Forecasting Air Quality Index Values

Disa Alda Naomi
Department of Statistics
Stanford University
disaalda@stanford.edu

Prarthna Khemka
Institute for Computational and Mathematical Engineering
Stanford University
pkhemka@stanford.edu

Abstract

Forecasting the level of air quality index has been a challenging and important undertaking in time series modeling. In this project, we conducted a comprehensive case study of predicting the AQI index at the next time step (next week or next day) involving methods in traditional time series, machine learning, and deep learning. We compared (S)ARIMA, GARCH, and state space models, as well as regularized and dimensionality-reducing regressions, followed by recurrent neural networks and its variants. Our experiments reveal that deep learning methods, particularly our CNN-LSTM model, outperform the other methods.

1 Introduction

With the development of cities over time, air pollution problems caused by burning of fossil fuels, agriculture, residential heating, and other causes are arising. Air pollution has a direct impact on human health, which has increased the interest in air pollution and its impacts among the scientific community. Forecasting air quality accurately then can play an important role in air quality management and public health [2].

Forecasting air quality has been a complex task due to the high volatility and high variability in the feature space of pollutants and particulates, as well as in the time dimension [2]. A few papers have discussed the performances of several predictive models for air quality [2, 8, 15], and we hope to extend their efforts in our project. AQI of 100 or higher is deemed unhealthy, especially for sensitive groups [1] and thus we hope to apply time series forecasting to produce accurate and informative forecasts to help people better prepare their daily lives.

2 Background

Air pollution is considered to occur whenever harmful or excessive quantities of substances such as gaseous pollutants or particulates are introduced into the atmosphere. Some of the most common air pollutants that contribute to the level of air quality are measured by the US Environmental Protection Agency (EPA). The Air Quality Index (AQI) is an indicator measuring air quality that focuses on health effects that can be experienced when exposed to polluted air [1,2].

The main dataset for the project is the EPA Daily Air Quality Index (AQI) [1], which contains the overall AQI scores and common pollutants like Ozone, particulate matters of two different concentrations (PM2.5 and PM10), Carbon Dioxide (CO), and Nitrogen Dioxide (NO2), collected through all major US cities from 1980 to 2022.

3 Methodology

We analyzed a 10-year period (2012-2021) of air quality in the Santa Clara county through data of its major pollutants, adding more features (e.g. day of week, day of month) as we see fit. We combined all the individual years into one data frame for daily AQI and restructured of each variable into their correct type, imputed missing data with its median, and excluded variables. The variables were removed only if they were fully correlated with another variable, if they had missing values greater than 30%, or if the variable had only one data point.

By plotting the time series and creating a histogram for the pollutant distributions, we found that the main pollutant majorly was either ozone or PM 2.5 (Fig. 5 in Appendix 6.1.2). Moreover, we found that aggregating daily data into weekly captured the variations well, which can be seen in Figure 1, and so we decided to proceed with the aggregated data for further analysis.

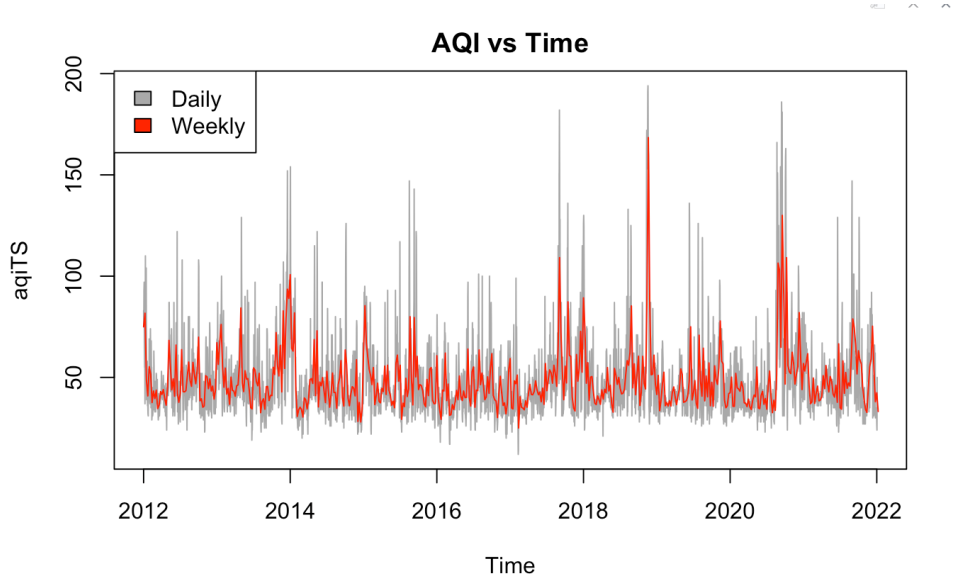

Figure 1: Daily and weekly AQI superimposed

We split the data into train and test sets using 80-20 split for our analysis. We took a three pronged approach, fitting time series models, machine learning models, and deep learning models to predict future AQI values. The models chosen in each category were fitted using the training set of the first 8 years of the data and then evaluation metrics were computed on the test set of last 2 years of the data.

4 Experiments

4.1 Time Series Analysis

The AQI index value is defined as the maximum of pollutant level amongst CO, NO2, Ozone, PM25 and PM10 levels for a given date (Fig. 4 in Appendix 6.1.1). Thus, the AQI index value is identical to exactly one of these pollutant levels at a given date. For this reason we compare two different approaches to predict the AQI index: 1) By using historical AQI index values and constructing a univariate time series model, and 2) By predicting the level of individual pollutants, then taking the maximum level to be the prediction of AQI index of a given date.

4.1.1 Univariate Analyses: AQI Index Time Series

To find the appropriate model for both daily and weekly data we followed these steps [14] using only historical AQI values in the model:

- Plot time series, ACF, and PACF to analyse the properties and check if differencing is needed.

- Difference time series and repeat step 1.
- Fit several models indicated through the analysis of the first two steps and run model diagnostics (ACF and normality of residuals and Ljung-Box statistic) for each of those models.
- Select the models that pass through the diagnostics and use for predictions.
- Compare models through prediction metrics of RMSE, MAE, MAPE etc., to select the best model.

Through these steps, we concluded that weekly data did a better job in both predicting AQI values and explaining trends in its time series, so we decided to continue the rest of the analysis using weekly data. Additionally, we verified the models chosen through diagnostics by also running the auto-arima() function in R, which finds the best model according to either AIC, AICc, or BIC value for a given data. We compared MSE, RMSE, MAE and MAPE of each model and proceeded with the model that outperforms the others in majority of the metrics.

Table 1: Performance of univariate time series models

Model	Parameters	MSE	RMSE	MAE	MAPE
ARIMA	(2,0,0) (with non-zero mean)	294.199	17.152	11.266	20.708
ARIMA	(0,1,1) (with drift)	354.805	18.836	11.831	19.663
ARIMA	(1,0,0)	294.2360	17.153	11.253	20.690
ARIMA	(1,1,1)	295.838	17.199	11.218	20.471

The analysis for the univariate time series resulted in ARIMA(1, 0, 0) being the best model for the undifferenced time series with RMSE of 17.15, MAE of 11.25, and MAPE of 20.69. The model diagnostics as well as ACF and PACF can be seen in Appendix 6.2.

4.1.2 Multivariate Analyses: Individual Pollutants Time Series

We used the historical values of each individual pollutant to fit models and then found the best-performing model for each pollutant through comparison of regression evaluation metrics of several time series models listed below.

