[Final Report] Enhancing Short Term AQI Predictions with Location & Meteorological Data

Kushagra Gupta
Department of Statistics
Stanford University
kushgpt@stanford.edu

Quentin Hsu
Department of Statistics
Stanford University
qhsu@stanford.edu

Abstract

Air quality index (AQI) data is very volatile and fails to exhibit long-term patterns useful in time series prediction. For this project, we sought to build a 3 day ahead prediction for the air quality index of Santa Clara County. We approached the problem with increasingly complex models and evaluated the performance increase with a sliding window cross-validation strategy. We specifically included AQI and meteorology data features from surrounding counties and improved performance when using relevant features. Our final 2 models of VARMA and LSTM perform better than ARIMA and our naive baseline, indicating that the models have learned some signal from the features we provided. We also used a hidden Markov model (HMM) to separate wildfire and non-wildfire data and train a multi-modal VARMA model. However, no one model performed the best in all situations, indicating that AQI prediction is volatile and difficult to predict.

Our code referenced in this report and presentation can be found at this Google Colab and Slides.

1 Introduction

In the past few years, wildfires have burned throughout California. This constant surge of fires coupled with climate change have created long periods of degraded air quality. Exposure to these pollutants have severe health consequences, so developing accurate predictions of air quality is important for individuals to avoid dangerous conditions and for government to make key policy decisions.

Although the problem is important, it is also challenging because of the severe volatility of the air quality index (AQI), which renders traditional time series methods useless for long term prediction. To tackle this problem, researchers have focused on short term predictions augmented by additional factors like geographical location, meteorological information, etc in conjunction with the historical AQI data to improve the explanatory power of AQI prediction models [4].

Meteorology factors have a large relationship with air quality measurements. Temperature and humidity can create air inversions that can trap pollutants under a layer of warm air [1]. More importantly, these factors lead to different types of wind currents that shift pollutants to nearby locations over a certain time period [7] [10]. This can be especially prominent during wildfires when large amounts of pollutants are expunged into the air and can be distributed across large distances [1] [2].

Our project explored the correlation of air quality in different locations around Santa Clara county created by changes in meteorological factors like wind currents. We explored different modeling techniques in increasing complexity to learn more nuanced relationships in an attempt to improve time series forecasts. We also tried to separate wildfire from non-wildfire data using HMMs, given the apparent distinction in the data from these periods. Our goal is to leverage these features to make 3 day predictions into the future of the AQI in Santa Clara county.

2 Data and Features

Our data is the daily AQI data from June 1, 2016 to December 31, 2020 publicly available on the EPA website. We are interested in using the AQI of surrounding counties, temperature, relative humidity / dew point and the wind speed / direction as features for our models.

2.1 Data Wrangling and Pre-processing

The raw data was given in the form of measurements across various sites in a county with some missing dates of measurements. We restricted the data to counties within California and averaged across the various monitoring sites in a county. The remaining missing dates were filled with linear interpolation. This process was followed for all time series (of all counties and for all meteorology factors). We use the Augmented Dickey-Fuller unit root test to test for stationarity of the data split in blocks. If any of the time series in a cross-validation window is non-stationary, we use the log transformation, Box-Cox transformation or first order differencing to achieve stationarity.

2.1.1 Nearby Counties

We believed that the AQI of nearby counties will improve the predictive power of our models since meteorology factors would shift air pollutants amongst nearby locations. We calculate the distance between Santa Clara and all other California Counties using the haversine formula and keep counties within a 100 miles radius.

2.1.2 Meteorology Factors

Wind is a large factor that can create conditional relationships between nearby locations, especially in times of increased AQI [3]. We attempt to leverage this physical property by creating a spatio-aware feature called wind projection based on the wind direction, wind speed, and county location data.

Imagine drawing a vector between a surrounding county (eg. Marin County) and Santa Clara county. This geographic feature has both a direction (bearing) and distance. Wind from Marin county is also another vector that has both a direction and distance (wind speed * time). We project this wind vector onto the geographic vector to obtain a wind projection that represents how far the wind is able to carry particles from Marin to Santa Clara county. Figure 1 depicts an example of how we derived a wind projection feature. We also use the temperature and relative humidity/dew point data in our model.

