**ECE496 Progress Report**

| **Project Title** | Physics-based machine learning models for indoor wireless localization |
| --- | --- |
| **Project Number** | 2021415 |
| **Supervisor** | Costas Sarris |
| **Administrator** | Tome Kosteski |
| **Name of students** | Prerna Anand  John Adrian Ambrad  Hiranya Maharaja  Deeksha Tewari |
| **Date of submission** | 24th January, 2022 |

**Executive Summary**

The following document contains the progress made by our team in performing indoor wireless localisation using machine learning models. The goal of the project is to create a machine learning model that uses the received WiFi signal strength to accurately predict the location of a static object. Two major requirements of this project were to (1) collect experimental data using Raspberry Pis and Raytracer simulations and (2) develop machine learning models to predict the receiver location.

Since the Interim Demo, the team has made significant progress towards achieving the requirements. Prerna and Hiranya were able to research and successfully prototype a K-Nearest Neighbors (KNN) by the stipulated deadline, and with nearly 2657 simulated data points to train the model, they have achieved localisation to within 4.9 meters of the target location. They are currently working on fine-tuning this model, to reduce the error margin even further. However, due to recent Covid-19 related building closures, John and Deeksha have been unable to access the Bahen 8th floor to obtain real-life RSSI data to enhance the model’s generalisability. Consequently, they have redirected their efforts to researching and prototype an Artificial Neural Network (ANN) model.

The team will compare this supervised model with the previous unsupervised model to investigate if labeling the data provides any improvement in prediction accuracy - that is, if we can achieve an accuracy of less than 4 m. To this effect, the team’s next steps will be to further hypertune the KNN and ANN models. Once the team settles on a model, we will then turn our focus to further hypertune the model's hyperparameters, and reschedule the in-person data collection to another date.

**Table of Contents**

[**Project Overview**](#_cgezwkkc5ok1) **4**

[**Team Progress Summary**](#_snue7bhphs12) **6**

[**Individual Progress and Contributions**](#_q3mapahzvsnx) **7**

[Prerna Anand](#_qzwnx5bh4pm3) 7

[Deeksha Tewari](#_5k06bpdeyly4) 10

[John Ambrad](#_sdcns4ymec45) 12

[Hiranya Maharaja](#_bk4comh750qb) 13

[**Conclusions/Project Assessment**](#_ueluryq8d8dg) **15**

[**Appendices**](#_fv5rzgomfjzc) **16**

[Appendix A: Gantt charts](#_il9ikskj2bga) 16

[Appendix A.1: Implementation Gantt Chart](#_u6sqpjivngcf) 16

[Appendix A.2: Updated Gantt Chart](#_cwiob5mh1d1i) 17

[Appendix B: Bahen stl file](#_1vijpfb9aby) 18

[Appendix C: Prerna Anand](#_ucd63ragxg4n) 19

[Appendix C.1 : System Context Diagram](#_got1s0o7cu2h) 19

[Appendix C.2 : Draft segmentation of Bahen Map](#_jkynoqx8rq23) 19

[Appendix C.3 : Python script for simulation of data points](#_isptvg341pks) 20

[Appendix C.4 : Convergence Analysis](#_dubhrwpc9kuf) 20

[Appendix C.5 : Data cleanup and splitting into training and testing](#_qetxm0ylsbu1) 21

[Appendix C.6 : scikit library’s K-Nearest Neighbor model](#_lj5drfuffnea) 22

[Appendix C.7 : Custom K-Nearest Neighbor model](#_w3ydwq27v5oj) 22

[Appendix C.8 : Prediction function](#_tth8alizzex1) 23

[Appendix D: Deeksha Tewari](#_nt7fpp40pgg7) 23

[Appendix D.1: Screenshot of NetSpot Application used to perform data collection](#_zeffzthhu7tj) 23

[Appendix D.2: Data points collected on Bahen 8th floor using NetSpot](#_39jfmtunpcua) 24

[Appendix E: John Ambrad](#_bwz4ei4ps3vs) 25

[Appendix E.1 Screenshot of Random Forest Code](#_9nsg35obscxj) 25

[Appendix F: Hiranya Maharaja](#_gemq0am76o5m) 25

[F.1: Plot of xy loss values for each point using MSE in the scikit-learn KNN](#_m7ws2y3hbwe) 26

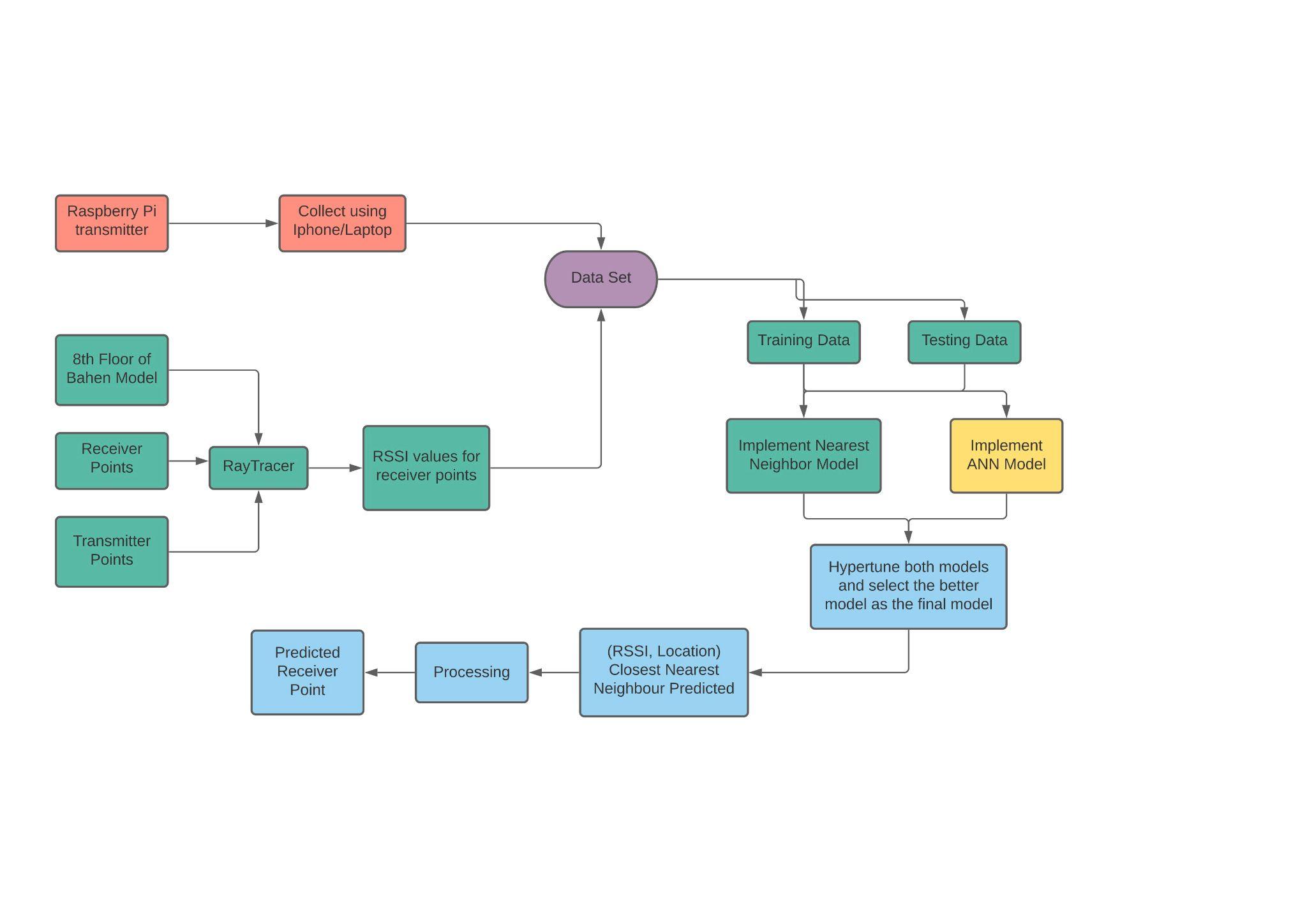
[F.2: K-means clustering algorithm](#_ye8zdsvc0w29) 27