- **SARIMA - Seasonal AutoRegressive Integrated Moving Average**: We experimented with different SARIMA processes, by analyzing the ACF and PACF plots of the time series and making use of auto-arima() function, noting that the strong seasonality in pollutant levels should be incorporated in the model. We conducted model diagnostics to examine the fit of the models and cross-checked with auto-arima() function.
- **GARCH - Generalized Autoregressive Conditional Heteroskedasticity**: As SARIMA models assume homoskedatic errors, GARCH models are able to relax such assumption by allowing for heteroskedatic errors and modeling conditional volatility/variance of the time series [6]. We fit the time series with commonly used GARCH(1,1) process.
- **ETS - Exponential State Space Smoothing**: This method compares a variety of state-space models with exponential smoothing. Exponential smoothing allows for weighted averages placing different weights on different observations. Additionally, the parameters in these models are estimated by maximizing the likelihood - not minimizing the sum of squared errors. The method chooses the best model with the lowest AICc in R [4].
- **Holt-Winters Filter**: The Holt-Winters filter is a type of exponential smoothing model that can also incorporate trend and seasonality in its specification. These three components together ("triple exponential smoothing"), have proven to be good-performing method to predict complex, noisy time series data, and thus we expect this method to perform well on our pollutant time series data [3].
- **TBATS**: The acronym stands for the key features of the model, i.e. Trigonometric Seasonality, Box-Cox transformation, ARIMA errors, Trend, and Seasonal components. The forecasting method aims to forecast time series with complex seasonal patterns using exponential smoothing [4]. Since the pollutant time series show strong seasonalities, we hope

that TBATS models could capture such seasonal complexity. The best model is chosen by choosing the model with the lowest AIC value in R.

We compared MSE, RMSE, MAE and MAPE of each model and proceeded with the model that outperforms the others in majority of the metrics for each pollutant (see model diagnostics in Appendix 6.3).

Table 2: Performance of best pollutant time series and combined models

Pollutant	Model	Parameters	MSE	RMSE	MAE	MAPE
CO	SARIMA	(1,0,1)(1,1,0)[52]	004.873	02.207	01.629	21.111
NO2	Holt-Winters	(0.143, 0, 0.371)	029.788	05.458	04.429	25.226
Ozone	Holt-Winters	(0.004, 0.067, 0.372)	083.257	09.125	06.861	18.186
PM2.5	TBATS	(0.009, 0,3, 0.869, <52,5>)	328.562	18.126	11.828	24.582
Final Model			277.256	16.651	10.371	18.201

Once we selected these best-performing models, we then predicted the AQI index value by taking the maximum of the four predictions for the next timestep. Such predictions gave comparable, slightly better error metrics than the univariate AQI index predictions as seen in Table 2. One may expect the error for each pollutant time series to propagate, but in this case, we find that combining several superior models on multiple pollutants improve our predictions. One can hypothesize that this gives more information for the predictions, as we are using information not only on the maximum value of pollutants but also on the past history of all pollutants in consideration.

4.2 Machine Learning

Machine Learning models have become increasingly popular in forecasting time series as they can be tuned to produce better models than statistical methods [11]. Since machine learning models are good at identifying patterns in multivariate context, we hypothesise that these flexible models can reduce the subjectivity that comes from determining time-series parameters (p,d,q) and be more accurate in predicting AQI at timestep t + 1. To do so, we converted the time series forecasting task into a supervised learning task, by creating subsequences of the time series containing past values of AQI and the main pollutants (at timesteps t, t - 1, t - 2, ... t - k) as inputs to the models. For all the models, we scaled our data using the optimal scaler (ex. Min-Max or Standard) and conducted hyperparameter tuning for each.

4.2.1 Lasso Regression

Lasso is a type of linear regression that performs L1 regularization by applying a shrinkage penalty equivalent to the absolute value of the magnitude of the coefficients. Lasso can perform variable selection since coefficients can shrink to zero, which promotes sparsity and could improve interpretability of the resulting statistical model. We tuned the shrinkage hyperparameter lambda, through cross-validation, to find an optimal value which minimizes MSE. Similarly, we tuned the length of the subsequences fed into the model. The optimal model for lasso resulted in 8 non-zero coefficients out of 25 (both exclusive of intercept), so significant variable selection occurred which improved accuracy from the baseline model.

4.2.2 Ridge Regression

Ridge is another type of linear regression that performs L2 regularization by applying a shrinkage penalty equivalent to square of the magnitude of the coefficients. However, unlike lasso, ridge does not shrink any coefficient exactly to 0, so it cannot perform variable selection. Similar to lasso, we tuned lambda and length of subsequences to give the lowest test MSE. Ridge in general performed better than lasso, indicating that the 8-variable model isn't sufficient.

4.2.3 Principal Component Regression ("PCR")

PCR is a regression method based on principal component analysis (PCA) which is a dimensionality reduction technique. The first principal component is the direction along which the observations vary

the most [6]. The second principal component is then orthogonal to the first and explains the most variability in the data that is unexplained by the first component, and so on. These components (new predictors) help summarize the data with fewer variables and thus performs dimensionality reduction. We performed cross-validation for tuning of M (number of principal components), which were in turn tuned through choosing subsequence length (Fig. 16 in Appendix 6.4.2).

4.2.4 Partial Least Squares Regression ("PLS")

PLS is similar to PCR, except that PLS identifies the principal components in a supervised manner by using the response variable to identify new features that not only approximate the old features well, but also that are related to the response. PLS was tuned similarly to PCR, but performed slightly worse.

4.2.5 Support Vector Regression ("SVR")

SVR is a regression counterpart to SVC/SVM, but instead of classifying data points into spaces divided by a hyperplane, SVR aims to find the best fit line (or hyperplane in higher dimensions) to the data, while giving some flexibility to how much error is acceptable as defined by the model practitioner. In contrast to ordinary least squares regression, the objective function of SVR is to minimize the coefficients. We fit SVR to our data and tuned for the hyperparameters C (tolerance for points far from the best fit line), gamma (distance of the influence of a single training point) and the kernel function (linear, polynomial, and radial basis function to perform the kernel trick). Additionally, we also tuned the length of each subsequence.

4.2.6 Random Forest Regression

Random Forest regression combines an ensemble learning method with decision tree framework to create multiple randomly drawn trees that each predict a value for the response. The final prediction from random forest regression is the average of predictions given from each tree. We tuned for the parameters of how many predictors should be considered for the split, how many trees should be grown, and length of the subsequences. We also tried bagging, which is the same as random forest except the number of predictors considered for the split are all the predictors involved. However, bagging performed worse than random forest so we have elected to not report it.

4.2.7 Boosting

While random forest and bagging involve creating multiple copies of the training set through bootstrapping and then fitting a separate decision tree to each dataset (i.e. each tree is independent), boosting grows trees sequentially [6]. Boosting essentially uses information from the previously grown trees and therefore learns from them and improves the performance of the current tree. We used two types of boosting models: XGBoost and Generalized Boosting Models ("GBM"). Through grid search for XGBoost, we tuned for the learning rate, the weight given to each child node when splitting, gamma (the amount of loss required when partitioning the tree), and max depth of the tree. For GBM, we tuned for lambda (shrinkage parameter) and interaction depth (highest level of variable interactions involved). Additionally, both models were also tuned for the length of the subsequences.

4.2.8 Results

The metrics for each optimal model can be seen in Table 3. The best model was PCR for Machine Learning models. See Appendix 6.4.1 for the best hyperparameters for each of these models chosen through cross-validation and/or grid search.

4.3 Deep Learning

Deep learning models have shown promise on time series forecasting tasks as evidenced in Wang, et al [15]. For our task, we hypothesize that deep learning models could learn complex trends in the AQI time series data by introducing nonlinearity representations, larger number of parameters, and relaxing some statistical assumptions in time series methods. We utilize different types of Recurrent Neural Networks ("RNNs") to predict the level of AQI for the next timestep. Since neural networks benefit from and are robust to a large number of data, we compared results from both weekly and

Table 3: Performance of best machine learning models

Model	MSE	RMSE	MAE	MAPE
Lasso	212.999	14.595	09.477	17.545
Ridge	202.437	14.228	09.374	17.427
PCR	196.998	14.036	09.416	17.693
PLS	200.189	14.149	09.327	17.414
SVR	207.186	14.394	10.161	19.250
Random Forest	222.530	14.917	09.798	18.626
XGBoost	252.771	15.899	10.263	19.621
Generalized Boosting Model	208.292	14.432	09.975	18.640

daily data. While we recognize that we may have limited data, we hope that deep learning approaches can make use of the complex trends that time series approaches have not, while also making use of the sequential nature of inputs.

