Predicting Air Pollution in Beijing with Multivariate Time Series Modeling

Thomas Nguyen nguy3817@umn.edu
Jeffrey Jia jiaxx215@umn.edu
David Hwang hwan0259@umn.edu
University of Minnesota
CSCI 8523: AI for Earth
Github Link: https://github.umn.edu/NGUY3817/BeijingAirPollution

Abstract

Our goal with this paper is to create a time series model to accurately predict PM2.5 in Beijing, China. Prediction of PM2.5 is an important task, especially in areas that do not contain an air quality monitoring station. If PM2.5 can be predicted from other factors or generalized for an area in Beijing, then citizens can be alerted of dangerous levels of PM2.5. Time series classification was done with LSTM, Bi-LSTM, Transformer, and a Wide Transformer architecture using 48 hours of input to predict the next 6 and 24 hours. After creating the models, we found that Bi-LSTM was able to predict PM2.5 data on the test dataset with 24 hours prediction with a R2 of 0.575 and 6 hour prediction with a R2 of 0.570. MAE and RMSE for Bi-LSTM was also lower than the other models. In 24 hours and 6 hour prediction ahead, the Transformer model performed worst than Bi-LSTM and LSTM and achieved a R2 of 0.538, and 0.531, respectively. The wide transformer performed even worse with a R2 of 0.520, 0.516. These results suggest that transformer architectures may not be fit for PM2.5 prediction and Bi-LSTM is the best time series model to predict PM2.5 data.

1 Introduction

In China, one of the major health and environmental concerns has been pollution in the atmosphere. These air pollutants are small microscopic solids or liquids that remain in the air like dust, and soot which are created by sources from cars, power plants, industrial facilities which emit particulate matter into the air [10]. Pollution can be further quantified into particles of PM10 or PM2.5, along with other chemicals including sulfur dioxide, nitrogen dioxide, carbon monoxide, ozone [4].

PM2.5 has been set as the standard in China for pollution, which defines an aerodynamic diameter of less than $2.5\mu m$ [10]. In 2013, there were reports of PM2.5 levels in China that were 40 times higher than the world health organization (WHO) standards [7]. PM2.5 has been linked to various health concerns as long term exposure has been revealed to worsen lung function and cause severe medical diseases [4]. In 2015, PM2.5 exposure was reported to have caused around 3 million deaths in China [8].

In 2013, China enacted strict policies to improve air quality in response to rising PM levels. Industrial emission standards were strengthened, clean fuels were promoted, and vehicle emissions were regulated [6]. Multiple air quality monitoring sites were created across China with the goal of monitoring PM levels [10].

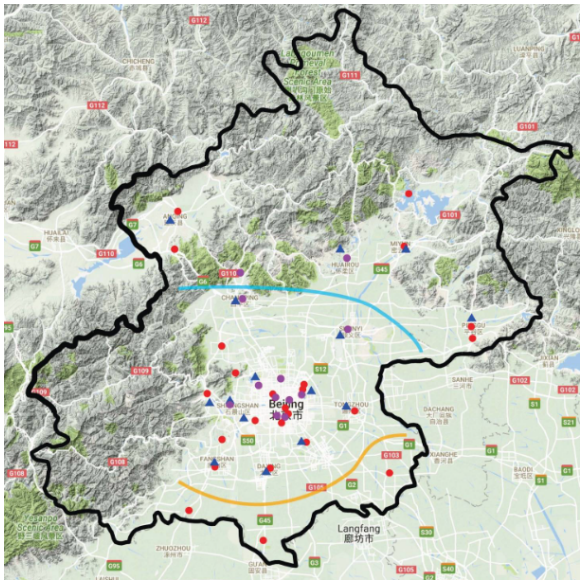

Figure 1: Air Quality Stations in Beijing (red/purple dots) and Meteorological stations (blue triangles). Purple dots mark the nationally controlled stations. The blue line denotes the north region and the orange line denotes the south region.

There are 12 nationally controlled air-quality monitoring sites in Beijing that report hourly readings for air pollution which consists of air pollutants that contain PM2.5 [12]. PM2.5 levels are

known to be affected by multiple factors such as the presence of other air pollutants, emissions, and also meteorological conditions. According to the US embassy in Beijing more than 70% of the variation in PM2.5 can be explained by Meteorological variables [13]. Thus our goal in this study is to utilize data from meteorological sites, which conduct hourly readings of the weather, combined with the data from the 12 nationally controlled air-quality monitoring sites to create a model to predict PM2.5. We utilize deep learning methods to create a time series classification model using the past two days as input to predict PM2.5 for 6 and 24 hours ahead.

Significant developments in the artificial intelligence space have been made with many deep learning architectures being created such as Recurrent Neural Networks (RNN), Long short-term Memory (LSTM), and Transformers. Deep learning methods have become more popular in forecasting over traditional methods like ARIMA and older machine learning algorithms like Random Forest and Support Vector Regression reporting better results [3]. As such, we utilize LSTM, Bi-LSTM, Transformer, and a Wide transformer (implementation explained later) in order to see what model best predicts PM2.5.

2 Related Works

Forecasting specifically with air pollution is a complex problem in which machine learning methods and deep learning methods have been previously applied. Using air pollution data in Seoul, South Korea the authors developed LSTM and autoencoder models to predict PM2.5 [14]. It was found that LSTM performed the best which had RMSE of 12.174 over the autoencoder model which had RMSE of 15.431. Other articles also suggest that LSTM performs the best in time series classification. In another study, the authors use ARIMA, CNN, FBProphet, and LSTM to predict PM2.5 in the environment. From their results, they found that LSTM outperformed all the other models in terms of MAE [11].

Deep learning methods were also conducted on the data in Beijing, China. The authors developed models on a wide variety of architectures such as GRU, CNN, Bi-GRU, CNN-LSTM, CNN-GRU, LSTM, Bi-LSTM. It was found that the Bi-LSTM model with RMSE of 8.947 outperformed all other models including the LSTM model with RMSE of 9.102 [9].

3 Methods

In our study, we implement the most successful time series methods from previous works with the addition of the transformer architecture, which was introduced in 2017. We describe the methods we use to predict PM2.5 which are Baseline, LSTM, Bi-LSTM, Transformer, and Wide transformer in this section.

3.1 Baseline

For the baseline methodology, we implemented a method that would make a prediction without the use of machine learning. We decided to take the average PM2.5 concentration value of the previous 48 hours and use that as the prediction for the next hour. This process was continued for either 6 or 24 strides/hours depending on how many hours ahead we planned to predict the PM2.5 concentration. The point of this baseline prediction is to see how our multivariate time series models perform compared to a naive method without machine learning.

3.2 LSTM

An LSTM (Long Short Term Memory) model is a variation of an RNN with the ability to learn relations between data over long periods of time using sequential data. LSTMs are commonly used in different domains, such as natural language processing and time series problems because the ordering of words or time matters in certain problem spaces.

The unique characteristics of LSTMs are cells that allow the LSTM to remember values over time, which are constantly propagated through various gates called the input, output, and forget gates. Input gates tell the cell which information from the current batch can be used within the network or which to pay attention more to than others. The output gate controls the information inside this cell and determines which values move to the next hidden layer. The forget gate decides which information from within the cell is to be discarded and which values from the prior time steps are needed by the cell.

3.3 Bidirectional LSTM

A Bidirectional LSTM, or Bi-LSTM, is a sequence processing model that consists of two LSTMs. The first LSTM takes in the input in the usual beginning-to-end manner, and the second LSTM takes in the input in an end-to-beginning manner. The Bidirec-

tional LSTM effectively increases the amount of information available to the network, which should create a more robust model than LSTM.

3.4 Transformer

LSTM and Bi-LSTM are common deep learning models to perform time series classification. With the release of transformers it provided a new architecture to improve over the previous models. Transformers was first introduced in June 2017 by Google originally for machine translation tasks but has been fast adopted to other applications for computer vision, natural language, and sequential input data. Transformers uses a sequence-to-sequence architecture that utilizes both an encoder and a decoder.

