Raspberry Pis: Visualization Library and Statistical Analysis From Sensor Calibration Through Machine Learning

Daniel Han

December 2023

Introduction

Measuring local air quality is an important aspect of atmospheric chemistry. However, obtaining accurate readings requires the use of expensive scientific instruments that are unaffordable for individuals and institutions. The aim of this project is to create a low-cost sensor that would lower the bar to participate in monitoring local air health. The long-term vision is to allow citizen scientists and academic institutions such as high schools create a network of sensors to gather and upload data into a collective database that would monitor changes in air quality around Williamsburg, Virginia. Specifically, the "Raspberry Pi" would measure temperature, humidity, and particulate matter (PM2.5 and PM10). However, since sensor performance positively correlates with cost, the performance of the low-cost sensors yielded inaccurate results.

A co-location method was implemented in order to gather data for calibration. From July 22, 2022 to July 27, 2022, two Raspberry Pis were placed near the most extensive air quality monitoring facility in Virginia at the Virginia Department of Environmental Quality's air quality monitoring station at the MathScience Innovation Center site in Richmond. These sensors were then calibrated using the readings from the Virginia Department of Environmental Quality (DEQ).

Data Preprocessing

After the co-location period ended, data was retrieved and preprocessed to generate a machine learning model. To create an accurate machine learning model, the DEQ and co-location data must be organized exactly the same. However, the raw data collected by both the DEQ and co-located Raspberry Pi units generated multiple incompatibilities. The first step to the preprocessing process was aligning the two datasets along an absolute timescale. The data from the co-location Pis were collected and logged onto a relative time scale, where time was represented as a repeating pattern of 0-60 seconds with data collection points at every 5 second interval, whereas the DEQ data had readings every minute lined up with the date. The co-location data also had a pattern of logging, where one-minute measurements would be made every 14 minutes, creating a 15-minute cycle. To match the readings onto an absolute timescale, we first averaged the readings of the co-location data, creating a single data point for each minute of measurements. We then filtered the DEQ data to only include measurements made at the same time as the co-location. This resulted in 95% of the DEQ data to not be included in the calibration as these measurements had no equivalent in the co-location data.

There were also unforeseen errors with the DEQ sensors during their data collection period. From July 22-24th, the PM10 and PM2.5 sensors malfunctioned and failed to record any measurements. To remediate these errors, data from nearby sensors at Bryan Park was spliced into the missing values from the MSIC sensors. Note that these sensors were in a different location which most importantly was closer to a highway than the DEQ or co-location sensors, which should be kept in mind for model inaccuracies

Model Calibration

To calibrate the co-location data to more closely match the DEQ data, we created three machine learning models: linear regression, multiple linear regression, and random forest. Both linear regression and multiple linear regression generated unsatisfactory results, with high mean squared errors and low R2 values. Random forest was the only model to calibrate the co-location data accurately. The general shape of the data is captured well by the Raspberry Pi as seen in figure 1.

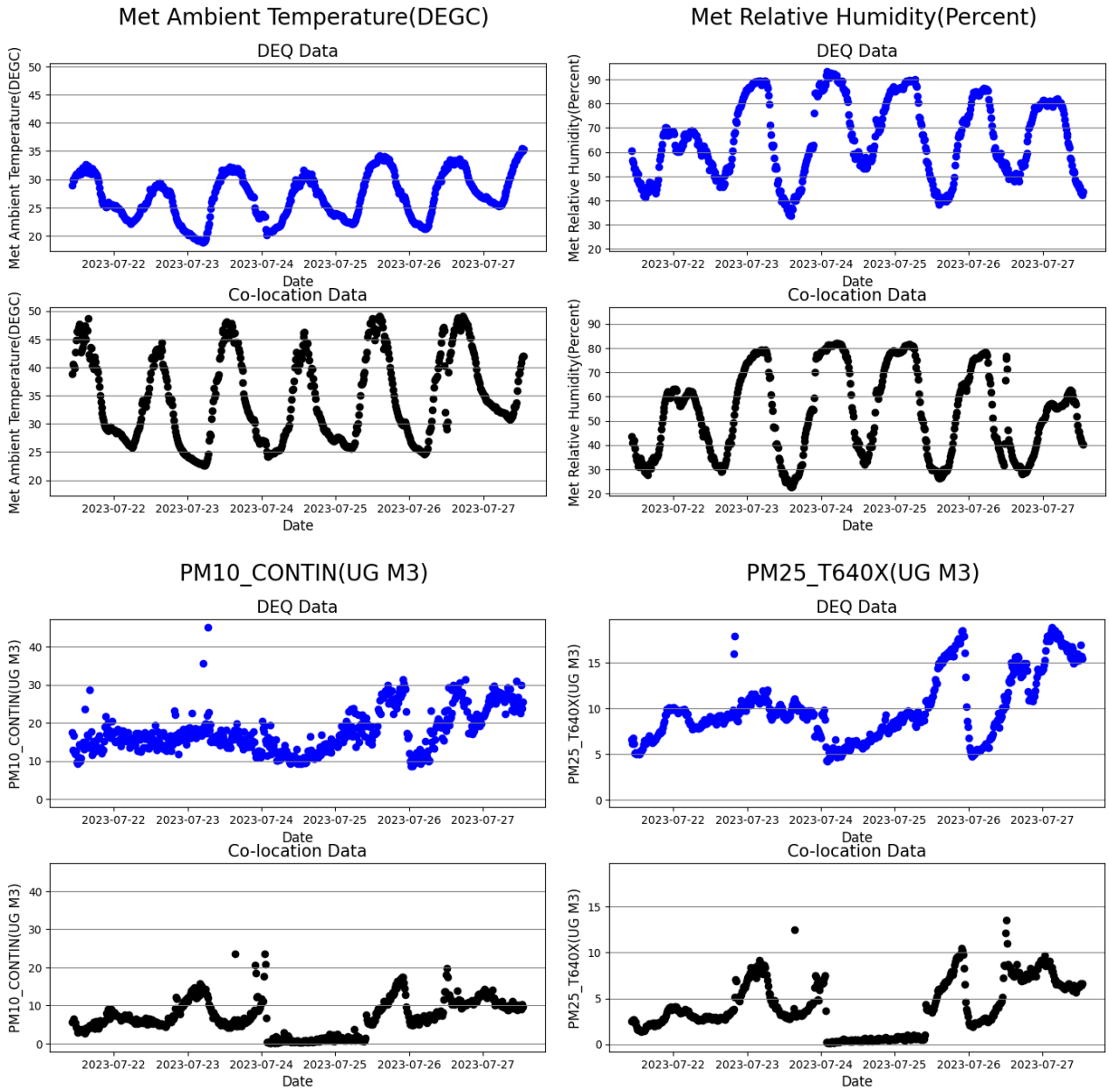

Figure 1. Scatterplots depicting the initial measurements of temperature, humidity, $PM_{10,}$ and $PM_{2.5}$ from Pi 1 compared to the corresponding data from the DEQ sensors. Note that for $PM_{10,}$ and $PM_{2.5}$ data was spliced in from sensors at Bryan Park from July 22-24.

