galaxy Note 3	Total
0	2885	2899	5784
1	5740	7663	13403
2	4387	2739	7126
3	4342	5380	9722
4	4626	5440	10066
5	3873	3783	7656
Total	25853	27904	53757

Progress Summary

- Collected data from the campus in the Electrical Engineering building using 5 devices on 2 carriers
- Experimented with multiple base architectures and produced initial results
- Implemented a baseline from literature (based on the SkyLoc paper)
- Experimented with different data setups
- Started investigating the heterogeneity problem and conducting analysis on the data
- Used popular time series classification architectures from literature
- Tuned all architectures used to maximize output accuracy
- Employed feature selection and data augmentation to increase accuracy

SkyLoc Baseline

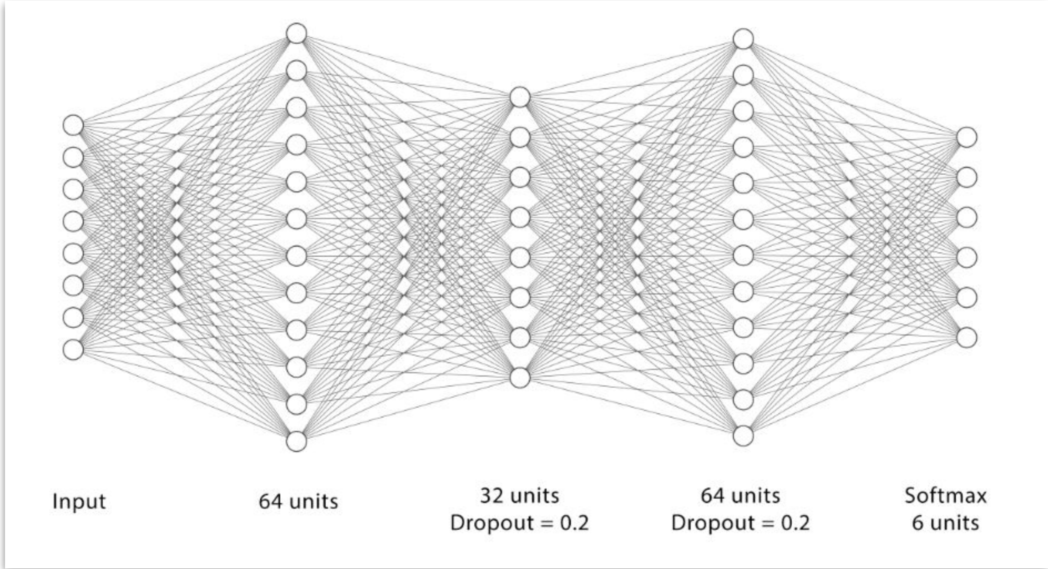
- Implemented the naive approach from the skyloc paper*
- Calculates euclidean distance in RSS space from collected samples to estimate the floor of new samples

$$\sqrt{(R_1^{t\,r} - R_1^{t\,st})^2 + (R_2^{t\,r} - R_2^{t\,st})^2 + (R_3^{t\,r} - R_3^{t\,st})^2}$$

*Varshavsky, Alexander & LaMarca, Anthony & Hightower, Jeffrey & de Lara, Eyal. (2007). The SkyLoc Floor Localization System. 125-134. 10.1109/PERCOM.2007.37.