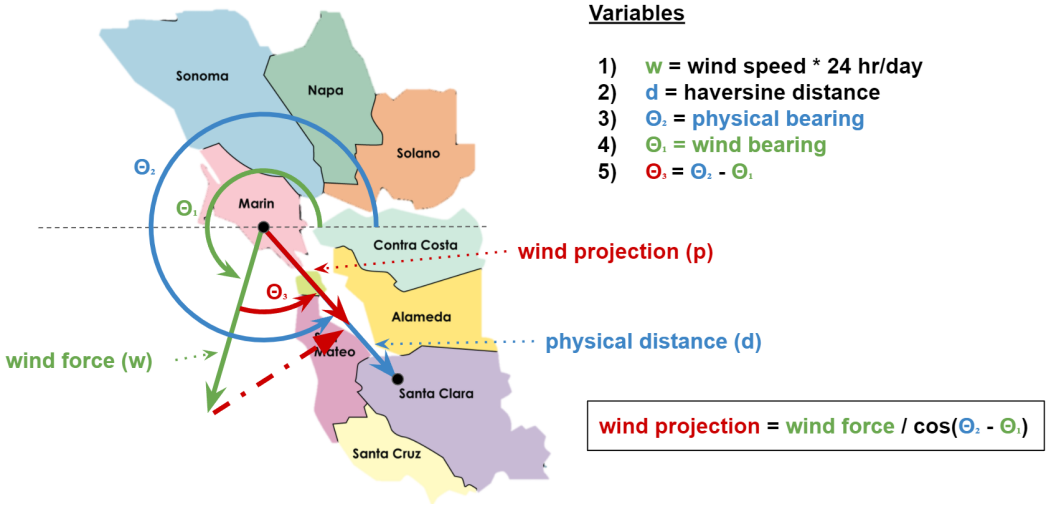


Figure 1: Wind Projection derivation

3 Methods

3.1 Overall Approach

We attempt to systematically test out custom features and increasingly complicated models in order to improve our ability to forecast Santa Clara County's AQI. We use time series specific cross-validation (CV) and calculate the average root mean squared error (RMSE) across all the CV blocks to evaluate the robust performance of our different methods. We also perform network Granger causality tests to identify if the AQIs of surrounding counties and/or Santa Clara's meteorological factors "Granger cause" the AQI of Santa Clara county.

3.2 Performance Evaluation

3.2.1 Rolling Cross Validation

We envision that our modeling methods in a production environment would be routinely retrained in batches. Therefore, we evaluate our models using sliding window cross-validation. Due to the nature of time series data, we require the test data to be strictly after the train data. Thus, we split our data into contiguous blocks, each with 120 days (4 months) of train and 3 days of test data. We allow each block to overlap with other blocks so that we essentially predict the full time period we are interested in evaluating [5]. Figure 12 depicts our cross validation setup.

For each block, we not only fit the parameters of the modeling method we use, but also reevaluate what order our model should be in cases of ARIMA and VARMA. This is important since AQI does not follow the same parameterization for longer time frames; constraining it to a fixed model specification greatly deteriorates the predictive performance.

3.2.2 Evaluation Time Periods

We stratified our samples into 3 evaluation time periods:

1. 2019 Data
2. 2020 Non-Wildfire Data - This includes all months except Aug-Oct 2020
3. 2020 Wildfire Data - This includes months Aug-Oct 2020

We split up 2020 into Non-Wildfire and Wildfire months since there was a large increase in absolute AQI as well as variability during wildfire months (see Figure 13). Splitting up those months helped us evaluate when our models performed better and whether there are future opportunities to improve in specific stratums.

3.3 Modeling Techniques

We tried out 5 types of modeling techniques of increasing complexity.

1. We set a **naive baseline** that uses the last observation in the training set as the 3 day prediction. This baseline has no information that is learned and thus, serves as a good benchmark for whether our modeling techniques are gaining anything out of the training data.
2. We fit a basic **ARIMA** model. This is a univariate technique, limiting us to making predictions using only the past AQI of Santa Clara County.
3. We expanded the model to a **VARMA** model in order to simultaneously fit additional time series such as the past AQI of surrounding counties or meteorology of Santa Clara County.
4. We explore **Long Short-Term Memory (LSTM)** models to see if the increased flexibility of neural networks enables learning of complicated non-linear relationships between our features [8]. LSTM also allows us to leverage longer historical data from 2016-2018 and start off the model with some weights that were influenced by the historical training data.
5. We also used a **hidden Markov model (HMM)** model to identify wildfire and non-wildfire data, and trained a multi-modal VARMA model catering to this classification.

3.4 ARIMA

We trained a univariate ARIMA model on the unprocessed as well as some transformations of the AQI data of Santa Clara. We tried a log transform, box-cox transformation and first order differencing to test if transformed AQI data is more conducive to time series forecasting.