# **Project Overview**

The goal of this project is to create a machine learning model that accurately predicts the location of a static object in a specified environment, based on the WiFi signal strength (RSS) received by it. The two main requirements for this project are data collection using Raspberry Pi and RayTracer and creating a machine learning model to predict the location.

The team has done all the configurations to collect data using a Raspberry Pi but due to Covid-19 and sudden closures the team was not able to collect data using Raspberry Pi and has decided to use the simulated RayTracer data for creating the models. If time permits and the situation with Covid-19 improves, the team will collect data using Raspberry Pi.

Over the last few weeks the team has been working on implementing machine learning models. The team split into two, Prerna and Hiranya are working on an unsupervised model (predict the result without learning the labels) and John and Deeksha are working on a supervised model (learn labels). The team has successfully implemented the unsupervised, K-Nearest Neighbor Model (KNN) with 4.9 m error and is currently working on completing the supervised, Artificial Neural Network Model (ANN).



*Fig 1: System Configuration Diagram*

*: Tasks skipped : Task done : Task in progress : Task to be done*

# **Team Progress Summary**

Since the demo, the team has completed a number of significant goals outlined in the Gantt chart in our implementation plan. We divided major tasks between the team members in order to parallelize the work and ensure efficiency. John and Deeksha were given the task of gathering live data with the Raspberry Pis, while Prerna and Hiranya were in charge of gathering simulated data and creating a custom KNN.

Prerna and Hiranya were responsible for researching and creating a custom KNN as per the Gantt Chart. Over 2657 data points have been simulated and organized into a usable dataset. Data collection was done by breaking the Bahen.stl map (Appendix B) into smaller sections and marking their x,y coordinates. Inputting those coordinates into a python script generated the 2657 data points. Convergence analysis was then done with these data points to account for unknowns such as the number of reflections and transmission. Prerna then ran the RayTracer for three different access points. With this dataset and the scikit-learn library’s pre-trained KNN functions, Prerna and Hiranya were able to complete a proof-of-concept KNN. However, the results indicated that the model needed to be fine-tuned, so the same dataset was used to complete the prototype for the custom KNN algorithm which predicts k closest neighbors. Using the k-closest neighbors and an averaging function, the model has a 4.9 m error margin. The team is currently working on reducing it to be under 4 m to meet the target for this project.

John and Deeksha were responsible for the configuration and programming of the Raspberry Pi WiFi hotspots and using them to gather real life data (signal strength and distance away from the hotspot). They were also given the task of exploring an ANN approach to WiFi localization and are currently completing research on how that can be done.

The team has come up against significant challenges for their respective tasks as well. John and Deeksha are facing issues with obtaining live data as it requires them to be on the 8th floor of the Bahen building. They could not complete this task as by the deadline as indicated in the Gantt chart due to the issue of Covid-19 restrictions blocking access to the building in the winter. Furthermore, the power of the WiFi signal emitted from each Raspberry Pi hotspot may not be appropriate for the purposes of this capstone project and must be reconfigured. Prerna and Hiranya faced issues when creating the KNN, they faced the issue of finding the best prediction algorithm based on the K closest neighbors found for each testing data set. They are working on multiple prediction algorithms to find the ideal one.

# **Individual Progress and Contributions**

## **Prerna Anand**

Prerna has worked on tasks including researching the need for indoor localization and different unsupervised machine learning model, segmenting bahen.stl file in blender into smaller sections, simulation of data points using python script, data collection using Raytracer and data cleanup.

She started by researching indoor localisation using the documents of the previous teams and online literature to identify the need for this project and the scope of work and the best machine learning models for this task, which she documented in the proposal. Post the proposal she worked heavily on data collection and K-Nearest Neighbor Model.

For data collection, she segmented the Bahen.stl map (Appendix B) in Blender into smaller sections and marked the x,y coordinates for each section. With the help of the coordinates, she worked with Hiranya to implement a python script to generate 2657 data points that are 0.7 feet apart in x direction and 1.7 feet apart in y direction throughout the map. Using these points collected and the RayTracer she ran convergence analysis to find the ideal number of reflections and transmission. Once all the unknowns were defined, she ran the RayTracer for 3 different access points. Finally, she cleaned and combined the data collected into a single file which can be accessed by the model. After data cleanup she moved to splitting the data into training and testing data.

Based on her previous research, she had chosen K-Nearest Neighbors as an ideal model for this task and for proof of concept she worked on implementing the sckit library’s K-Nearest Neighbor model to predict k closest neighbors for each testing data set of RSSI values. The team identified a pattern in the sckit libraries K-Nearest neighbor model and she went on to implement a custom K-Nearest Neighbor model to predict K closest neighbors using which she was able to get a base error of 4.9 m and is currently working on a better prediction function.

