Improving Air Quality Modeling in Salt Lake City through Visualization and Machine Learning

Dylan R. Wootton, *Member, IEEE*

***Abstract*— Poor air quality impacts public health due to increased incidence of cancers, heart diseases, and various respiratory disorders. Many Utah cities have hazardous air quality episodes as a result of inversions and forest fires; however, despite its impact on our community, the data obtained about PM 2.5 levels is often course- often being measured for an entire zip code. Recent research suggests that microclimates of pollution exist that are not captured by these measurements. As such, a model was built to provide finer resolution PM 2.5 estimates throughout Salt Lake Valley; however, many analysts noticed that this model overestimates point sources of pollution. A tool to visualize these spikes in pollution was built and a Random Forests Classifier was trained to detect and remove these points from the model in an effort to improve model performance and elucidate how the model behaves in response to these spikes.**

***Index Terms*—Data Visualization, Machine Learning, Random Forests, Air Quality, Distributed Sensors, Pollution.**

I. Introduction

CURRENT estimates from the World Health Organization (WHO) indicate approximately 9 out of 10 people are regularly exposed to high levels of air pollution [1]. In particular, fine particulate pollution (PM 2.5) is hazardous to human health. This pollution, with a particulate size less than 2.5 microns, is small enough to pass through the human respiratory tract and accumulate in the lungs [2]. From here, fine particles enter the body and can result in a variety of negative health effects: strokes, heart disease, cancer, and respiratory diseases. Poor air quality is estimated to lead to the premature deaths of 2,000 Utah residents each year [3].

Given the concerns related to public health, the University of Utah College of Engineering launched the AirU Initiative. The goal of the Air U project is to deploy low-cost air quality sensors and create a cyber-physical approach to gather data from these sensors and produce reliable, neighborhood-level estimates of air quality [4]. The purpose of this approach is to provide a systematic method to integrate sensors to generate air quality insights; current research suggests that these sensors are unreliable when not integrated into a system that corrects for known problems of accuracy and precision [5].

Manuscript received April 25th, 2019. This work was supported in part by the National Science Foundation under Grant IIS-1350896.

D. R. Wootton is with the department of Biomedical Engineering at the University of Utah. UT 84112. E-mail: woootton.dylan@gmail.com

While the network of sensors is expected to provide point readings of air quality data for zip codes, recent research indicates that microclimates of PM 2.5 variability exist on a much smaller scale, often due to geographic features that surround the area [6]. This evidence indicates that significant PM 2.5 differences can exist in distances as small as 50 meters.

The current sensor network (n = 241) is not capable of creating a complete and detailed map of PM 2.5 pollution across the Salt Lake Valley. To augment the readings from the deployed sensors, an Air Quality Mapping System (AQMS) was created to provide air quality estimates across the Salt Lake Valley [9]. The AQMS produces these estimates by using a model that incorporates data about land use, topology, weather, and sensor readings.

While the AQMS model can provide fine resolution estimates, point sources of pollution in close proximity to sensors can skew the models estimates for disproportionately large areas. The behavior of the model was noticed in multiple situations as reported from sensor maintainers, and an example of this irregularity is displayed in Figure 1. These

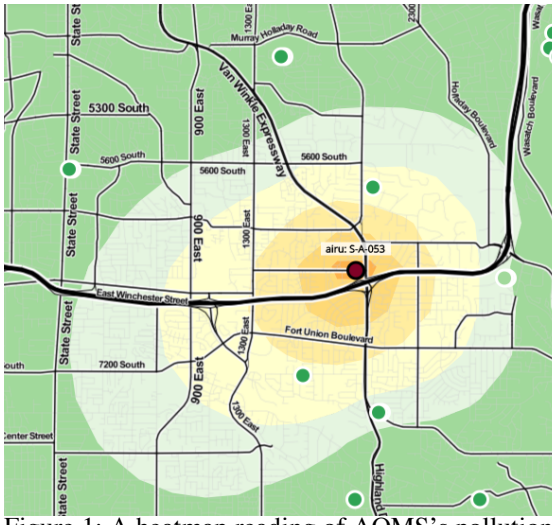

Figure 1: A heatmap reading of AQMS's pollution estimates in a subsection of the Salt Lake valley. The above spike in air quality demonstrates how a small event such as a barbeque could skew the model output drastically.

events are sudden (less than 5 minutes) increases in PM 2.5 readings that can be caused by a variety of activities: a surge of traffic, the use of fireworks, or a barbeque. Such events are part of the uncertainty around low cost sensors, and pose a limitation to their usefulness in policymaking and personal

health decisions [24].

