Air Quality Forecasting using Neural Networks

Abhranil Chandra
(19MF10002) IIT Kharagpur
abhranil.chandra@gmail.com

Manav Nitin Kapadnis
(19EE30013) IIT Kharagpur
iammanavk@gmail.com

Kolla Ananta Raj
(19EE30012) IIT Kharagpur
rajanant49@gmail.com

Parth Mane
(19MF10022) IIT Kharagpur
mane.parth@gmail.com

Akshat Patidar
(18MA20004) IIT Kharagpur
akshatpatidar7@gmail.com

Amit Anand
(18CE36002) IIT Kharagpur
amitanand7375@gmail.com

Abstract

Due to excessive urbanization and industrialisation, the quality of air has been drastically degrading over the years. Monitoring and preserving air quality has become an essential activity in most industrial and urban areas. The quality of air is being harmed by many types of pollutants from transportation, electricity, and fuel consumption, amongst other things. Harmful gas emissions are posing a severe danger to the quality of life in smart cities. With rising air pollution, we need to establish effective air quality monitoring models that gather data on pollutant concentrations and offer air pollution assessments in each location. Hence, air quality evaluation and prediction has become an important research area. It is influenced by a variety of multi-dimensional factors, such as location, time, and uncertain variables. Being a major worldwide issue, tackling air pollution requires meticulous planned policy and decision making. This is where data based AI systems and specifically neural network based models perform extremely well to learn generalizable trends from past data and gives highly accurate predictions over future data, thus giving policy makers and governments essential insights to make effective decisions to curb air pollution and enforce effective air quality control measures thereby protecting the environment and the citizens. We leveraged existing neural network based models such as RNN (Schuster and Paliwal, 1997), LSTM (Hochreiter and Schmidhuber, 1997) and state of the art models such as Facebook's ProphetNet (Taylor and Letham, 2017) to perform air quality forecasting on three cities of Beijing's Multi-Site Air-Quality Data Set. We then performed detailed evaluative analysis between these three models. The ProphetNet model outperforms the rest of the two models on forecasting of most of the pollutants of the cities. Additionally, all the code to generate reproducible results on our models is available on Github[1].

1 Introduction

The concentration of various air contaminants in the atmosphere at a given time and location is referred to as Air Quality. Sulphur Dioxide (SO2), Ammonia (NH3), Nitrogen Dioxide (NO2), Carbon Dioxide (CO2), Carbon Monoxide (CO), Ozone (O3), and particulate matter are examples of air pollutants (PM 2.5 and PM 10).

Air pollution is a major concern in many developing economies. Different pollutants show a variety of impacts on humans and their surroundings. PM is known to have detrimental impacts on human health (Agarwal et al., 2020) (Burnett et al., 2014); ozone (formed as a secondary pollutant through reactions of NOx and VOCs) is known to retard agricultural productivity significantly (Sharma et al., 2016); and in addition pollutants like SO2 also impacts buildings (Mallik et al., 2019), population growth, rapid increase in urbanization, vehicular ownership, energy demand, and industrialisation are the major drivers of rising air pollution levels in urban areas. Following these patterns, Indian cities have also reached alarming levels of pollution (WHO, 2018), which has led to a significant number of mortalities in India (HEI, 2019). Impacts are caused due to both acute and chronic exposures to air pollutant concentrations (WHO, 2018), and hence, knowing the short-term pollution forecasts could be very beneficial to reduce acute exposures to pollutant concentrations.

Air quality forecasts can play a vital role in air quality management by providing timely health alerts, supplementing existing emission control programs, operational planning (e.g. aviation) and emergency intervention to counter high pollution episodes (NOAA, 2001). To predict pollutant concentrations for the next several

[1] `https://github.com/manavkapadnis/AirQualityForecastingAI60002Group3`

days, short-term air quality forecasting is done using a variety of methodologies. Chemical transport models with emissions and meteorological inputs can be as simple as linking recurring weather patterns with pollution levels, or as sophisticated as employing chemical transport models with emissions and meteorological data. Forecasts for air quality can also be made using statistical approaches like regression and artificial neural networks (ANN). There are various studies across the world that demonstrate the use of ANN for air quality forecasting(Hooyberghs et al., 2005) (Alimissis et al., 2018), however, there are limited studies available in the Indian context (Singh et al., 2012)(Prakash et al., 2011)(Mahapatra, 2009)(Jain and Khare, 2010) where air pollutant levels are found to be extremely high and also with significant daily and seasonal fluctuations. There are specific efforts in India like the SAFAR (System of Air Quality and Weather Forecasting and Research) which is one of those air quality forecasting programs started by the Indian Ministry of Earth Sciences. However, the reach of forecasting programs remained limited due to their cost-intensiveness. With more than 474 urban centres (MoHUA, 2017), and widespread polluted conditions, there is a need for a low resource-intensive forecasting model for the prediction of air quality in India. Most of the Indian cities are highly polluted cities in terms of particulate matter concentrations and are influenced by multiple emission sources, and specific models are required to be tested which can take into account varying conditions.