Fully Connected Architecture

-



Using Homogeneous Datasets

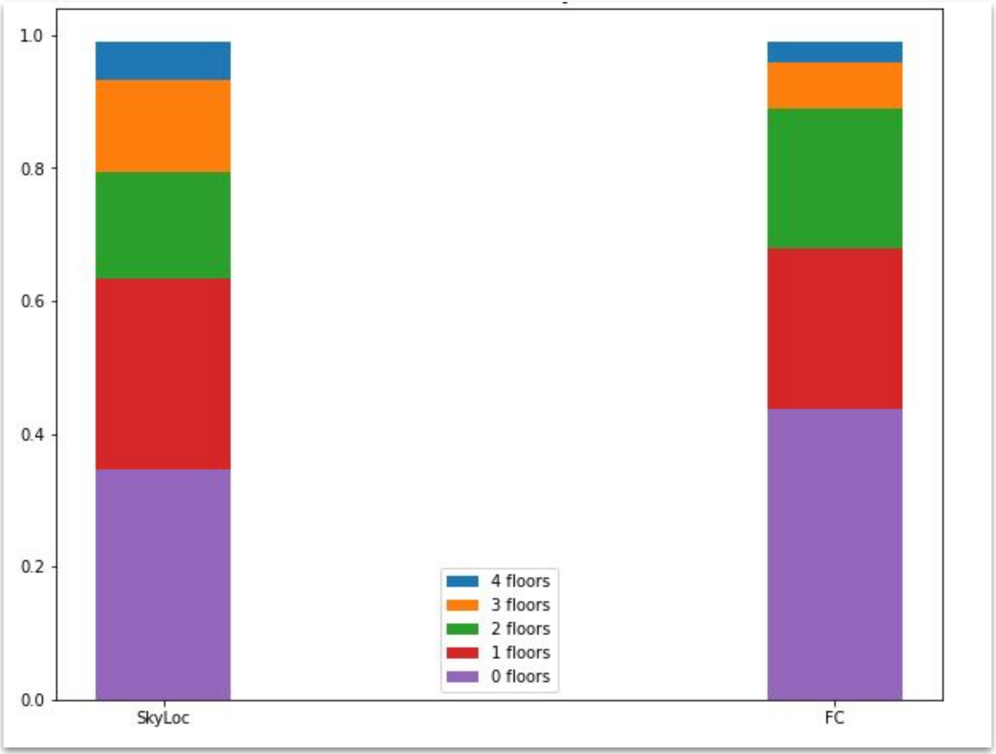
- As remarked in the previous milestone, we started by testing our models against homogeneous datasets
- Train and Test data are both sampled from the same collection/distribution
- Each dataset used was partitioned as 80% training, 10% validation and 10% test.

Dataset	SkyLoc	FC
HTC X9	100	99.8
Moto G5	100	99.8
Oneplus 6	100	100
Combined Dataset	99.9	99.8

Using Heterogeneous Datasets

- We moved on to investigating how to solve the heterogeneity problem
- This involves training a model using data from some phones and attempting to classify data from other "unseen" phones
- We used a "leave on out" approach where we train our model using data from all phones except one, and use unseen data from that remaining phone for validation and test
- Most of our experiments used data from the HTC X9 and Moto G5 for training and data from the Oneplus 6 for validation and testing

Within-k-floors	SkyLoc	FC
0	34.6	43.8
1	63.3	67.8
2	79.4	88.9
3	93.2	95.9
4	99.1	99
5	100	100



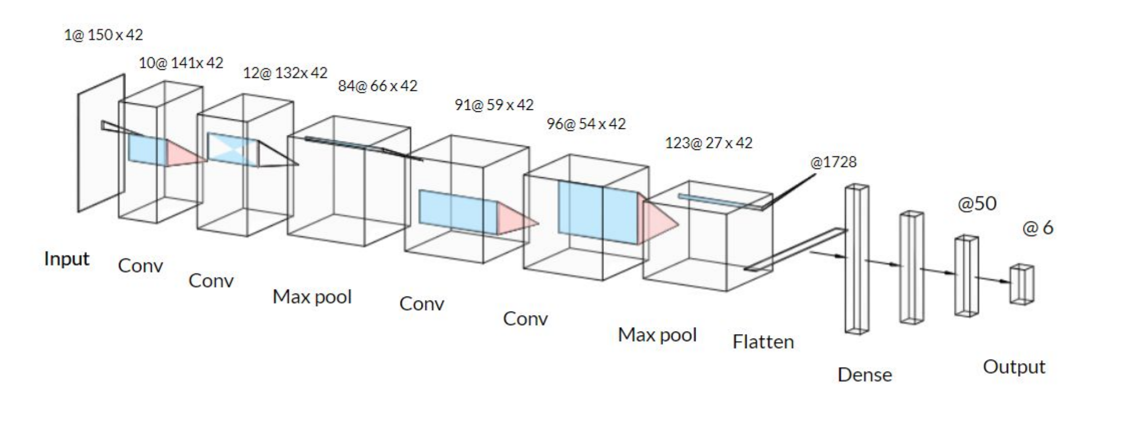
- Accuracy of SkyLoc and FC using a heterogeneous dataset at different levels of error

