Benchmarking Low-cost Air Quality (PM2.5) Sensors-Examining Their Potential to Complement Existing Pollution-Measurement Frameworks in Pittsburgh

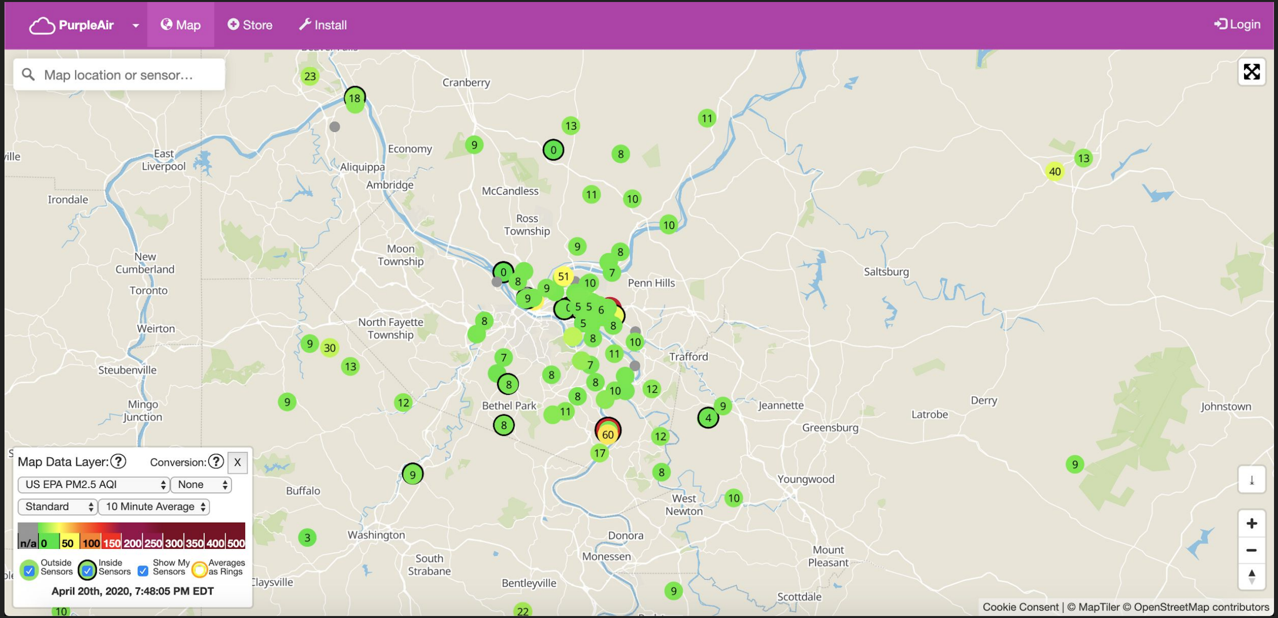
Abhishek Viswanathan, Vasco Xu
Instructor: Dr. Amy Babay
CS 3551: Advanced Topics in Distributed Information Systems
Spring 2020

Problem

- Not enough air quality data on local scales
- Uncertainty about accuracy of low-cost sensors
  - If the data is inaccurate, how can we adjust/calibrate its values?
- How to incorporate low-cost sensor data into environmental models
- How to get citizens to participate with easy-to-use devices
- How to automatically visualize citizen-science data