**Proposal/Research Work:**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Identify need for the project | Old | Completed | 17th September 2021 |  |
| 2 | Identify the scope of work | Old | Completed | 17th September 2021 |  |
| 3 | Create system context diagram (Appendix C.1) | Old | Completed | 17th September 2021 |  |
| 4 | Research on K-Nearest Neighbor Model | Old | Completed | 5th October 2021 | 9th October 2021 |

**Technical Work:**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Segmentation of Bahen map (Appendix C.2) | New | Completed | 15th October 2021 |  |
| 2 | Python script for selection of data points (Appendix C.3) | New | Completed | 25th October 2021 |  |
| 3 | Convergence Analysis (Appendix C.4) | New | Completed | 5th November 2021 | 12th November 2021 |
| 4 | Data collection4 using RayTracer | Old | Completed | 15th November 2021 | 20th November 2021 |
| 5 | Data Cleanup and Split into Training and Testing (Appendix C.5) | Old | Completed | 20th November 2021 | 22nd November 2021 |
| 6 | Implement scikit libraries K-Nearest Neighbor  (Appendix C.6) | New | Completed | 24th November 2021 | 30th November 2021 |
| 7 | Implement custom nearest neighbor model (Appendix C.7) | Old | Completed | 10th January 2022 | 13th January 2022 |
| 8 | Implement prediction function (Appendix C.8) | Old | In Progress | 28th January 2022 |  |
| 9 | Compare ANN vs K-Nearest Neighbors | Modified | In progress | February 13th, 2022 | February 28th, 2022 |

## **Deeksha Tewari**

Deeksha’s technical contributions to this project was configuring the physical data collection of WiFi Strength (RSSI) vs location values, researching new software tools and their setup, and implementing a supervised machine learning algorithm (Appendix A.1 and Appendix A.2).

For the project proposal, Deeksha worked to develop the problem statement and goal for the project proposal. She also helped generate and edit the Bahen .stl file using Blender (Appendix B). This helped us enter receiver points for the RayTracer simulation input, and decode the locations and RSSI value measurements output by the Raytracer simulation.

Deeksha also researched, formulated, and tested the experimental data collection strategy. The strategy involved measuring the x,y distance from an origin point determined in the Bahen .stl file and then measuring the strength of the Wi-Fi signal strength in dBm at that location. Along with John, she researched open-source applications like NetSpot and iPhone’s Airport Utility to measure the RSSI strength (Appendix C.1). The RSSI values can be read off the software and then manually entered into an .xlsx file (Appendix C.2). This will be the labeled experimental data we can use to train and validate the machine learning model. However, as indicated in Figure 1, the in-person data collection has been skipped for now due to delays caused by Covid-19 related building closures. This prevented us from collecting this data in December as previously planned.

At present, Deeksha is focusing on researching and prototyping the ANN (Artificial Neural Networks).

**Proposal/Research work (Appendix A.1 and Appendix A.2)**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Project Proposal | Old | Completed | 20th September |  |
| 3 | Research ANN model | New | In Progress | - | 18th January |

**Technical Work (Appendix A.1 and Appendix A.2)**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Blender setup and .stl synthesis | Old | Completed | 10th October 2021 | **-** |
| 2 | Research applications to measure signal strength (RSSI). | Modified | Completed | 24th October 2021 |  |
| 3 | Test Data Collection | Old | Completed | 21st November, 2021 |  |
| 3 | Full Data Collection with Raspberry Pi | Modified | Delayed | December 27th, 2021 | February 28th, 2022 |
| 4 | Data Augmentation | Modified | Delayed | January 1st, 2021 | February 28th, 2022 |
| 5 | Prototype ANN | New | In progress | - | January 31st, 2022 |
| 6 | Compare ANN vs K-Nearest Neighbors | Modified | In progress | February 13th, 2022 | February 28th, 2022 |

## **John Ambrad**

John’s primary task has been to configure the Raspberry Pi as well as gather live data from the 8th floor of Bahen.

He attended and scheduled meetings with the team to discuss how to configure the Raspberry Pi’s and what the data output from the Raspberry Pi would look like.

John has utilized openWRT software to configure three Raspberry Pi 3B+ microcomputers to broadcast a Wi-Fi signal. He has also worked with Deeksha to research available software that can measure RSSI. He also obtained RSSI and location data from the 8th floor of Bahen (Appendix D.2). However, due to Covid-19 restrictions this task could not be completed the team may choose to continue live data gathering as the situation continues to evolve. He has begun work on random forest regressors as an alternate model to the KNN model and has achieved a Mean absolute error of 4.42 (Appendix E.1).

**Proposal/Research work**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Project Proposal | Old | Completed | 20th September |  |
| 2 | Final Proposal | Old | Completed | 17th September |  |
| 3 | Research ANN model | New | In Progress | - | 18th January |

**Technical Work:**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Configuring Raspberry Pi | Old | Delayed |  |  |
| 2 | Data Collection on Raspberry Pi (Appendix D.2) | Modified | Delayed | 21st November, 2021 |  |
| 3 | Data Augmentation | Modified | Delayed |  | February 28th, 2022 |
| 4 | Prototype Random Forest (Appendix E.1) | New | In progress | - | January 13th, 2022 |
| 5 | Create custom Random Forest (Appendix E.1) | New | In progress | - | January 13th, 2022 |

## 

## **Hiranya Maharaja**

Hiranya has worked on researching and implementing the K-Nearest Neighbors model with Prerna, as well as creating the dataset using the RayTracer and the Bahen stl file (Appendix B). She helped create a python script to generate over 2657 data points by using points that are 0.7 and 1.7 feet apart in either direction on the Bahen.stl map (Appendix C.3). Along with Prerna, she created a proof-of-concept K-Nearest Neighbors model, but found that the accuracy obtained by the pre-trained model from the scikit-learn library was not sufficient, so she then helped create a custom model. She also created a plot to visualize the model’s error, to aid in hyperparameter choices for the custom model (Appendix F.1).

For the custom K-Nearest Neighbors model, Hiranya designed a k-means clustering algorithm (Appendix F.2). This function was designed to take in the K-Nearest Neighbors predicted by the distance function created by Prerna, and output a final prediction of the x-y location most likely to correspond to the given RSSI passed into the overall model. Currently, she is working on fine-tuning this model to improve its overall accuracy.