The encoder converts the input sequence into a set of representations and then the decoder takes this latent space and converts it into the prediction. This architecture has been seen before with LSTM and RNN; however it has a couple of major differences. The input sequence is processed non sequentially as the input sequence is taken all at once instead of being taken one after another. This is possible due to the self-attention mechanism that is unique to transformers and allows it to create an embedding of each input in the sequence which takes into account the previous and future inputs that are most important which is added to a positional embedding that takes into account where the input is in the sequence. This is in contrast to LSTM, as they only take into account previous inputs in the sequence. Although Bi-LSTM also uses previous and future inputs when computing an embedding, the transformer has been argued to be able to better contextualize the input through what is called multi-headed self-attention [1].

The transformer uses a stack of parallel attention layers called multi-headed self-attention that helps with parsing the sequences and keeps long-term dependency between inputs which LSTM and Bi-LSTM struggle with. Multi-headed attention is calculated to improve the performance of the attention layer. In our transformer, we utilize four heads. This allows the model to better understand the relevance of other words in the sequence and add more Query, Key, Value weights which creates a more robust representation from the embedding. Using four heads would create four different self-attention matrices which would need to be merged together to one self attention matrix. This can be done by concatenating all the attention heads together and multiplying them by a weight matrix to get the final attention matrix which is sent to the feed-forward neural network which computes a representation of the sequence that feeds into the encoder [1]. In our implementation, we stack four encoder blocks together.

The decoder we implement deviates from the original implementation as it only consists of a pooling layer and a feed-forward layer followed by a fully connected layer, essentially a pooling layer followed by two dense layers. This takes in the output from the encoder and then predicts the PM2.5 for 6 or 24 hours.

3.5 Wide Transformer

In order to create a comparison against the base transformer, we have also developed another transformer model that we call the wide transformer which doubles the number of heads and encoder blocks but leaves the rest of the architecture unchanged. This would in essence allow for a more robust model and allow us to compare our results to the transformer model. The boosted transformer is expected to have better results than the regular transformer model.

3.6 Ensemble

To obtain a generalization of the PM2.5 level in Beijing we average or ensemble all the predictions from the individual stations. Ensemble learning utilizes the multiple model's predictions to form a singular prediction. An advantage of using ensemble learning is that it utilizes all the data from the individual stations, thus allowing us to gain a more generalizable prediction over Beijing. The downside of using an ensemble is that instead of training one model, we are required to train over 12 models which is a substantial increase in training time.

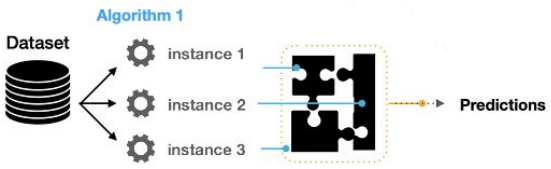

Figure 2: Architecture of Ensemble Learning where instances are our models.

4 Experiments

4.1 Datasets

Beijing Multi-Site Air-Quality Data Set

The dataset we use for training our models takes PM2.5 data from 12 nationally-controlled air-quality monitoring sites in Beijing and the meteorological data from the weather station closest to each of the 12 monitoring sites. Meteorological data consists of air temperature, wind direction and speed, pressure, relative humidity, and precipitation. The time period of the dataset ranges from March 1st, 2013 to February 28th, 2017.

Airnow Dataset

The Airnow dataset takes in PM2.5 data from the US embassy complex in Chaoyang district, Beijing. We use this dataset to see how well our ensemble performs to generalize PM2.5 data to the region of Beijing. We match the predictions for PM2.5 made from the ensembles to the PM2.5 values in the Airnow dataset.

4.2 Preprocessing

To preprocess the datasets, we took each of the datasets from the 12 different monitoring sites and individually preprocessed them. In the preprocessing step, we needed to ensure that we had all our data in numerical form and that we did not have any missing values. Each of the 12 original datasets had wind direction as cardinal (N, E, S, W), ordinal (NE, SE, SW, NW), and secondary intercardinal (NNE, ENE, ESE, SSE, SSW, WSW, WNW, NNW) directions so we had to change them to numerical values ranging from 0 to 360 each separated by 22.5 degrees before feeding the data into our machine learning models.

After examining the datasets, we also noticed some missing data for each monitoring site. The missing data on average only made up about 4% of each dataset and was largely scattered throughout the dataset, so we decided to use a spline linear interpolation method to fill in the missing data. We decided to split our dataset into training on the first three years and testing on the fourth year.

We had a hypothesis that the season would have a major effect on the PM2.5 concentration, so we decided to also create four different models, one for each season of winter, spring, summer, and fall, to test if it would produce better results. This task requires splitting the dataset from each station into four smaller datasets containing data from a specific season. We denote March, April, and May as Spring; June, July, and August as Summer; September, October, and November as Fall; and December, January, and February as winter.

We train each model to predict 6 hours and 24 hours ahead using the last two days. Thus, we create a dataset where every datapoint contains the last two days of input with both PM2.5 data and meteorological data and the next 6 or 24 hours of PM2.5 data. We split the Beijing Multi-Site Air-Quality data set into a 75% train/25% test. Thus, data is trained from 3/1/2013 to 3/1/2016 and tested from 3/2/2016 to 2/28/2017.

4.3 Fine-tuning details

For all of our models, we use the same hyperparameter settings. Each model is compiled with RMSE as the loss function and Adam as the optimizer with a learning rate of 1e-4. The models were fit to 50 epochs, batch size of 72, and validation split of 20% of the training data employing early stopping. Early stopping stops the model if validation loss does not improve over 10 epochs. We generally found 50 epochs was sufficient for all the models.

4.4 Evaluation Metrics

Models are created for each of the 12 stations for all of our methods (LSTM, Bi-LSTM, Transformers, Wide Transformer). We take in input from the test set and either 6 or 24 hours are predicted from that time step. We average all predictions from the same time step and compute metrics from our predictions to the actual values of that station. Then the metrics are averaged across to get the results for each method. Metrics are also computed on the Airnow dataset we ensemble each of the methods by averaging all of the predictions across the stations and compute the metrics from our ensembled prediction to the PM2.5 values of the Airnow dataset.

Mean Absolute Error (MAE)

$$MAE = \sum_{n=1}^{n} \frac{|\hat{y}_i - \overline{y}|}{n}$$

MAE measures the average of the absolute difference between $\hat{y}_i$, predicted values, and $\overline{y}$ the sample mean. The value we get from MAE tells how off our predictions are on average.

Root Mean Squared Error (RMSE)

$$RMSE = \sum_{n=1}^{n} \sqrt{\frac{(y_i - \hat{y}_i)^2}{n}}$$

RMSE measures the square root of the squared difference of the actual and predicted values. RMSE ranges from 0 to infinity with a value of 0 showing no difference from the predicted and actual values and larger values showing more variation between the actual and predicted values. RMSE, unlike MAE, penalizes large errors so RMSE would be higher than MAE in the case that frequently predicted values are farther off from the actual values than expected.

R-Squared (R2)

$$RSquared(R^2) = 1 - \frac{\sum_{n=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{n=1}^{n}(y_i - \overline{y_i})^2}$$

R2 measures the proportion between the sum of squared regression over the total sum of squares. This metric gives information on how much variance is accounted for in the model and how well our model predictions fit to our actual values. Usually this is between values 0 and 1 however there are cases of values being negative if the model does not fit the actual values.

5 Results

5.1 Air Quality Dataset Metric Analysis

Our team wanted to see the effect of predicting 24 hours into the future compared to 6 hours in the future. According to our metric tables 1 and 2 for the whole Beijing air quality dataset, we found that predicting fewer hours in the future resulted in lower MAE, RMSE scores, and higher R2 scores. Lowering the prediction window from 24 hours to 6 hours improves the MAE and RMSE by about 40%. It seems we are predicting too far into the future for our models. We infer that if we were to predict a smaller prediction window of 1 or 2 hours into the future, we would be able to get more accurate predictions of raw concentration. In both the prediction of 24 hours and 6 hours, we saw that the Bi-LSTM performed the best of all the models. Compared to the baseline model, all other models had better RMSE, MAE, and R2 proving that our models are learning during training to be able to predict PM2.5 concentration.