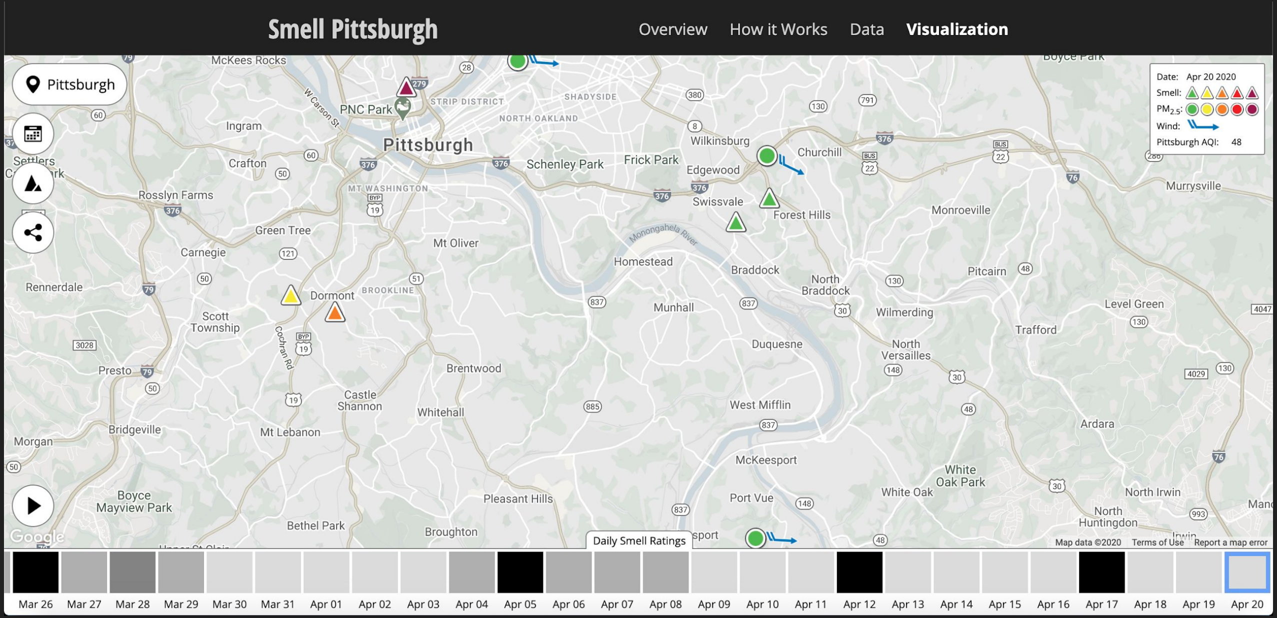
Related Work

PurpleAir



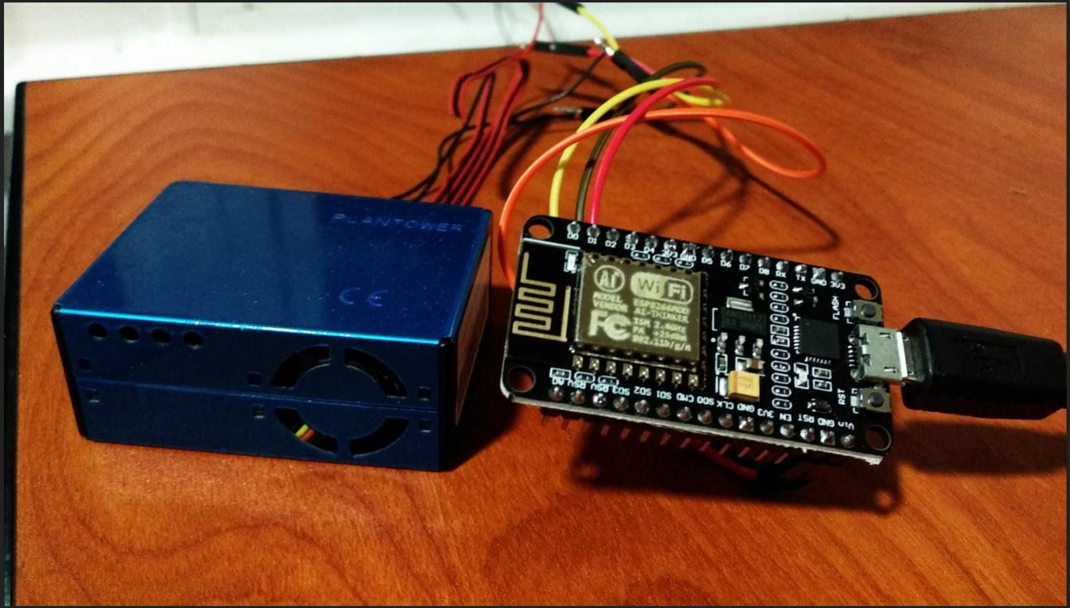
Related Work

SmellPGH



Approach

- Built prototype with inexpensive PMS5003 sensor to automatically post data to web server.