2 Related Works

Atakan Kurt (Kurt et al., 2008) developed an online air pollution forecasting system for the Greater Istanbul Area. The system predicted three air pollution indicator (SO2, PM10, and CO) levels for the next three days (+1, +2, and +3 days) using neural networks. +1, + 2, and + 3 days' pollution levels were first predicted independently using same training data, then +2 and +3 days were predicted cumulatively using predicted values for the previous days, which led to better prediction results. The effect of the day of week as an input parameter was investigated, leading to the conclusion that better forecasts with higher accuracy were observed using the day of week as an input parameter.

A neural network technique based on the air quality time series as well as external meteorological information is presented in (Zhao et al., 2016) for air quality forecasting. The core model for the forecasts is a regularized version of the Extreme Learning Machine, and feature selection is utilized to identify the most relevant model inputs. The suggested method is tested against other ways for accomplishing spatial data fusion. Experiments demonstrate that including meteorological data improves accuracy; that how the geographical component of the problem is handled matters a lot to the model; and, ultimately, that the model is typically capable of selecting relevant inputs and providing accurate air quality forecasts.

In another study from 2017 by P A Rahman, A A Panchenko, and A M Safarov (Rahman et al., 2017) investigated the use of artificial neural networks for the ecological prediction of state of the atmospheric air of an industrial city for the capability of operative environmental decisions. They developed of two types of prediction models for determining of the air pollution index on the basis of neural networks: a temporal (short-term forecast of the pollutants content in the air for the nearest days) and a spatial (forecast of atmospheric pollution index in any point of city). It established that the obtained neural network models provided a sufficient reliable forecast, which meant that they were an effective tool for analyzing and predicting the behavior of dynamics of air pollution.

Air quality forecasting in (Cordova et al., 2021) proposes the use of MLP (Multi-layer perceptron) and LSTM based neural networks which efficiently predicted one-hour ahead PM10 concentrations where the models were evaluated under two validation schemes: the Hold-out (HO) and the Block Nested Cross-Validation(BNCV). In this study, artificial neural networks have been implemented to model time series data collected from five meteorological and air quality monitoring stations from Lima, Peru. They investigated the geographical and meteorological divergence of the forecast results from the five air quality monitoring areas in LIM using data collected from two years.

A study from 2021 (Sakhrieh et al., 2021) proposed the use of a Nonlinear Autoregressive Exogenous (NARX) model to anticipate pollution levels in Amman, Jordan. The Marquardt–Levenberg learning method was used in the model. Its performance was demonstrated using many indices, including R2 (Coefficient of Determination), R (Coefficient of Correlation), NMSE (Normalized Mean Square Error), and plots comparing network predictions to actual data. Four of the air quality monitoring sites provided historical measurements of air contaminants.

Meteorological data from three years (2015, 2016, and 2017) were utilised to train the Artificial Neural Network (ANN). The findings indicated good performance for forecasting SO2, O3, CO, and NO2 at the provided four sites, and adequate performance when forecasting Particulate Matter (PM10).

3 Dataset

In this term project, we perform a detailed analysis of Neural Network based methods for air quality forecasting. We have chosen Beijing Multi-Site Air-Quality Data Set(Zhang et al., 2017) which is publicly made available on UCI Machine Learning Repository. This dataset is developed by collecting hourly air pollutants data from 12 nationally-controlled air-quality monitoring sites at multiple sites in Beijing. The dataset contains time series data of 6 main air pollutants and 6 relevant meteorological variables spanning over a period of 4 years from March 1, 2013 to February 28, 2017. Furthermore, The air-quality data are from the Beijing Municipal Environmental Monitoring Center. The meteorological data in each air-quality site are matched with the nearest weather station from the China Meteorological Administration.

3.1 Data Preprocessing

The dataset contains air pollutants data from 12 nationally-controlled air-quality monitoring sites from multiple sites in Beijing. We then selected top three cities out of these twelve on the basis of lowest percentage of null values present in their data. These three cities are namely, Guncheng, Nongzhanguan, and Wanshouxigong. We then set the date time values of each of these datasets as the index of the dataframe using *Pandas* python package.

Furthermore, we restricted our analysis to six major components of air pollution such as PM2.5, PM10, SO2, NO2, CO, and O3. We then performed time series forecasting on these six individual time series data for each city separately to predict the quality of air in each of three sites.