The daily cycle of temperature and humidity are represented as well as a silhouette of PM10 and PM2.5. However, the distribution and range of data points left much to be desired. Our first two models, linear regression and multiple linear regression, produced similar results. The ranges of data were more closely matched but these models failed to accurately change the distribution of data points.

The final model, random forest, generated the most accurate model both visually and mathematically speaking. However, due to the limitations of random forest random forest regression the extremities of the data are not possible to model because the model is not able to extrapolate data.

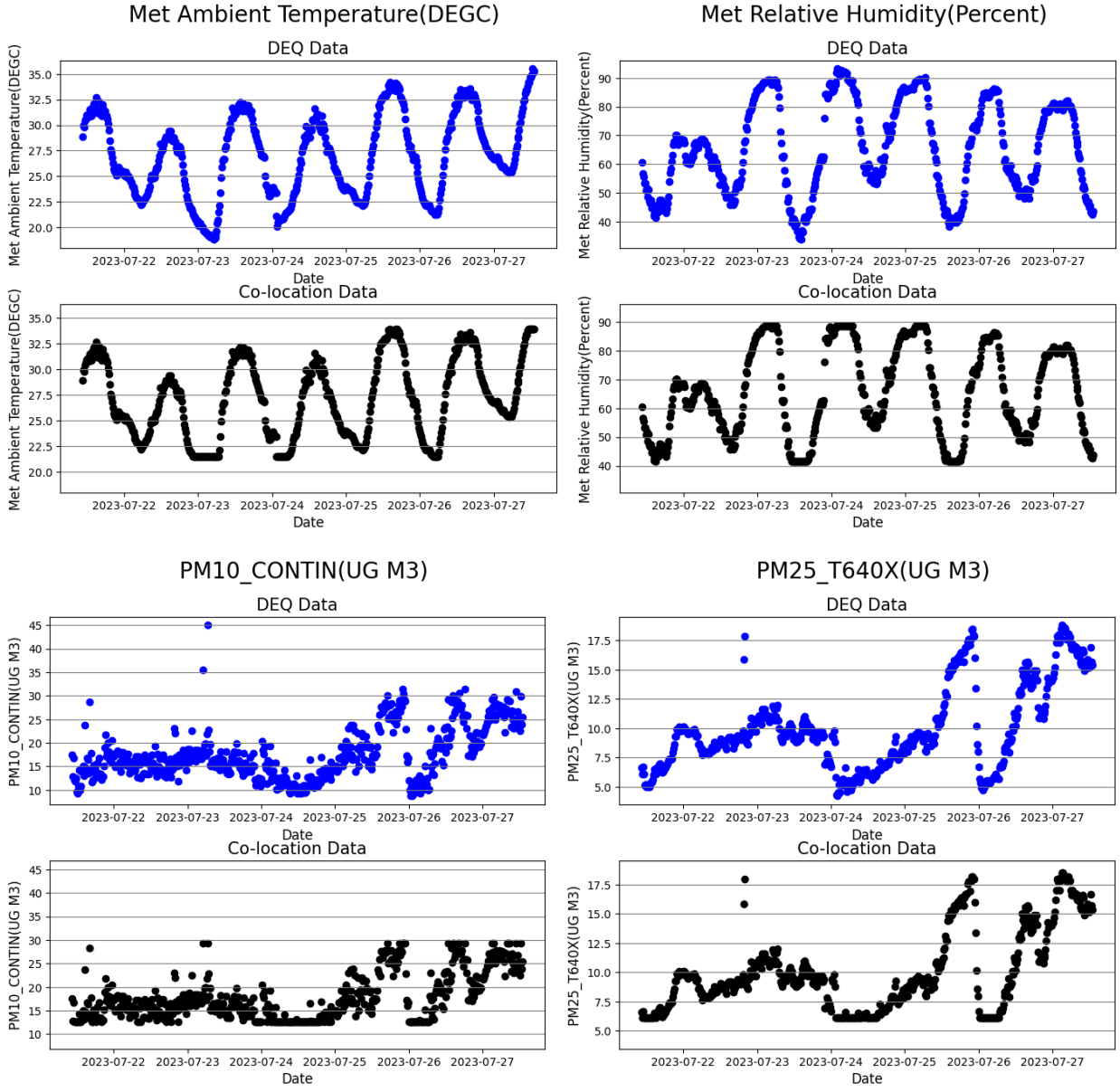

Figure 2. Results of a combination of linear regression and random forest models. Note the model's limitations seen at the extreme ends of data where it generates flat data values.

Random forest is similar to KNN in that it predicts values based on an average of previous data points, meaning that random forest regression is bound by the highest and lowest value found in its training set. This resulted in an accurate model for central data points but a complete inability to model the upper and lower bounds with flat data points where the ranges maxed out.

To remedy these limitations, we created a two-step process using both linear regression and random forest. We first processed the co-location data through linear regression as it increased the range of data to more closely match the DEQ data. Then this data was used in random forest regression to create the final model. The R2 and MSE values generated by this model were satisfactory despite the persisting limitations in calibrating extreme values and outliers. Although it is not known how this model would perform in a different season such as winter or fall, the applicability of this model however is limited. The data this model is trained on only covers a week in the summer of Virginia so it would have to be "re-calibrated" for each season and climate. A longer co-location period as well as more varied locations could solve this issue, combining the data and models from multiple locations and time periods into a comprehensive model.

A combined visualization was created to help put into perspective the gains made in data accuracy after calibration. The model created by random forest regression was applied to data collected from Pi 1. The DEQ points (red circles) are the target data points that the model (blue triangles) is trying to reproduce, note the significant change from the initial co-location data (gray squares) in figures 3, 4, 5, and 6.