Approach

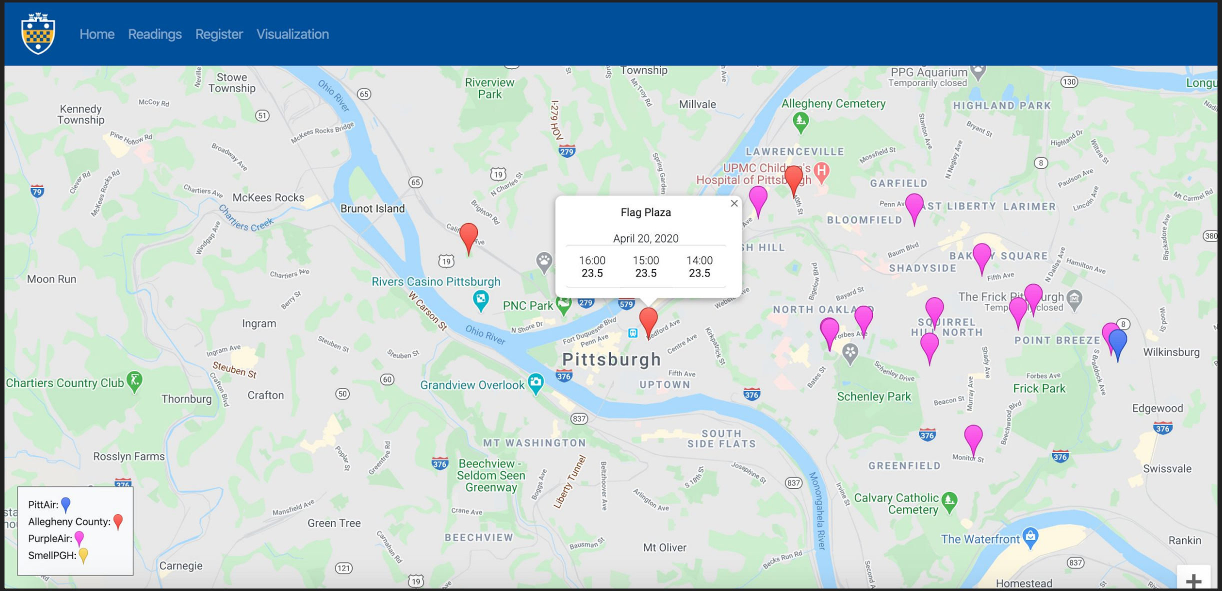
- Built web-server hosted on Heroku to receive and store sensor data.

Home Readings Register Visualization

Device Name	Time	PM0.3	PM0.5	PM1.0	PM2.5	PM10
Maple	20/04/2020 19:54:18	351	102	1	2	3
Pine	20/04/2020 19:49:03	348	104	1	2	2
Maple	20/04/2020 19:44:10	378	104	1	2	2
Pine	20/04/2020 19:38:54	282	84	1	1	1
Maple	20/04/2020 19:34:00	264	75	1	1	1
Pine	20/04/2020 19:28:53	288	76	1	1	1
Maple	20/04/2020 19:23:59	357	95	2	2	3
Pine	20/04/2020 19:18:44	288	90	1	2	2
Maple	20/04/2020 19:13:50	267	83	1	1	2
Pine	20/04/2020 19:08:42	330	98	1	2	2