3.5 VARMA

Given that long term trends are not very informative (or present) in AQI ARIMA modeling, we augment the model with AQI data from neighbouring counties and meteorology of Santa Clara.

3.5.1 Auto-ARIMA for VARMA

Since a grid search on the AR and MA order (p,q) for each CV window is prohibitively expensive, we design a custom approach to estimate p and q. We use the `AutoARIMA` function from `pmdarima` to estimate the order of each VARMA component, and then use the values of p and q for VARMA from the component that gives the least RMSE [9]. This limits the scope of possible values for p and q and gives a good approximation of the best case scenario.

3.5.2 Granger Causality and Co-integration Test

We performed network Granger causality tests on the AQI data from nearby counties and meteorology data from Santa Clara to identify which time series can improve prediction of Santa Clara's AQI. We also performed co-integration tests to spot relationship between Santa Clara's AQI and the components we want to add to our VARMA model.

3.6 Long Short-Term Memory (LSTM)

We structured our problem as a supervised learning problem in order to apply LSTM. Each row in our dataset was converted to have 3 days of future AQI values (t, t+1, t+2) and also include features from t-1 to a certain amount of lag that we specified. This lets LSTM predict 3 days worth of AQI using all the features + lagged features that we chose to include. Features were scaled to be between 0 and 1.

We tuned various parameters by calculating the Test RMSE under different configurations. For tuning, we used data from 2016-2018 with the first 80% of data as train and the last 20% of data as test. See the results of the tuning in Section 4.4.1.

To evaluate the final performance of our LSTM model, we first trained our LSTM model on full dataset from 2016-2018. This gave us a base model with weights that learned from the historical data. We then iterated through each cross validation block. In each block, we updated the base model with the training set from each cross validation block, made a 3 day prediction, then calculated the RMSE for the prediction. Our final metric is the average RMSE of all of the CV blocks. Note that each CV block starts with the base model and is only updated with the training set in that CV block. For 2020 CV, we also included 2019 data in training the base model.

3.7 Hidden Markov Models

We trained ARIMA and VARMA models on the wildfire and non-wildfire classes identified by a Gaussian HMM. Since the values in the two classes are very disparate, a multi-modal model with HMM log-likelihood identifying the class allows the two sets of model to capture trends of their own class [6]. The classification results can be seen in figure 14.

4 Experiments

4.1 Summary of Modeling Performance

Figure 2 summarizes the performance of our models in the cross validation time periods we tested.

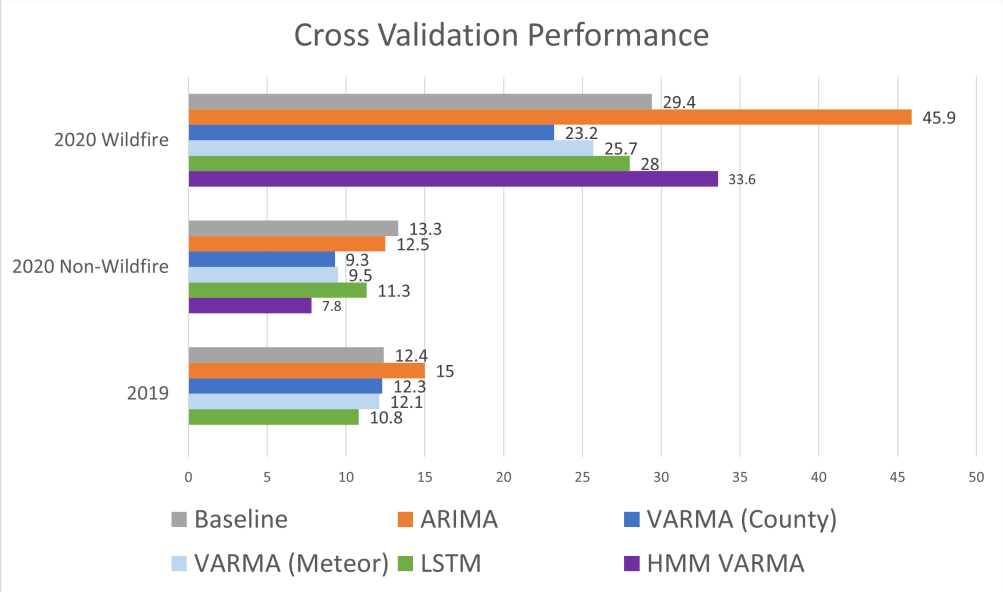


Figure 2: Summary of Modeling Performance

4.2 ARIMA Model

We performed basic checks (ACF, PACF, etc) to fit a suitable ARIMA model to the AQI time series of Santa Clara county (see Appendix 6.1) and obtained a ARIMA(3,0,1) parameterization using `Auto-Arima` from `pmdarima` package in python. We stuck to unprocessed AQI values as applying log/box-cox transformations or first order differencing did not improve performance. We obtain an average RMSE of 20 from our cross-validation setup. The AQI scale increases in a category of concern every 50 AQI, so 20 AQI is 40% of a category of concern and is quite high. If we refit the model for every block, the RMSE reduces to 15.8. This indicates that we cannot use the same parametrization for the entire time series and that our time series changes order depending on the time period. We also fit various transformations of the Santa Clara AQI data and compare the performance of the transformations in Table 1.