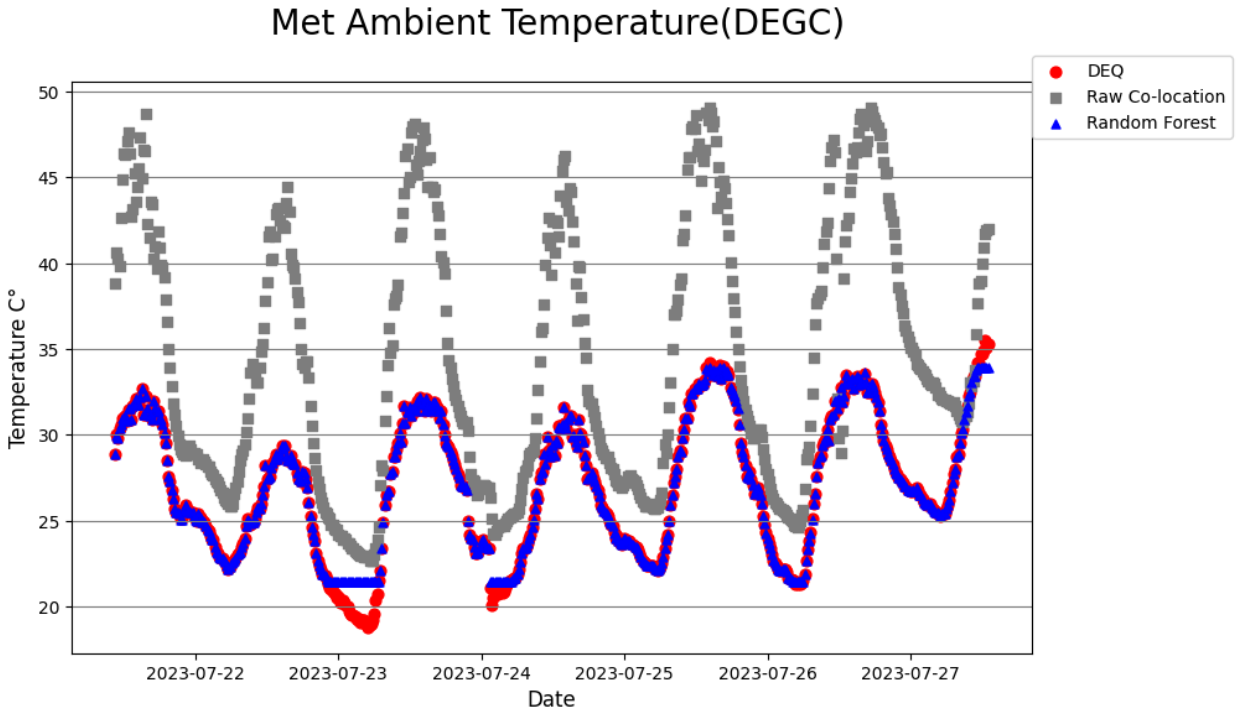

Figure 3. Results from the two-step process in regards to temperature. R2 = 0.988, MSE = 0.200 in the random forest model.

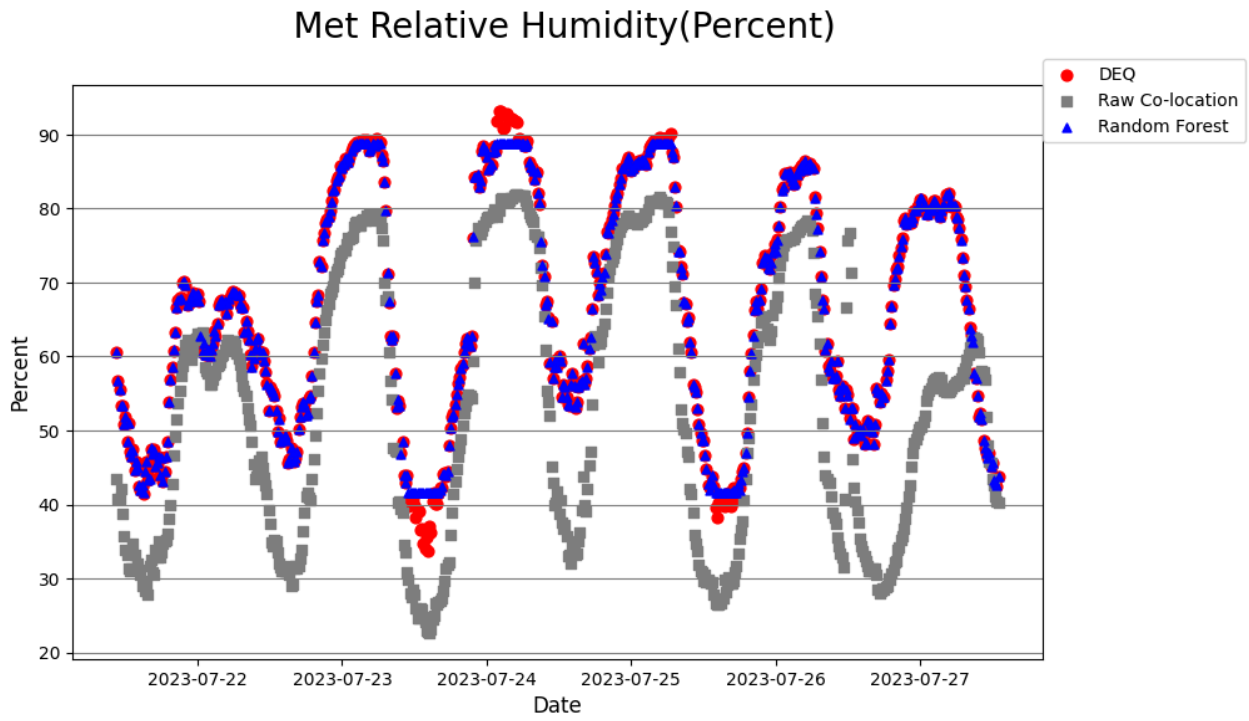

Figure 4. Results from the two-step process in regards to humidity. R2 = 0.996, MSE = 0.927 in the random forest model.