4.3.1 Data Processing

Like with machine learning models, we created subsequences of the time series of various lengths for AQI and the pollutants as inputs to our RNNs, with the length of inputs as a hyperparameter that we tune on. We also compared different types of scalings of the input data, e.g. Min-Max and Standard Scaling, when available.

In processing the daily time series data, we experimented with creating time features such as the day of week, day of month and month variables as static variables to be fed into the network. However, to avoid the non-cyclical, high cardinality problems of using categorical variables, we transformed these variables into sine and cosine features, which have been shown to be effective in encoding cyclical variables such as time [13]. We then fed the transformed features in place of the original features into our networks that utilize daily time series.

4.3.2 Long Short-Term Memory ("LSTM")

LSTM is a type of RNN that takes in sequential inputs and can perform both regression and classification tasks. Each LSTM unit has three gates that control the flow of information being propagated: Forget gate, update gate, and output gate. LSTM was designed to avoid vanishing and exploding gradient problem, and to enable RNNs to 'remember' information from long sequences. Depending on the length of input sequences, our time series dataset can benefit from this feature. While LSTM is one of the most commonly used gated RNNs, we want to compare its performance with other RNNs that are simpler, e.g. GRUs, and those that are more complex, e.g. CNN-LSTM.

4.3.3 Gated Recurrent Unit ("GRU")

Another type of RNN with gating mechanisms, but with one less gate than LSTM. With less parameters for the same architecture configurations, GRUs are simpler models than LSTM, which means that we can build bigger models compared to LSTM with the same number of parameters. This could improve performance on our data since it is currently limited in size. However, LSTM could be more flexible in terms of controlling how information passes through the recurrent units. Thus, it is important for our project to experiment with both gated RNNs to infer the trade-off between model size and performance.

4.3.4 Convolutional Neural Network-LSTM ("CNN-LSTM")

We can utilize the convolution mechanism of CNNs into the LSTM network by adding a 1D convolution layer to process the inputs before passing on the data into LSTM layer. We can view the 1D convolution layer as a 'smoothing' pre-processing step for a pollutant/AQI level across time steps. The size of the convolution filter/kernel can be interpreted as the number of time steps considered to be processed for a given time step t. Learning the parameters of this convolution filter could be helpful to smooth out noisy data points in a time series, in order to extract meaningful signals from neighboring time steps. We hypothesize that the convolution mechanism could be efficient in

denoising our time series data, where there are layered trends and seasonality from each of the main pollutants in both weekly and daily data.

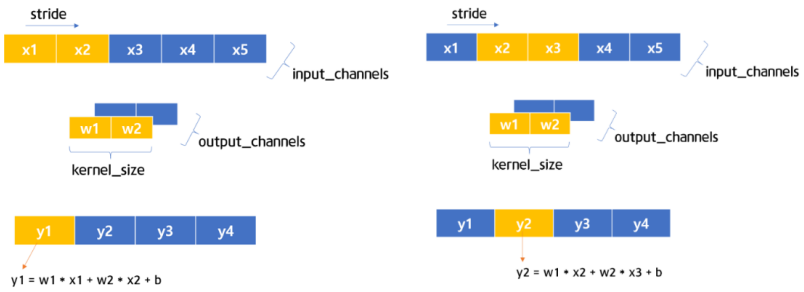

Figure 2: 1-D convolution operation on time series data

4.3.5 DeepAR

DeepAR is an auto-regressive recurrent network model to produce probabilistic forecasts, designed to train on a large number of related time series. It aims to learn a global model from historical data of multiple, potentially related time series. What sets this model apart from classical time series approaches are its minimal feature engineering and its ability to provide forecasts for items with little or no history at all [12].

Despite its advantages, DeepAR does not perform on the AQI dataset as well as a few of the deep learning approaches discussed previously. This is somewhat expected as the Amazon documentation warns that "DeepAR algorithm starts to outperform the standard methods when your dataset contains hundreds of related time series" [10] and we currently only have five time series consisting of the AQI index and major pollutants.

4.3.6 Temporal Fusion Transformers ("TFT")

In addition to the more 'traditional' RNN approaches above, we experimented with the novel attention-based architecture TFT [7]. The TFT architecture utilizes canonical components such as gating mechanisms and skip connections, variable selection networks, static covariate encoders, temporal processing, and prediction intervals for its forecasts. The architecture attempts to learn temporal relationships at different scales - using recurrent layers for local processing and interpretable self-attention layers for long-term dependencies.

This transformers-based model enables valuable interpretability to identify globally-important variables for the prediction problem, as shown by persistent temporal patterns and significant events as shown in the attention plots and feature importance plots in Appendix 6.5.2.

We found that on average across all predictions, the most attention for each prediction is given to the most recent t-1 timestep, and the attention decreases with time into the past. Analyzing the attention weights of individual predictions, however, we also notice that the model pays more attention to the past time steps when there are drops, sometimes overcoming the attention to the most recent timestep. This shows how the model could adapt its predictions to significant events such as drops in the AQI/pollutant levels.

Despite its novel features and model architectures, however, TFT models generally do not outperform other deep learning methods. Since some of its features, such as Transformers architecture and skip connections, greatly benefit from large amount of data and longer input sequences, we may need to accumulate more data in order to benefit from these features in our project.

4.3.7 Hyperparameter Tuning

For the GRU, LSTM, and CNN-LSTM models that we considered, we optimized for the hyperparameters listed below, while we optimize for some of these hyperparameters for the TFT and DeepAR models when available. We chose the best set of hyperparameters by first defining lists (or

grids) of candidate hyperparameters, then computing evaluation metrics of models fitted with each combination of the hyperparameters, and finally recording models with the lowest regression errors.

- Length of Sequence/Timesteps
- Learning Rate
- Batch Size
- Number of Epochs
- Dropout Rate
- Weight Decay
- Hidden Layer Dimension
- Layer Dimension
- Scaling
- Window/Filter Size (for CNN-LSTM)

4.3.8 Results

We concluded that the CNN-LSTM model performed the best, and the GRU model takes the second place. See Appendix 6.5.1. for the best hyperparameters for each model chosen by grid search.

Table 4: Performance of best deep learning models

Model	Time Aggregation	MSE	RMSE	MAE	MAPE
LSTM	Weekly	196.845	14.030	10.855	20.976
GRU	Weekly	189.289	13.758	09.298	18.099
CNN-LSTM	Weekly	169.871	13.033	09.327	17.257
DeepAR	Daily	270.857	16.458	11.544	23.265
TFT	Daily	238.865	15.455	09.656	19.451

We see that the forecasts generally predict the weekly AQI index well, as it generally fits the average values and some moderate spikes quite well. However, we observe a few instances where it underpredicts high spikes in the AQI index. We also observe one instance where the model overpredicts a values close to the average with a high value. Thus we might want to improve our models particularly on predicting extreme events such as spikes giving high AQI index values.