4.3 VARMA

4.3.1 Nearby Counties

The results of the VARMA models are summarized in Table 2.

We first tested including AQI of San Benito, San Joaquin, and Stanislaus in the VARMA model as nearby counties (I) of Santa Clara. Using the F-test for network Granger causality on the entire period (June 2016 - December 2020) concludes that the three counties "Granger cause" Santa Clara AQI. Including them yields an improvement over univariate AQI modeling.

To be more rigorous, we then restricted the set of counties available to those in a 100 mile radius (II) of Santa Clara. We found that using Calaveras, Sacramento, San Benito, and Yolo gives the maximum Granger statistic. The average RMSE after including these counties was 13.9, which is an improvement over randomly picking nearby counties.

4.3.2 Santa Clara Meteorology Features

Finally we include only the meteorology features of Santa Clara as components of the VARMA model (without surrounding counties' AQIs). The network Granger causality F-test again concluded that these features "Granger cause" Santa Clara's AQI. The average RMSE with including meteorology data improves the predictive power compared to the univariate case.

Transformation	RMSE
no transformation	15.81
log transform	15.38
Box-Cox transform	15.25
1st order differencing	17.26

Table 1: AQI prediction for transforms

VARMA	RMSE	Test Stat	Crit Value
I set of counties	14.1	6	2.61
II set of counties	13.9	9.72	2.37
Meteorology factors	14.5	9	2.37

Table 2: AQI prediction for VARMA models

4.4 Long Short-Term Memory (LSTM)

4.4.1 Parameter Tuning

We looked into adjusting the following parameters for our LSTM model in the following order of priority:

1. Direct Features
2. Nearby Features
3. Lag
4. Neurons
5. Epochs

Since searching for all possible combinations of all those parameters would be infeasible, we took a prioritized greedy approach where we adjusted parameters one at a time and kept that configuration for the rest of the tuning. This makes our parameter search dependent on the order in which we explore our parameters.

From trial and error before our final tuning procedure, we had a sense of what were some decent configurations for our LSTM model on this dataset. We started our configurations with 100 epoch (to get a large enough range since we knew the minimum is usually within 70 epochs), 10 neruons, 1 batch size, and lag of 7. We ran each tuning config 5 times, took the average test RMSE at each epoch, and report the epoch of the lowest average test RMSE.

For features to include, we tested the following combinations (both for direct and nearby):

- AQI - AQI Only
- Meteor (Meteorology) - AQI with Temperature and Relative Humidity/Dew Point (RHDP)
- Wind - AQI with Wind Speed and Direction
- WindProject (Wind Projection) - AQI with Wind Projection
- MeteorWind - AQI with Temperature, RHDP, Wind Speed, and Wind Direction
- MeteorWindProject - AQI with Temperature, RHDP, and Wind Projection

The full list of tuning can be seen in the Appendix in Table 7.

Direct Features

Table 3 shows the tuning results for direct features. We kept the other configs as 10 neurons, 1 batch_size, 7 lag, and no nearby features.

Since our wind projection converts wind speed and wind direction into a value toward Santa Clara County, it did not make sense to have a wind projection feature for Santa Clara County itself. We can see this in how wind speed + direction performed better than wind projection as a direct feature. However, it seems like the best set of direct features was only using AQI with meteorology features without wind information. We continue our tuning with this combination.

Nearby Features

Table 4 shows the tuning results for nearby features. We kept the other configs as 10 neurons, 1 batch_size, 7 lag, and meteor direct features.