**Proposal Work:**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Define the project requirements | Old | Completed | 17th September 2021 |  |
| 2 | Complete a feasibility assessment on the project | Old | Completed | 17th September 2021 |  |

**Technical Work:**

| **Task #** | **Task Title** | **Category** | **Status** | **Old Completion Date** | **New Completion Date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Python script for dataset generation | New | Completed | 25th October 2021 |  |
| 3 | Implement a prediction algorithm for K-Nearest Neighbors model from scikit-learn | New | Completed | 26th November 2021 | 2nd December 2021 |
| 4 | Implement a k-means clustering algorithm for the custom K-Nearest Neighbors model | New | Completed | 12th January 2022 | 15th January 2022 |
| 5 | Identify the best parameters (distance equation, clustering algorithm) to predict location - i.e., hypertune model | Old | In Progress | 28th January 2022 |  |

# 

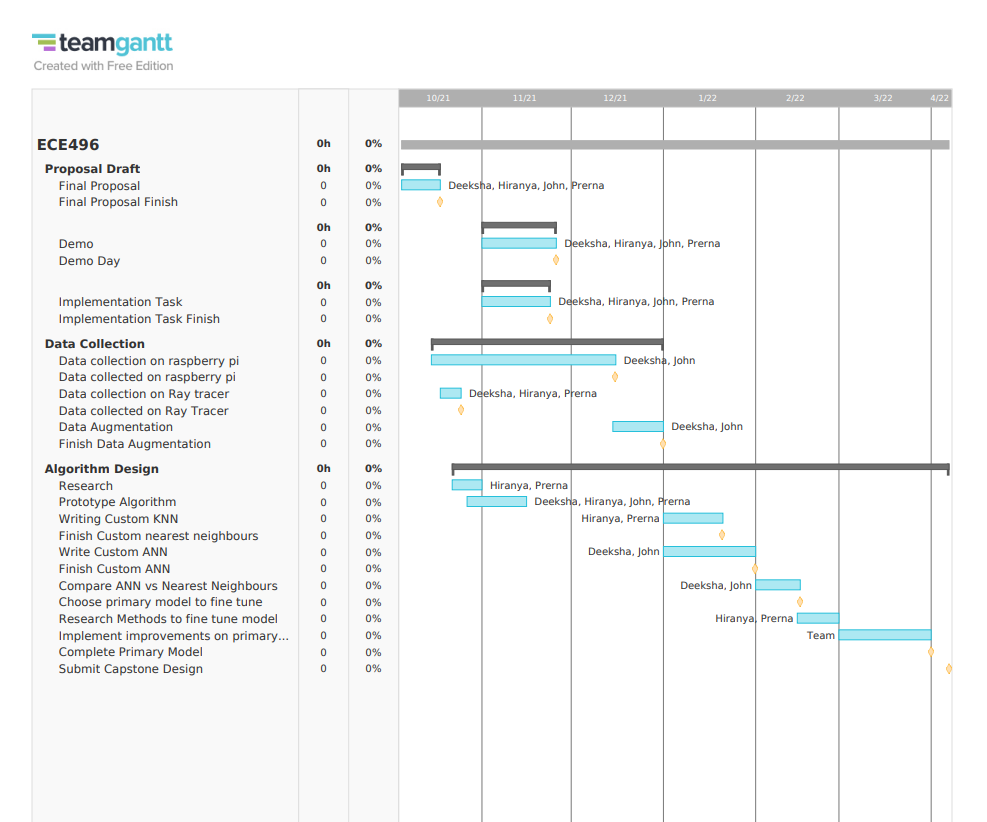
# **Conclusions/Project Assessment**

So far, Prerna and Hiranya have been able to complete a K-Nearest Neighbours model using scikit-learn’s pre-existing functions as a proof of concept, as well as train a custom K-Nearest Neighbours model to fine-tune for more accurate predictions. Also, Deeksha has been able to configure the raspberry pi’s along with the openWRT software along with John, and is currently working on an ANN model to compare with the K-Nearest Neighbors model. These tasks are all progressing smoothly and remain aligned with our Gantt chart. While additional in-person collection of data had to be postponed due to Covid-19 restrictions, Prerna and Hiranya’s custom model used the RayTracer data. However, depending on how the Covid-19 restrictions evolve, Deeksha and John may be able to complete the in-person data collection. The data will be fed into the ANN and K-Nearest Neighbor model, after which the team will fine-tune the best model to achieve the highest possible accuracy.