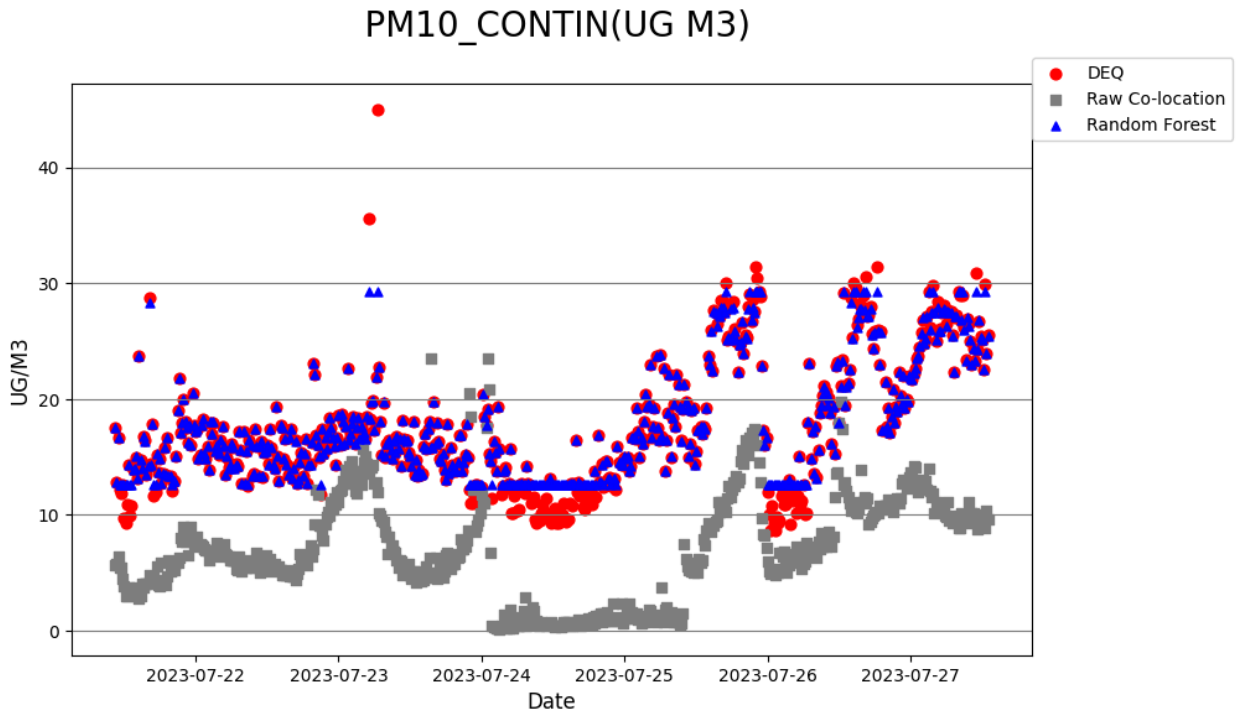

Figure 5. Results from two-step process in regards to PM10. R2 = 0.960, MSE = 1.225 in the random forest model.

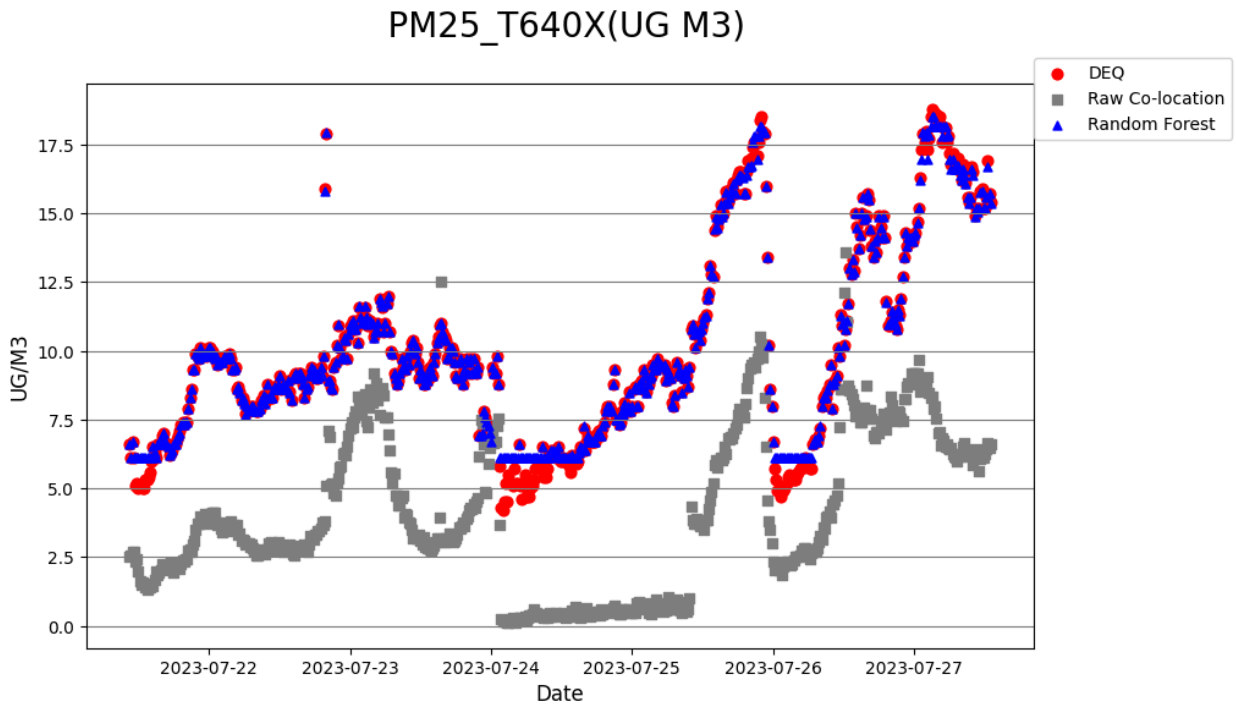

Figure 6. Results from two-step process in regards to PM25. R2 = 0.992, MSE = 0.105 in the random forest model.

Another way to visualize the shortcomings of the machine learning model is shown in figure 7, where points that lie on the horizontal red line represents a practical 1 to 1 match from the machine learning model to the DEQ data. Here, the errors are more pronounced, as the data is "normalized" such that correct values are uniform along the x axis. This also shows how the

