Time Series Project

Delhi Air Quality Between 2015 and 2020

Technion - Israel Institute of Technology

MELLOUL Naomie - 342445871
JORNET Jeremy - 931215248

March 19, 2025

Submission Deadline (early submission): April 19, 2025.

Contents

1 Introduction

The Air Quality Index (AQI) is a standardized measure used to assess air pollution levels. It is computed using a formula that aggregates several key air pollutants. In this project, we analyze the daily AQI time series for Delhi, India from 2015 to 2020, sourced from the Central Pollution Control Board via Kaggle. We aim to explore its statistical properties and predictive potential. Our key questions focus on identifying trends, seasonality, evaluating the forecasting performance with three models learned in class, assessing whether incorporating an exogenous variable improves predictions and detecting structural shifts overtime. We first conduct a visual analysis and time series decomposition, followed by model fitting and forecasting, then adding an exogenous variable, and conclude with change point detection to determine if significant shifts occurred in AQI distribution over time and discuss their possible causes.

2 Preliminaries and Data Visualization

In this section, we conduct an initial approach based on the visual analysis of the time series. The objective is to visually identify seasonality, trends, and patterns within the data.

2.1 High-Level Overview

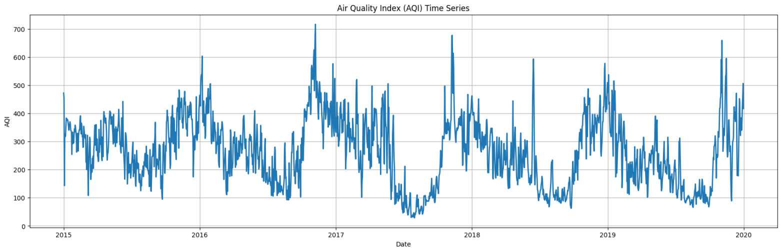


Figure 1: Air Quality Indicator Time Series from 2015 to 2020 in Delhi, India

A preliminary visual analysis reveals recurring patterns over time, indicating potential seasonality. Values appear in distinct blocks at regular intervals, though their amplitudes may vary slightly. The series is highly volatile and there is no clear long-term increasing or decreasing trend. Several sharp peaks are present, but their regular occurrence over time suggests a seasonal pattern rather than true outliers.

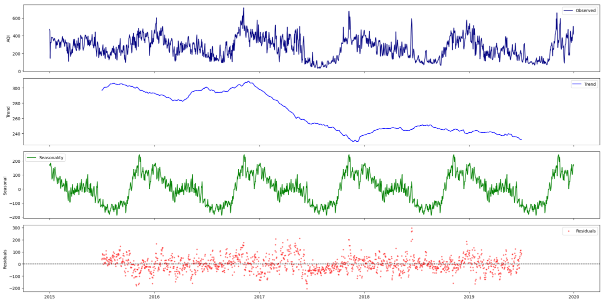


Figure 2: Air Quality Indicator Time Series Seasonal Decomposition

Figure 2 shows a primary seasonal decomposition of the time series, using `statsmodels`. The trend plot will be discussed in the next *Trend* subsection. The seasonal plot exhibits the isolated seasonal component of the time series and shows its recurring oscillatory structure. The residuals plot represents the remaining variation in the data after removing the trend and seasonal components. We can observe that residuals are centered around zero and do not display clear pattern : the random noise assumption may be reasonable.

2.2 Trend

In the trend plot of Figure 2, there is no clear trend observed between 2015 and 2017. However, a sharp downward trend emerges in early 2017, which appears to stabilize from 2018. As mentioned in Tirgul 4, if a trend is observed, the times series is likely not stationary because the expectation value varies with time.

2.3 Seasonality

The visual analysis of the seasonality component suggests an annual pattern, meaning that similar periods across different years exhibit comparable behavior. The times series is likely not stationary because of the strong seasonality. Based on this observation, we first examined the distribution of AQI values across months over multiple years to confirm the presence of annual seasonality.