Direct Features	epoch	Test RMSE
aqi	67	25.46
meteor	67	25.26
wind	60	25.89
windproject	94	26.04
meteorwind	52	26.61
meteorwindproject	75	25.77

Table 3: Tuning Direct Features

Nearby Features	epoch	Test RMSE
none	67	25.263
aqi	36	27.16
meteor	54	27.97
wind	53	29.00
windproject	66	26.61
meteorwind	54	29.50
meteorwindproject	70	27.25

Table 4: Tuning Nearby Features

Lag	epoch	Test RMSE
1	35	25.62
2	57	25.59
3	42	26.88
4	64	25.73
5	56	25.22
7	66	26.61
14	58	29.23

Table 5: Tuning Lag

Lag	Neurons	epoch	Test RMSE
2	1	34	28.43
2	2	80	26.87
2	5	51	25.74
2	10	57	25.59
2	20	41	24.79
2	50	30	23.93
2	75	39	23.09
2	100	44	23.34

Table 6: Tuning Neurons

When we added features from nearby counties, performance seems to have degraded. However, this may be due to suboptimal configurations from the other parameters, so we will ignore the option of leaving out nearby features completely.

Wind projection performed better than wind speed + direction and adding nearby temperature + rhdp seemed to degrade the performance even more. The best version including nearby counties seemed to be only including AQI with wind projection.

Lag

Table 5 shows the tuning results for lagging our features. We kept the other configs as 10 neurons, 1 batch_size, meteor direct features, and wind projection nearby features.

It looks like lags 1, 2, and 5 performed the best. We went with lag 2 since most of the trials reached the lowest peak and was relatively stable compared to 1 and 5. See Appendix Figures 15 and 16 to see the performance of lag 2 and lag 5 over epochs. Only one of the trials was consistently high, pulling up lag 2's average. Lag 2 is also the order selected by our ARIMA model.

Neurons

Table 6 shows the tuning results for increasing the number of neurons. We kept the other configs as 1 batch_size, meteor direct features, and wind projection nearby features. We tested this for both lag 2 (Table 6) and lag 5 (Appendix Table 7, since they were both good from the previous step.

After running through several neuron combinations, we got the lowest RMSE with lag 2 at 75 neurons.

4.4.2 LSTM Results/Insights

The best config that we selected was:

Direct Features	Nearby Features	Lag	Neurons	Epochs
AQI, Temp, and RHDP	AQI and Wind Projection	2 days	75	40

From the tuning exercise, we notice several insights:

- Temperature and RHDP of neighboring counties were not useful. Wind seems to be the most important meteorology factor for relating nearby counties. Temp and RHDP did not have enough signal and likely added to the noise of predictions.
- Lag 2 was the best to use and is similar to the best order discovered from ARIMA. However, AQI is very volatile and so a dynamic lag would likely work even better.
- Wind projection worked better than wind speed and direction independently. This means we chose a good approach for combining the two features.
- LSTM seems to produce more conservative estimates when dealing with periods of volatility. We can see in Figure 3 that LSTM's predictions were more aligned with the general trend and did not seek to capture large spikes as the naive baseline had done.

The predictions vs actuals charts can be seen in Figure 3.

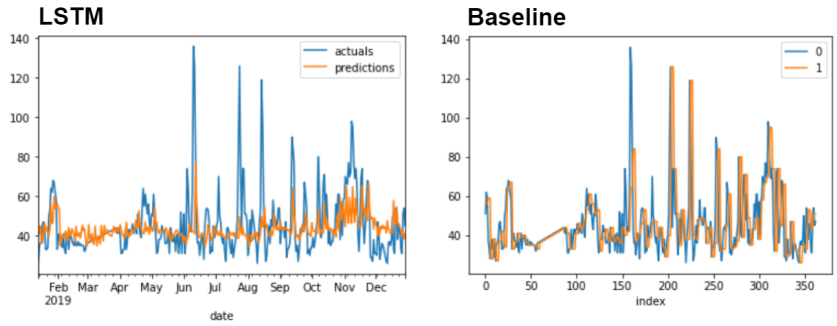


Figure 3: 2019 Predictions vs Actuals

4.5 HMM

Splitting the data into wildfire and non-wildfire classes with HMM improved the performance on non-wildfire ARIMA and VARMA models for 2020 to 8.8 and 7.8 respectively. However, the model performed worse for wildfire data and the RMSE actually increased to 37. and 33.6 for ARIMA and VARMA respectively. The increase in RMSE for wildfire periods can be understood as the past data being used to predict the wildfire AQI might be from the previous year (especially for the early wildfires in 2020). The lack of wildfire data affects the performance of the ARIMA and VARMA.

5 Conclusion

Air quality is quite volatile and cannot be modeled by a simple stationary model since different weather conditions drastically change air quality.

Adding meteorology and spatial features did help our more complex models, but we also needed to be careful in how many features we include since irrelevant/low power features create additional noise and decrease the performance of our models.