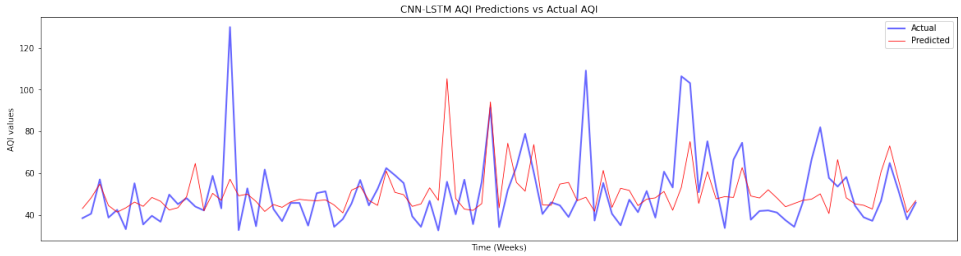

Figure 3: Forecasts from best CNN-LSTM model

5 Conclusions and Next Steps

In summary, we built models in the time-series domain using univariate and multivariate approaches to predict AQI at t+1 timestep. We found that multivariate time series approaches had better performance than univariate approaches, and proceeded to work on multivariate models. To reduce the subjectivity in determining time-series models, we modelled the time series forecasting tasks into supervised learning regressions and saw an improvement in accuracy, which led us to investigating more refined and complex models like neural networks. The deep learning models were able to utilize complex, nonlinear representations as well as the sequential nature of inputs and resulted in producing the most accurate predictions through CNN-LSTM on the weekly aggregated data.

For next steps, we would like to expand the scope of the project by procuring related datasets for wildfires or asthma rates to integrate into this model. We believe that by enriching our datasets, either by increased length of time period or the addition of related time series inputs, we can make use of some of the more complex models and architectures where we've seen opportunities of improved performance.

References

[1] “Air Quality Index Daily Values Report.” EPA, Environmental Protection Agency, https://www.epa.gov/outdoor-air-quality-data/air-quality-index-daily-values-report.

[2] Castelli, Mauro, et al. "A machine learning approach to predict air quality in California." Complexity 2020 (2020).

[3] Chatfield, C. “The Holt-Winters Forecasting Procedure.” Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 27, no. 3, 1978, pp. 264–79. JSTOR, https://doi.org/10.2307/2347162. Accessed 14 Nov. 2022.

[4] De Livera, A.M., Hyndman, R.J., & Snyder, R. D. (2011), Forecasting time series with complex seasonal patterns using exponential smoothing, Journal of the American Statistical Association, 106(496), 1513-1527.

[5] Engle, Robert F. “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation.” Econometrica, vol. 50, no. 4, 1982, pp. 987–1007. JSTOR, https://doi.org/10.2307/1912773. Accessed 14 Nov. 2022.

[6] Gareth, James, et al. An introduction to statistical learning: with applications in R. Spinger, 2013.

[7] Lim, Bryan, et al. "Temporal fusion transformers for interpretable multi-horizon time series forecasting." International Journal of Forecasting 37.4 (2021): 1748-1764.

[8] Liu, Hui, et al. "Intelligent modeling strategies for forecasting air quality time series: A review." Applied Soft Computing 102 (2021): 106957.

[10] Mishra, Abhishek. “Machine Learning in the AWS Cloud: Add Intelligence to Applications with Amazon Sagemaker and Amazon Rekognition.” Amazon, John Wiley & Sons, 2019, https://docs.aws.amazon.com/sagemaker/latest/dg/deepar.html.

[11] Parmezan, Antonio Rafael Sabino, Vinicius MA Souza, and Gustavo EAPA Batista. "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model." Information sciences 484 (2019): 302-337.

[12] Salinas, David, et al. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks." International Journal of Forecasting 36.3 (2020): 1181-1191.

[13] “Three Approaches to Encoding Time Information as Features for ML Models.” NVIDIA Technical Blog, 21 Aug. 2022, https://developer.nvidia.com/blog/three-approaches-to-encoding-time-information-as-features-for-ml-models/.

[14] Shumway, Robert H., David S. Stoffer, and David S. Stoffer. Time series analysis and its applications. Vol. 3. New York: springer, 2000.

[15] Wang, Jingyang, et al. "An air quality index prediction model based on CNN-ILSTM." Scientific Reports 12.1 (2022): 1-16.