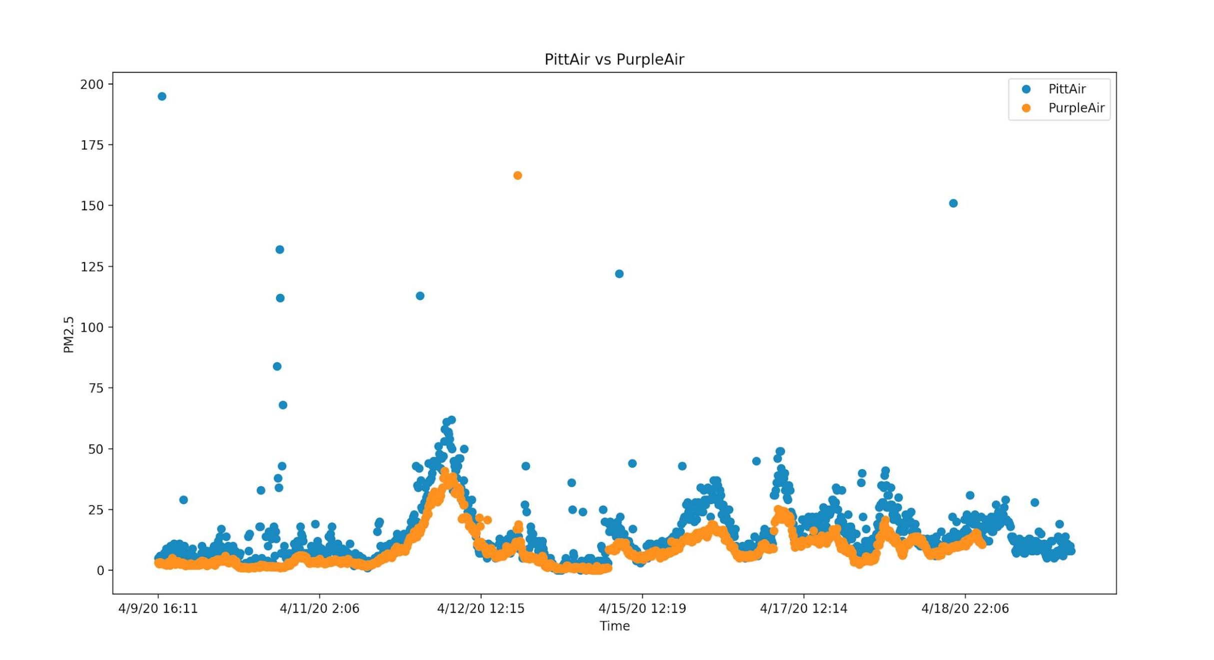
Approach

- Visualize data from other projects - SmellPGH, PurpleAir, Allegheny County Official Data



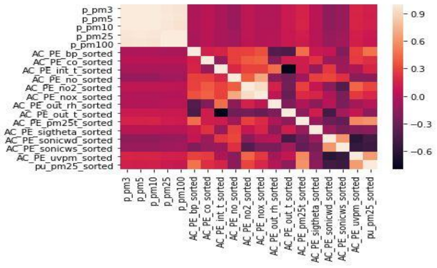
Approach

- ML + Data Analysis



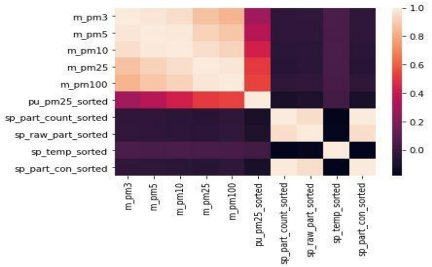
Analysis

```
In [5]: sns.heatmap(outdoor.corr())
Out[5]: <matplotlib.axes._subplots.AxesSubplot at 0x1f5d5e22b00>
```




```
In [11]: from sklearn.metrics import r2_score
         r2_score (y_test, y_pred)
Out[11]: 0.4977079199997898

In [13]: from sklearn.ensemble import RandomForestRegressor
         from sklearn import metrics

         regressor = RandomForestRegressor(n_estimators=1000, random_state=0)
         regressor.fit(X_train, y_train)
         y_pred = regressor.predict(X_test)

         print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
         print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
         print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

         Mean Absolute Error: 0.33816015625
         Mean Squared Error: 0.40376442578125005
         Root Mean Squared Error: 0.6354246027509873
```