The recording effect of these events can be dramatic- often affecting geographic areas on the order of miles. Previous methods for assessing model performance relied primarily chance conversations with sensor maintainers. While this approach was sufficient for a small set of data, an automated approach was necessary to ensure that model performance could be accurately assed and these spikes could be detected.

As there didn't exist a quantitative definition for what characterizes a spike, algorithmic techniques for the detection of these spikes wasn't initially possible. Tools for data visualization have historically been used as a means of exploratory data analysis [10], and as there was little information about these spikes, a visualization tool that demonstrate the sensor reading and the corresponding model output would play a key role in determining more information about these spikes.

No functionality existed for visualizing the differences between the input (sensor readings) and output (AQMS estimates). Thus, the beginning aim of this study was to design an interface for visualizing the AQMS estimates during these sudden spikes in pollution data. To this aim, the Air Quality Explorer (AQ Explorer) was created to visualize how AQMS models these irregular spike events. This tool identifies spikes of interest, pulls AQMS and sensor data for those spikes, and displays the data to a user.

Through the development of AQ Explorer, analysts were able to determine commonalities behind these spikes. These commonalities were used to automate spike detection on a compilation of sensor data through various methods: k-means clustering [11], Support Vector Machine (SVM) Classification [12], Decision Trees [13], and Random Forests [21] and various signal processing techniques. Additionally, AQ Explorer served as a guide to build a separate AQ Labeler tool that allowed analysts to quickly label detected air quality events as spikes, which is a central methods contribution of our work. After hyperparameter tuning [14], the Random Forest model outperformed other supervised and unsupervised machine learning models, which is consistent with other studies involving the calibration of sensors [21].

Random Forest Classifiers (RFs) are a type of supervised machine learning model that utilize multiple decision tree estimators to classify a sample. These decision tree estimators classify a sample through the use of branched tests, similar to a flow chart diagram. At each tree node, the decision tree tests an attribute (also described as a feature) of the sample, and the route of the path down the tree will be determined by whether or not the sample passes the tests. Once the sample has reached the last node on its branch, it is classified [22].

The creation of AQ Explorer allowed engineers to analyze AQMS behavior and enabled the modeling and analytical methods that were applied to classify these spikes. The identification of these events allowed for an assessment of AQMS performance and a classifier capable of detecting point source pollution events. These tools enabled the generation of more accurate estimations of air pollution across the Salt Lake Valley- through which, we can develop more informed public health strategies, policies, and personal health decisions.

II. Methods

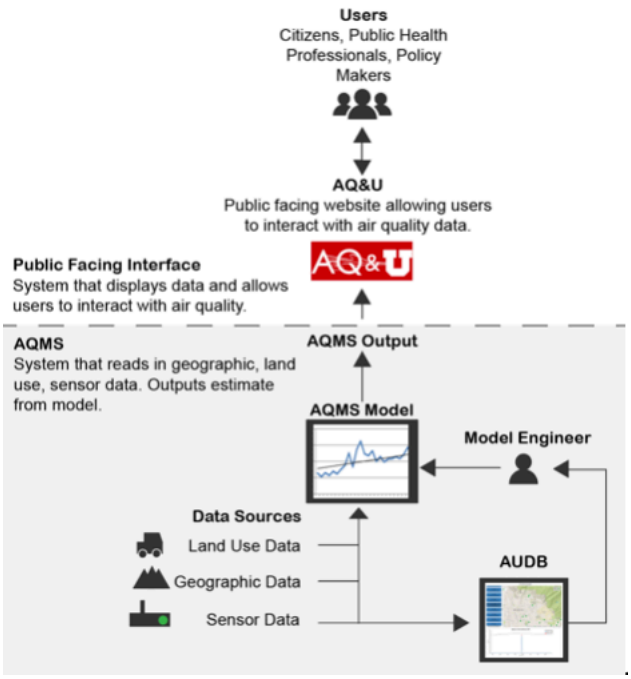

Figure 2: Relational Map. The example relational map between the different systems and AQ Explorer. AQ Explorer works by connecting the model engineer to the sensor data to understand trends in the data.

A. AQ Explorer Development

User Research

Interviews were conducted with four air quality researchers who work on various parts of the Air U Project. The interviews were utilized to determine the design requirements for AQ Explorer (Figure 4) and develop a better understanding of how the tool fits into the researcher's workflows. The interviews were recorded and transcribed to allow for further thematic analysis. The requirements displayed above were used to develop a relational map of the systems, as displayed in Figure 2.