6 Appendices

6.1 Exploratory Data Analyses

6.1.1 Correlation Matrix

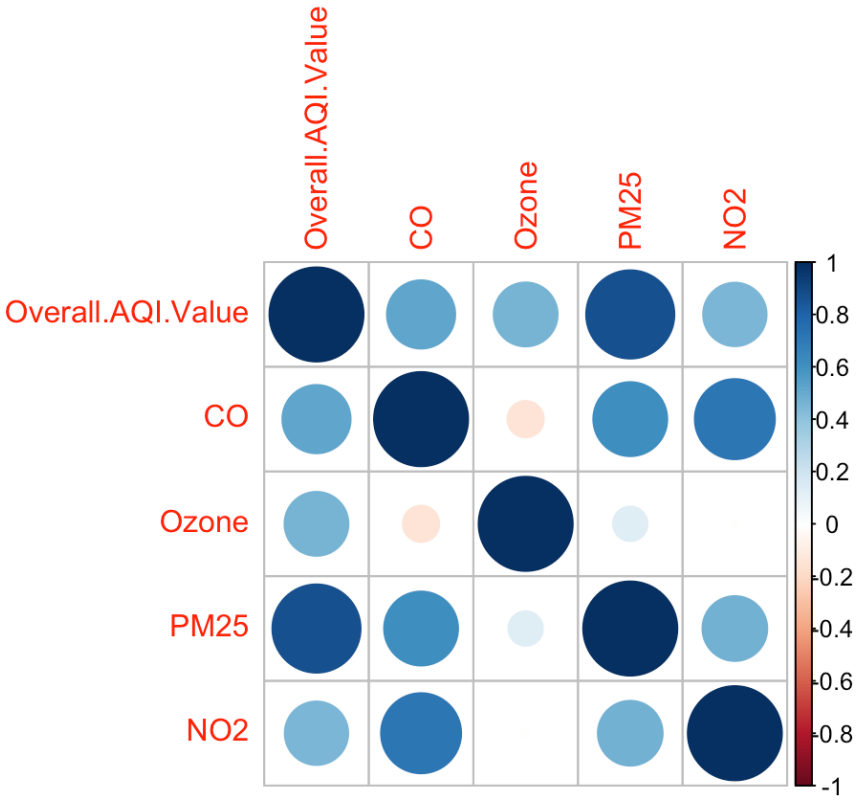

Figure 4: Correlation matrix of AQI dataset

6.1.2 Main Pollutant Analysis

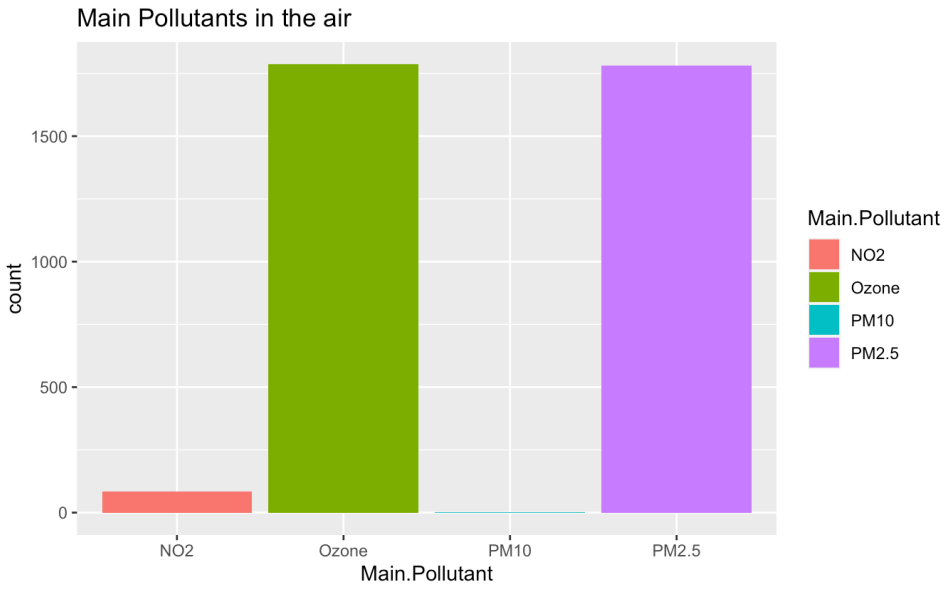

Figure 5: Main pollutant distribution

6.1.3 AQI Weekly

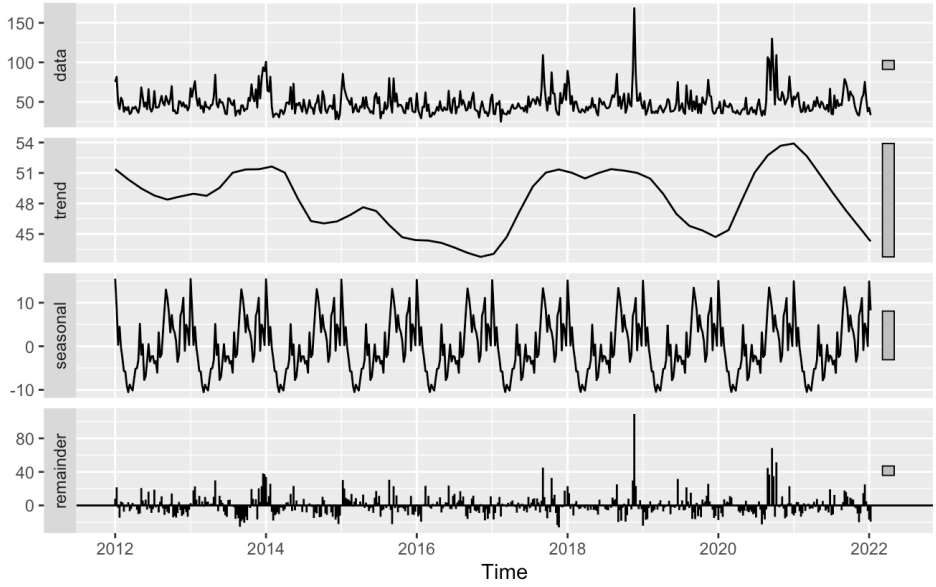

Figure 6: Weekly AQI decomposed.

6.2 Univariate Analyses: AQI Index Time Series

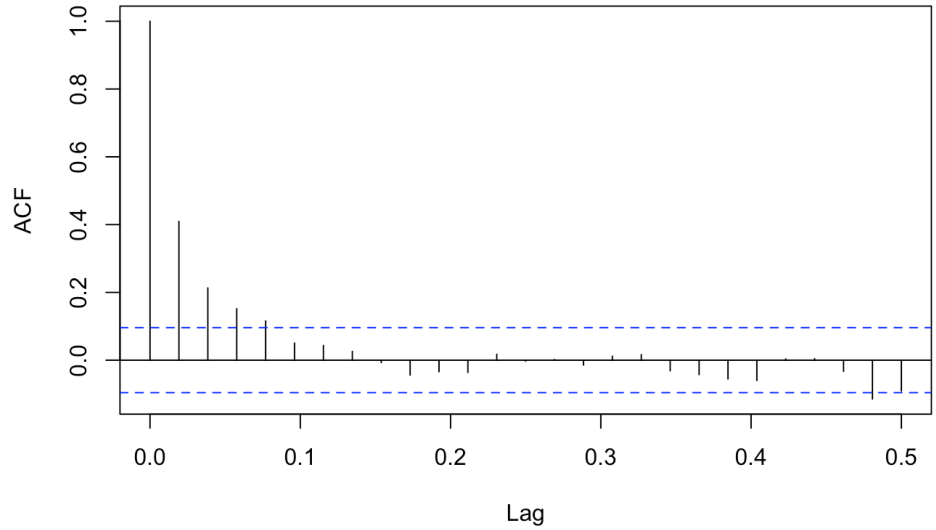

Figure 7: Sample ACF of weekly AQI

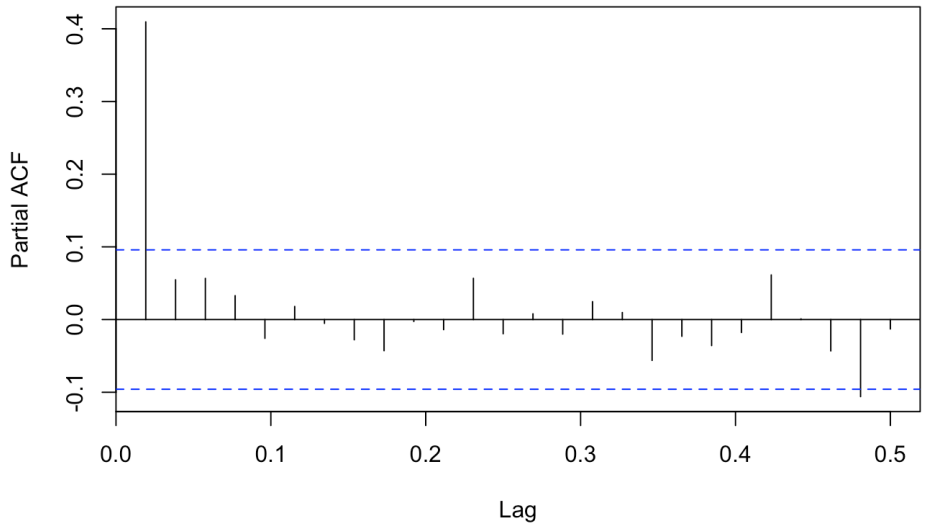

Figure 8: Sample PACF of weekly AQI

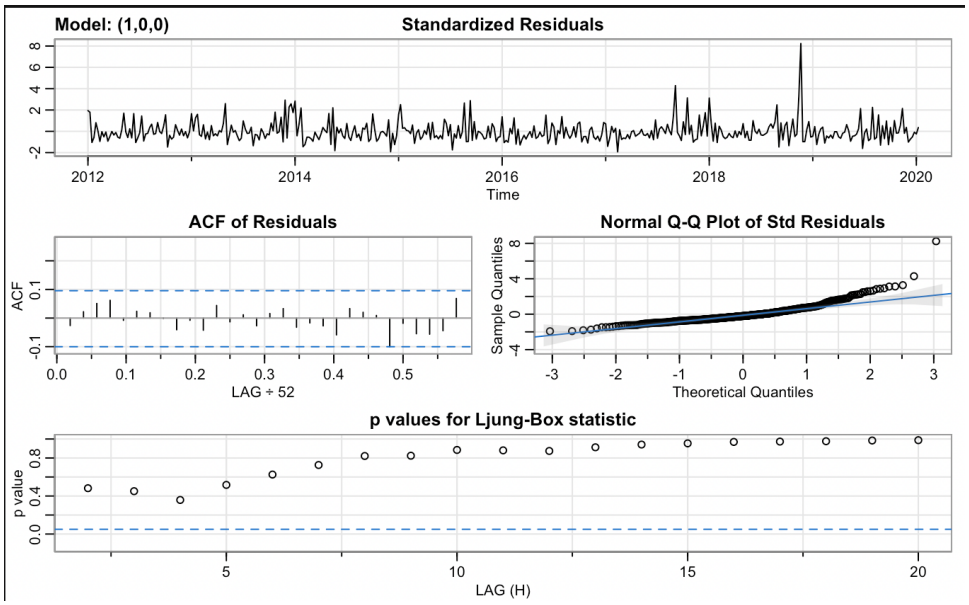

Figure 9: Model diagnostics for best model ARIMA(1, 0, 0)