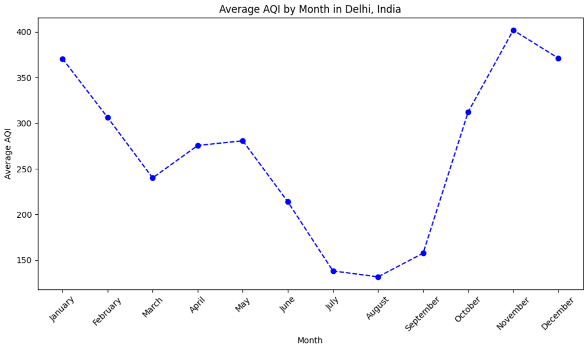


Figure 3: Average AQI Over Different Months

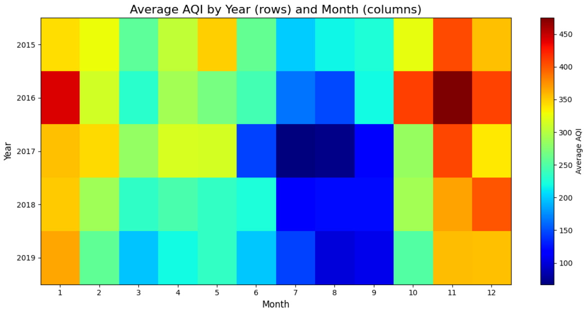


Figure 4: Heatmap of AQI Over Months and Years

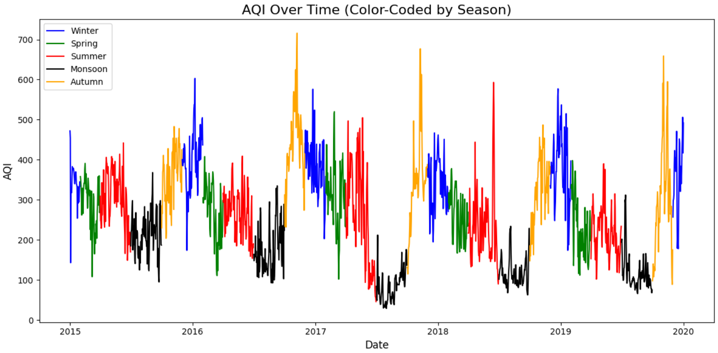


Figure 5: AQI Time Series with Colored Seasons

Figure 3 highlights significant variations in average AQI values across months, with adjacent months displaying more similar values. The average AQI is notably lower in June, July, August, and September, while it peaks in November, December, and January. Figure 4, the heatmap, provides a clearer visualization of this pattern: across multiple years, months exhibit consistent AQI averages color region, forming blocks of months with gradual transitions between them. We then chose to segment the time series according to Delhi's seasons, as referenced from Wikipedia (Winter, Spring, Summer, Monsoon, Autumn). The seasonal color-coding in the plot clearly reveals a recurring pattern across years.

We intuitively suspected a strong weekly seasonality, as weekday activities could influence air composition, potentially affecting air quality.

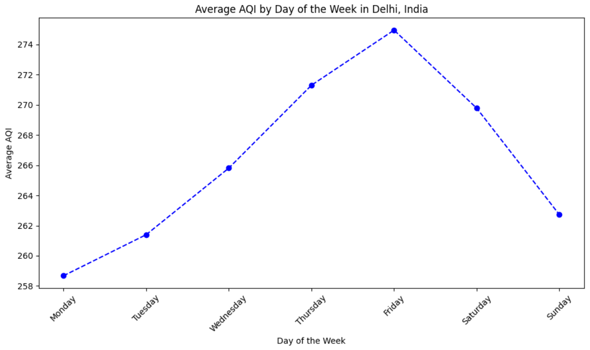


Figure 6: Average AQI Over Different Days

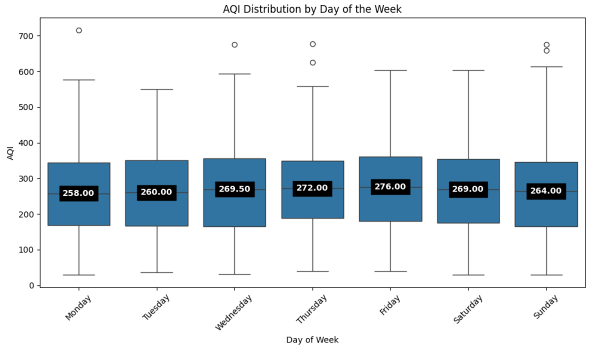


Figure 7: AQI Box Plot Distribution Over Days

As shown in Figure 6, the average AQI is highest on weekdays, gradually increasing from the beginning of the week until Friday, as expected, before dropping on the weekends. However, Figure 7 provides a clearer view, revealing that the change is very subtle. The distribution remains consistent, with a slightly increasing median throughout the weekdays and a decrease on the weekends, similar to the average trend.

2.4 Autocorrelation and Partial Autocorrelation

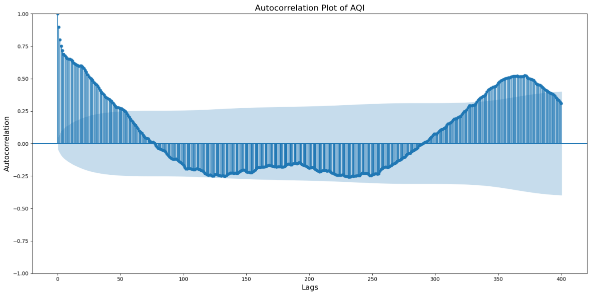


Figure 8: Autocorrelation Over 400 Lags

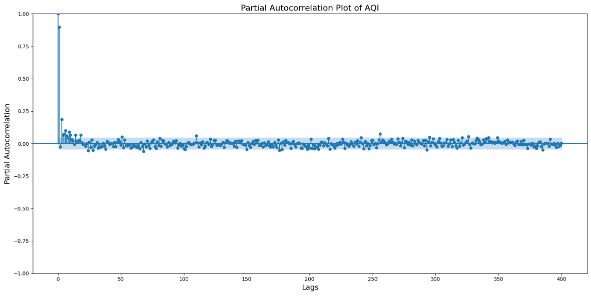


Figure 9: Partial Autocorrelation Over 400 Lags

The ACF plot in Figure 10 shows strong positive autocorrelation at small lags, which gradually decreases. A clear seasonal pattern is visible since the autocorrelation drops and then rises again outside the confidence bounds around lags of a year.

The PACF plot in Figure 11 shows a sharp drop-off after the first lag, being the most significant. Then, some of the first few lags are outside the confidence bounds and later almost all lags are within the confidence bounds. It suggests that only the first or two lags are significant.

3 Model Fitting

In this section, we present three different models fitted to the time series: SARIMA, Prophet, and a Neural Network. The project notebook includes detailed implementations, additional visualizations, and other modeling attempts, such as Regression with Fourier Terms. We selected SARIMA for its status as a classic time series model, capable of capturing complex relationships while remaining relatively simple. Prophet was chosen for its widespread use, scalability, and minimal parameter tuning nature, which is highly desirable in model fitting, as parameter selection is a significant challenge. We chose the Neural Network because we wanted to evaluate its performance on a Time Series, as we had previously implemented it for entirely different tasks.

3.1 SARIMA

As suggested by the project guidelines, because of computational challenges, we aggregated our time series by weeks and computed the weekly average AQI to reduce the number of rows.