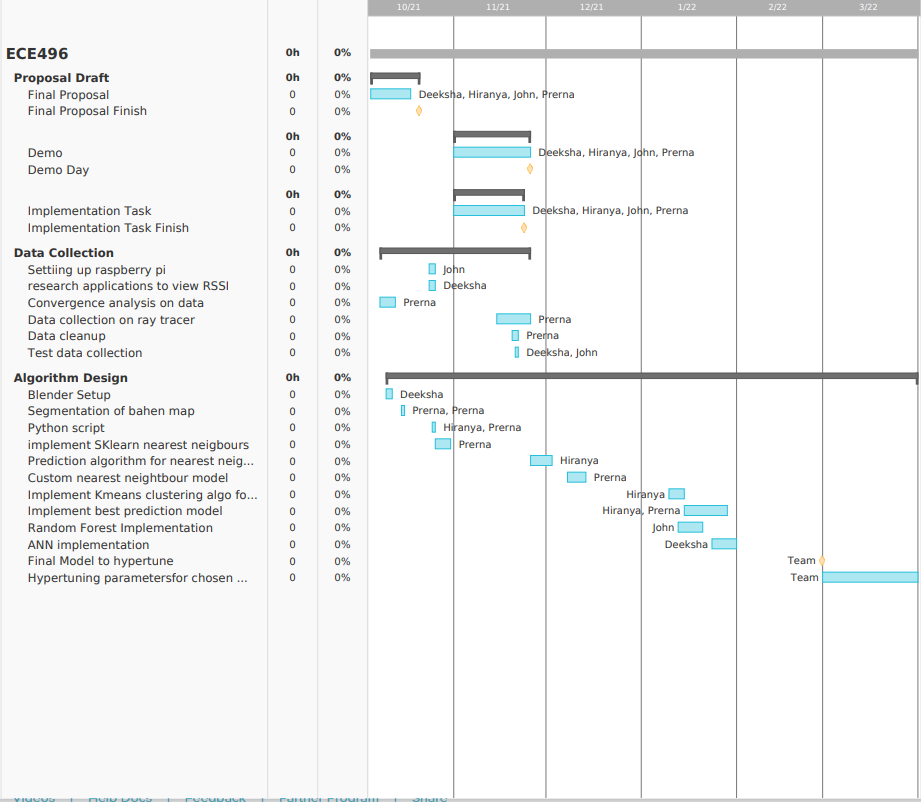
# **Appendices**

## **Appendix A: Gantt charts**

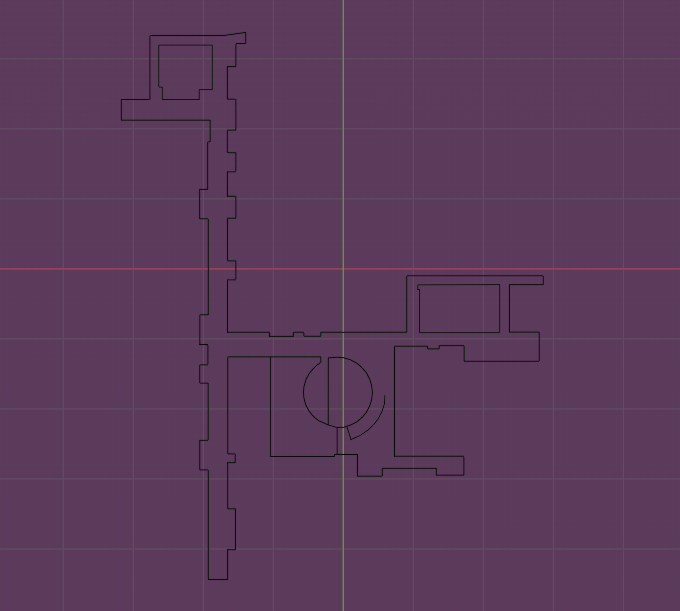
### Appendix A.1: Implementation Gantt Chart