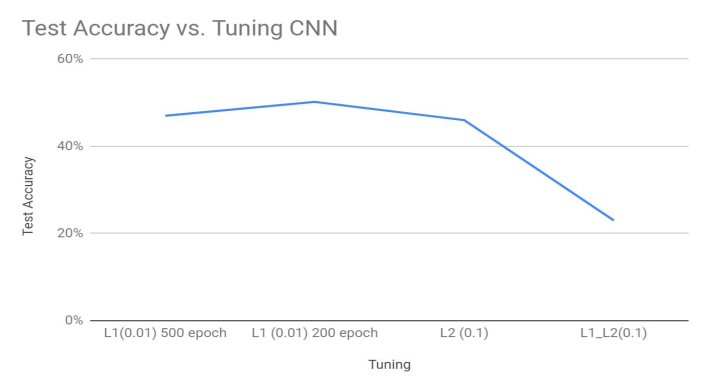
Time Series Classification Architectures

- We formalized the problem as a sequence model, leveraging a history of RSS scans to produce better predictions
- We tested with multiple time series CNN architectures from literature
    - 1D CNN
    - SeriesNet
    - InceptionTime
    - 1D ResNet

CNN

- Tuned our 1D CNN model, the full model is shown below
- Tuned the model using different regualizers to reduce overfitting





SerieNet

- An architecture used in time series forecasting
- We adapted the model to our problem by replacing the last layer with a softmax layer
- Uses dilated convolution

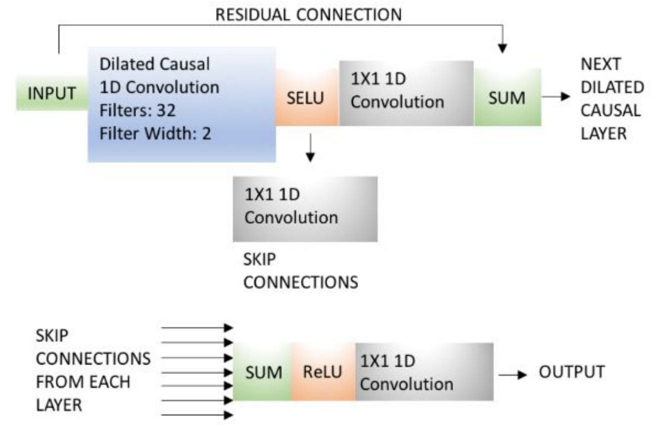
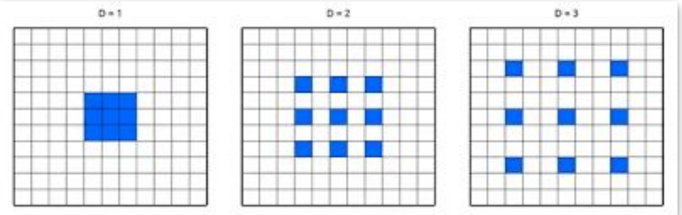
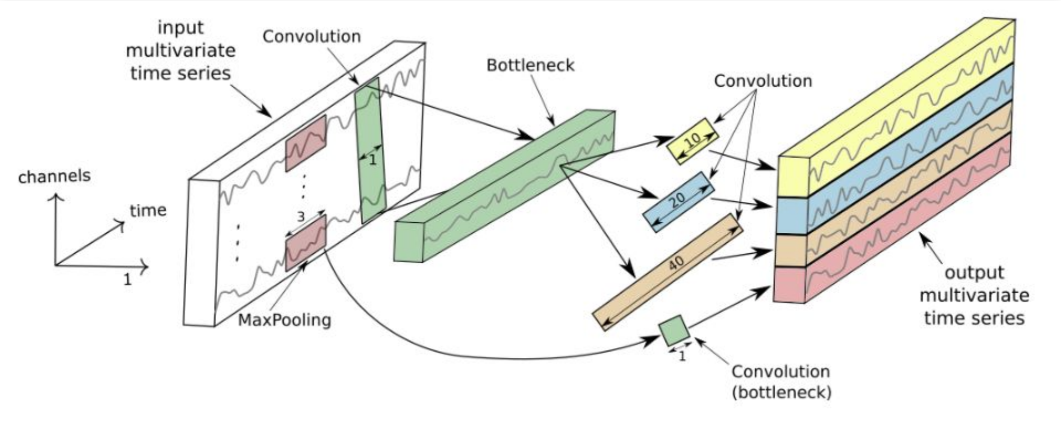