6.3 Multivariate Analyses: Individual Pollutants Time Series

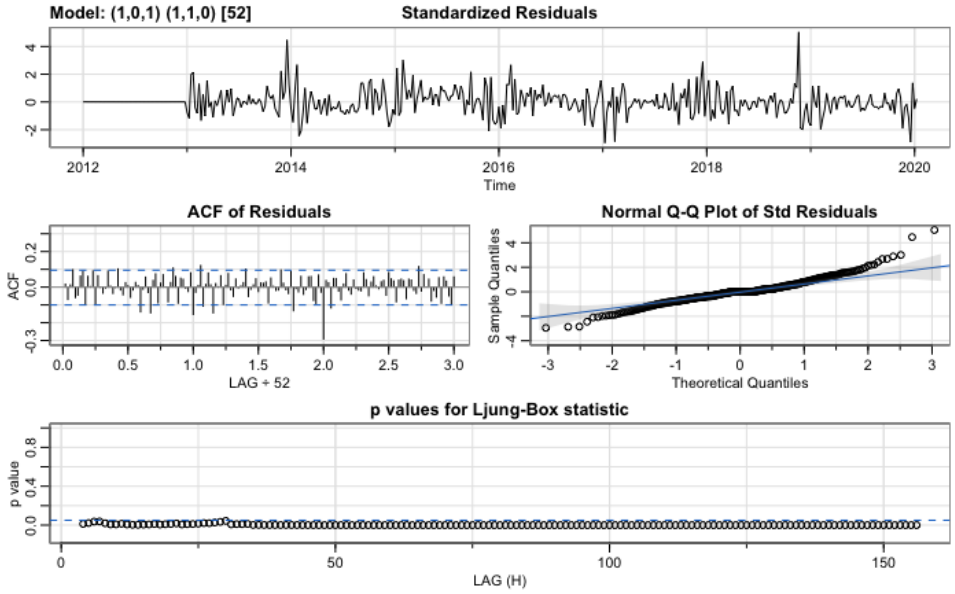

Figure 10: Residuals diagnostics of best CO model - SARIMA(1,0,1)(1,1,0)[52]

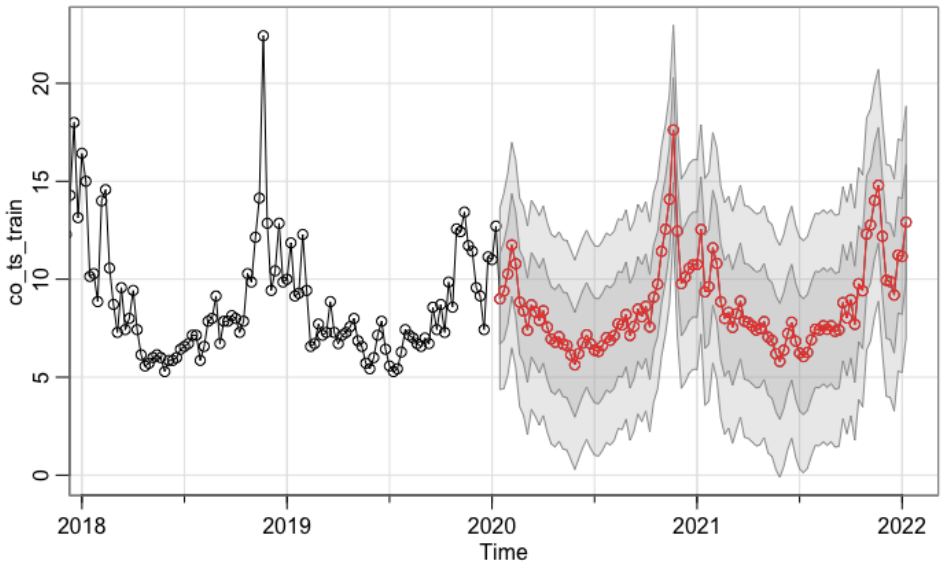

Figure 11: Forecasts of best CO model - SARIMA(1,0,1)(1,1,0)[52]