5.1 Future Directions

There are many additional things we could explore to further improve AQI predictions:

- Hourly data (as we mentioned AQI data is very time dependent and so there is likely a lot of information we are missing by using daily data)
- Multi-modal LSTM to have separate parameters/weights trained specifically for the states identified by Hidden Markov Models
- More complex architecture for LSTM based models.
- Using the geography along with proximity data to identify related counties.

References

[1] UCAR 2020. How weather affects air quality, 2020.

[2] Walter F. Dabberdt Et al. Meteorological research needs for improved air quality forecasting: Report of the 11th prospectus development team of the u.s. weather research program*. *Bulletin of the American Meteorological Society*, 85(4):563 – 586, 2004.

[3] Sienna Bishop. Air quality measurements series: Wind speed and direction, 2021.

[4] Yue-Shan Chang, Hsin-Ta Chiao, Satheesh Abimannan, Yo-Ping Huang, Yi-Ting Tsai, and Kuan-Ming Lin. An lstm-based aggregated model for air pollution forecasting. *Atmospheric Pollution Research*, 11(8):1451–1463, 2020.

[5] HS Hota, Richa Handa, and AK Shrivas. Time series data prediction using sliding window based rbf neural network. *International Journal of Computational Intelligence Research*, 13(5):1145–1156, 2017.

[6] Piyush Jain, Sean C.P. Coogan, Sriram Ganapathi Subramanian, Mark Crowley, Steve Taylor, and Mike D. Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4):478–505, 2020.

[7] Eric Hsueh-Chan Lu and Chia-Yu Liu. A spatial-temporal approach for air quality forecast in urban areas. *Applied Sciences*, 11(11), 2021.

[8] Dewen Seng, Qiyan Zhang, Xuefeng Zhang, Guangsen Chen, and Xiyuan Chen. Spatiotemporal prediction of air quality based on lstm neural network. *Alexandria Engineering Journal*, 60(2):2021–2032, 2021.

[9] Yugesh verma. A guide to varma with auto arima in time series modelling, 2021.

[10] Guyu Zhao, Guoyan Huang, Hongdou He, Haitao He, and Jiadong Ren. Regional spatiotemporal collaborative prediction model for air quality. *IEEE Access*, 7:134903–134919, 2019.