Figure 5: SeriesNet architecture



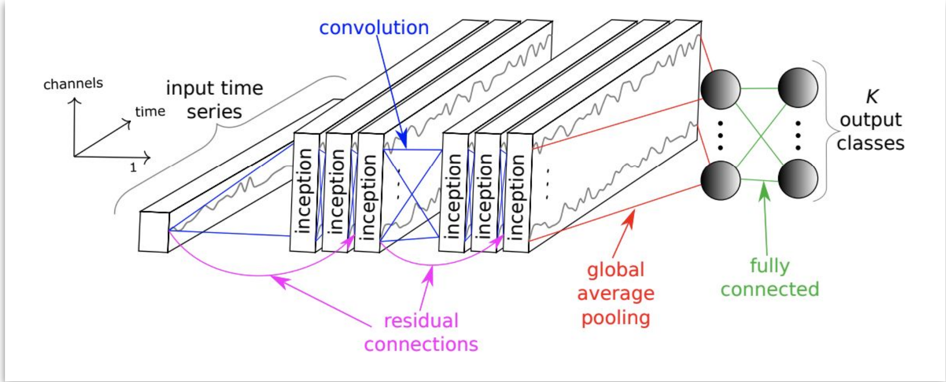
InceptionTime

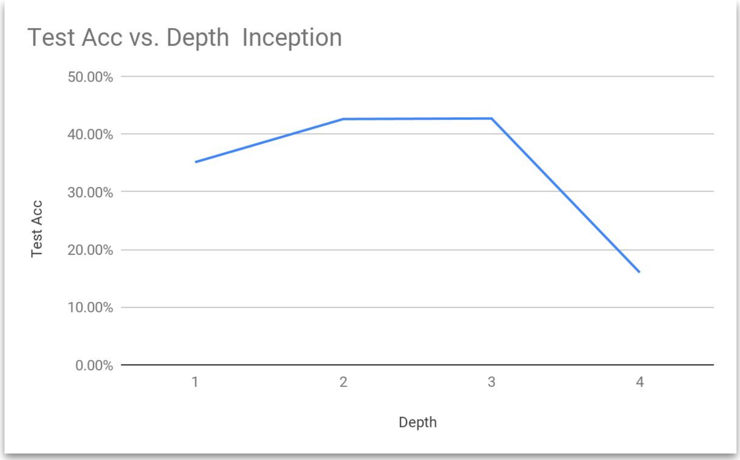
- Inspired from the Inception network architecture, using different kernels for convolution and concatenating their results
- This figure show 1 inception unit



InceptionTime

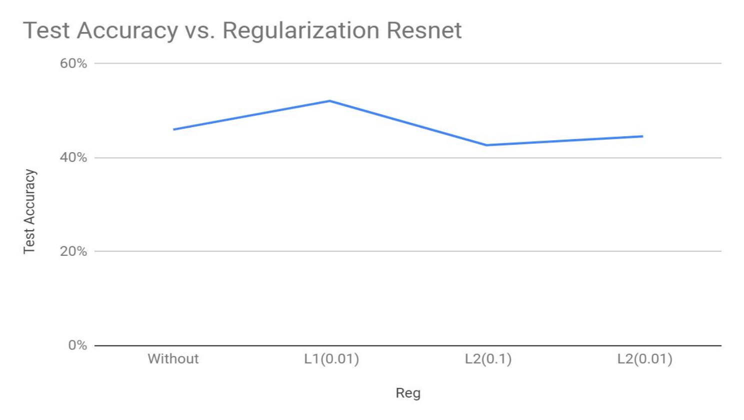
- The full network uses successive inception units and employs skip connections similar to ResNet
- We attempted tuning on the number of the units (Depth)