****

### Appendix A.2: Updated Gantt Chart

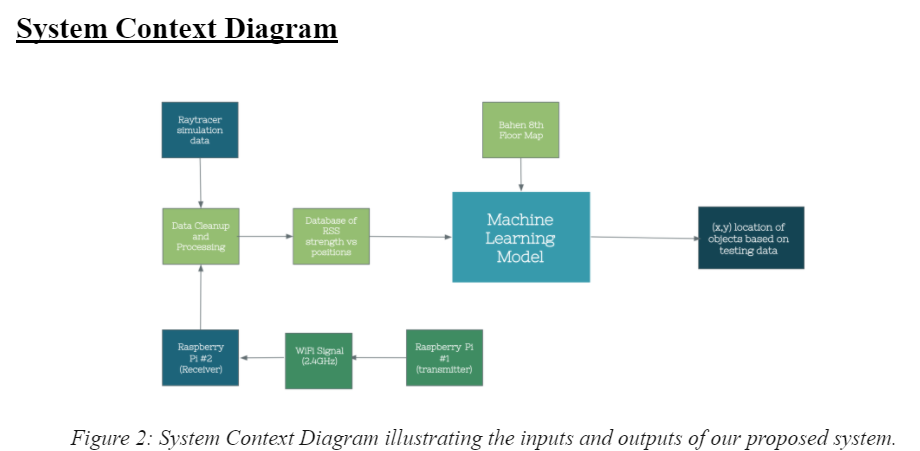


## **Appendix B: Bahen stl file**

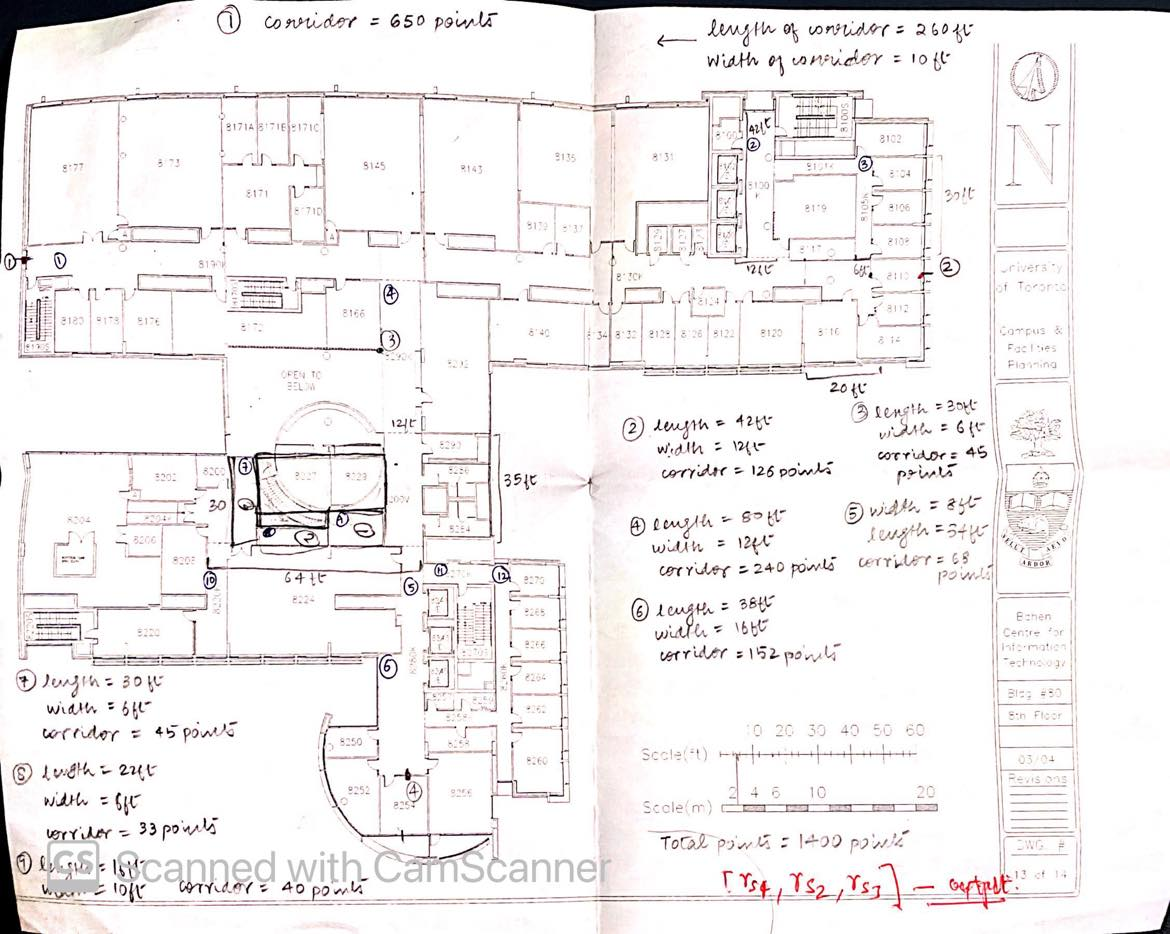
****

## **Appendix C: Prerna Anand**

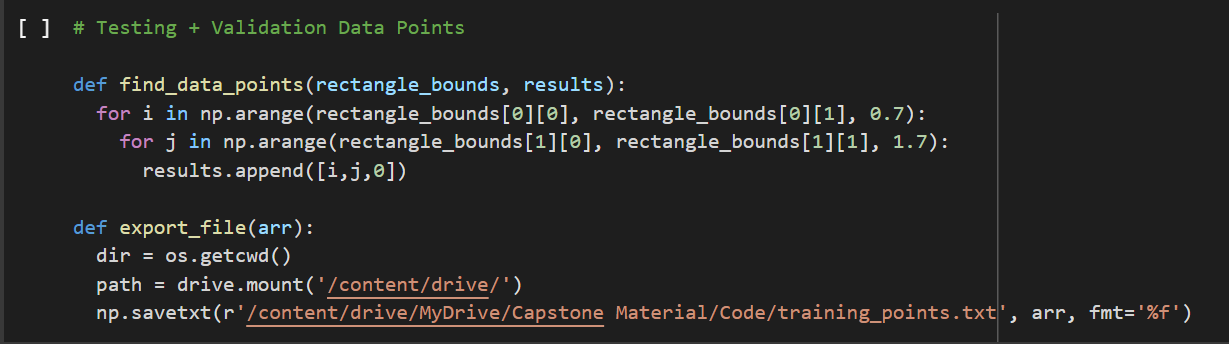
### Appendix C.1 : System Context Diagram



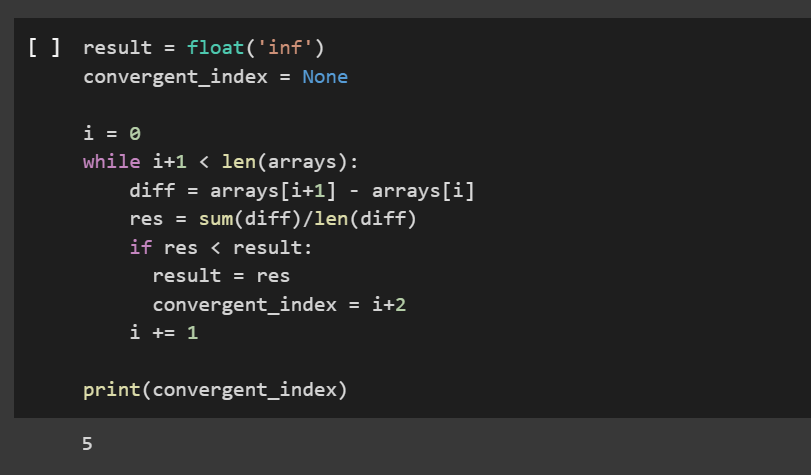
### Appendix C.2 : Draft segmentation of Bahen Map



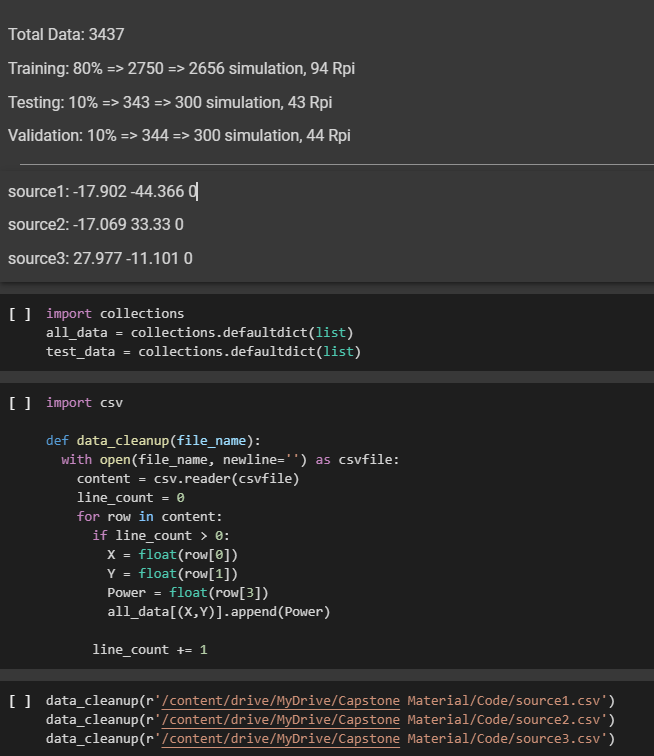
### Appendix C.3 : Python script for simulation of data points