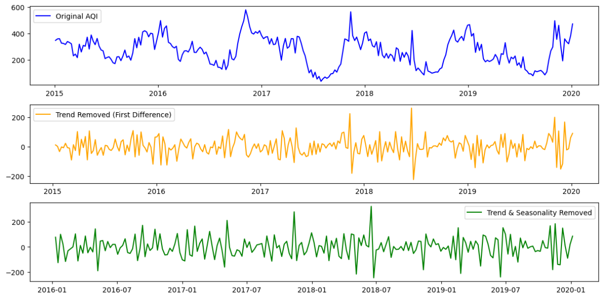


Figure 10: Seasonal Decomposition of the weekly time series

We discussed that the our time series is likely not stationary because of the trend and has inherent strong seasonality. Differencing of lag 1 to remove trend and of lag 52 to remove annual seasonality are needed, see Figure 10. The ADF test on differentiated Time Series confirmed stationarity.

We estimated the model parameters manually based on the methods learned in class. As mentioned during the reception hour on March 10th, the interesting part is to leverage our theoretical knowledge to estimate optimal model parameters rather than testing all possible combinations.

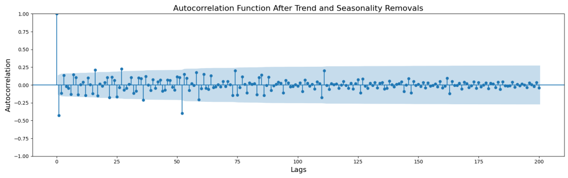


Figure 11: ACF of the weekly time series

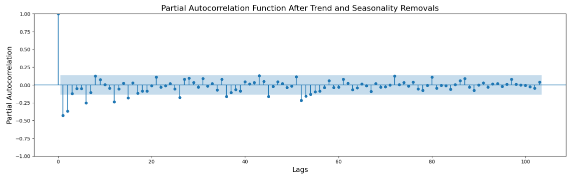


Figure 12: PACF of the weekly time series

In Figure 11, the ACF exhibits a sharp cutoff after the first lag, with nearly all subsequent lags falling within the confidence bounds or only slightly exceeding them, except for lag 52, which is significant but cuts off afterward.

In Figure 12, the PACF does not show a crystal clear pattern. We can either consider a sharp cutoff after the first and second lags or maybe a gradual decay to 0 with nearly all subsequent lags falling within the confidence bounds or only slightly exceeding them.

We consider two plausible scenarios, following the methodology we learned in Tirgul 4 :

1. if the ACF exhibits a sharp cutoff after the first lag and the PACF also exhibits a sharp cutoff after lag 2, this suggests an ARMA process involving both $MA(1)$ and $AR(2)$ components.

2. if the ACF exhibits a sharp cutoff after the first lag while the PACF decays gradually, this suggests an $MA(1)$ component.

Due to constraints on report length, we will only detail the second scenario, as it provided the best results: slight lower BIC, lower MAE and RMSE, better residual diagnostics, and a better visual forecasting fit. Moreover, the second scenario has a reduced number of parameters since it primarily considers a main MA component, which simplifies the model. As discussed throughout the semester, model simplicity is a highly desirable attribute.

Note : The first scenario analysis is available in the notebook we submitted.

Let's recall that a SARIMA model is defined by the parameters $(p, d, q) \times (P, D, Q, m)$.

Parameter	**Choice and Explanation**
p	$p = 0$ because we do not consider a non-seasonal AR pattern.
d	$d = 1$ to remove the trend.
q	$q = 1$ from the first significant spike in the ACF plot.
P	$P = 0$ because we do not want to add too many parameters to the model to keep it simple and relying on the dominant component.
D	$D = 1$ to remove the seasonality.
Q	$Q = 1, 2$ since the ACF shows a significant spike around lag 52 and a notable rise around the two-year lag, though still within the confidence bounds.
m	$m = 52$ since we identified annual seasonality and we transformed our time series into weekly observations.

Table 1: Second Scenario SARIMA Parameters Selection

We obtained the following results, ranked by BIC criteria :

	order	**seasonal order**	**AIC**	**BIC**
1	(0, 1, 1)	(0, 1, 2, 52)	554.15	561.80
2	(0, 1, 1)	(0, 1, 1, 52)	1112.88	1120.75

We noticed that the first scenario best model ((2,1,1)x(1,1,2)[52]) gave similar BIC of 567.81 even if it held many more parameters.

Moreover, the parameters (0,1,1)x(0,1,2)[52] surprisingly provided the best BIC criteria by far for the second scenario. We did not anticipate this result, given that the two-year lag displayed in Figure 11 fell within the confidence bounds despite exhibiting a noticeable increase.

After exploring possible explanations, we hypothesize that these unexpected outcomes might arise from the SARIMA implementation assigning zero coefficients to certain parameters (and then do not penalize BIC on these parameters) when insufficient training data is available. In our case, we indeed have small train data and particularly very small seasonal data due to the annual seasonality, so seasonal parameters might be impacted.

The two scenarios best models provided the following error metrics :

Metric	**1st Scenario Model**	**2nd Scenario Model**
Mean Squared Error (MSE)	10635.54	5496.09
Root Mean Squared Error (RMSE)	103.13	74.14
Mean Absolute Error (MAE)	86.27	57.86

Table 2: Best Models Test Error Metrics of Both Scenarios

While the BIC did not decisively distinguish between the best models from the two scenarios, the error metrics provide a clear indication. The second scenario, in addition to having a slightly better BIC and fewer parameters, shows significantly better error metrics on the test set.