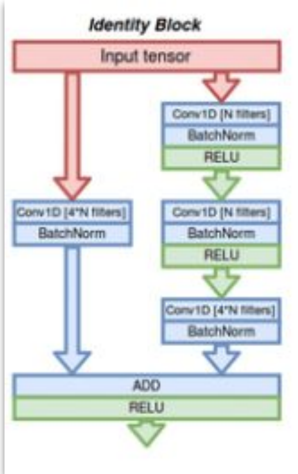




ResNet

- Follows the same architecture of the original ResNet but uses 1D convolutions
- We attempted tuning the model on regularization to reduce overfitting

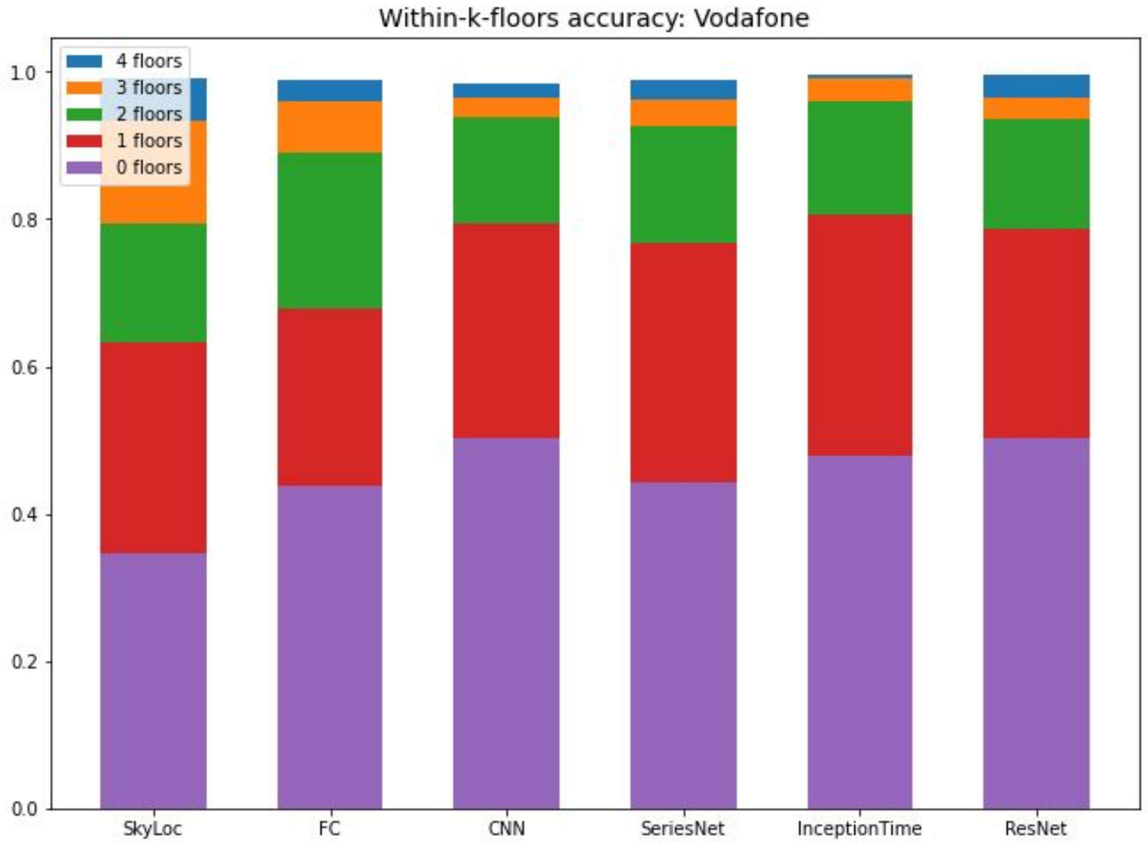




Maximum Accuracy at k floors

Within-k-floors	SkyLoc	FC	CNN	SeriesNet	InceptionTime	ResNet
0	34.6	43.8	50.2	44.4	47.8	50.2
1	63.3	67.8	79.3	76.7	80.7	78.6
2	79.4	88.9	93.8	92.7	96.1	93.5
3	93.2	95.9	96.4	96.2	99.2	96.5
4	99.1	99	98.3	98.9	99.7	99.5
5	100	100	100	100	100	100