5.2 AirNow Data

To test the model's generalizability, we wanted to find another real-time dataset near Beijing but not covered by one of the 12 weather stations. We grabbed the same historical time frame that we used in our Beijing Multi-Site Air-Quality test Dataset (March 2nd, 2016 - February 28, 2017). We concatenated two datasets together—the 2016 AirNow dataset from March to December and the 2017 AirNow dataset from January to February. This replicates the test dataset in the Beijing Air Quality dataset. The AirNow dataset was collected in the Chaoyang district located near the heart of Beijing. Since we wanted to achieve an idea of the overall PM2.5 concentration in Beijing as a whole, we wanted to compare the US Embassy's historical record of PM2.5 data to use as a ground truth representation of Beijing as a whole. If we can train our models on the 12 weather stations, we will have a better idea of whether this is enough data to predict the overall PM2.5 concentration in Beijing altogether.

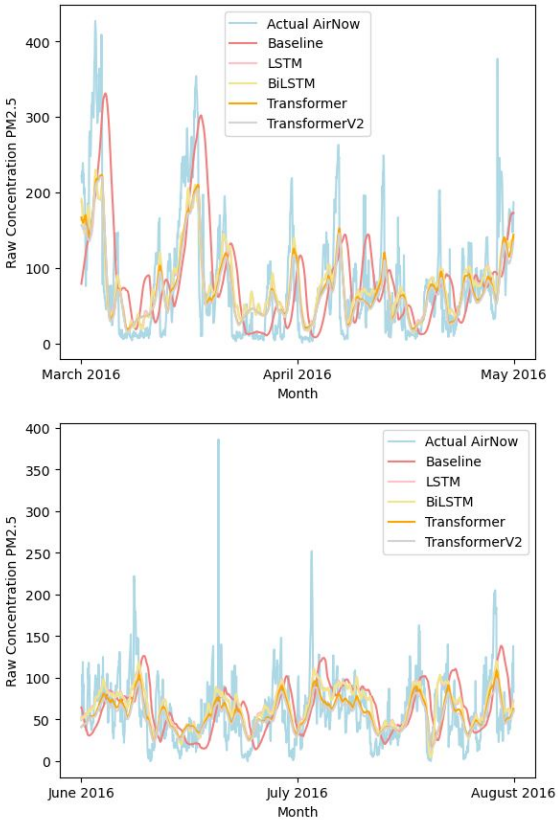

Figure 3: March 2016 - August 2016 All Models

Firstly looking at the metrics at table 3 and 4, we noticed the same trend of a performance boost of around 40% is evident when reducing the predic-

Test	Baseline	LSTM	Bi-LSTM	Transformer	Wide Transformer
MAE	55.437	36.065	35.960	37.251	37.779
RMSE	78.873	53.560	53.449	55.670	56.715
R2	0.072	0.573	0.575	0.538	0.520

Table 1: Metrics on the test set using previous 48 hours to predict next 24 hours

Test	Baseline	LSTM	Bi-LSTM	Transformer	Wide Transformer
MAE	48.234	22.635	20.937	21.246	22.057
RMSE	70.783	33.815	33.638	34.282	35.973
R2	0.253	0.830	0.832	0.825	0.806

Table 2: Metrics on the test set using previous 48 hours to predict next 6 hours

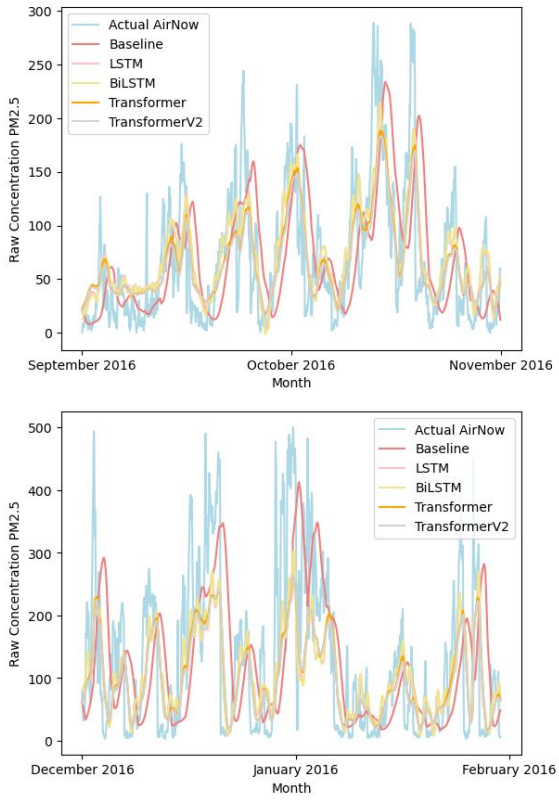

Figure 4: September 2016 - February All Models

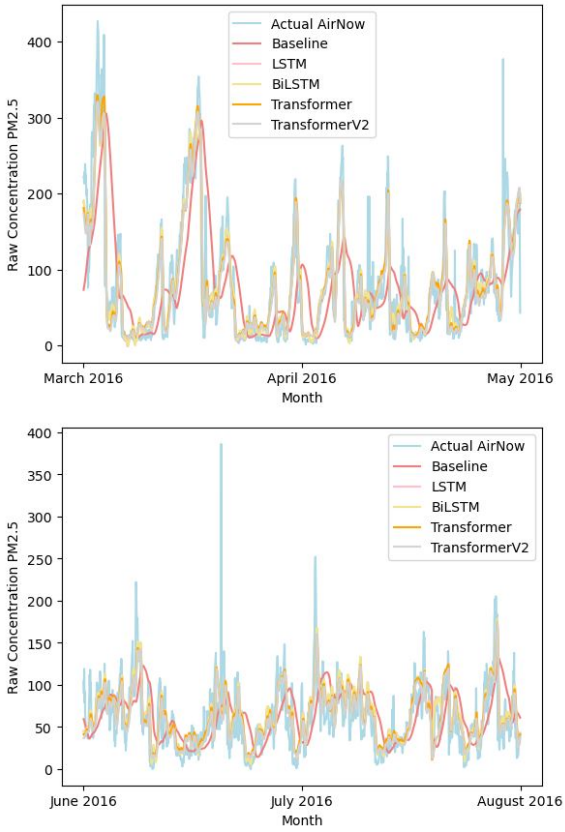

Figure 5: March 2016 - August 2016 All Models

tion window from 24 hours to 6 hours, in addition with performance increases when comparing deep learning models to our naive baseline approach. For visibility reasons we decided to graph the first 2 months of every season on the x axis with the PM2.5 concentration on the y axis. In the paper we included the graphs for 24 hour predictions on figure 3 and 4 going from March 1st to February 1st. We also did the same thing for the 6 hour predictions located at figure 5 and 6. We have additional graphs located in the appendix showing 24 hour and 6 hour predictions for AirNow vs Baseline, AirNow vs LSTM vs Bi-LSTM vs Transformer, and AirNow vs Transformer vs WideTransformer (TransformerV2).