Data Procurement and Cleaning

SQL (Structured Query Language) was used to query the database containing the AQMS data; however, to ensure access to the data without the need of credentials, a Application Programming Interface (API) route was created to access this data. The air quality sensor data was obtained by querying the API. After the data was obtained, it was cleaned and transformed into a JSON object so it could be manipulated by AQ Explorer. Unit tests were completed to ensure that the queried data met the requirements of the query that obtained the data.

Signal Detection

After the data was prepared, the z-scored thresholding algorithm [10] was used to detect uncharacteristic spikes in the sensor data. This algorithm works by using a moving mean and comparing the value of a signal at a given point to that standard deviations away from that mean. If the new point was observed at a set amount of standard deviations away from the moving mean, a spike was recorded. Previous research in our group suggested that this algorithm was able to capture the necessary spikes but not noise in the data. In practice, this algorithm was insufficient to correctly identify all of the spike events, and as such, a thresholding algorithm was used to limit the identified events to larger changes in air pollution.

Dashboard Creation

Using the identified air quality events, a dashboard (Figure 4) was created to display a list of the air quality events (the spikes of PM 2.5) with the other tool components. When a spike is selected, a map zooms to the location of that sensor and displays the reading on the time chart. Additionally, the sensor and AQMS data displays on a time series chart. The selection list and time series chart were created using JavaScript, D3.js, Bootstrap, and jQuery- a common technology stack for many visualizations [15]. The geographic map was created using JavaScript's Google Maps Library. The code for this tool can be found on the following GitHub repository: github.com/dwootton/AQ-Explorer.

B. Data Labeling and Handling

Data was obtained through the AirU API using Python's request library [16]. This data was then stored inside of Pandas [17] data frames and utilized for further analysis. Statistical tests were completed using the Scipy [18] and Stats Model [19] libraries.

After extensive use of the AQ Explorer tool, analysts were able to determine that these spikes in the data often occurred when sensor readings became higher than 350 parts per million (ppm); however, analysts had noted that not all sensor readings over 350 ppm caused the model to become skewed. As such, a simple thresholding algorithm was not sufficient to accurately detect and remove these spikes from our data. As such, spike detection was treated as a classification problem, and a dataset of 1400 detected spikes was manually labeled using the AQ Data Labeler Tool. This tool leveraged the map and slider component of the AQ Explorer Tool (seen in Figure 3).

After the data was classified as causing an overestimation of the model or not, these labels were associated with their corresponding sensor reading. This dataset was then split into training and test sets and then appropriately scaled using the SK Learn package [20].

Both supervised and unsupervised machine learning models were used as classifiers in this study. 6 different models were trained and utilized to classify whether or not a sensor reading would cause a model to significantly over