We will now look at the residuals diagnostics of the best model :

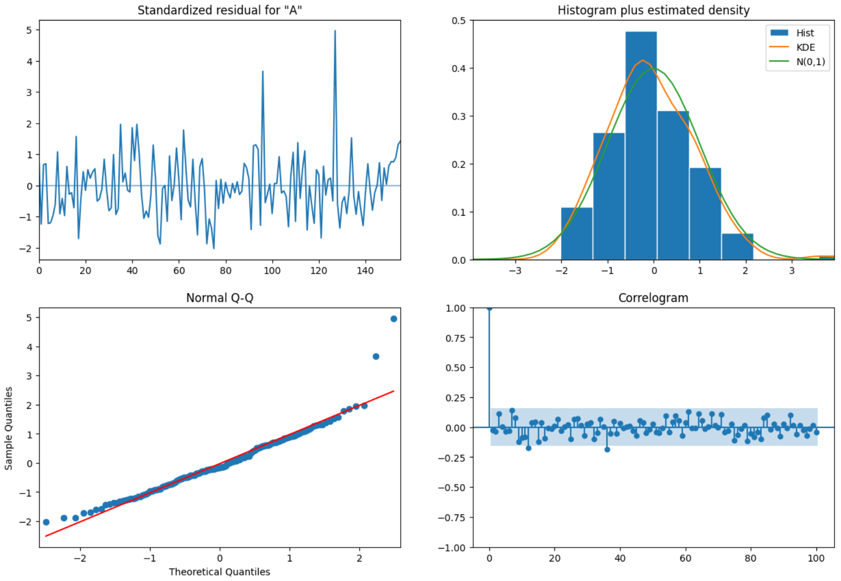


Figure 13: Second Scenario Best Model Residuals Diagnotics

The residuals appear centered around zero, which is a positive indication. The histogram appears roughly normal given the observations. Most residuals fall along the red theoretical normality line, suggesting approximate normality. The autocorrelation function shows no significant autocorrelation, with most lags falling within the confidence bounds. It is a good indicator for the model we got : it might have effectively captured most of the time-dependent structure.

We obtained the following forecast for the entire year of 2019:

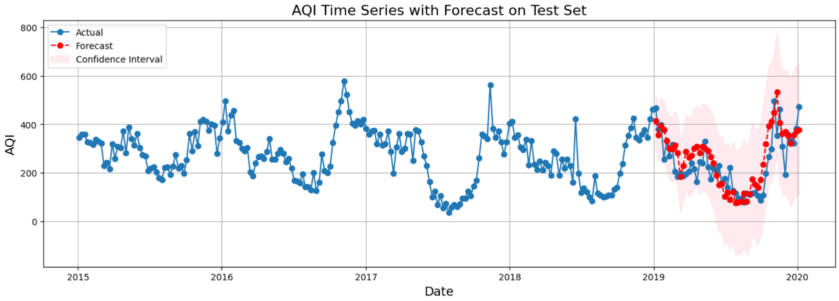


Figure 14: Actual vs. Forecasted AQI values for 2019 by First Scenario Best Model

We can observe in Figure 14 that the forecast follows the overall trend of the actual AQI values well and captures seasonal fluctuations and general movement. Some extreme peaks and drops are not perfectly captured, suggesting a slightly underestimated volatility. To conclude, the model performs well in capturing the general trend and seasonality but seems to struggle with greatly capturing extreme short-term fluctuations.

3.2 PROPHET

Prophet is a time series forecasting model developed by Facebook. Prophet is widely used for its simplicity and computational efficiency. Notably, we observed that its model fitting process was the fastest among all the approaches tested.

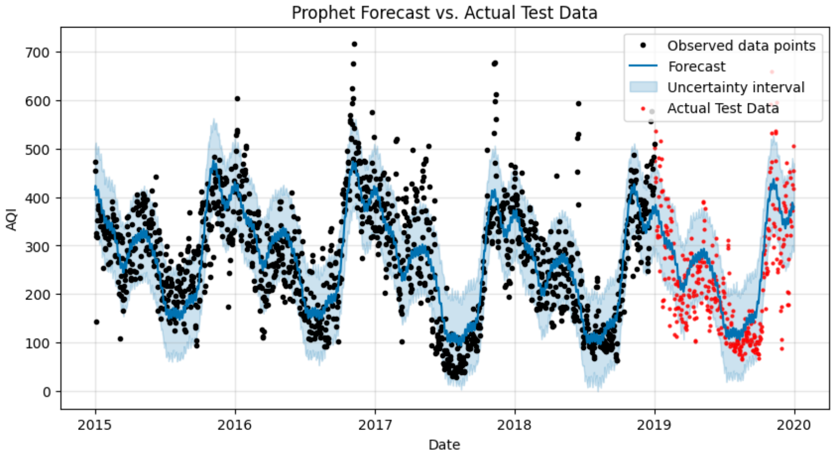


Figure 15: Prophet Forecasting on the Time Series

According to Figure 15, we can observe that Prophet forecasts periodic seasonal patterns and follows the general trend of the actual test data reasonably well even if it slightly overestimates. The majority of the actual test data is within the uncertainty interval.

About the parameter selection, Prophet automatically detects and models daily, weekly, and yearly seasonality without requiring manual feature engineering. We chose to use Prophet with its default settings, allowing it to handle parameter selection automatically.

3.3 Neural Network

We primarily based our implementation on Tirgul 11 and approached the task through three successive attempts, progressing from simpler to more advanced methods.

Initially, we used basic date-based features as in Tirgul, without any hidden layers. Surprisingly, this approach yielded good generalization, accurately following the overall data pattern, though it largely overlooked short-term variations.

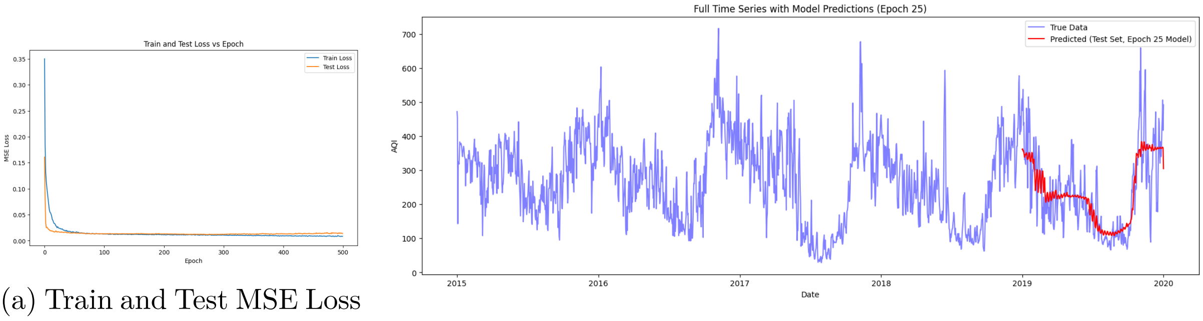
Next, we incorporated more sophisticated date-based features, which led to a more intricate forecast that more captured short-term fluctuations. However, despite this improvement, the model's performance metrics were unexpectedly poor.