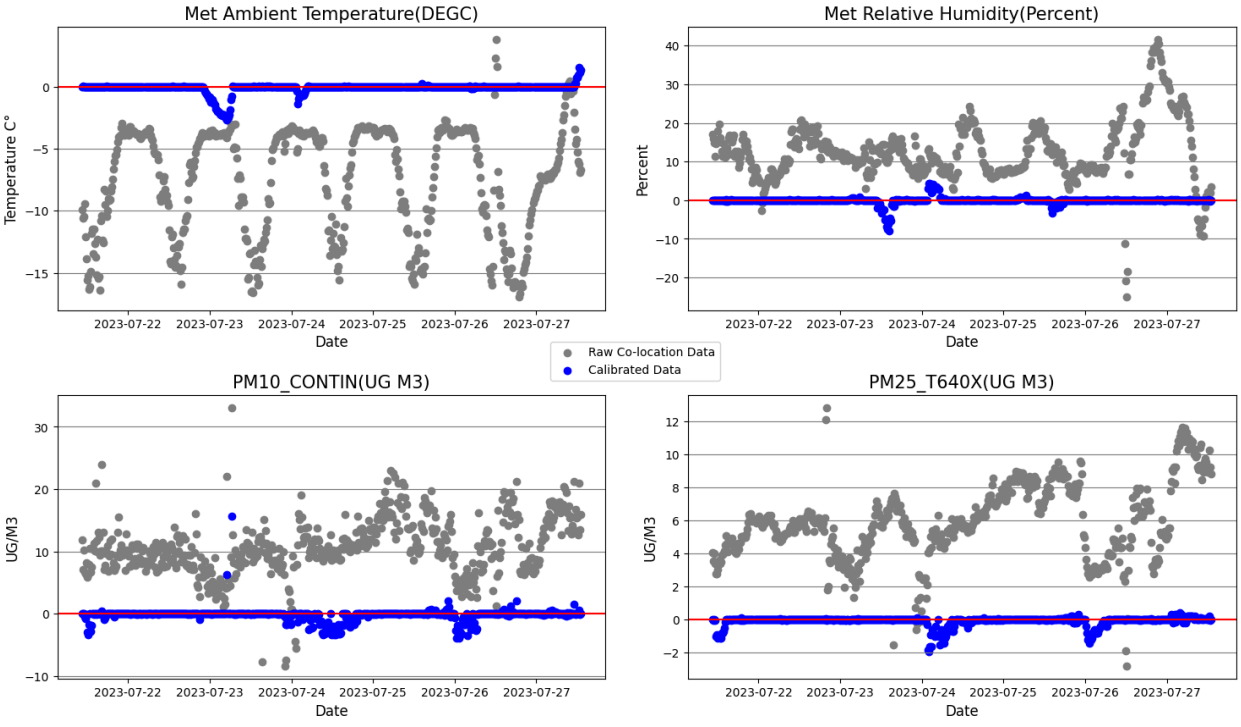

Figure 7. Graph of data points "normalized" along 0 axis, where 0 axis represents 0 deviation from DEQ data.

specific use to find high and low temperatures will not be a statistic that the sensors can measure, since the model is not able to predict values outside of its training range.

To further validate the results of the calibration, a correlation matrix was created of both the DEQ data and calibrated Pi 1 data. Comparing their correlations can give another perspective on the accuracy of the machine learning model across variables. This accuracy across variables is also important in the overall accuracy of the data, since there are physical relationships between each of the feature variables. Although the correlation values only differ by a few points, these relationships between variables could mislead scientists. For example, the 0.02 inaccuracy in PM10 and Humidity between the DEQ and co-location data could be misread as a chemical imbalance in the air due to pollution. Thus, the Raspberry Pi sensors would not be suited towards detecting specific relationships between variables as it is not accurate enough to warrant these inferences. Note that there are other factors that could also affect these variables and the application of this comparison is limited to this specific co-location.

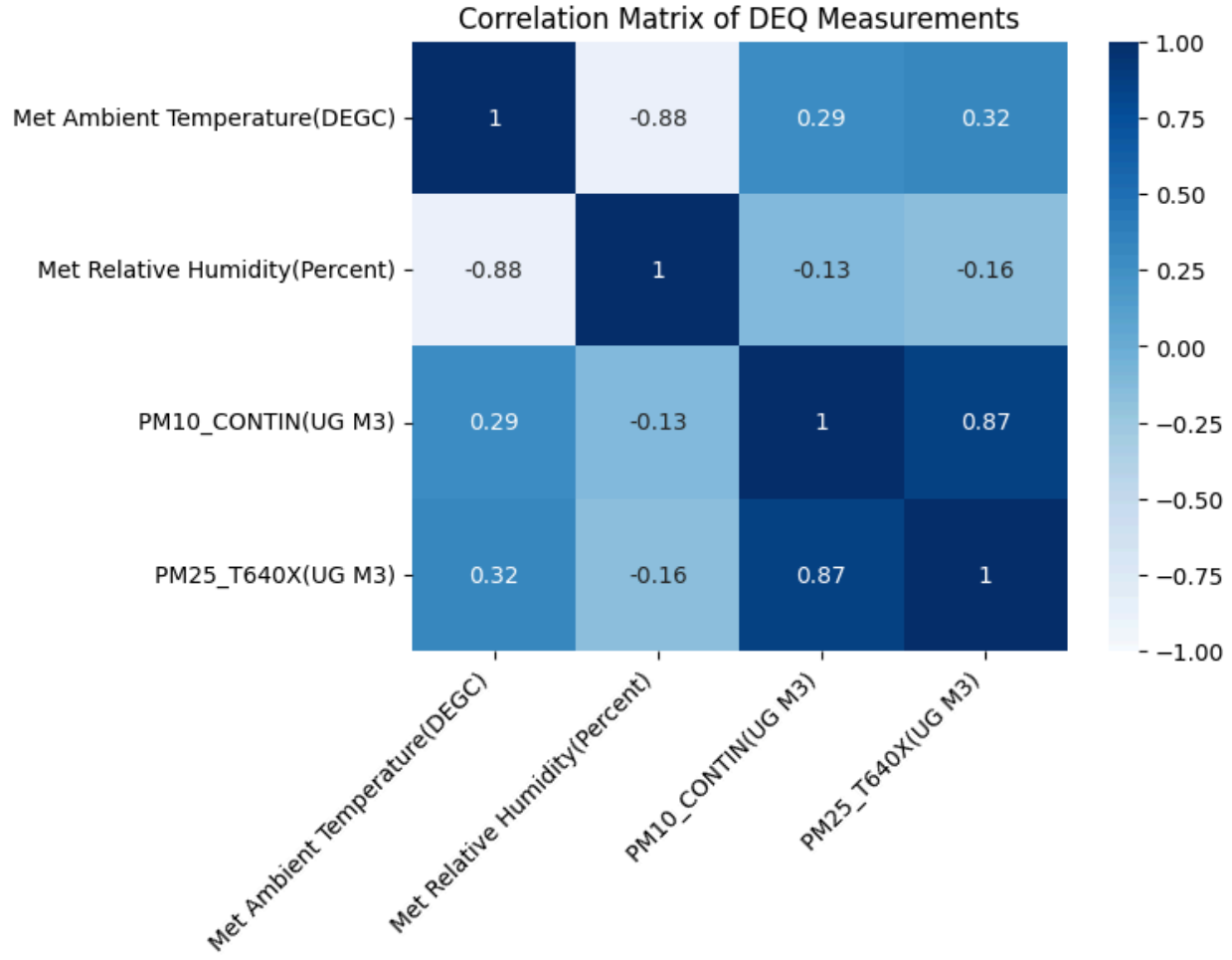

Figure 8. Correlation matrix of the DEQ measurements. Useful for comparing accuracy between datasets, but outside factors such as pollution could limit the applicability of this visualization.