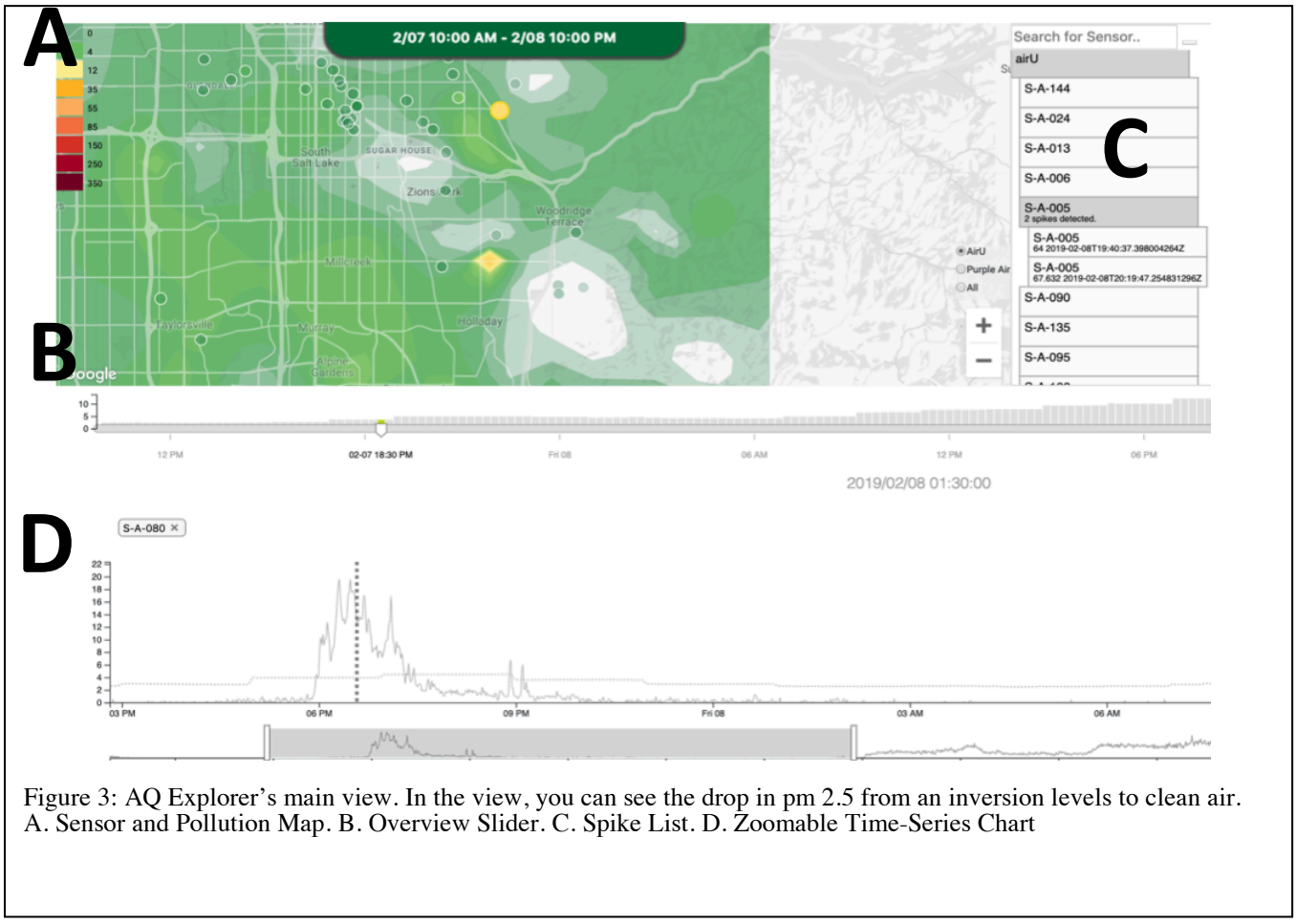

Figure 3: AQ Explorer's main view. In the view, you can see the drop in pm 2.5 from an inversion levels to clean air. A. Sensor and Pollution Map. B. Overview Slider. C. Spike List. D. Zoomable Time-Series Chart

estimate. The metric utilized to analyze model performance was precision, referring to the model's ability to minimize for false negatives. This metric was chosen as it would be preferable to discard sensor readings rather than allow for a possible spike to get through undetected. Additionally, the log loss of each model was determined and utilized as a measure of accuracy when selecting the model.

After an initial analysis of a variety of models, a Random Forest Model (RFM) was selected as the best performing model when considering both the log loss performance of the model and its precision. Additionally, as Random Forest Models can be easily visualized as a decision tree, the results could inform further analysis of the important parameters affecting whether or not a spike causes the model to overestimate. This RFM was then tuned to the training data through the use of cross validation grid search (CVGridSearch) [20], and the tuned RFM was tested against the test set.

III. Results

Tool Development:

The primary contribution of this thesis project is the design of AQ Explorer as this visualization tool guided the construction of the following classification models. The design of AQ Explorer was guided through discussions with multiple engineers who are involved on the Air U project. The dashboard is split into multiple sections: an interactive map, an overview slider, a time series chart, and a list of detected spikes. The function of the tool is characterized by Figure 4, and the corresponding tool components are described below

Geographic Map: The map provides the geographic context for the user and is displayed in figure 3A. The map allows the user to look at the sensor's location in relation to other sensors while seeing pm 2.5 values for each of the sensors at a given time point. Additionally, a user can see the contour map created by the AQMS model estimates. Additionally, the map allows the user to investigate the surrounding geography of the sensor- allowing the user to incorporate domain knowledge about how the geographic surroundings may affect the output. Finally, the map plays a role as the main interactive component that allows the users to select and unselect sensors.

Overview Slider: The overview slider lets a user choose a specific time they are interested in investigating. When a set of dates are selected, the average pollution for every 15-minute interval over that date range is visualized on the top of the slider. The user can change this from an average to a max estimate through a custom command. This visualization indicates to the user where points of interest in the data might be and allows for the user to interact with the slider to change the data that is displayed on the Geographic map.

Time Series Chart: A time series line chart was used to display both the AQMS estimate and the actual sensor data. This chart, displayed in figure 3B, overlays the model (gray) and the sensor reading (colored). This chart allows for the user to compare both the model and sensor data, and the chart supports the ability to zoom in, pan, and navigate through the data. The time series chart also provides the user the ability to change the contour map and sensor data that is displayed on the Geographic map. Through clicking on the time chart, the time indicator will move to the clicked time and the map view will re-render to show the data from that time.

Spike Selector: The spike selector, displayed in figure 3A, lets a user choose a detected air quality event to display the data for. When one of the events is selected, the sensor readings and model data load on to the backend, the data are displayed in the time series plot and the map zooms to the location of the sensor that detected the event.