6 Appendix

6.1 ARIMA ACF and PACF plots

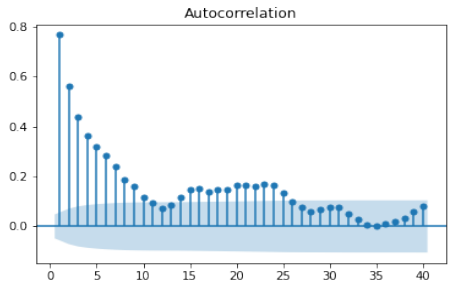


Figure 4: ACF for AQI

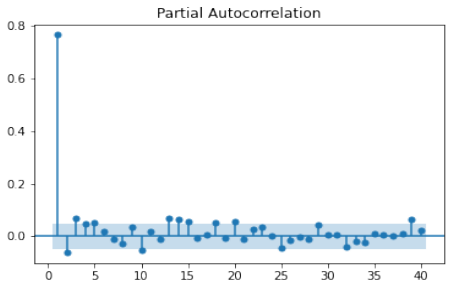


Figure 5: PACF for AQI

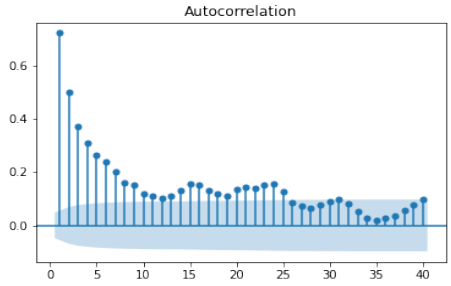


Figure 6: ACF for log transform of AQI

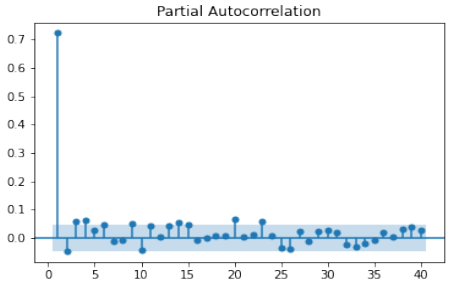


Figure 7: PACF for log transform of AQI

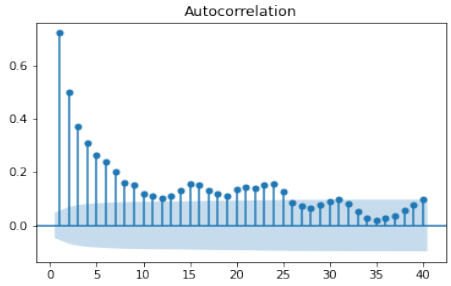


Figure 8: ACF for Box-Cox transform of AQI

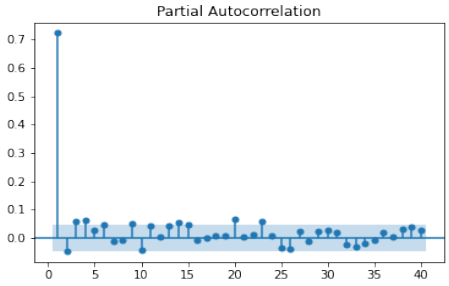
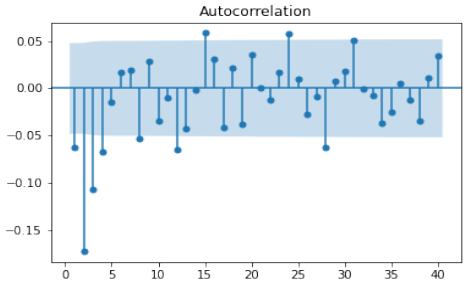


Figure 9: PACF for log transform of AQI



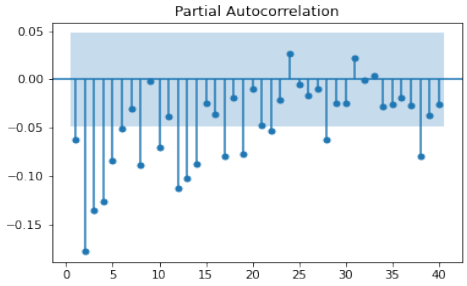


Figure 10: ACF for 1st order differencing of AQI Figure 11: PACF for 1st order differencing AQI

6.2 Cross Validation Figure

Figure 12 is a visual depiction of our sliding window cross validation structure.