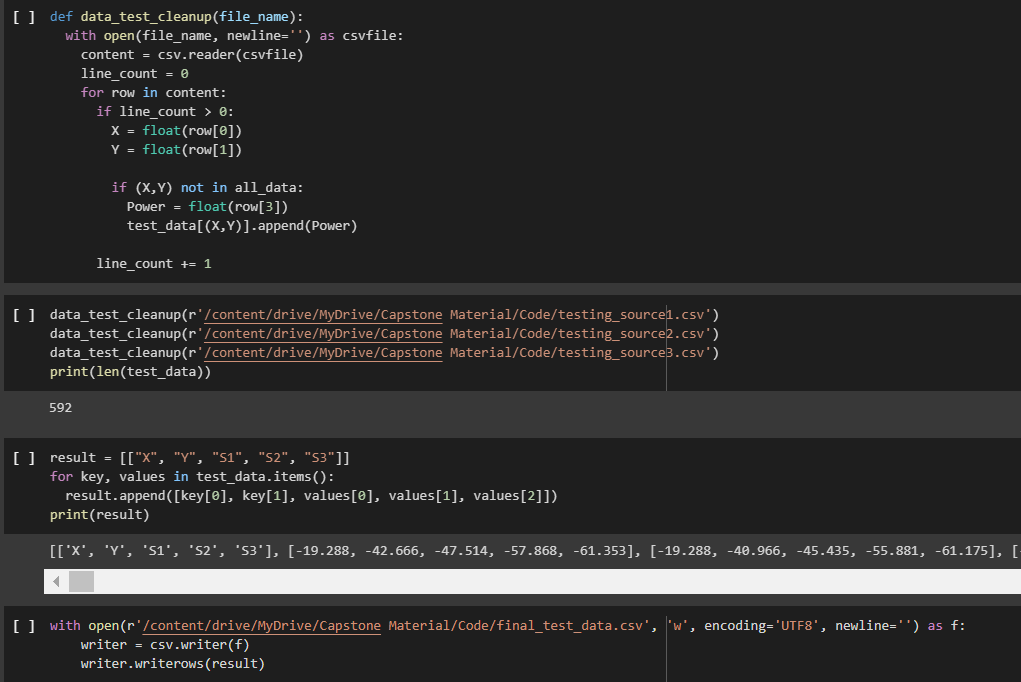


### Appendix C.4 : Convergence Analysis

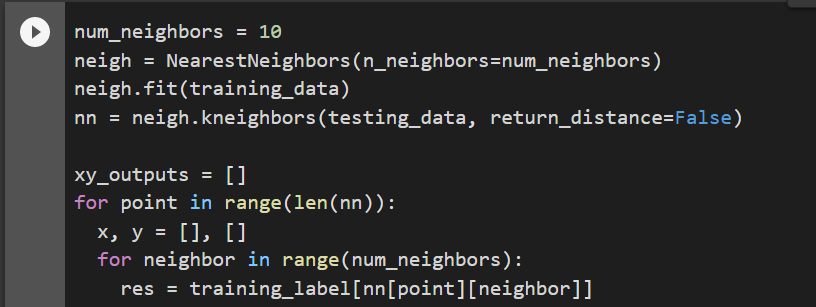


### Appendix C.5 : Data cleanup and splitting into training and testing

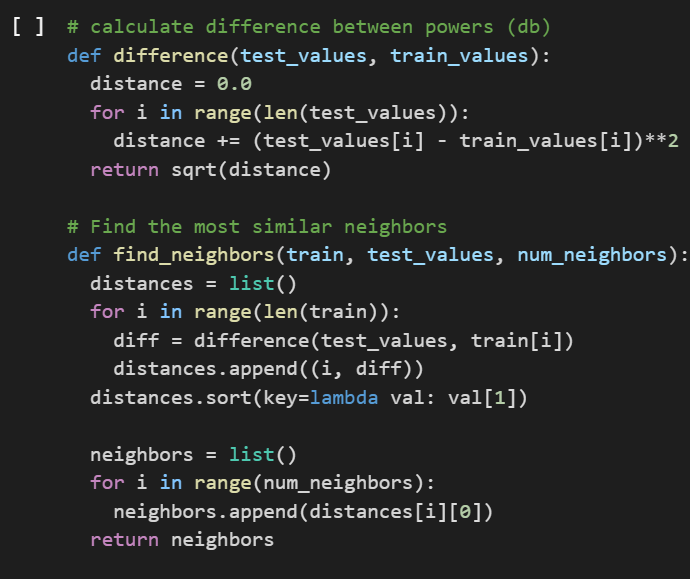




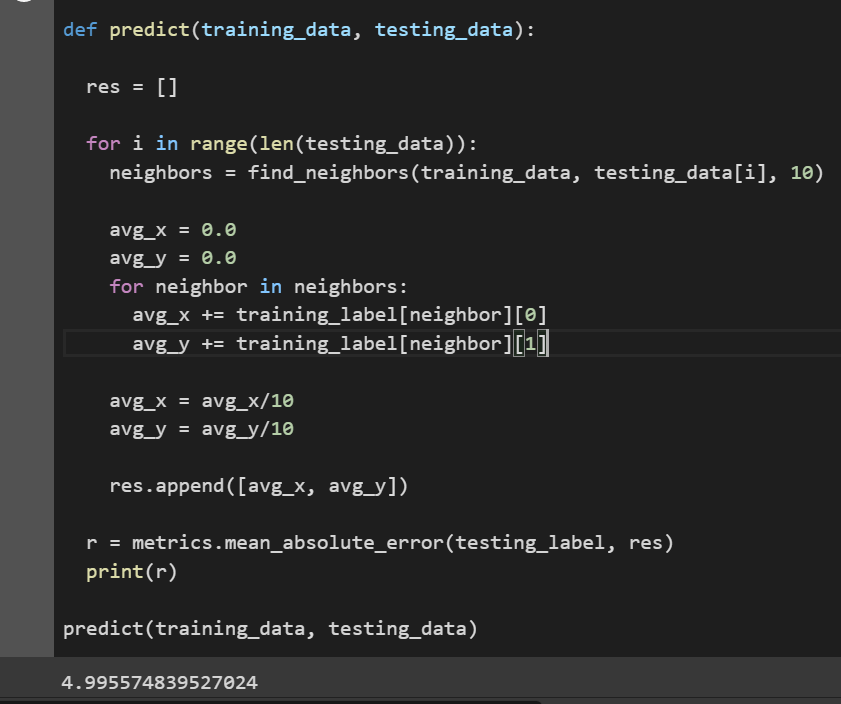
### Appendix C.6 : scikit library’s K-Nearest Neighbor model