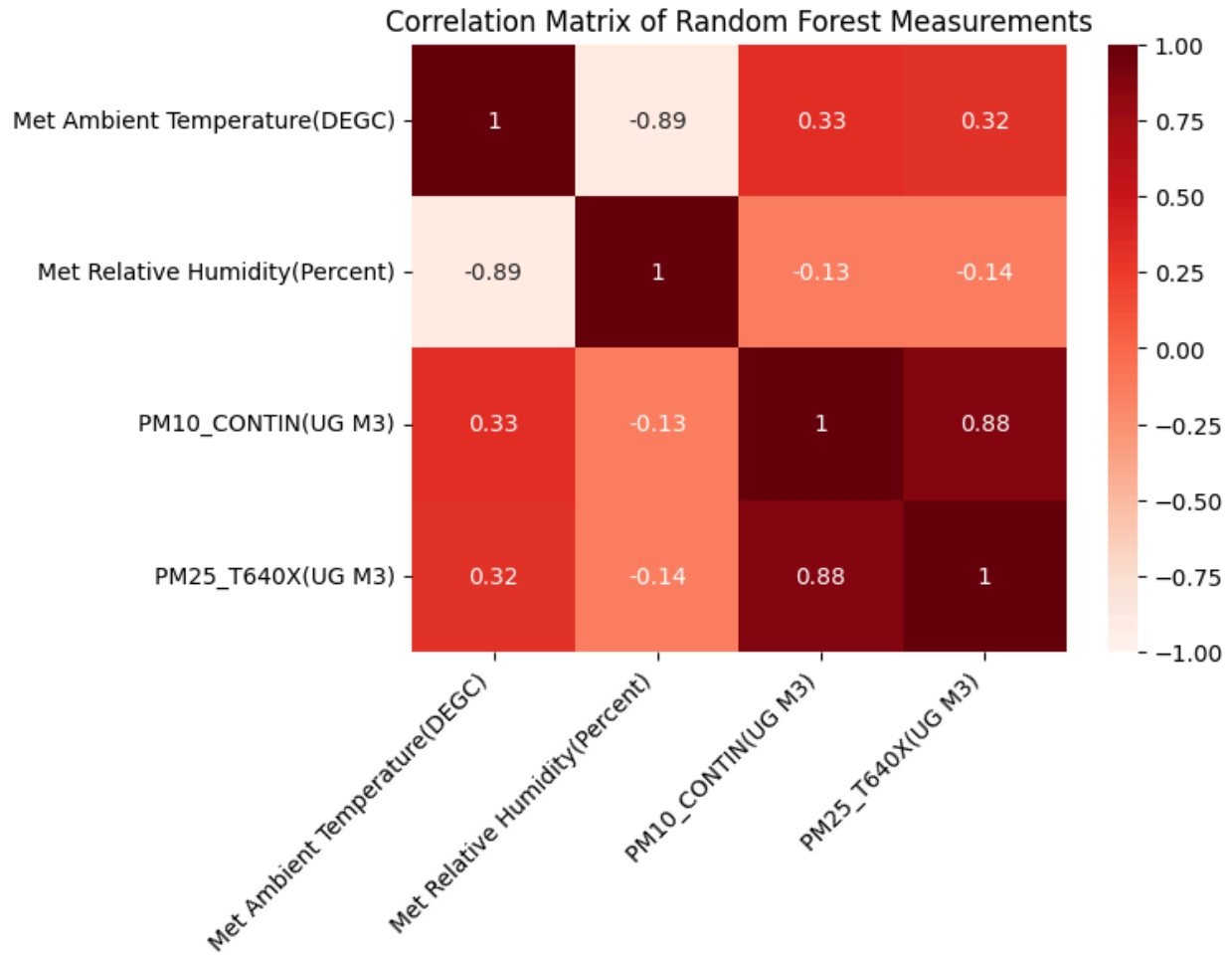

Figure 9. Correlation matrix of Random Forest model.

An important aspect of model validation is verifying the normality of the plot of residuals. If the residuals follow a normal distribution, it suggests that the model's errors follows a normal distribution. This indicates that the model is not only matching the training data set, but also capturing the underlying patterns in the data. This allows for deeper statistical analysis to take place such as confidence intervals and t-tests which could increase the validity of the model. As seen in figure 10, the distribution of humidity, PM10, and PM25, seem to follow a normal distribution while the temperature has a right skew.

The normality of three out of the four feature variables suggests that the model is robust and could be used in similar environments. The skew of the temperature could be due to physical mechanical oversights, since the housing of the Raspberry Pi would be exposed to direct sunlight as well as heat generated by its own electronic components. This suggests that further development in the design of the Raspberry Pi are needed before scientific usage is possible.