Testing

Testing was completed to ensure the accuracy of the data acquisition and visualization steps in the tool. These tests revealed that the data obtained from the database queries were accurate and delivered data corresponds to the time and location attributes of the database query. Integration tests

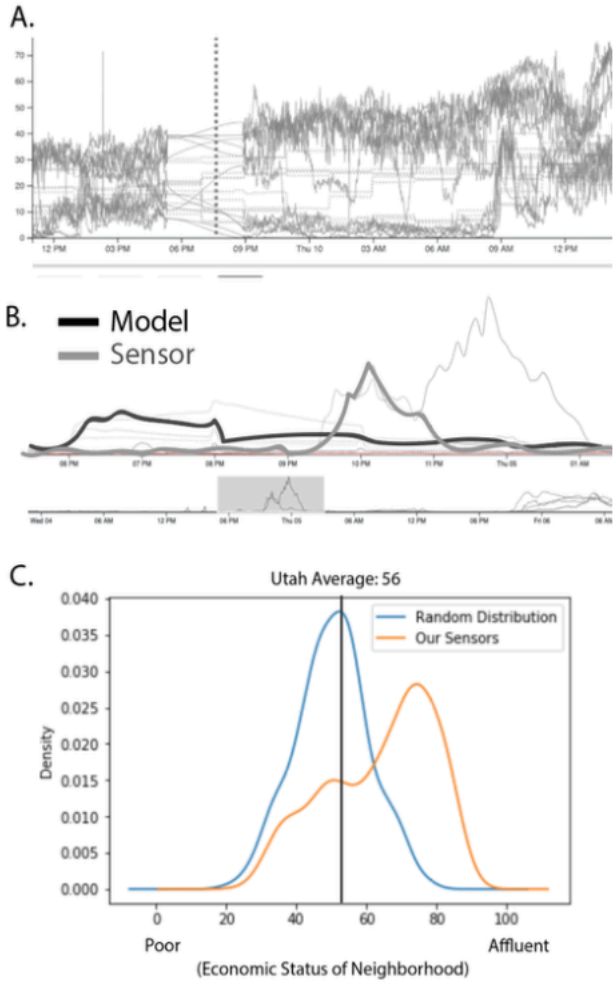

Figure 5: Additional Findings. A. All sensors failed to record pollution for a period of 3 hours. B. The Model incorrectly 'precomputes' large events such as fireworks – offsetting them by 7 hours. C. The Sensors are located in neighborhoods, significantly overrepresenting those who are wealthy

were completed through the use of Selenium IDE. Macros were created to mimic human interaction with the visualization and the corresponding event dispatches were used to determine if the proper behavior was expected.

Usage

AQ Explorer was useful in developing an understanding of model behavior and indicating general problems with the sensor network. From observing pollution spikes using the tool, a general understanding that these spikes are caused by pollution values over 350 ppm was determined. Furthermore, the use of this tool indicated that not all values over 350 ppm result in a corresponding change to the model. As a result, the RF Classifier was built to classify these high value events as either affecting the model or not.

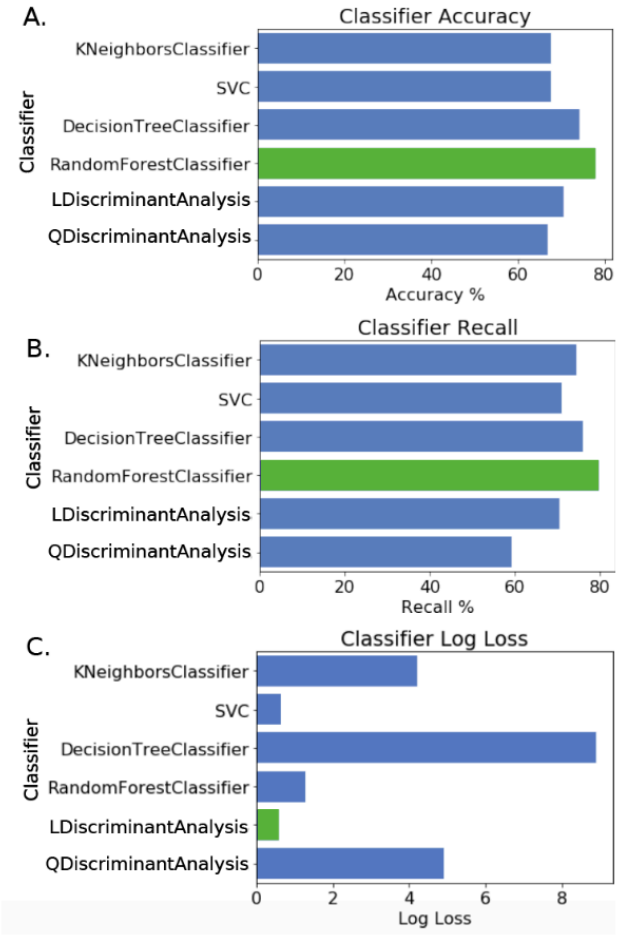

Figure 6: Metric scoring plots for the 6 model survey. A. Accuracy Scores. B. Recall Scores. C. Log Loss Scores.