Looking at both figure 3 and 4, the baseline, which is our naive prediction method, is essentially a running average of the previous 48 hours' PM2.5 concentration levels. The figure reveals that this naive baseline method of predicting PM2.5 is ineffective. The LSTM and Bi-LSTM tend to follow the same pattern as the other, and the Transformer and Wide Transformer models tend to follow a sim-

Test	Baseline	LSTM	Bi-LSTM	Transformer	Wide Transformer
MAE	54.700	37.020	36.532	38.018	38.294
RMSE	84.103	57.095	56.690	59.211	60.060
R2	0.111	0.564	0.570	0.531	0.516

Table 3: AirNow Metrics on the test set using previous 48 hours to predict next 24 hours

Test	Baseline	LSTM	Bi-LSTM	Transformer	Wide Transformer
MAE	48.234	22.635	22.534	22.869	23.821
RMSE	76.567	39.129	38.830	39.326	41.530
R2	0.263	0.795	0.798	0.793	0.769

Table 4: Air Now Metrics on the test set using previous 48 hours to predict next 6 hours

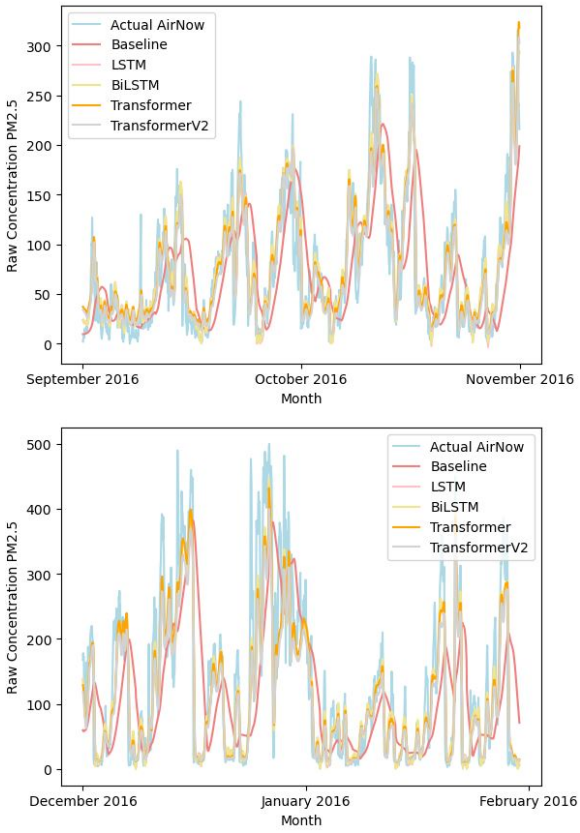

Figure 6: September 2016 - February All Models

ilar pattern as well. Through observing the graphs, we notice Bi-LSTM follows the actual values the best since it is able to predict higher PM2.5 levels better than the other models.

When looking at the various graphs, there is high variability in the data which results in our models being unable to predict PM2.5 accurately. This could be because our models weren't exposed to enough training data with high spikes above 300 PM2.5. Because of this, we believe 3 years might be insufficient training data to fit the model. To put this into context, in our test dataset, there exists seemingly random spikes in PM that have a PM2.5 value close to 400 and then drastically lowers the next day such as in the end of June 2016 of figure 17 in the appendix. There is a lot of unknown factors that could have caused the PM2.5 values to sky rocket or plummet. Ultimately, these spikes are hard to capture by our models and result in lower performance.

5.3 Weather Station Exclusion

As a group, we thought that the location of the stations could play a key role in determining our predictions. Since there are 4 air quality stations that we include that are hovering around the mountainous area of Beijing while most of the stations are located in central Beijing. The four weather stations that are located in this mountainous regions is Changping, Dingling, Huairou, and Shunyi. We hypothesized that the mountainous setting could play a role where PM2.5 readings in those areas could be significantly be lower than those in central Beijing. Since the mountainous areas has a different terrain than that of central Beijing it is a possibility that PM2.5 could be drastically different.

We decided to retrain our models excluding these 4 weather stations. We decided to test only with LSTM to see if there were any significant changes in our results. We decided to compare the standard LSTM model versus the LSTM that excludes certain stations for 24 hours prediction.

Based on the results above, the standard LSTM without exclusion had a RMSE of 57.095 and MAE of 37.020. LSTM with exclusion performed similarly with MAE of 37.168 and RMSE of 56.284. This suggests that the setting and location of where these stations are located do not generate more or

Metric	LSTM	Exclusion (LSTM)
MAE	37.020	37.168
RMSE	57.095	56.284
R2	0.564	0.576

Table 5: LSTM vs Exclusion (LSTM) on various metrics

less PM2.5 than the locations at the center of Beijing. Perhaps the air quality in the center of Beijing and around the mountainous stations are very similar or identical to allow similar PM2.5 predictions with or without them. Looking at the meteorological data, for some of the dates, we noticed that some of the weather conditions are very similar with one another. So another possibility is that the weather stations selected for the air quality station could be closer then the actual distance between the air quality stations.

5.4 Season Analysis

From prior research, our prior experiments, and intuition we expected that seasons would have a major effect on PM2.5 concentration. This is due to PM2.5 pollution having a high correlation with temperature and dew point which changes with the seasons. We decided to take our best performing model thus far, the Bi-LSTM model where we predict the PM2.5 concentration for the next 6 hours, and create four smaller Bi-LSTM models each corresponding to a season. Taking a look at our graphs in Figure 7, it appears that the season specific models appear to follow a similar trend as the full year model, which does make sense since both temperature and dew point are inputs to our model. However, our quantitative results told a different story that was quite interesting. We found that both the MAE and RMSE decreased significantly on the models specifically trained for the Spring, Summer, and Fall seasons. We hypothesis that the improvement in the Spring, Summer, and Fall models can be attributed to the fact that the training and testing datasets are more homogeneous and allows the season specific models to learn standard seasonal patterns and trends. On the contrary, the model trained to handle the Winter season actually performed worse than the model encompassing the entire year. I think we can attribute this to Winter being a more unpredictable season with snow and the largest temperature range depending on the year. However, taking a look at the table we did have some results relating to summer that confused us. The Summer model had the best MAE and RMSE scores but also had the worse R2 score at 0.472. We think this error is likely due to unexpected value in our dataset or a bug in our code but we could not find the source of this issue.

5.5 Wide Transformer Model Analysis

Since the base transformer model performed the worst over LSTM and Bi-LSTM we decided to explore if the transformer implementation was not robust enough to fit to the training data. Thus, we created the wide transformer model which contains double the amount of self-attention heads and encoder blocks.

From the results we find that the wide transformer performs the worst over all the models in the test set and airnow dataset. From the test dataset the MAE is 37.779 and the RMSE is 56.719 which are worse than the base transformer with MAE 37.251 and RMSE 55.670. We can see from the R2 results that the wide transformer performs worse than transformer as R2 is 0.520 compared to 0.538 of the transformer. We can also see that metrics from the airnow dataset are worse as the wide transformer R2 score is 0.516 and the R2 score on the airnow dataset on the base transformer is 0.531.

It is interesting to see that the wide transformer model performs worse than the base model. This seems to suggest the transformer implementation or self-attention itself are not the correct fit for a time-series model. More research will need to be done on this topic to see what is the exact issue here but there could be a multitude of reasons why the predictions are worse.

6 Conclusions

In this project, we were able to obtain predictions on PM2.5 concentration for 6 hours and 24 hours in the future based on the 48 hours prior using LSTM, Bi-LSTM, Transformer, and Wide Transformer models. Unsurprisingly, when only predicting the PM2.5 levels for 6 hours in the future, we were able to achieve better results across all multivariate time series models. We can attribute this to PM2.5 concentration being affected by numerous environmental factors including but not limited to temperature, dew point, wind speed, and wind direction. These environmental factors vary significantly throughout 24 hours and become more difficult to predict the further in the future we go. We found all of our multivariate times series mod-