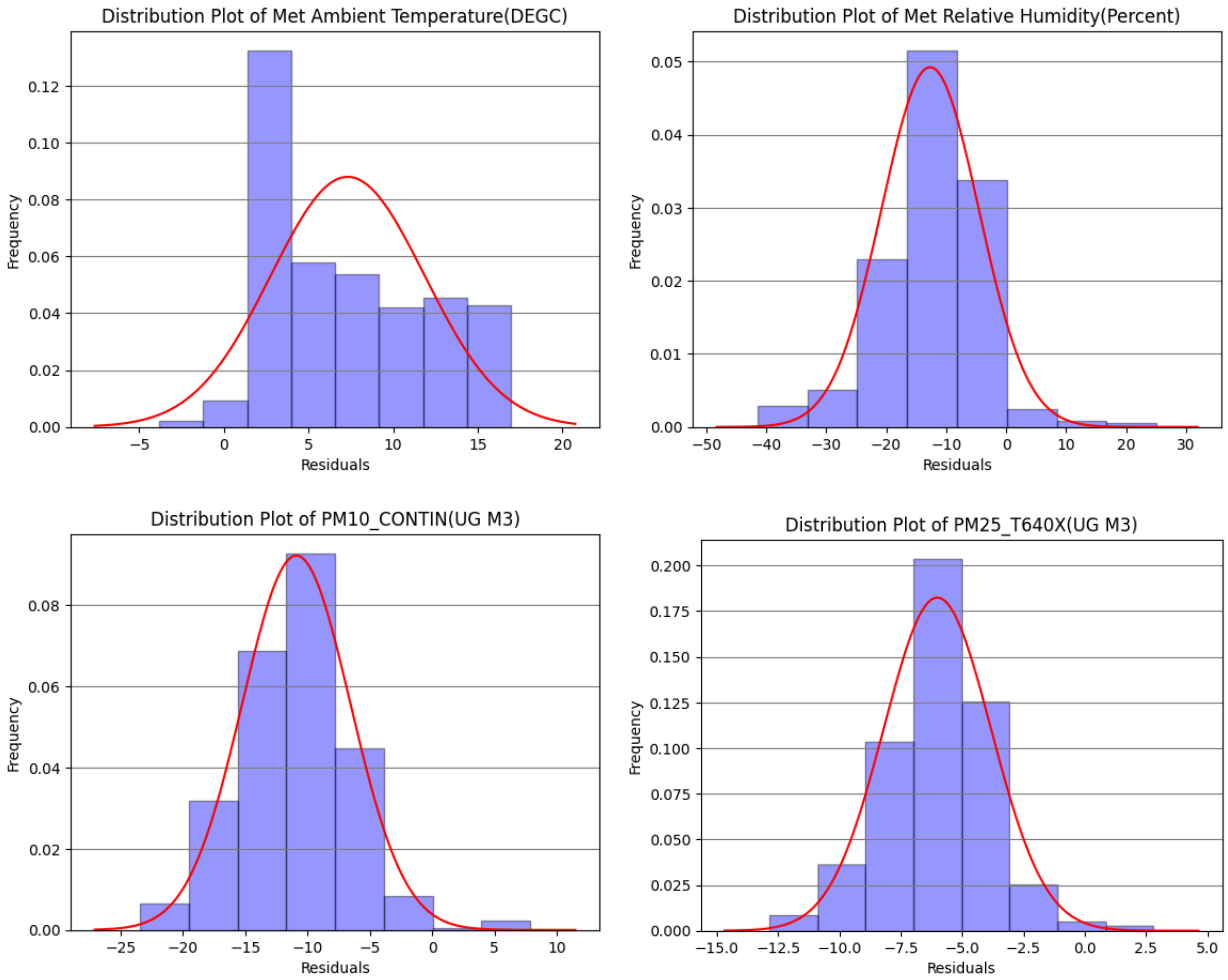

Figure 10. Plot of residuals from the random forest model.

Statistical Analysis

Statistical analysis such as hypothesis testing, variance testing, and other statistical measures can be performed on the data and can offer additional evidence to the efficacy of the models and data. The Shapiro-Wilk test will test the normality of the DEQ data and Pi 1 datasets which is an important factor that will be used in later statistical analyses.

$$H_o: DEQ\ data\ is\ normally\ distributed$$
$$H_a: DEQ\ data\ is\ not\ normally\ distributed$$

Python code was created to test these hypothesis where each feature variable goes through the Shapiro-Wilk test and creates a Q-Q plot depicted in figure 11. The data tested is from the DEQ dataset and all of the p-values are below the alpha value of 0.05, which means that the null hypothesis is rejected in all cases. This same test was also run on raw co-location data and model predicted data with the same results.

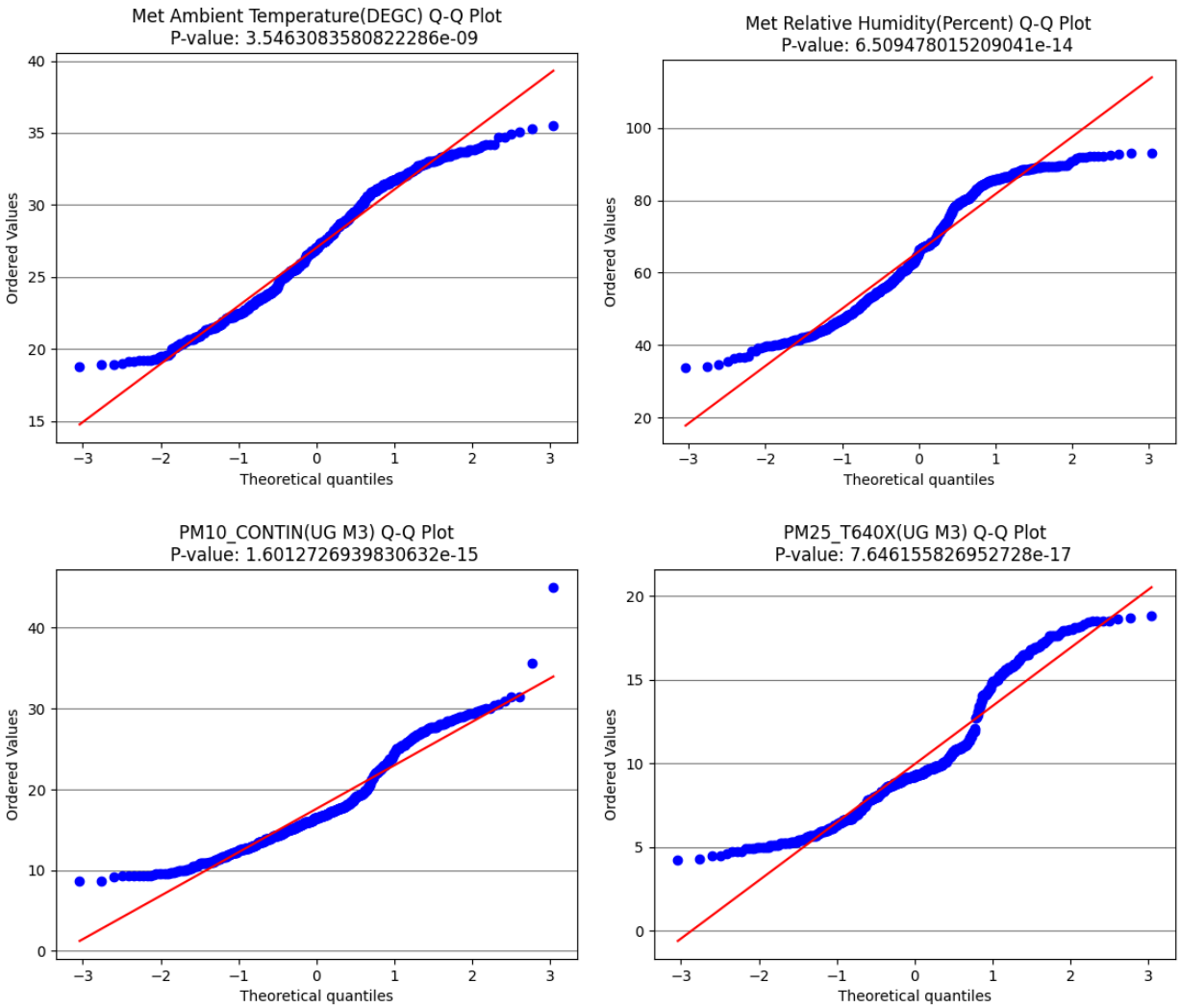

Figure 11. Q-Q plots of each of the feature variables of the DEQ dataset. Other results from raw data or machine learning data not depicted.