Beyond the use of this tool to investigate spikes in air pollution, the tool also was useful in investigating other behaviors including sensor network logging failures, model precomputation, diurnal canyon behavior, and recognizing the skewed socioeconomic distribution of sensors (Figure 5).

Modeling:

6 different types of classifiers were trained on the training data and were scored against three metrics: accuracy (% of correct guesses), recall (% of true positives over the number of true positives and false negatives), and their log loss (a measurement of the uncertainty of predications). The performance of these models in this survey is depicted in Figure 6. As the Random Forest Classifier (RFC) performed well across all three metrics, the RFC was chosen as the model of choice for detecting these false events.

The hyperparameter tunings for this random forest determined that the optimal hyperparameters for this classification problem involve a class weights ratio of 1:25, a maximum depth of 100 nodes, 3 max features per tree, 5 minimum samples per leaf, 100 decision tree estimators in the forest, and a minimum split sample requirement of 8.

The trained model exhibited a correct classification rate of 97.5% on identifying sensor readings that caused the model to overestimate and a 43% rate of identifying sensor readings that were over 350 ppm but didn't trigger the model (Figure 7 A). RFC trained based upon accuracy had classification scores in the upper 80's; however, these classifiers had unacceptably high levels of false negatives, which we were optimizing for.

The features importance's were then extracted from the model and used to show which features of the data were important in classifying if a sensor reading would produce a corresponding change to the model (Figure 7 B)

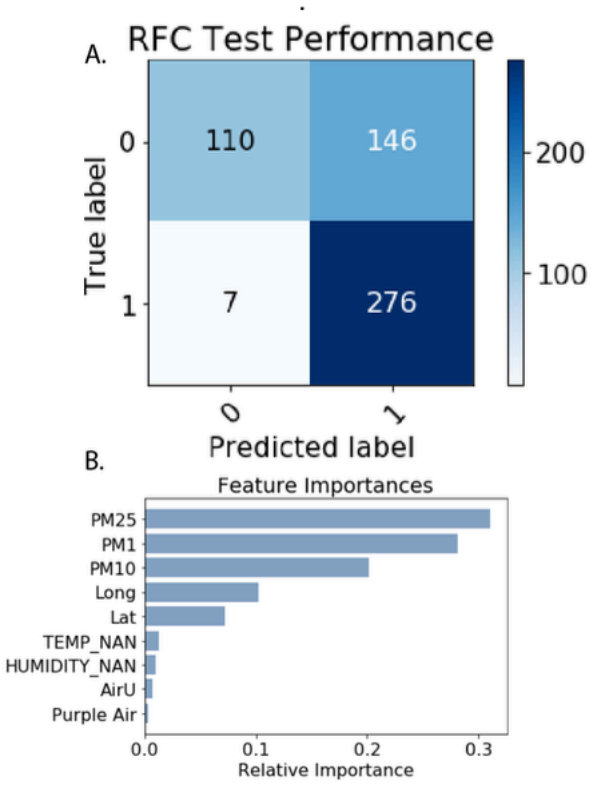

Figure 7: A. Model performance on test data with 43% accuracy identifying false negatives and 97.5% accuracy identifying spikes that activated the model. B. Importance of features on classification.

IV. Discussion

Air pollution plays a hidden effect on the health of many Utah citizens, and while such effects may be negligible over short periods of time, the longer-term impacts of high pollution exposure shorten lives, cause developmental delays [7], and cause a variety of health effects [8]. Despite the public health consequences that air pollution has on Utah residents, we have a poor understanding of the spatial temporal patterns of air pollution. To improve our understanding of how air quality varies in the Salt Lake Valley, the Air U project draws data from over 400 sensors to produce air quality estimates across the valley. While this project has been successful in integrating sensors with a centralized data base, the model can produce inaccurate estimates- sometimes generalizing one sensor's values across much larger regions as displayed in figure 2. This is known problem with many low-cost air quality sensor projects and is a major limitation to wide scale adoption of low-cost sensors [25].