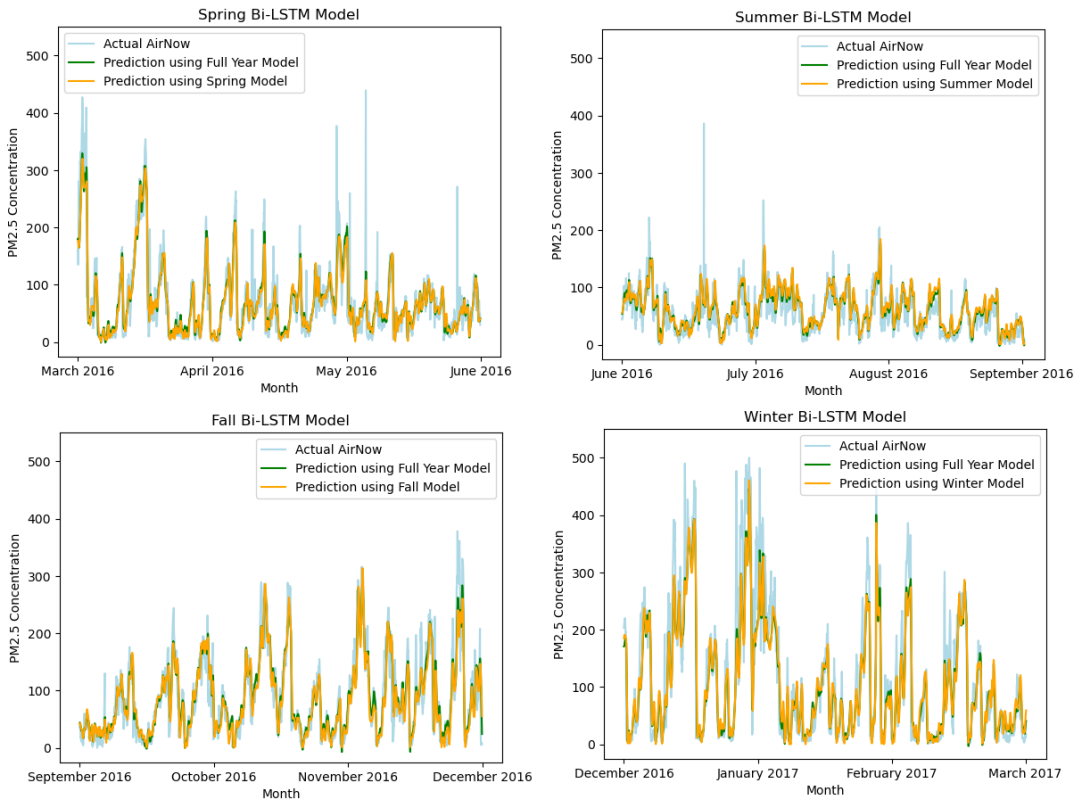

Figure 7: Seasonal Bidirectional LSTM Models

Metric	Full Year	Spring	Summer	Fall	Winter
MAE	22.534	19.990	19.097	22.980	32.881
RMSE	38.830	31.576	26.245	33.736	50.517
R2	0.798	0.795	0.472	0.783	0.818

Table 6: Seasonal Bi- LSTM models on various metrics

els outperformed the baseline method. Comparing the evaluation metrics of all our models, we discovered that the Bidirectional LSTM model performed the best overall, achieving Mean Absolute Error(MAE) of 22.534, Root Mean Square Error(RMSE) of 38.830, and R-squared (R2) of 0.798. The LSTM model came in at a close second, achieving MAE of 22.635, RMSE of 39.129, and R2 of 0.795. Both transformer models did not perform as well as the LSTM and Bi-LSTM on PM2.5 concentration prediction, suggesting that transformers are not a good fit for time series forecasting. Our exclusion method could lead one to assume terrain has little to no effect on the PM2.5 concentration as when we removed the stations in the mountainous areas of Beijing the prediction accuracy remained virtually unchanged; however, our exclusion method was not comprehensive and there could be some underlying factors that can cause different geographical areas to have abnormally higher or lower PM2.5 concentration values. When building a model for each season independently, we found that the Spring, Summer, and Fall models achieved more accurate predictions, while the Winter model actually performed worse than the model that encompassed the whole year. Overall, we found that the best model to predict PM2.5 was the Bi-LSTM. Predicting 6 hours in the future yields an MAE that improves by nearly 40% (36.532 -> 22.534) over predicting 24 hours ahead, and that seasons likely do have an effect on the levels of PM2.5 in the air.

7 Limitations

During our time together working on this project, we were able to try out a couple of different methods to see how close our predictions were to the actual measure of PM2.5 in Beijing. Upon working

through this problem, we realized that were a couple of limitations restricting us from getting better results. We feel if we had a lot more time, we could get the following future work implemented.

7.1 Chinese Policy Advances

China is a growing populous nation with indications in the news for improving air quality daily with new technological advances (solar power, electric vehicles, etc.). With new policy changes implemented after 2013 to improve PM2.5 level. it's hard to account for certain events in the models such as non-quantitative events that are currently happening in the world, such as EV policies will reduce PM2.5 concentrations by about 18.8% in Beijing by the year of 2030 [5]. We have thought of putting weights on the time frame when these policies have been implemented, but then we would introduce more bias to the later years to come and neglect earlier years when producing predictions. Having a bias toward the later years would capture more attention to what the current policy is, but would ignore other natural phenomena that would increase or decrease the raw concentration. These phenomena would potentially include what season we are in and how much rainfall affects PM2.5, among others.

7.2 Dataset

Another limitation of our project is the limited data we could find for our dataset. We were able to use meteorological data along with PM2.5 as input features to our models; however, there are many factors that affect PM2.5 levels rather than just meteorological data. Factors like the natural terrain (mountains, bodies of water, etc.), factories near the station, or the station being close to a highway producing emissions. Finding hourly data that contained these extra features or metadata besides meteorological data was extremely difficult, and we could not obtain access to the few we could find even after we tried emailing the providers for access.

8 Future work

In the future, we would want to explore adding weights to our inputs in our model to see how our "policies" or location affect the overall outcome of the data. For example, the DingLing station is near mountains, so we could potentially add weights to the model specifically for this air quality station. Adding weights to location specific weather stations would be an interesting plan, but a tedious matter to tackle without any formal way to assign weights for each policy or each location the station resides in. Therefore, we would need to look deeper into this topic and experiment with different methods of weighing the stations differently.

8.1 Parameter Tuning

Another work we could do in the future is performing cross-validation on our dataset to see how our model handles the dataset without taking formal seasons into account. By performing cross-validation, we could effectively use all our data and predict the different years or parts of years instead of just the last year. This, in turn, may make our model more accurate by being able to generalize on different times in the dataset.

8.2 Visualization

Lastly, we would have liked to create a mini-map of our air quality stations on a single graph and plot a heatmap to see which station was more polluted with PM2.5 than others to show if location and terrain matter when during a year or even an individual season. We would also be able to see which weather station was more biased towards during the averaging amongst the weather stations. Below is a graph that [2] plotted out when using linear regression to predict pollution in Beijing.

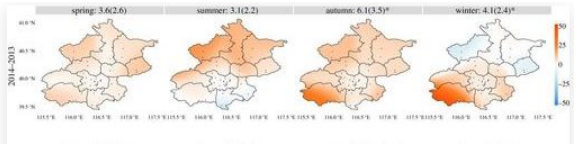

Figure 8: HeatMap intentional example

9 Reflection/Contribution

9.1 David Hwang

In the project I implemented the Bi-LSTM, LSTM, transformer, and boosted transformer models in keras. I also decided on which evaluation metrics to use with the models and trained and tested the boosted transformer model. Through this project I learned that there were a lot of aspects where we were lacking in the project, especially more so with the conceptualization of our idea. In the initial phase of our project we were just thinking of just comparing Bi-LSTM, LSTM, and transformer

models on the dataset and comparing the results together, however through the interim presentations and the final presentation we found that there were several holes in our project. We decided to add a more concrete baseline without any deep learning to ensure that our models are actually effective. We also decided to use our Bi-LSTM, LSTM, and transformer models on a province in China to see how well our models actually perform when predicting PM2.5 not in our dataset. There were definitely more aspects that I did not talk about but I definitely learned a lot about how to structure a project. Through creating the models and debugging the models I definitely learned more about how to create deep learning models especially for time series classificaiton which I have never worked with before. Also through running the boosted transformer model I learned how long it can take to train and test on the dataset. Just with the boosted transformer alone it took around a day to get all the results. Altogether, although there were still more things that I wanted to do with the project that were limited by time and knowhow I feel like we accomplished the goals that we set out for this project.

9.2 Thomas Nguyen

In this project, my main contributions were creating the Ensemble learning method, training and evaluating on the LSTM and Transformer models using various input and output values (input = 48 hours, outputs = 24hours, 6hours) multiple times, model creation, helped with implementing the seasonal predictions of Beijing, post-processing of resulting graphs and analysis on the Beijing Airquality dataset and AirNow dataset, and helped with preprocessing, along with the brainstorming and ideation along with creation of the baseline method without machine learning. The Transformer took 125 minutes total (10 minutes to train for each weather station) each time we needed the Transformer ran. LSTM took 30 minutes to train fully for all 12 weather stations.