Finally, in our third and most refined attempt, we introduced hidden layers along with an overall index feature to help the model capture the trend more effectively. The following details are specifically about this final implementation ("Third Attempt" in the notebook). A summarizing table of the model implementation such as features definition, model architecture and optimization strategy is available in the appendix, see Table 3.

The cyclic features were included to explicitly capture seasonal and periodic patterns inherent in the time series. Normalization was applied to ensure each feature contributes proportionally.

The use of multiple hidden layers provided smoother and more stable training and enabled to achieve lower test loss. The chosen layer sizes (256, 128, 128, 64, and 32 neurons) are common design patterns in neural network architectures. The ReLU activation function is even more relevant when modeling highly non-linear relationships typical of time series data. Dropout regularization was employed specifically because we learned in the ML2 course that it tends to good generalizations guarantees.

The Adam optimizer is the current SOTA in numerical optimization on this kind of problems. A standard learning rate of 1×10^{-3} was adopted. The Mean Squared Error loss function was utilized since it directly measures the average squared deviations of predictions from actual values, thus heavily penalizing larger errors and it is particularly relevant for time series forecasting tasks, where large prediction errors are often more undesirable.



(a) Train and Test MSE Loss Against Epochs

(b) The NN Model Forecast

Figure 16

According to Figure 16 (a), at the beginning, both the training and testing losses exhibited exponential decay, indicating stable learning and good generalization with low errors for both datasets. However, from epoch 400, we observe the early signs of typical overfitting behavior, as indicated by the test loss surpassing the training loss and starting to rise, whereas the training loss continues its decrease.

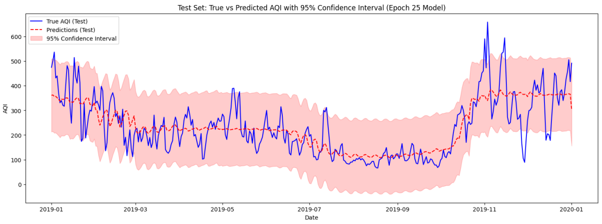


Figure 17: The NN Model Forecast Focus on 2019

Based on Figure 16 (b) and Figure 17, we observe that the Neural Network accurately captures the overall trend and seasonal patterns present in the actual data; however, it fails to adequately model shorter-term fluctuations and local variations. The prediction line appears smooth and almost straight.

3.4 Other Attempts

We also explored additional models using Fourier transformations, which provided interesting results and behaviors. However, due to length constraints on this report, we do not present them here. The detailed implementations are available in the submitted notebook.

Note : A summarizing table of the metrics of the models we discussed in the report is available in Appendix.

4 Incorporating an Exogenous Variable

Note : because length constraints, all the figures of this part are available in the Appendix.

We have obtained an additional dataset from Kaggle (link in Appendix) containing five daily average exogenous candidates: temperature, humidity, dew point (the temperature at which water vapor condenses into liquid water), wind speed, and pressure. Instead of testing each candidate individually and comparing performance through a brute-force approach, we will try to find the best candidate.

According to Figure 18, the average dew point variable shows the highest correlation with AQI. However, as highlighted in Human Factors in Data Science, correlation does not necessarily imply causality. To support our choice, we reviewed relevant research and found the article *Recognizing Dew as an Indicator and an Improver of Near-Surface Air Quality* by Yingying Xu and Xinyue Zhu, which suggests that dew can serve as an indicator of near-surface air quality. Based on this evidence, we selected the average dew point as the exogenous variable for this analysis.

We fitted a SARIMAX model using the same parameters as in the SARIMA section of the Model Fitting part, where the second scenario yielded the best results. As shown in Table 4, there is a clear improvement in MAE and MSE.

In Figure 19, we observe that the SARIMAX model with the exogenous variable provides a better fit to the actual data compared to the model without it. The predicted values are almost always closer to the real AQI values and exhibit less overestimation. The short-term fluctuations seem better captured. One possible explanation for this improvement is that incorporating the dew point as an exogenous variable helps capture additional atmospheric influences on air quality. This additional information likely enables the model to make more precise predictions.

5 Change-Points Detection

Note : because length constraints, all figures of this part are available in the Appendix.

In this section, we aim to identify changes in the distribution of the series over time. We previously observed that in early 2017, the trend shifted to a decreasing pattern, whereas no clear trend was present before 2017 and then from 2018, the decreasing trend seem to stabilize. The seasonality seems to remain unchanged according to Figure 5. It is not clear if the residuals structure change overtime.

Our objective is to apply statistical tools to confirm these changes we visually identified and to detect additional change points that may have gone unnoticed.

We firstly applied Shewhart detection on the non-stationary time series to observe interesting behaviors. Then, we applied detection on isolated trend and residuals components using STL seasonal decomposition.

5.1 First Method : Shewhart

We begin by a naive approach, applying the Shewhart method on the original non-stationary series in an offline setting since all observations in the series are already available. We followed the classic bounds selection that we saw in class : the Control Limit is set to three times the standard deviation of the time series.

5.1.1 Overall Change-Points Detection with Shewhart

In Figure 20, the Shewhart method detected change points that align with peak AQI observations in early November during 3 out of 5 years. This recurring pattern likely reflects seasonal effects or outlier events during high pollution periods rather than true shifts in the underlying distribution. When applied to the full non-stationary time series, the Shewhart method appears to capture extreme values rather than real change-points because of seasonality or/and outliers.