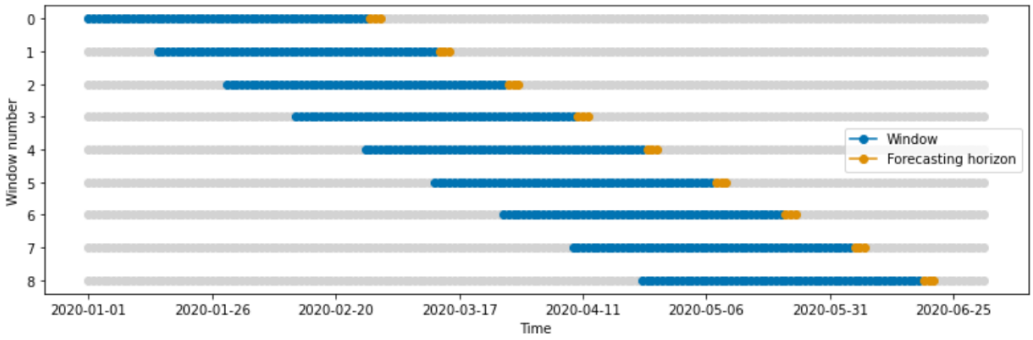


Figure 12: Block rolling cross validation

6.3 Evaluation Time Period

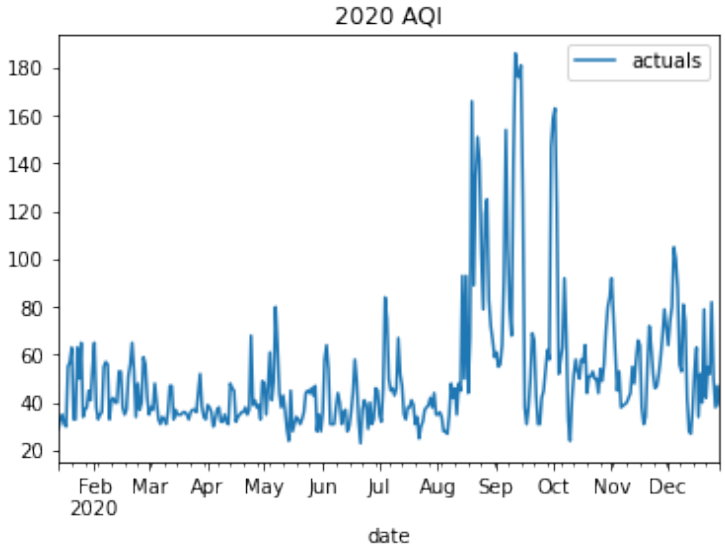


Figure 13: 2020 AQI with Wildfire Spike

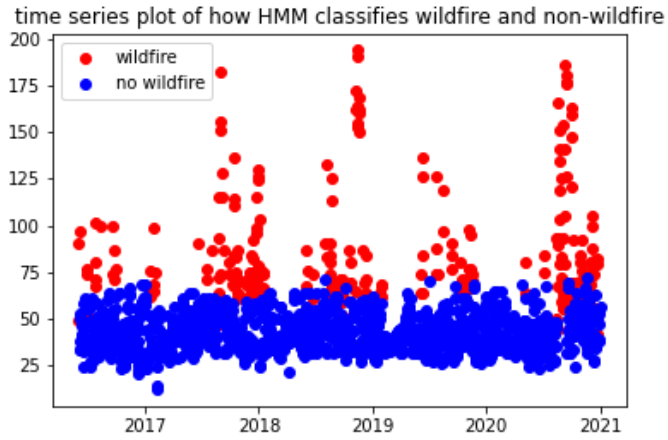


Figure 14: HMM classification of wildfire and non-wildfire data

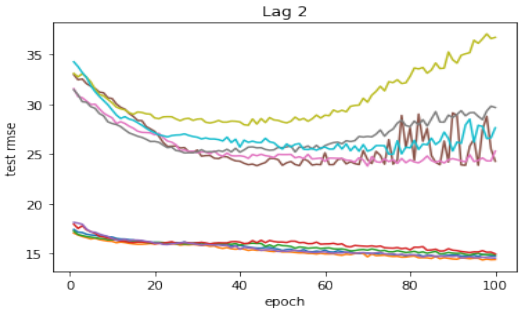


Figure 15: Lag 2 Performance over Epochs

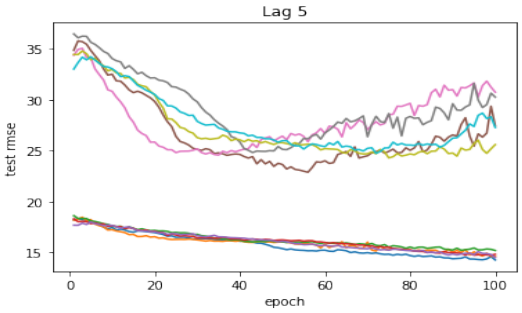


Figure 16: Lag 5 Performance over Epochs

6.4 LSTM Tuning

Neurons	Lag	Direct Features	Nearby Features	epoch	Train RMSE	Test RMSE
10	7	aqi	none	67	15.73918977	25.45672264
10	7	meteor	none	67	15.8051889	25.26477073
10	7	meteorwind	none	52	15.85994385	26.6145692
10	7	meteorwindproject	none	75	15.70233547	25.75666319
10	7	wind	none	60	15.80120224	25.89160272
10	7	windproject	none	94	15.58992427	26.03657839
10	7	meteor	aqi	36	15.95361049	27.16167587
10	7	meteor	meteor	54	15.13510196	27.97340131
10	7	meteor	wind	53	15.44366366	28.9999691
10	7	meteor	windproject	66	15.59304756	26.60827286
10	7	meteor	meteorwind	54	15.19661403	29.50006604
10	7	meteor	meteorwindproject	70	14.86675297	27.25394738
10	1	meteor	windproject	35	15.7906596	25.62292252
10	2	meteor	windproject	57	15.37477264	25.59162129
10	3	meteor	windproject	42	15.92021728	26.8786739
10	4	meteor	windproject	64	15.4190479	25.72642804
10	5	meteor	windproject	56	15.81333307	25.21539367
10	14	meteor	windproject	58	16.12330798	29.23350531
1	2	meteor	windproject	34	16.14252245	28.43357884
2	2	meteor	windproject	80	15.68555446	26.8681189
5	2	meteor	windproject	51	15.69736859	25.74204914
20	2	meteor	windproject	41	15.24387566	24.78608989
50	2	meteor	windproject	30	15.33068015	23.93242847
75	2	meteor	windproject	39	14.93920315	23.09266012
100	2	meteor	windproject	44	14.8198142	23.33582963
1	5	meteor	windproject	97	16.27385178	28.41902393
2	5	meteor	windproject	100	16.13827547	26.71099619
5	5	meteor	windproject	57	16.35187237	27.08850311
20	5	meteor	windproject	53	15.84929292	24.58762524
50	5	meteor	windproject	45	15.84466571	24.43962834

Table 7: Full LSTM Tuning