I learned a lot from this project. It really made me think outside the box when it comes to the data given. From my experience in data science and machine learning, we have just been given a dataset from Kaggle (or a precollected dataset) and perform some machine learning algorithms and then explain our results. Usually from the data alone I would be able to come to conclusions or find patterns in the data that can allude to certain behaviors the model outputs. However, that isn't the case for this project. For this project, our dataset features were limited to meteorlogical data and the PM2.5 concentration for that period in time. When our results came through, I couldn't go off the features alone as to why our results are what they were. I had to really think on a societal scale of Beijing as to what they are experiencing during that time. Whether that is the different seasons, new policy changes to how to combat pollution during that time, the setting or terrain visible around the weather station, or even just natural spikes in PM2.5 randomly. There is a lot more features to consider that isn't included in our dataset that we had to be mindful of, so it was a fun time discovering or thinking of new ways that are data may have been mis represented. Upon presenting our topic to the class it was really fun hearing the feedback or new things we could try that may impact PM2.5 readings in Beijing as there are so many different reasons why PM2.5 could spike or decline during certain times of the year.

9.3 Jeffrey Jia

In this project I handled the preprocessing for each of our methods, standard, cross validation, and seasonal models. I learned that although preprocessing data does not directly use machine learning techniques it is a very important step that can make using the models much easier. I was not familiar with many interpolation methods before this course and often just threw out rows with missing value but through my work on this project I learned about and experimented with different interpolation methods like linear, polynomial, and spline to fill the missing pieces of data. I also assisted with the model creation and evaluation metrics brainstorming and did the training and evaluation on the Bidirectional LSTM model each of the 12 stations, each taking about 5 minutes to train. I also did the comparison of our generated PM2.5 predictions with ground truth results from AirNow data, data from the US embassy in Beijing. Since we predicted for 6 or 24 hours ahead and used stride of 1 we ended up with multiple predictions for each timestep, I implemented a method to take the average of the 6 or 24 predictions from each timestep. After receiving valuable feedback from the class, we decided to implement a naive baseline method that does not use machine learning to better illustrate the improve-

ment in PM2.5 concentration prediction our model was able to achieve. I learned a lot from the feedback on the interim presentation specifically about the importance of effectively presenting results in an easy to understand and intuitive way. I learned that even if your methods preform well, if you are unable to effectively communicate the results, to the audience your work may appear trivial. Another important lesson I learned is that sometimes it is important to be flexible when working on a complex and new project that may not be in your area of expertise. When we first thought of the project idea we though we had a fairly concrete set of tasks and goals but as we continues the project we realized some tasks were unachievable given our time/resources restraints and and others needed additional elements to succeed.

References

[1] Samira Douzi Khadija Douzi Abdellatif Bekkar, Badr Hssina. Air-pollution prediction in smart city, deep learning approach. 2021.

[2] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. 2017.

[3] Lochandaka Ranathunga Eranga De Saa. Comparison between arima and deep learning models for temperature casting. 2020.

[4] Yana Garcia. Inhalable particulate matter and health (pm2.5 and pm10). 2022.

[5] Xie Y. Ma C, Madaniyazi L. Impact of the electric vehicle policies on environment and health in the beijing-tianjin-hebei region. int j environ res public health. 2021.

[6] China Ministry of Ecology and Environment. Report on the state of the ecology and environment in china. 2018.

[7] Dan Tong Jiming Hao Qiang Zhang, Yixuan Zheng. Drivers of improved pm2.5 air quality in china from 2013 to 2017. 2019.

[8] Mieczyslaw Szyszkowicz Joseph Spadaro Richard Burnett, Hong Chen. Global estimates of mortality associated with long-term exposure to outdoor fine particulate matter. 2018.

[9] Himanshu Jindal Satvik Garg. Evaluation of time series forecasting models for estimation of pm2.5 levels in air. 2021.

[10] Renyi Zhang Song Guo, Min Hu. Elucidating severe urban haze formation in china. 2014.

[11] Giyeol Lee Thanongsak Xayasouk, HwaMin Lee. Air pollution prediction using long short-term memory (lstm) and deep autoencoder (dae) models. 2020.

[12] Shuyi Zhang Hui Huang Song Xi Chen Xuan Liang, Shuo Li. Pm2.5 data reliability, consistency, and air quality assessment in five chinese cities. 2016.

[13] Tao Zou Shuo Li Haozhe Zhang Hui Huang Song Chen Xuan Liang, Bin Guo. Assessing beijing's pm2.5 pollution: severity, weather impact, apec and winter heating. 2015.

[14] Tao Zou Shuo Li Haozhe Zhang Hui Huang Song Chen Xuan Liang, Bin Guo. Assessing beijing's pm2.5 pollution: severity, weather impact, apec and winter heating. 2015.