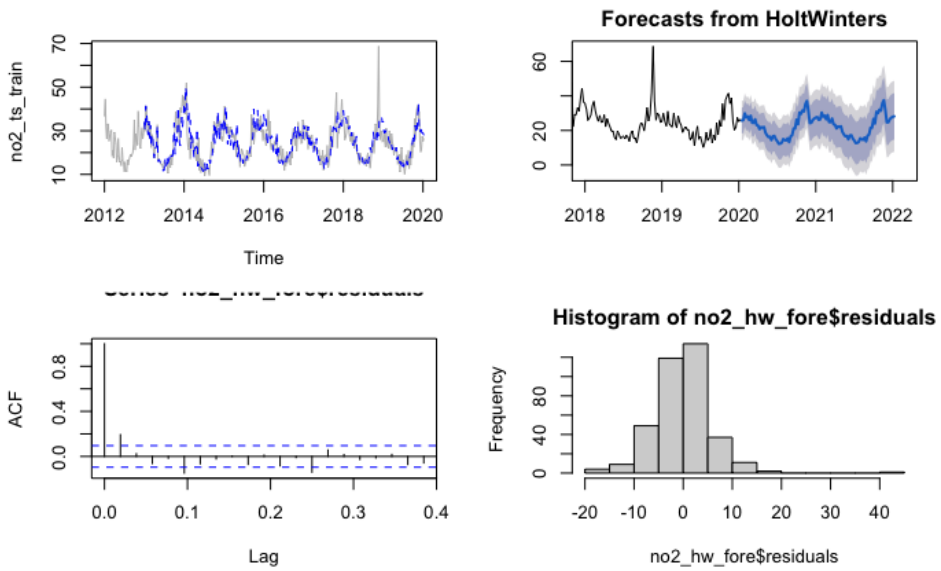

Figure 12: Diagnostics and forecasts of best NO2 model - Holt-Winters

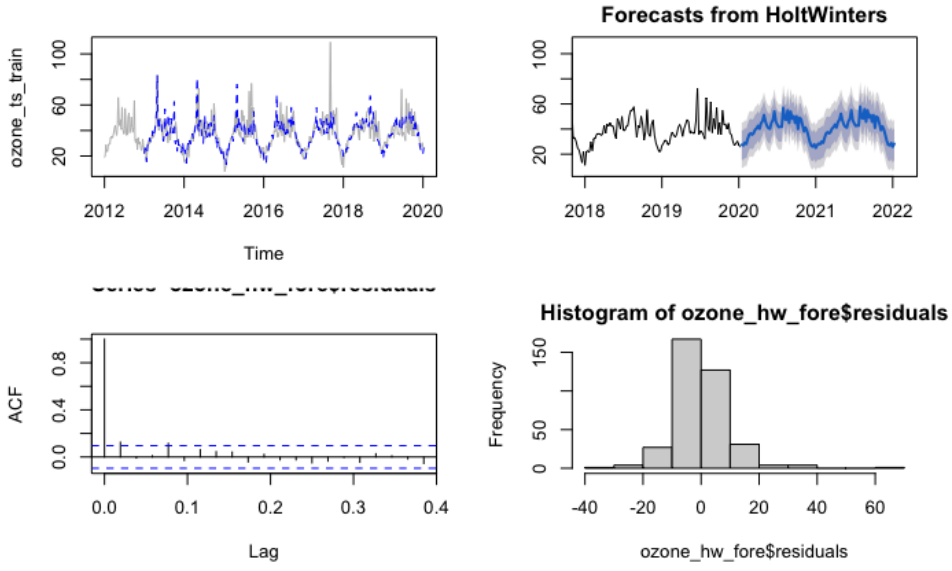

Figure 13: Diagnostics and forecasts of best ozone model - Holt-Winters

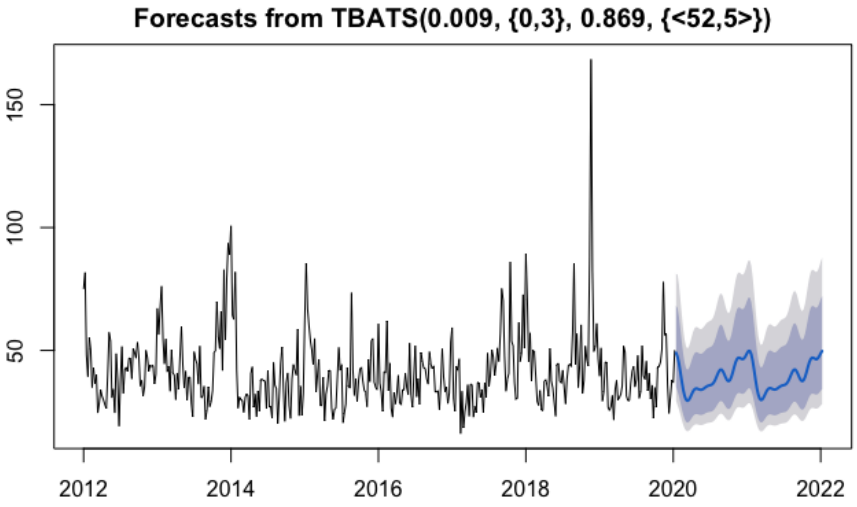

Figure 14: Forecasts of best PM25 model - Holt-Winters

6.4 Machine Learning

6.4.1 Hyperparameter Tuning

Table 5: Hyperparameters of best machine learning models

Model	Scaling	Timesteps	Hyperparameter Values
Lasso	Standard	5	Lambda = 0.04037
Ridge	Standard	3	Lambda = 0.37649
PCR	Standard	3	M = 6
PLS	Standard	3	M = 3
SVR	Min-Max	5	Kernel = RBF C = 100 Gamma = 0.01
Random Forest	Standard	3	Mtry = 3 Ntree = 3
XGBoost	Standard	10	Learning Rate = 0.1 Min. Child Weight = 4 Gamma = 10 Max. Depth = 3
GBM	Standard	5	Lambda = 0.07943 Interaction Depth = 2

6.4.2 PCR Results

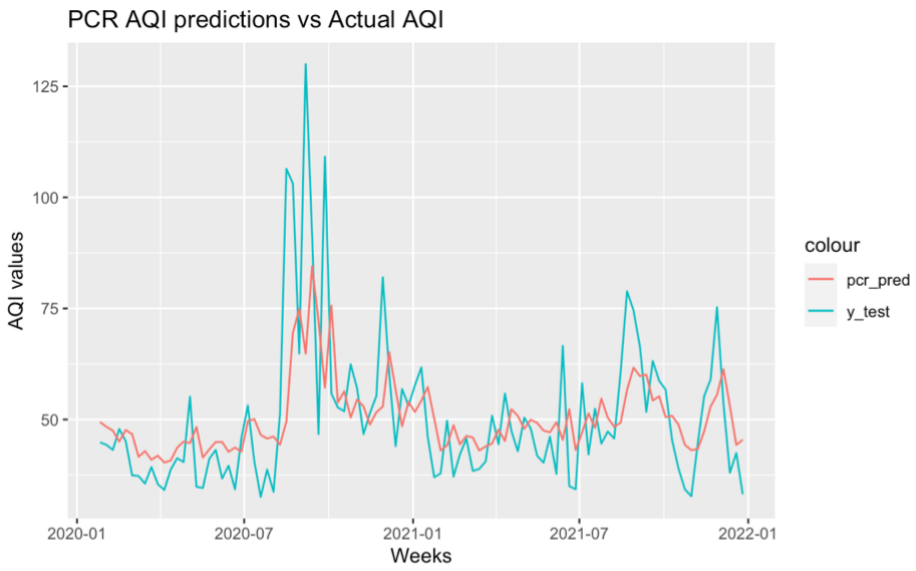

Figure 15: PCR test predictions plot

6.4.3 Boosting Results

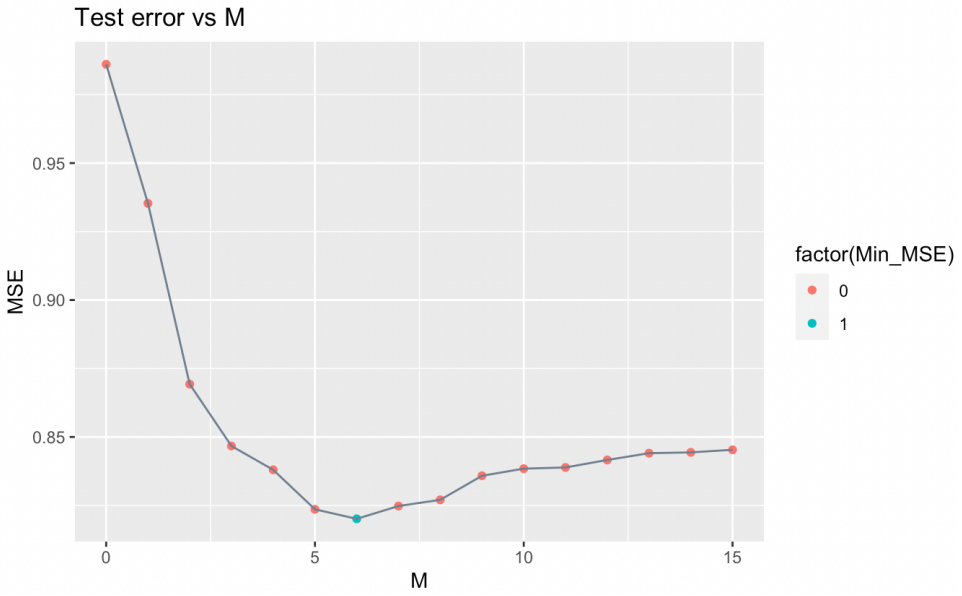

Figure 16: PCR cross-validation plot to tune M

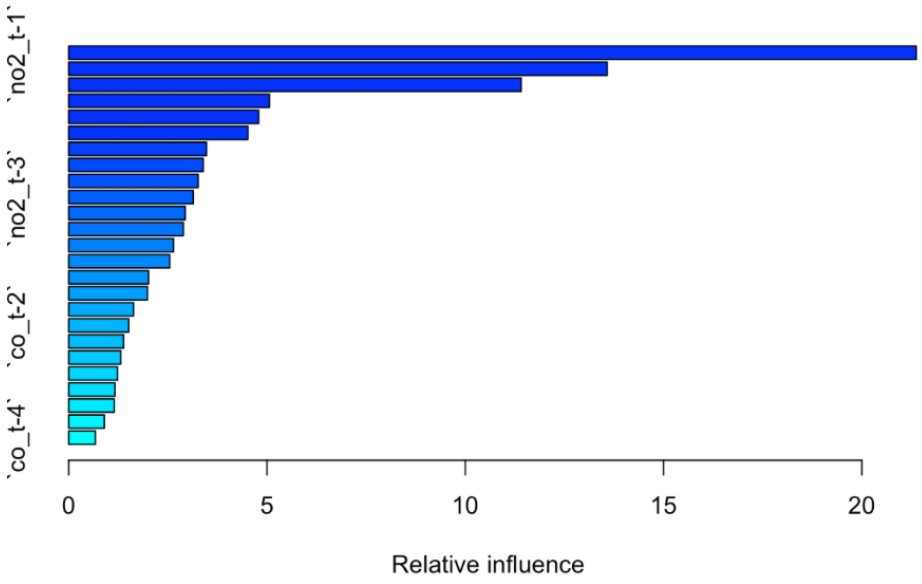

Figure 17: Variable importance plot from GBM

6.5 Deep Learning

6.5.1 Hyperparameter Tuning

Table 6: Hyperparameters of best deep learning models

Model	Time Aggregation	Scaling	Timesteps	Batch Size	Learning Rate	Epochs	Dropout
GRU	Weekly	Standard	3	64	0.001	50	0.5
LSTM	Weekly	Standard	5	32	0.001	50	0.1
CNN-LSTM	Weekly	Standard	7	32	0.001	50	0.7
TFT	Daily	Standard	5	32	0.03	30	-
DeepAR	Daily	-	5	64	0.001	30	-