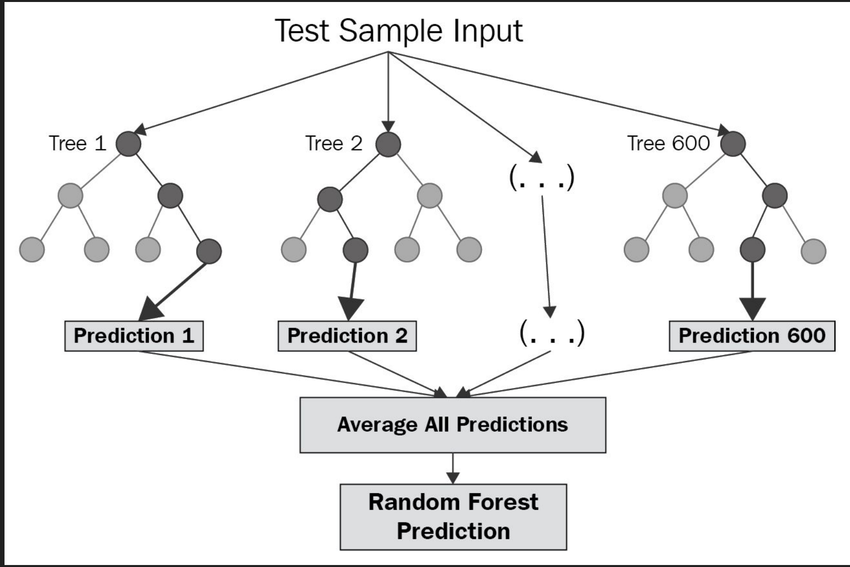
```
In [97]: sns.heatmap(indoor.corr())
Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x27496582dd8>
```




```
In [11]: from sklearn.metrics import r2_score
         r2_score (y_test, y_pred)
Out[11]: 0.6516819916142947

In [16]: from sklearn.ensemble import RandomForestRegressor
         from sklearn import metrics

         regressor = RandomForestRegressor(n_estimators=1000, random_state=0)
         regressor.fit(X_train, y_train)
         y_pred = regressor.predict(X_test)

         print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
         print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
         print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

         Mean Absolute Error: 1.082593021577374
         Mean Squared Error: 2.5691376767934293
         Root Mean Squared Error: 1.6028529804050742
```

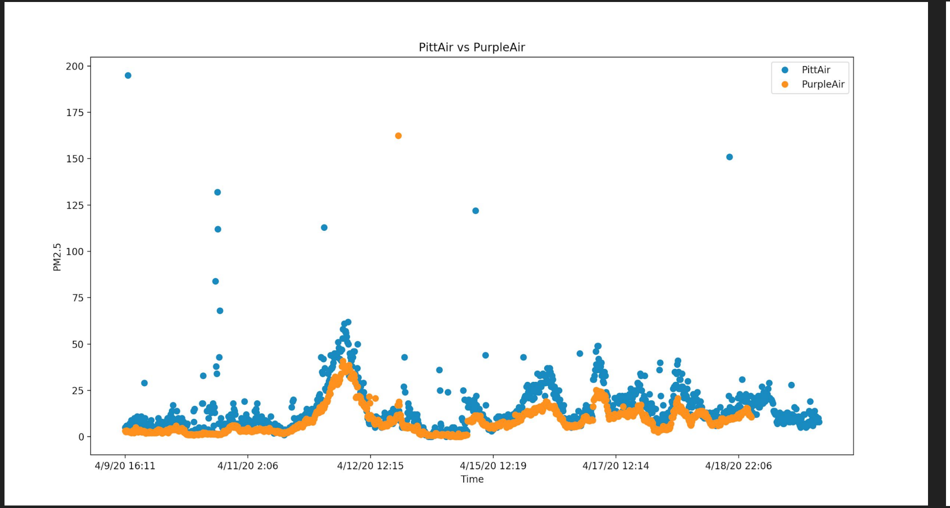
Analysis

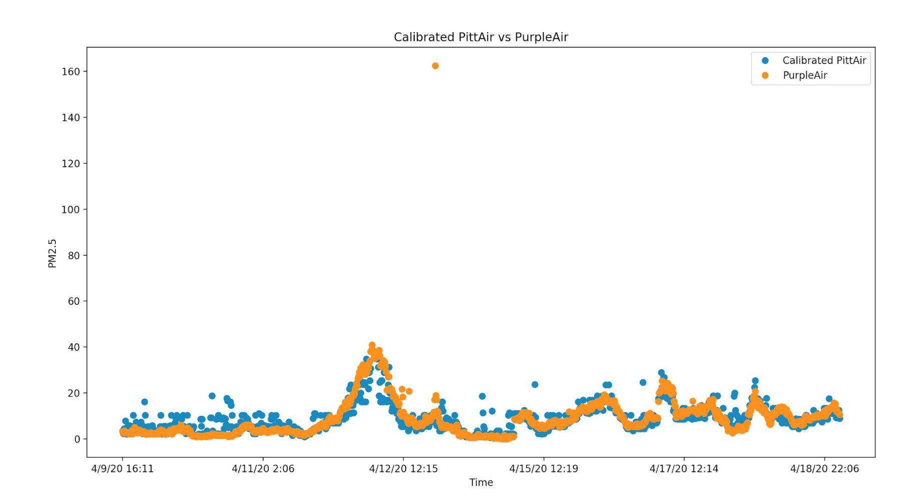
- Random Forest Regression (random forests are run in parallel)
  - Attempts to avoid overfitting.
  - Useful for non-linear data.