10 Appendix

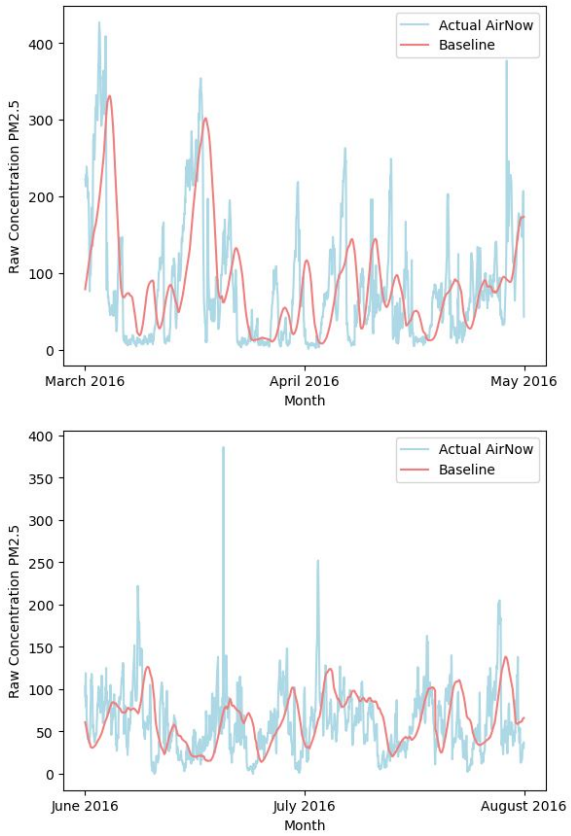

Figure 9: 24 Predictions - March 2016 - August 2016 Actual vs Baseline

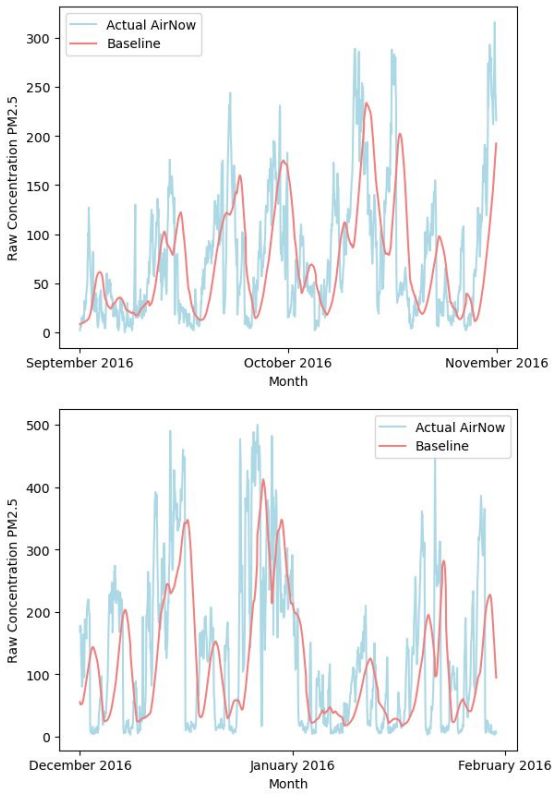

Figure 10: 24 Predictions - September 2016 - February 2017 Actual vs Baseline

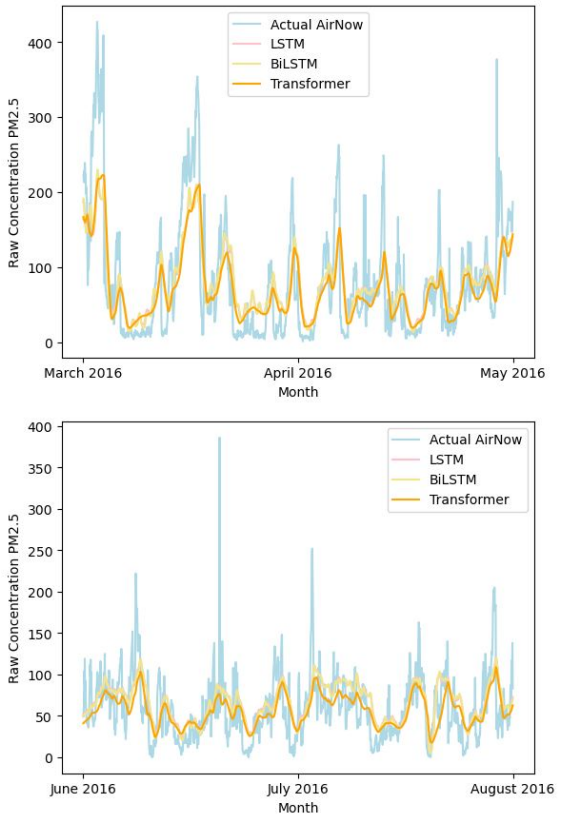

Figure 11: 24 Predictions - March 2016 - August 2016 Actual vs LSTM vs Bi-LSTM vs Tranformer

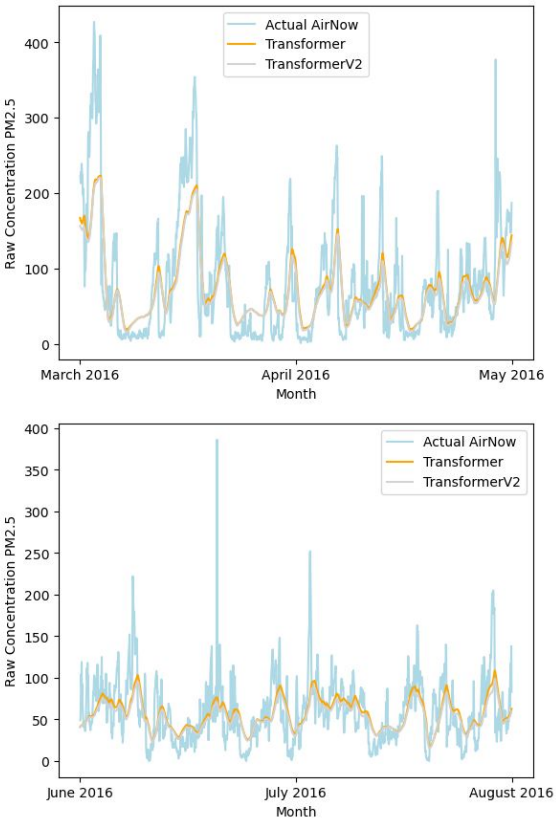

Figure 13: 24 Predictions - March 2016 - August 2016 Actual vs Transformer vs Wide Transformer

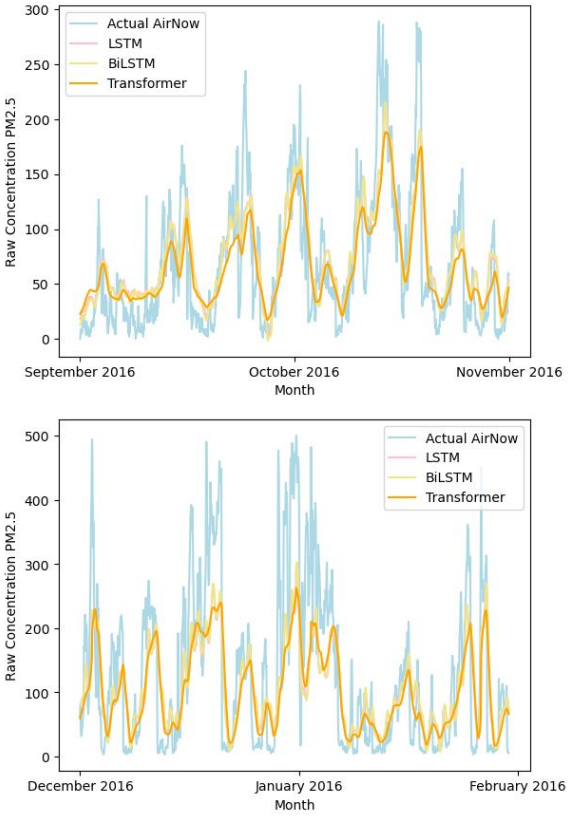

Figure 12: 24 Predictions - September 2016 - February 2017 Actual vs LSTM vs Bi-LSTM vs Tranformer

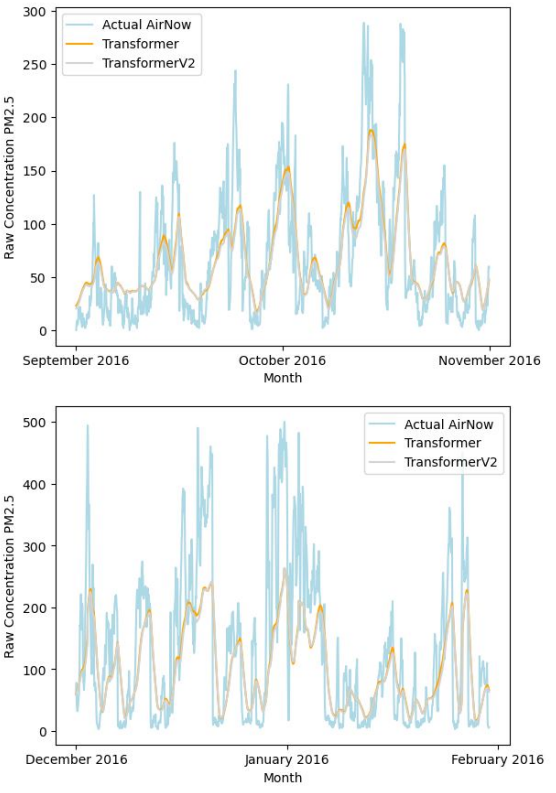

Figure 14: 24 Predictions - September 2016 - February 2017 Actual vs Transformer vs Wide Transformer