Table 7: Hyperparameters of best deep learning models(continued)

Model	Weight Decay	Hidden Dim	Layer Dim	Window/Filter Size	Attention Head
GRU	-	128	2	-	-
LSTM	-	128	3	-	-
CNN-LSTM -	128	2	6	-	-
TFT	-	64	2	-	3
DeepAR	-	128	2	-	-

6.5.2 TFT Plots and Analyses

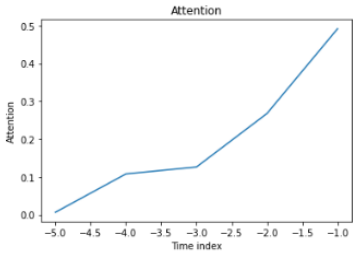

Figure 18: Average attention weights for predictions of best TFT model

Figure 18 shows the average attention given to past timesteps in making predictions at timestep t+1 of the best TFT model. On average we notice that the most attention for each prediction is given to the most recent t-1 timestep, and the attention decreases with time into the past.

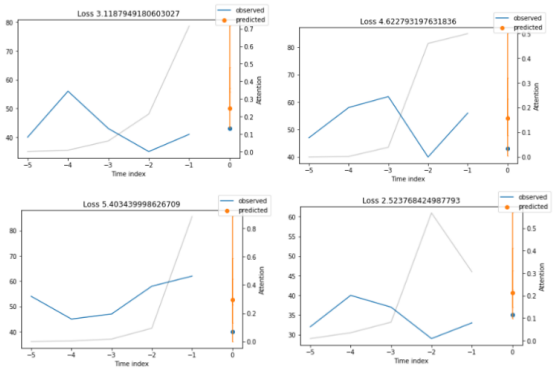

Figure 19: Attention weights for predictions of best TFT model

Analyzing predictions on the validation set, for example, we observe that the model usually pays more attention to most recent time steps. Additionally, we observe that the model pays more attention to the past time steps

when there are drops, sometimes overcoming the attention to the most recent timestep. This shows how the model could adapt its predictions to significant events such as drops in the AQI/pollutant levels.

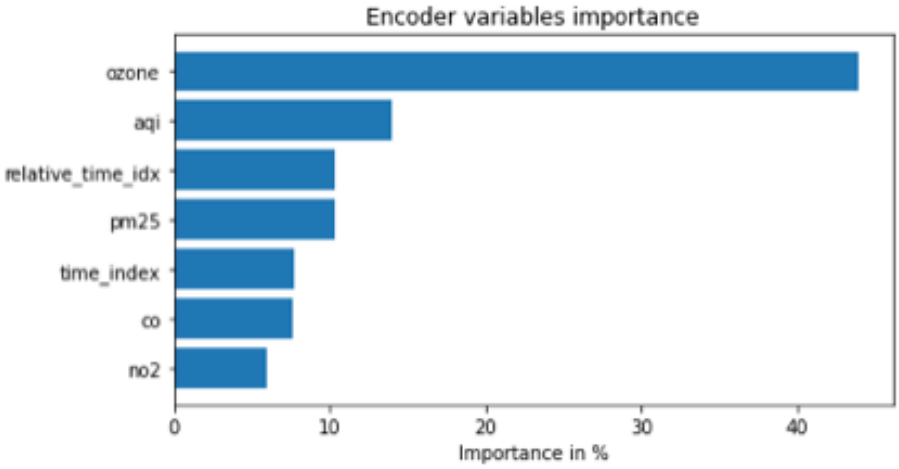

Figure 20: Variable importance plot of best TFT model

6.5.3 DeepAR Plots and Analyses

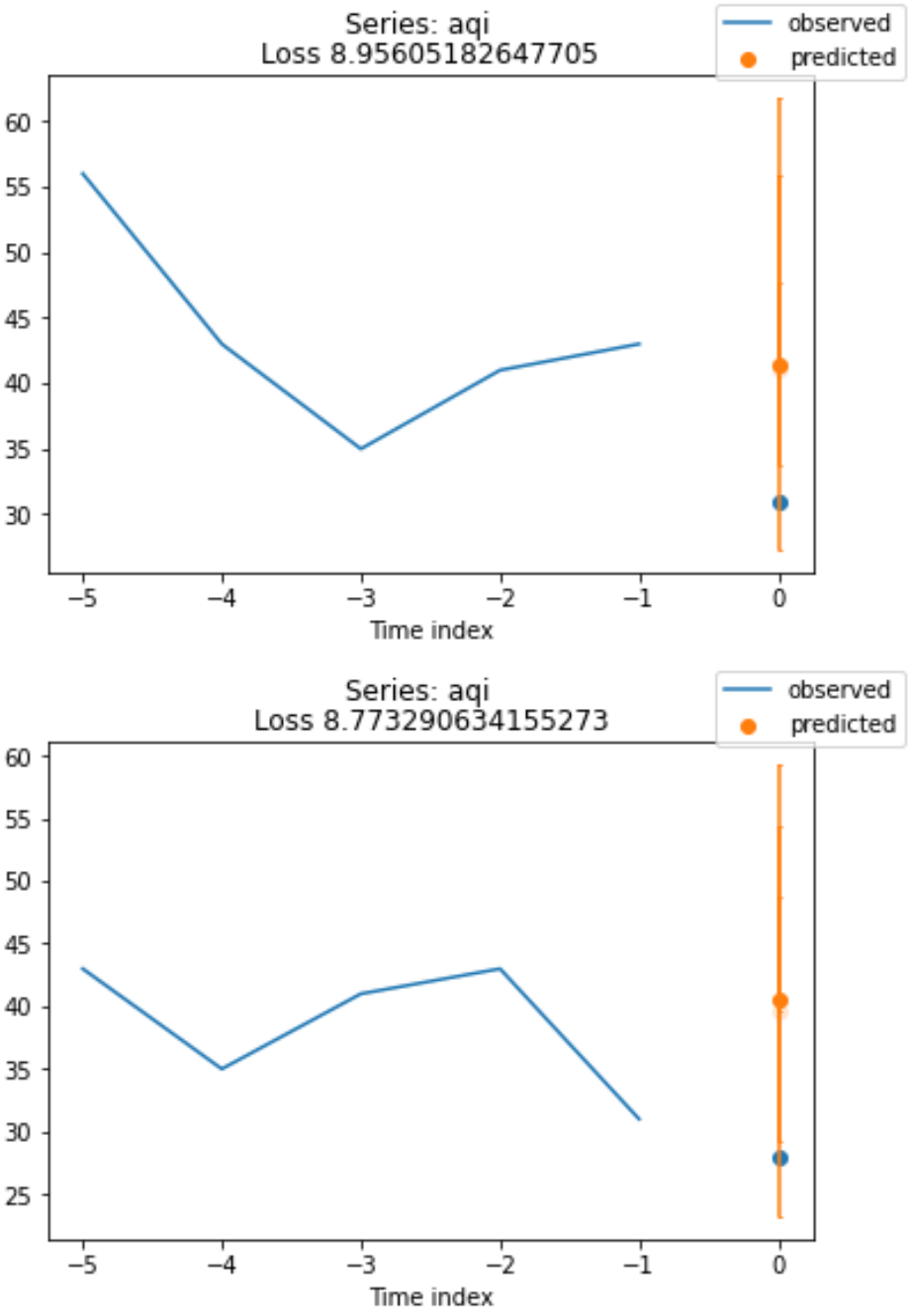

Figure 21: Sample predictions of best DeepAR model