5.1.2 Trend Change-Points Detection with Shewhart

The Shewhart method did not detect any change points in the trend as seen in Figure 21, even though there is a noticeable abrupt shift in the trend distribution. This suggests that the Shewhart method may struggle to identify gradual or subtle distribution changes, especially when the shift does not exceed three times the standard deviation, as we discussed this phenomenon in class.

5.1.3 Residuals Change-Points Detection with Shewhart

When applied to the residual component, see Figure 22, the Shewhart method identified a few change points. However, these points do not seem to indicate changes in the distribution of the residuals but are more likely outliers, with detections occurring in both June and November.

After investigating potential external factors of the recurring extreme values in early November, we found that they coincide with the Indian festival of Diwali. This celebration, typically occurring in late October or early November, is accompanied by fireworks, which can significantly impact air quality in India, and of course in Delhi. The article

"Multi-year Comparison of Air Quality during the Diwali Period" by the SAFAR Team establishes a link between the pollution peak during this period and Diwali festivities.

5.2 Second Approach : Segmentation

We then explored more sophisticated methods and discovered the concept of segmentation in time series. We used the `ruptures` Python library, which provides various segmentation algorithms. Given that our time series is relatively short, we can afford the computational cost of the dynamic programming algorithm that provides the optimal segmentation.

Dynamic Programming Segmentation (DPS) Algorithm : The DPS Algorithm detects change points by finding the optimal way to partition a time series while minimizing a predefined cost function (the MSE). It systematically explores all possible segmentations using Dynamic Programming technique. The DPS Algorithm ensures a globally optimal segmentation in terms of minimizing the total segmentation cost.

5.2.1 Trend Change-Points Detection with Segmentation

We set the minimum segment size (the smallest allowed length for a segment) to 370, ensuring that each segment is able to capture significant distributional changes on a year timeline. The jump parameter (step size for evaluating change points) was set to 5 to accelerate computation. The number of breakpoints was fixed at 2, based on our visual interpretation of trend shifts, though we acknowledge potential limitations in this approach.

As illustrated in Figure 23, this method identified a first change point in early to mid-2017 and a second one in mid-2018. During the initial period, the trend appears neutral. In the second segment, a strong linear decline is observed, while in the third period, the decline persists but at a much more moderate rate.

The detected change points align with the previously discussed trend shift, which began in 2017 and stabilized in 2018. This suggests that the method effectively captured the distributional change in the trend. However, we notice a latency in detecting the transition, as the algorithm "fires the alarm" with some delay. Upon closer inspection, the actual shifts in distribution seem to have occurred in late 2016 and early 2018, suggesting that the algorithm detects changes later than expected. This behavior reminds the Detection Delay vs. False Alarm Trade-off discussed in class. Interestingly, we initially associated this trade-off with online detection methods, so observing the same visual behavior in an offline segmentation approach was unexpected.

We investigated which external factor may explain the observed and detected changes in distribution. We found out that between late 2015, 2016 and 2017, Delhi introduced measures to combat air pollution, including for example Odd-Even Traffic Scheme and restrictions on diesel vehicles. Emergency actions in November 2016 were taken in reaction to the extreme pollution peak. In 2017, the Supreme Court added bans through new regulatory decisions. The implementation of these measures, along with government efforts to reduce air pollution, may account for the observed change in the time series distribution. After this period of decreasing trend, the third segment of the time series trend shows a more gradual decline rather than a sharp drop. This stabilization could be attributed to the long-term effects of regulatory actions or the possibility that these measures have reached their full potential, meaning there may be weaker improvements from these policies alone.

5.2.2 Residuals Change-Points Detection with Segmentation

Here, we chose a shorter minimum segment size to allow for shorter structural changes, since residuals distribution shifts are typically not as long-term as trend changes. We set the jump size to 5 as before.

For the residuals change points, we were unable to visually determine the optimal number of breakpoints. To address this, we plotted the segmentation error as a function of the number of change points and selected the first point where the rate of error decrease began to slow down. We chose 4 change points, see Figure 24, though 6 could have been an alternative choice, but 4 provided a better visual fit.

According to Figure 25, the detected change points occurred in late 2015, as well as early, mid, and late 2017.

The first change point is somewhat surprising, as it appears early and does not visually seem to correspond to a clear shift. This may be attributed to the low minimum segment length parameter, which allowed for earlier detections and potentially false detections.

Before early 2017, in the second segment, the residuals had a stable distribution with a mean of -5.72 and a variance of 3776.49. In the third segment, we observed a significant rise in both mean and variance, reaching 58.48 and 5037.92, respectively. Subsequently, in the fourth segment, the residuals showed a marked decrease, with a mean of -46.30 and a lower variance of 1901.76. Finally, in the last segment, the residuals returned to their earlier stability, displaying a mean of 2.30 that is closed to zero and a variance of 3477.96 which is very similar to that of the second segment.

This interesting fluctuation suggests that before the abrupt shift (third segment, early 2017), the residuals distribution was relatively stable and centered around zero. A short-term increase and subsequent sharp decrease in mean and variance respectively occurred before returning to a stable state comparable to the initial condition. This pattern coincides closely with the series of regulatory measures implemented before and throughout 2017, which likely influenced the residual distribution, causing temporary but noticeable shifts. Eventually, the residuals returned to stability, aligning temporally with the stabilization observed in the trend component.