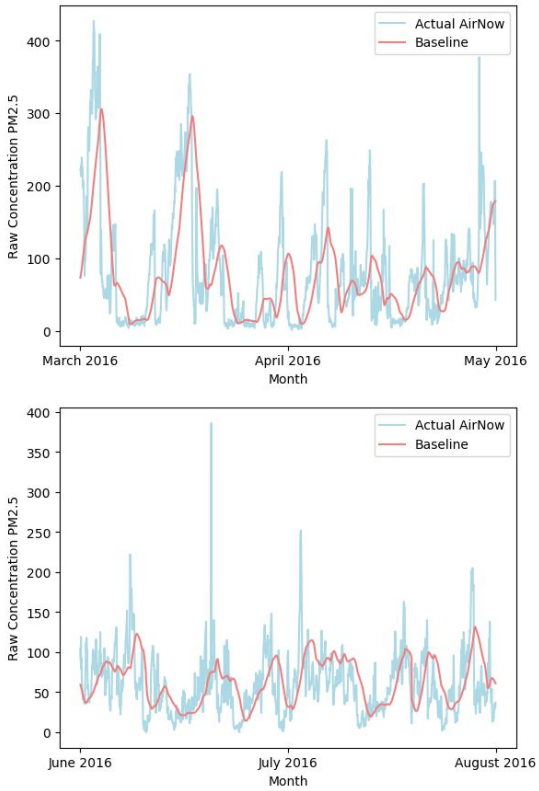

Figure 15: 6 Predictions - March 2016 - August 2016 Actual vs Baseline

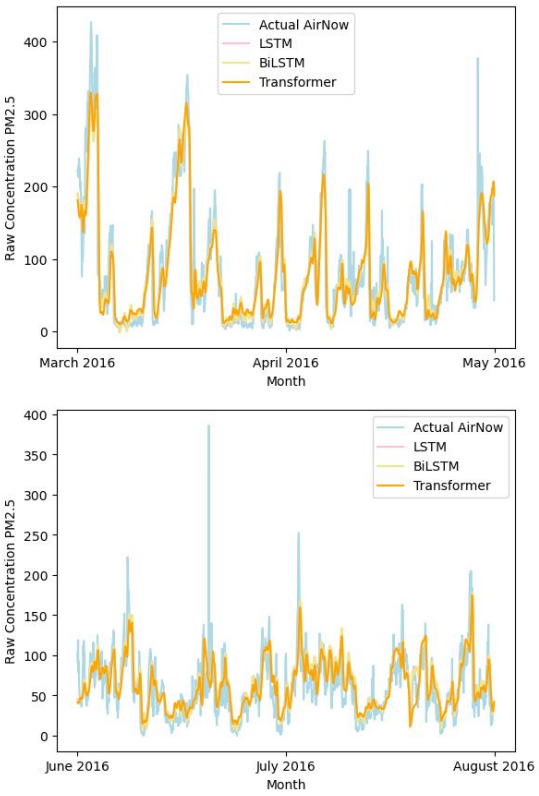

Figure 17: 6 Predictions - March 2016 - August 2016 Actual vs LSTM vs Bi-LSTM vs Tranformer

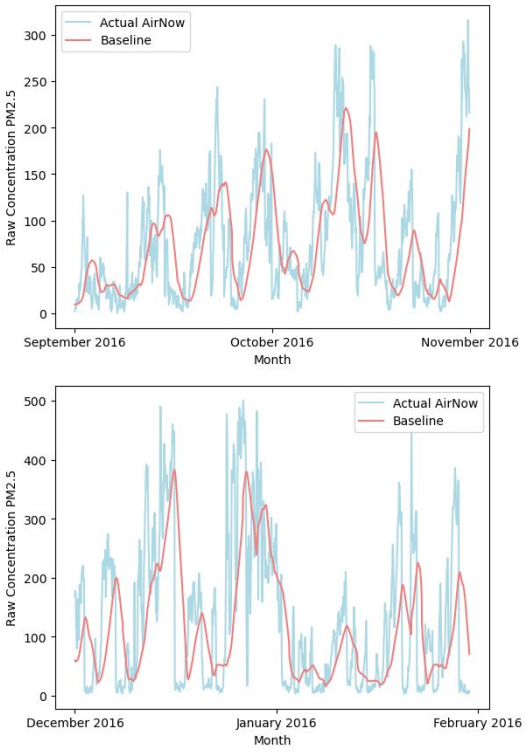

Figure 16: 6 Predictions - September 2016 - February 2017 Actual vs Baseline

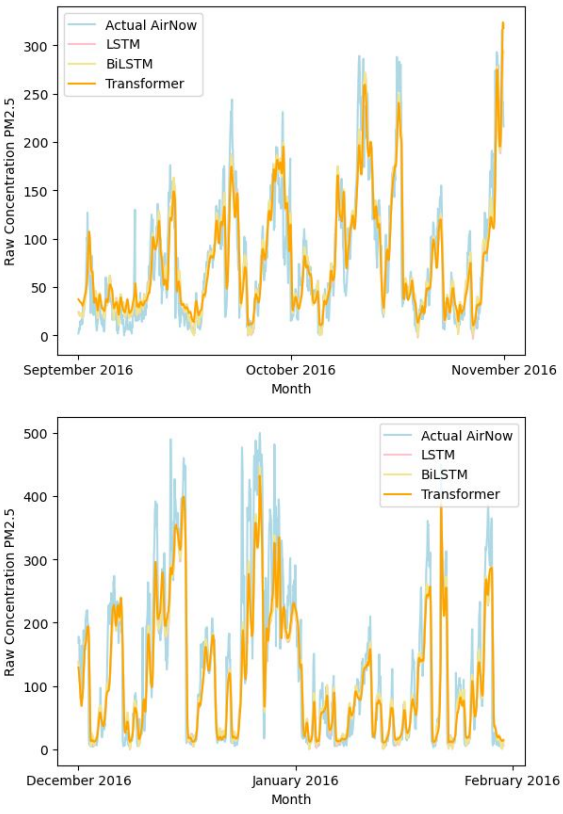

Figure 18: 6 Predictions - September 2016 - February 2017 Actual vs LSTM vs Bi-LSTM vs Tranformer

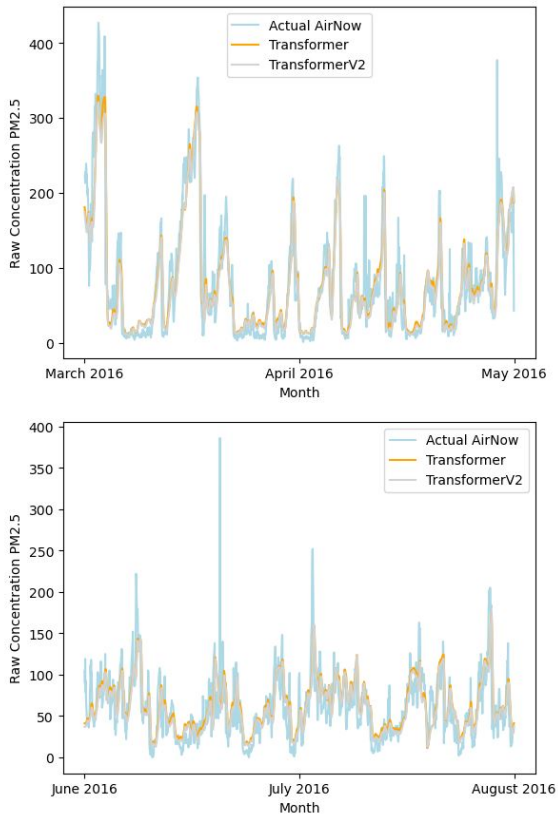

Figure 19: 6 Predictions - March 2016 - August 2016 Actual vs Transformer vs Wide Transformer

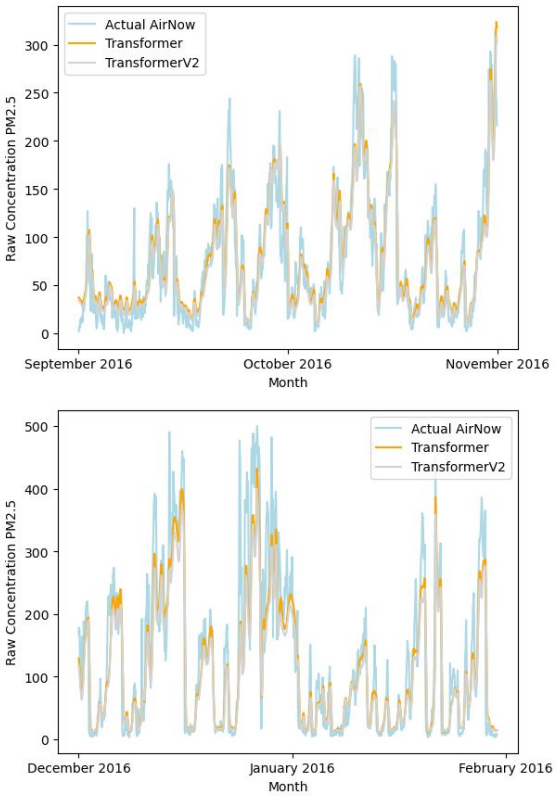

Figure 20: 6 Predictions - September 2016 - February 2017 Actual vs Transformer vs Wide Transformer