Model engineers had a difficult time investigating the cause of such spikes as the data was difficult to access; requiring dense SQL queries and complicated visualization code to simply look at small chunks of data. In order to speed up the process of analyzing and understanding the model estimations, a new tool AQ Explorer was created to visualize both model estimates and sensor pm 2.5 pollution readings. This tool gave users a simple interface to select time ranges of interest and visualize air pollution sensor readings and model estimates.

The development of the AQ Explorer tool was a multistage process. The first stage was to conduct user research and understand what the requirements of this tool were. The transcripts from user interviews were recorded and the information from the interviews was used to create a list of tool requirements.

We conducted interviews and crafted software requirements, and then began prototyping. Initially, AQ Explorer was built as a tool to focus solely on signal and event detection; however, through iterative testing with the tool users, there was a stronger desire to have flexibility when selecting and viewing air pollution data. This was added to the design requirements and the tool was rebuilt to accommodate flexibility- allowing users to select any date range and investigate any sensor on the map.

The third stage focused on the development of tests to verify the error-free nature of the data visualization tool. This involved the creation of unit tests to ensure that the correct data was fetched from the database and integration tests to ensure that the data was properly passed between objects. These tests demonstrated that AQ Explorer was able to accurately obtain, transform, and visualize the air quality data.

The tool enabled the discovering of multiple problems with the sensor network and the corresponding modeling system (AQMS). This included a logging error where sensors failed to record data for a three-hour period and a demonstration of how the model didn't accurately predict large events such as forest fires or fireworks. The model would incorrectly predict these events as being offset from the actual event occurrence, which represented a failure to correctly align the time values of the model with the sensor readings. These insights played an important role in improving the accuracy of the system.

Additionally, through looking at the sensors on a geographic map, users noticed the distribution of sensors across the Salt Lake Valley. Exploration through AQ Explorer helped analysts determine that there may be a skew in sensor distribution. This following study revealed a significant skew in where the sensors were being stored with 70% of all sensors being stored in neighborhoods classified as being affluent or very affluent.

The central focus of the tool was to investigate these spikes and produce a general heuristic for these events. Through tool use, a baseline of 350 ppm was established and utilized for further analysis; however, it was noted that a simple threshold of 350 ppm would cause the misclassification of numerous sensor readings that exhibited a higher level of pollution yet didn't alter the model.

The AQ Data Labeler was used to manually classify 1400 pollution spikes that were over 350 ppm, and this tool came directly from the components of the AQ Explorer tool. This visual analysis allowed the labeler to label 1400 points in under 20 minutes. This data labeler represents the second contribution of this thesis, and is a model independent way of creating a classified dataset of these spikes.

A Random Forest Classifier was trained, tuned, and tested using this data, and in the end, was able to identify the vast majority of the spikes that affected the model. The classifier was capable of a 97.5% reduction of these spike events that lead to an overestimation inside of the model while maintaining 43% of the other sensor events that would have been discard through a simple thresholding algorithm. This classifier can be utilized inside of the data processing workflow to ensure that any sensor readings that are used in the model wouldn't cause an overestimation of any individual spike in the data.

From analyzing feature importances, pollution estimates of PM25, PM10, and PM1 were the most impactful for the classifier. Following this are latitude and longitude, which likely allowed the classifier to detect sensors that frequently exhibited these spikes in pollution and classify them as likely causing a problem with the model. Through analysis of individual trees in the classifier, it was revealed that highest recordings of pollution (47 out of the top 50 detected readings) actually were associated with no change to the model. Such a result points to the AQMS model having learned that such pollution values were likely a mistake in readings or in data handling.

While the classifier was able to detect problematic points and classify them as incorrectly activating the model or not, there were some limitations of this study. First while the heuristic of 350 ppm was developed through use of the AQ Explorer tool, it's possible that not all overestimations in the model were caused by sensor readings over 350 ppm. As such, these overestimations would not be detect by the classifier, and thus could still skew the results of the model. Additionally, given the limited amount of data collected (1400 spikes detected over the 2017 and 2018 winters), it is possible that our RFC is overfit to our training data. This could make our model not generalizable to other spikes that

occur during the summer or to spikes that occur in other years.

Through more extensive data labeling, we hope to improve the performance of our classifier and reduce uncertainty about its generalizability. Our future work will involve utilizing all of the sensor data that has been collected over the past two years (not just the winter data). Additionally, other testing and training data sets will be created with thresholds of 200 and 250 ppm. These will be used to determine if model overestimations are possible below our currently set threshold of 350 ppm.

Future work on this AQ Explorer tool involves conducting more user interviews as well as support additional features on AQ Explorer. Furthermore, through the addition of extra features such as integration with Google maps routing service and address lookups, AQ Explorer can serve as a public facing tool- allowing for users to develop a better understanding of their personal exposure to pollution and make informed decisions about their respiratory health.