6 Conclusion

In conclusion, we clearly identified a decreasing trend and strong annual seasonality in Delhi's AQI time series. The three fitted models yielded decent results, and parameter selection was crucial to the modeling process. Among the models, SARIMA provided the best performance, although direct comparisons were limited due to weekly aggregation, see Tables 5 and 6. Surprisingly, the neural network outperformed Prophet and captured general patterns greatly. Incorporating average dew point as an exogenous variable significantly improved predictions with SARIMAX. Change point detection was challenging : Shewhart methods seemed inadequate, whereas segmentation methods clearly indicated structural shifts in early 2017 and mid-2018, coinciding with legal measures against air pollution taken before and during 2017. Additionally, residual analysis revealed interesting distributional stability before and after the 2017 shifts, further supporting the impact of regulatory actions hypothesis.

7 Appendix

7.1 Model Fitting - Neural Network Implementation Details Table

Category	Implementation Details
Features	Basic date features: `dayofweek`, `month`, `year`, `dayofmonth`.
	Cyclic features for seasonality: `dayofyear_sin`, `dayofyear_cos`, `month_sin`, `month_cos`, `dow_sin`, `dow_cos`.
	Trend feature: Normalized overall day index.
	Normalization: Scaling the target 'AQI' by 700 and other features by their respective ranges.
Model Architecture	Deep feed-forward neural network with 5 hidden layers.
	Layer sizes: 256, 128, 128, 64, and 32 neurons.
	Activation function: ReLU.
	Regularization: Batch normalization and dropout of 0.3 after each linear layer.
Optimization	Optimizer: Adam with a learning rate of 1×10^{-3}.
	Loss Function: Mean Squared Error (MSE).
	Training strategy: up to 500 epochs to plot train and test errors as a function of epochs but predictions were made with the best model at epoch 270.

Table 3: Summary of Implementation Choices

7.2 Incorporating Exogenous Variable Figures

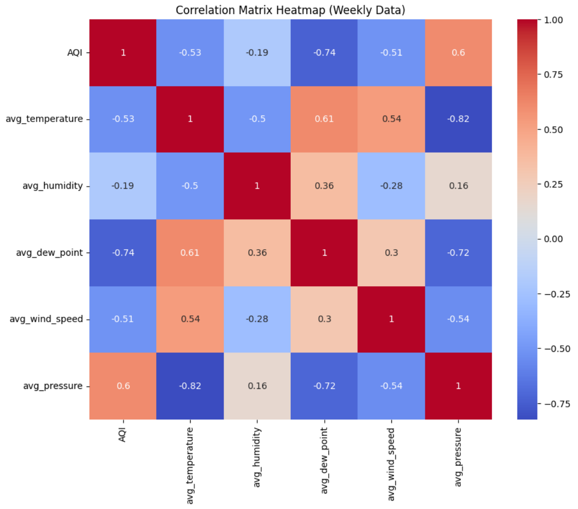


Figure 18: Correlation Heatmap Between AQI and Exogenous Candidates

Model	MSE	RMSE	MAE
SARIMAX (Endogenous only)	5496.09	74.14	57.86
SARIMAX (Endogenous and Exogenous)	4286.89	65.47	51.32