### Appendix C.7 : Custom K-Nearest Neighbor model

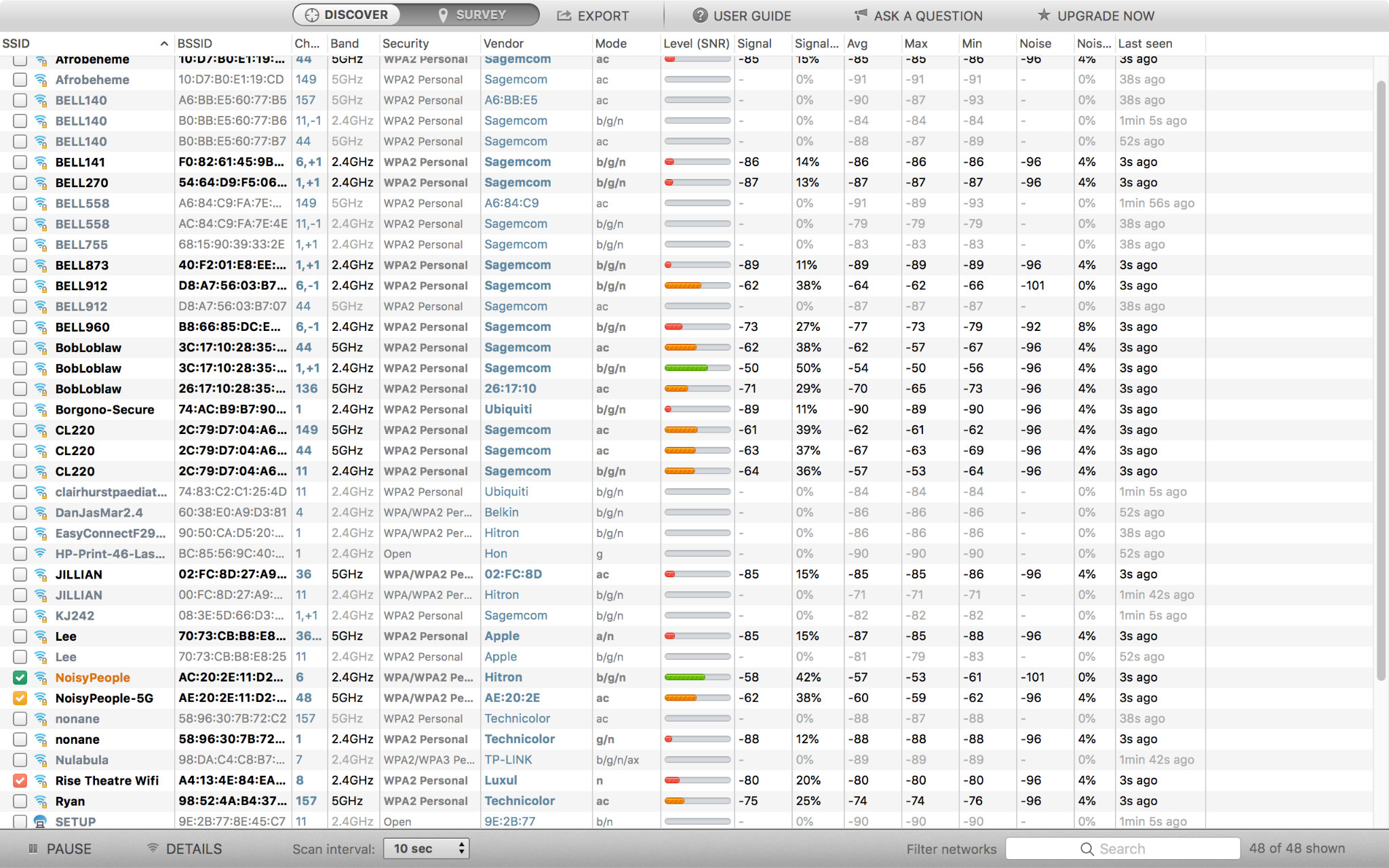


### Appendix C.8 : Prediction function



## **Appendix D: Deeksha Tewari**

### Appendix D.1: Screenshot of NetSpot Application used to perform data collection



The following screenshot illustrates the basic functioning of the NetSpot application - the application scans for networks every 10 seconds and shows a list of available networks. As shown, the coloured check marks show the network the device is connected to.

### Appendix D.2: Data points collected on Bahen 8th floor using NetSpot

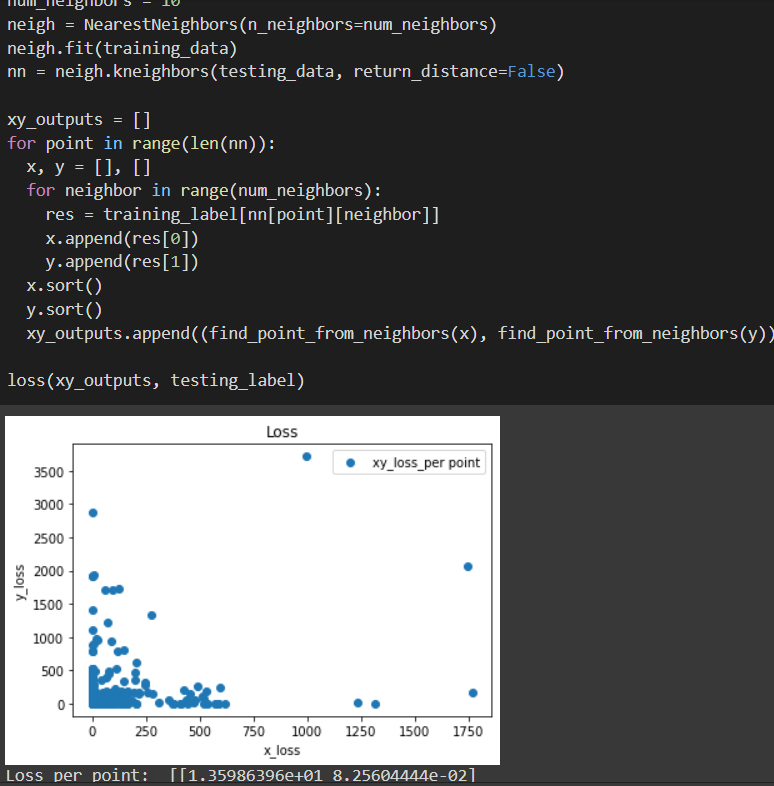
| **X** | **Y** | **Strength (RSSI)** |
| --- | --- | --- |
| 3.7 | 5 | -42 |
| -1.5 | 17 | -50 |
| 9 | 27 | -50 |
| -4.5 | 36 | -52 |
| 1.5 | 70 | -58 |
| 1.5 | 90 | -62 |
| 4.5 | 160 | -63 |
| -4.5 | 170 | -71 |
| -7.5 | 170 | -75 |
| 5 | 180 | -72 |
| 4.5 | 180 | -71 |
| -4.5 | 180 | -72 |

## **Appendix E: John Ambrad**

### Appendix E.1 Screenshot of Random Forest Code

## **Appendix F: Hiranya Maharaja**

### F.1: Plot of xy loss values for each point using MSE in the scikit-learn KNN



### F.2: K-means clustering algorithm