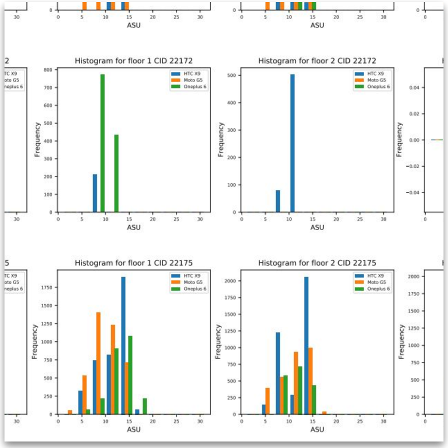
Accuracy table for various architectures using a heterogeneous vodafone dataset



Accuracy Plots for various architectures using a heterogeneous vodafone dataset

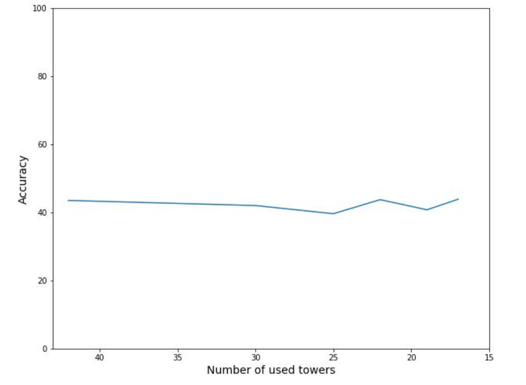
Investigating Heterogeneity

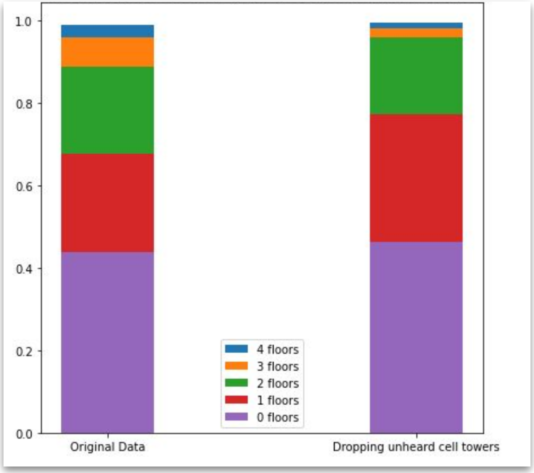
- Seeing that accuracy was still low, we proceeded to analyze the dataset to gather some insights
- We plotted histograms describing the distribution of RSS received in each floor from each cell tower for all phones
- This figure is a sample of the generated plots, full plots can be found here
- Insights obtained
  - Some cell towers were only picked up by one or two phones, not all
  - Some cell towers were heard only in a few number of scans
- Using these insights, we applied two feature selection techniques



Feature Selection

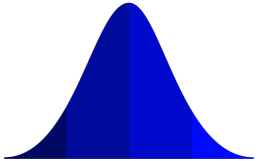
- Method 1:
    - Drop cell towers with a fewest number of samples
    - This method had no significant effect on accuracy
    - Plot show the sensitivity of FC to the number of towers
- Method 2:
    - Drop cell towers that were not heard by all phones in the dataset
    - This gave a slight increase in accuracy from 43% to 46% within-0-floors accuracy and from 67% to 77% within-1-floors accuracy
    - Plot shows the effect of dropping these cell towers





Data Augmentation

- Generated new samples using the gaussian noise method from literature*
    - Involves sampling random gaussian noise and adding it to existing data to generate new examples
- This had no significant effect on accuracy

[21, 0, 14, 3, 0] +  → [23, 1, 13, 6, 2]

Old Sample | Gaussian Noise | Generated Sample

* H. Rizk, A. Shokry and M. Youssef, "Effectiveness of Data Augmentation in Cellular-based Localization Using Deep Learning," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, 2019, pp. 1-6.

Next Steps

- Combining all insights gained to compile the best model achievable
- Conduct further investigation on how to overcome heterogeneity
- Creating website and poster

Thank you.