REFERENCES

[1] World Health Organization. May 2nd 2018. "News Release: 9 out of 10 people worldwide breath polluted air". http://www.who.int/airpollution/en/

[2] Yu-Fei Xing, Yue-Hua Xu, Min-Hua Shi, and Yi-Xin Lian. The impact of PM2.5 on the human respiratory system. Journal of Thoracic Disease. January 2016. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4740125/

[3] Brian Moench. Utah docs say inversion contributes to premature deaths. Jan 2013. http://archive.sltrib.com/article.php?id=55686497&itype=cmsid

[4] Miriah Meyer, Ross Whitaker, Kerry Kelly, Pierre-Emmanuel Gaillardon, CPS: Synergy: A Layered Framework of Sensors, Models, Land-Use Information and Citizens for Understanding Air Quality in Urban Environments. April 2018.

[5] K. Kelly, J. Whitaker, C. Widmer, A. Dybwad, and A. Butterfield. Ambient and laboratory evaluation of a low-cost particulate matter sensor (submitted). Environmental Pollution.

[6] S. Steinle, S. Reis, and C. E. Sabel. Quantifying human exposure to air pollution–moving from static monitoring to spatio-temporally resolved personal exposure assessment. The Science of the total envi- ronment, 443:184–93, jan 2013.

[7] Jama Global Health. Air Pollutants Undermine Infant Brain Development. February 20, 2018.

[8] Marilena Kampa, Elias Castanas. Human health effects of air pollution. June 2007. Environmental Pollution. 362 -367.

[9] K. Le, K. Tingey, T. Becnel, P. Giallardon, T. Butterfield/K. Kelly Building Air Quality Sensors & Citizen Scientists, Chemical Engineering Education. Proc. ASEE Annual Meeting, June 23 – 27, Salt Lake City, UT.

[10] Ware, C.. Information Visualization: Perception for Design. Morgan Kaufmann. June 2001. ; 3rd Edition.

[11] G. R. Kingsy, R. Manimegalai, D. M. S. Geetha, S. Rajathi, K. Usha and B. N. Raabiathul, "Air pollution analysis using enhanced K-Means clustering algorithm for real time sensor data," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 1945-1949.

[12] Cortes, Corinna, and Vladimir Vapnik. "Support-vector networks." Machine learning 20.3 (1995): 273-297

[13] Quinlan, J. R. (1987). "Simplifying decision trees". International Journal of Man-Machine Studies. 27 (3): 221–234. CiteSeerX 10.1.1.18.4267. doi:10.1016/S0020-7373(87)80053-6.

[14] Bergstra, J. and Bengio, Y., Random search for hyper-parameter optimization, The Journal of Machine Learning Research (2012)

[15] Bostock, M., Ogievetsky, V., and Heer, J. D3: Datadriven documents. In IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis), 2011

[16] Reitz K. Requests: HTTP for Humans. Release v2.21.0. 2018

[17] Wes McKinney. Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, 51-56 (2010)

[18] Jones E, Oliphant E, Peterson P, et al. SciPy: Open Source Scientific Tools for Python, 2001-, http://www.scipy.org/ [Online; accessed 2019-01-22].

[19] Seabold, Skipper, and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python." Proceedings of the 9th Python in Science Conference. 2010.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay. Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, 12, 2825-2830 (2011)

[21] Leo Breiman. Random Forests. Machine Learning. Volume 45 Issue 1, October 1 2001. Pp 5 -32.

[22 Zimmerman, N., Presto, A. A., Kumar, S. P. N., Gu, J., Hauryliuk, A., Robinson, E. S., Robinson, A. L., and R. Subramanian: A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring, Atmos. Meas. Tech., 11, 291-313, https://doi.org/10.5194/amt-11-291-2018, 2018.

[23] Vili Podgorelec, Peter Kokol, Bruno Stiglic, and Ivan Rozman. 2002. Decision Trees: An Overview and Their Use in Medicine. J. Med. Syst. 26, 5 (October 2002), 445-463. DOI: https://doi.org/10.1023/A:1016409317640

[24] Lewis A, Edwards P. Validate personal air-pollution sensors. Nature. 2016;535(7610):29–31. doi: 10.1038/535029a

[25] A. C. Rai , P. Kumar , F. Pilla , A. N. Skouloudis , S. Di Sabatino , C. Ratti , A. Yasar and D. Rickerby , End-user perspective of low-cost sensors for outdoor air pollution monitoring, Sci. Total Environ., 2017, 607–608 , 691 - 705