Although the normality of data is a necessary condition to conduct hypothesis testing, for the sake of analysis, we will assume normality. All data is sufficiently large however, with n = 587.

The matched pair test compares two groups of data and determines if there is a statistically significant difference between them. We can use this test to determine if the data generated by the machine learning model is the same as the DEQ data. All feature variables were determined to be dependent with correlation values close to 1 and each variable has the same amount of values.

Two-tailed test:

$$H_o: \mu_d = 0$$

$$H_a: \mu_d \neq 0$$

Where the t test statistic formula is

$$t_{ts} = \frac{\bar{d}}{s_d/\sqrt{n}}$$

Inputting the data for machine learning and DEQ data produces these t-test statistics:

$$\text{Temperature } t_{ts} = 5.1829$$
$$\text{Humidity } t_{ts} = 1.082$$
$$\text{PM10 } t_{ts} = 5.0528$$
$$\text{PM25 } t_{ts} = 7.1680$$

Using R to evaluate each of these test statistics using this formula, with degrees of freedom 586:

$$2 \bullet p(t > t_{ts})$$

$$\text{Temperature: } 2 \bullet p(t > 5.1829) = 0.0000003$$
$$\text{Humidity: } 2 \bullet p(t > 1.082) = 0.2797$$
$$\text{PM10: } 2 \bullet p(t > 5.0528) = 0.00000058$$
$$\text{PM25 } 2 \bullet p(t > 7.1680) = 2.3 \bullet 10^{-12}$$

At alpha levels 0.01 and 0.05 the p-values of temperature, PM10, and PM25 reject the null hypothesis, while humidity fails to reject the null hypothesis at these values. This means that besides humidity, the data generated by the machine learning model is statistically different from the correct DEQ data. This reveals that the model is not able to create an accurate 1 to 1 dataset. However, given the low-cost nature of the sensors, some margin of error is expected from the data it collects, so this does not necessarily mean that the model is unusable for some scientific applications.

A slightly different test, the two sample mean hypothesis test, instead measures the average of the sample instead of a pairwise test. This could be a more realistic measure of accuracy because the low-cost Raspberry Pi sensor can not be expected to perform at the same level as the MSIC sensors.

Two-tailed test:

$$H_o: \mu_m - \mu_n = 0$$
$$H_a: \mu_m - \mu_n \neq 0$$

Where the z test statistic formula is

$$Z = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{m} + \frac{s_2^2}{n}}}$$

Inputting the data for machine learning and DEQ data into python produces these z-test statistics:

$$\text{Temperature } z_{ts} = 0.4032$$
$$\text{Humidity } z_{ts} = 0.0460$$
$$\text{PM10 } z_{ts} = 0.7326$$
$$\text{PM25 } z_{ts} = 0.4470$$

Using R to evaluate each of these test statistics using this formula:

$$2 \bullet p(t > z_{ts})$$

$$\text{Temperature: } 2 \bullet p(z > 0.4032) = 0.6868$$
$$\text{Humidity: } 2 \bullet p(z > 0.4060) = 0.9633$$
$$\text{PM10: } 2 \bullet p(z > 0.7326) = 0.4639$$
$$\text{PM25 } 2 \bullet p(z > 0.4470) = 0.6550$$

The p-values of all the variables are greater than alphas of 0.05 and 0.01, this means that we fail to reject the null hypothesis. In this context, this means that there is no significant difference between the means of model and DEQ data. This can be interpreted as the Raspberry Pi being suited to draw long-term patterns and conclusions, rather than a precise minute to minute measurements.

To further verify that the Raspberry Pi is able to draw broad long term conclusions, a two sample variance test would test if the variance is statistically significant. This would mean that the variances as well as the means are statistically equivalent.

Two-tailed test:

$$H_o: \sigma_1^2 - \sigma_2^2 = 0$$
$$H_a: \sigma_1^2 - \sigma_2^2 \neq 0$$

Where the formula for the test statistic formula is:

$$F_{ts} = \frac{s_1^2}{s_2^2}$$

Inputting the data for machine learning and DEQ data into python produces these test statistics:

$$\text{Temperature } F_{ts} = 0.1626$$
$$\text{Humidity } F_{ts} = 0.0021$$
$$\text{PM10 } F_{ts} = 0.5368$$
$$\text{PM25 } F_{ts} = 0.1997$$

Using R to evaluate each of these test statistics using this formula:

$$2 \bullet p(t > F_{ts})$$

$$\text{Temperature: } 2 \bullet p(F > 0.1626) = 0.6869$$
$$\text{Humidity: } 2 \bullet p(F > 0.0021) = 0.9633$$
$$\text{PM10: } 2 \bullet p(F > 0.5368) = 0.4639$$
$$\text{PM25 } 2 \bullet p(F > 0.1997) = 0.6550$$

The p-values of all of these tests are greater than alphas 0.05 and 0.01, rejecting the null hypothesis for each metric. This means that not only is the mean statistically the same, but so is the variance.

Conclusion

The low-cost nature of the Raspberry Pi raises multiple issues, encompassing not only the reliability and use life of the sensor, but ability to handle more extreme environments. Data from co-location and by extension model calibration is limited in its scope. Although not tested, a short seven day collection period can not be used to create a comprehensive model. However, this project serves as a proof of concept of the efficacy of the Raspberry Pi, the statistical analysis shows that the Raspberry Pi can be used in some applications. Specifically, research drawing on long-term changes can consider low-cost sensors as a viable option to conduct legitimate scientific research at a hugely reduced price. More data collection and adjustments to model calibration processes are needed before an effective model that can be used across multiple weather conditions, environments, and times of year can be created.

References

Kidwell, Nathan, et al. *Raspberry Sky: Vertical Profiling the Chemical Composition Impacts on Aerosol Formation with a Drone-Enabled Raspberry Pi Multisensor*. Accessed 13 Dec. 2023.