Table 4: With and Without Exogenous Performance Comparison on Test Set

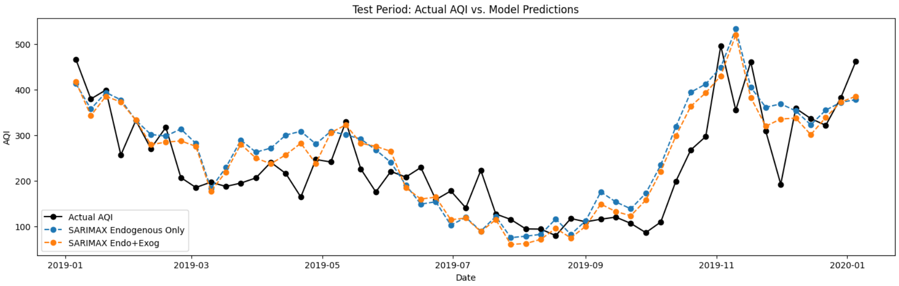


Figure 19: Real Test Data Against SARIMAX With and Without Exogenous Variable

7.3 Change-Points Detection Figures

7.3.1 With Shewhart

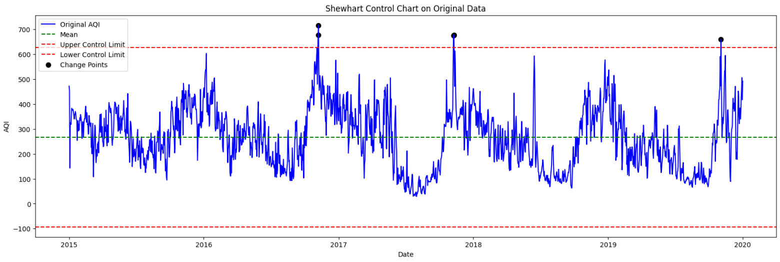


Figure 20: Original Time Series Shewhart Change-Points Detection

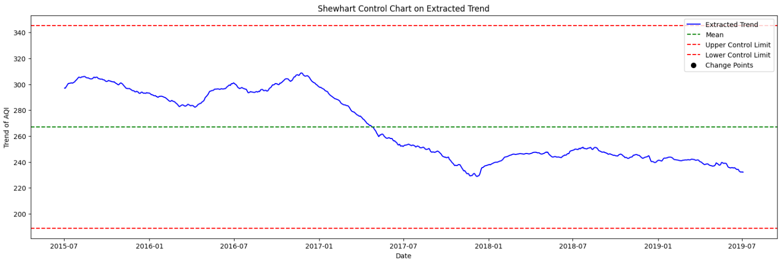


Figure 21: Isolated Trend Time Series Shewhart Change-Points Detection

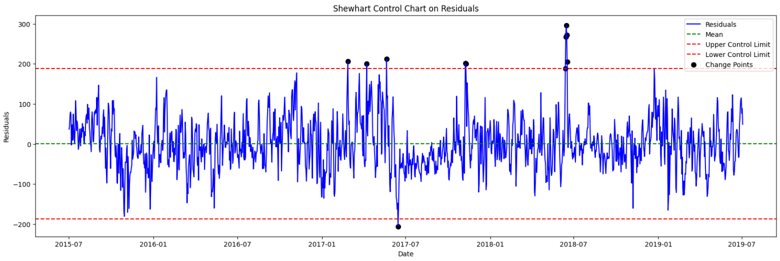


Figure 22: Isolated Residuals Time Series Shewhart Change-Points Detection

7.3.2 With Segmentation

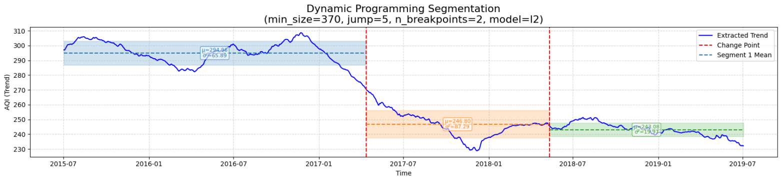


Figure 23: Isolated Trend Time Series Segmentation Change-Points Detection

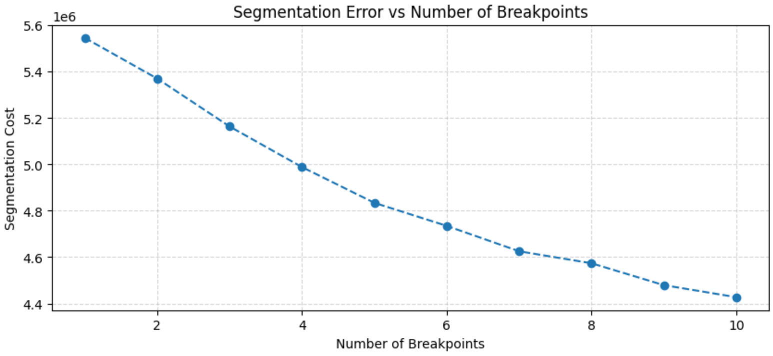


Figure 24: Segmentation Error as a Function of Number of Change-Points on Isolated Residuals Time Series

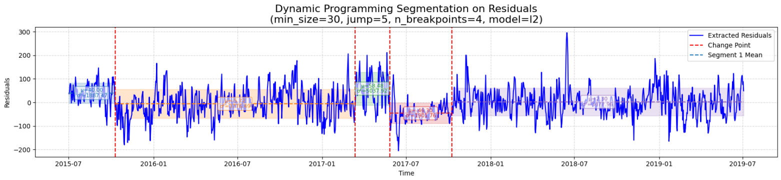


Figure 25: Isolated Residuals Time Series Segmentation Change-Points Detection

7.4 Metrics Summary Table

Metric	Prophet (Daily)	NN Model (Daily)
MSE	6015.97	5722.60
RMSE	77.56	75.65
MAE	60.31	55.57

Table 5: Comparison of Prophet and Neural Network Models (Daily Time Series)

Metric	SARIMA (Weekly)	SARIMAX (Exogenous, Weekly)
MSE	5496.09	4286.89
RMSE	74.14	65.47
MAE	57.86	51.32

Table 6: Comparison of SARIMA and SARIMAX Models (Weekly Time Series)

7.5 Datasets Source Links

Original Time Series

The original time series dataset is available at:

- `https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india`

Exogenous Variables

The exogenous variables are sourced from the dataset available via:

- `https://www.kaggle.com/datasets/ashx010/weather-patterns-and-trends`

References

[1] Arlt, J., Trčka, P., Arltová, M. (2018). *The Problem of the SARIMA Model Selection for the Forecasting Purpose.*

[2] Suhartono (2015). *Time Series Forecasting by Using Seasonal Autoregressive Integrated Moving Average: Subset, Multiplicative or Additive Model.*

[3] Machine Learning for Time Series. *Lecture 5: Change-point and Anomaly Detection*, Paris-Saclay.

[4] Time Series. *Episode 1: How to select the correct SARIMA parameters.*

[5] Malouche, N. *How can I select the best SARIMA model?.*

[6] Hyndman, R.J. and Athanasopoulos, G.
Forecasting: Principles and Practice.

[7] Cleveland, R.B., Cleveland, W.S., McRae, J.E., and Terpenning, I.
STL: A Seasonal-Trend Decomposition Procedure Based on Loess.
Journal of Official Statistics, 6(1):3–73, 1990.

[8] Box, G.E.P., Jenkins, G.M., Reinsel, G.C., and Ljung, G.M.
Time Series Analysis: Forecasting and Control (5th ed.).
Wiley, 2015.

[9] Skipper Seabold, Josef Perktold, *Statsmodels: Econometric and Statistical Modeling with Python*, Proceedings of the 9th Python in Science Conference, 2010.

[10] SAFAR Team, *Multi-year Comparison of Air Quality during the Diwali Period*, System of Air Quality and Weather Forecasting and Research (SAFAR), India, 2020.

[11] Yingying Xu, Xinyue Zhu, *Recognizing Dew as an Indicator and an Improver of Near-Surface Air Quality*, Environmental Science & Technology, vol. 55, no. 19, pp. 12999-13010, 2021.

[12] Anju Goel, Prashant Kumar, *Impact of Odd-Even Traffic Scheme on Air Quality in Delhi, India*, Journal of Transport & Health, vol. 10, pp. 133-142, 2018.

[13] Charles Truong, Laurent Oudre, Nicolas Vayatis, *ruptures: Change point detection in Python*, Journal of Machine Learning Research, vol. 21, no. 53, pp. 1–6, 2020.

[14] Sean J. Taylor, Benjamin Letham, *Forecasting at Scale*, The American Statistician, vol. 72, no. 1, pp. 37-45, 2018.

[15] Wikipedia, *Climate of Delhi*, Wikipedia, The Free Encyclopedia, 2024.

[16] *Time Series* Course, Technion, Yair Goldberg.