Results/Analysis

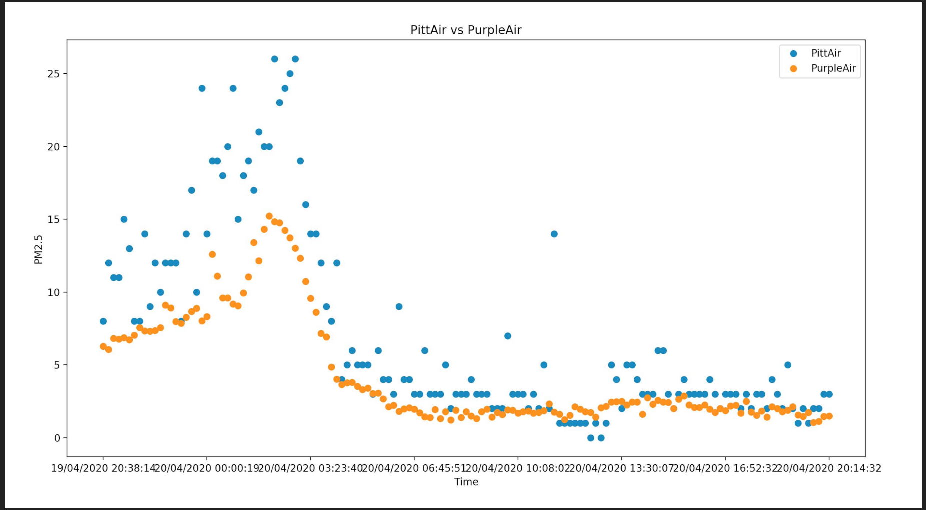
- Left figure: PittAir sensor values (placed inside a house) vs. PurpleAir sensor values (presumably also inside).
- Right figure: Calibrated PittAir sensor with Random Forest Regression ML (of 77% accuracy).
    - Total values used: 1132.
    - 75% of values used for training and 25% used for testing.

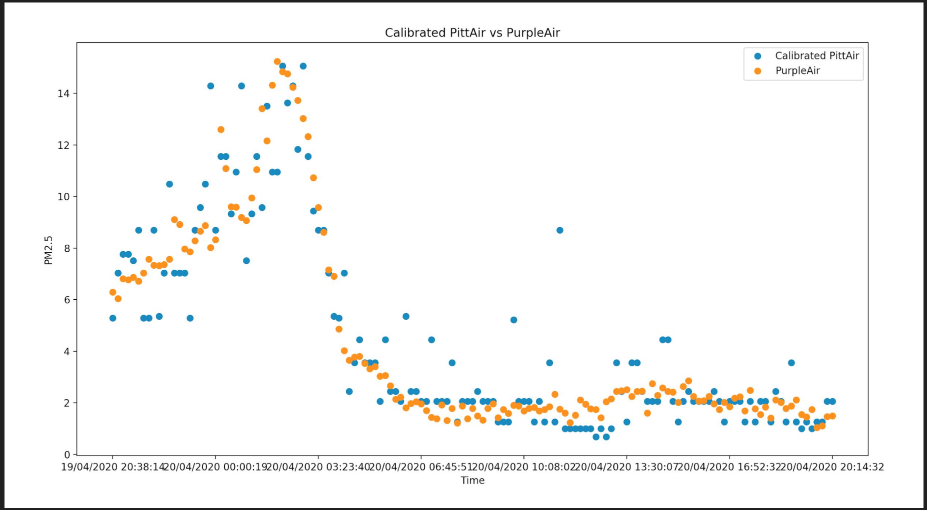




Results/Analysis

- Tested the accuracy of the model by using values not used in building or testing the ML model.
- Left figure: PittAir sensor values vs. PurpleAir sensor values.
- Right figure: Calibrated PittAir sensor values with ML model from previous slide (141 values used).





Future Work

- Improve outdoor models by calibrating near existing sensors hourly and analyzing its accuracy.
- Making collected data available in different formats for analysis.
- Automatically visualize the data on a graph on the website.
- Posting analysis/data automatically to Slack via an API.

Questions?

Thank you!