Since we had carefully selected the specific cities' datasets with a very small proportion of null values, we directly used median value of a particular pollutant's time series data to fill its null values. Thus, we used median value imputation to take care of the null values.

The hourly data was then resampled to daily and monthly data. Daily resampling was done by taking average value of pollutants all over the day. For monthly resampling, first the maximum concentration of pollutants all over the day was calculated. Then the data is resampled to monthly data by taking average of the maximum concentration of pollutants throughout the day.

For the Daily data, the training data was used from 1 March, 2013 to 29 February, 2016 and the remaining one year data was used as testing. For the Monthly data, we sliced monthly data from March 2013 to February 2016 to train-set and remaining one year data for test-set. The train-test split ratio was roughly around 66.67:33.3 for both daily and monthly datasets. In the study in section 5 we have only focused on the monthly air quality prediction.

3.2 Exploratory Data Analysis

We now perform detailed analysis of the numerical features of the dataset using graphical representations to perform preliminary investigations on data in order to uncover patterns, and detect anomalies.

Figure 1 shows the combined boxplot for all the variables of the dataset. We can observe that almost all of the pollutants' distribution contain outliers, however, the "CO" pollutant contains more widely spread outliers as compared to others, which can further lead to poor performance of the forecasting models.

The correlation heatmap (Figure 2) shows a high correlation amongst the variables, "CO", "NO2", "SO2" and "PM2.5" and also between "O3" and "TEMP" variables. This implies that any independent variable from the groups of high correlated variables can be predicted from another independent variable from the same group in a regression model. Also these correlations are backed by physical reasons like fossil fuel combustion leads to formation of CO, NO2, SO2 and PM2.5 together in varying percentages. Moreover with variation in temperature the ozone cycle also gets disrupted and leads to extra O3 formation.

We see a relatively higher pollutant concentration on weekdays than on weekends in most of the cases as seen from Figure 3. This again has practical reasons as major pollution creating sources like automobiles are used

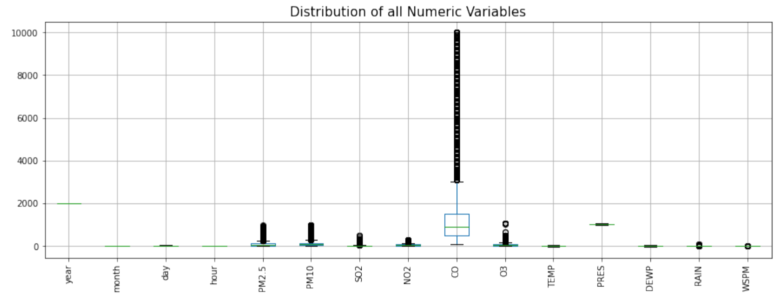


Figure 1: Outlier Analysis of Numerical Variables

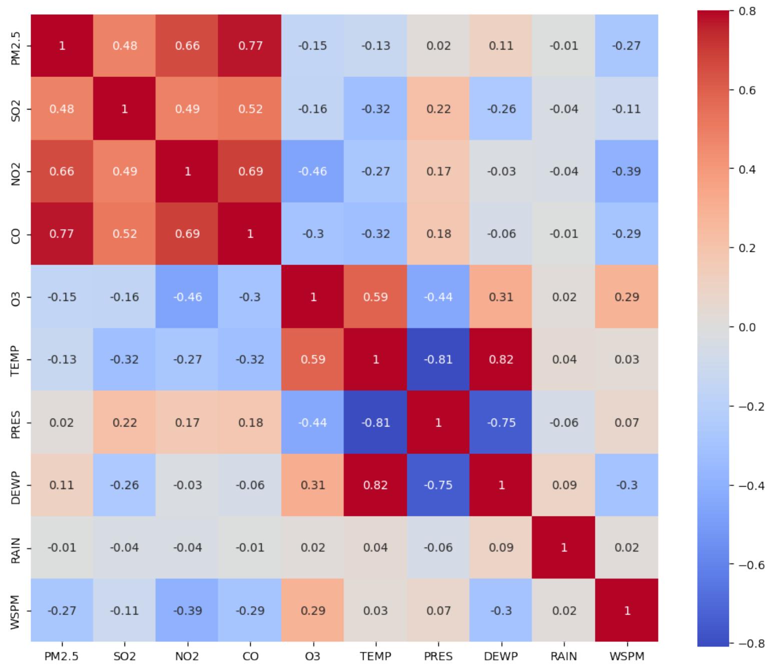


Figure 2: Correlation Heatmap of Variables

less on weekends than on weekdays thus there is a direct correlation of the amount of pollutants to the day of the weeks specifically weekdays and weekends.

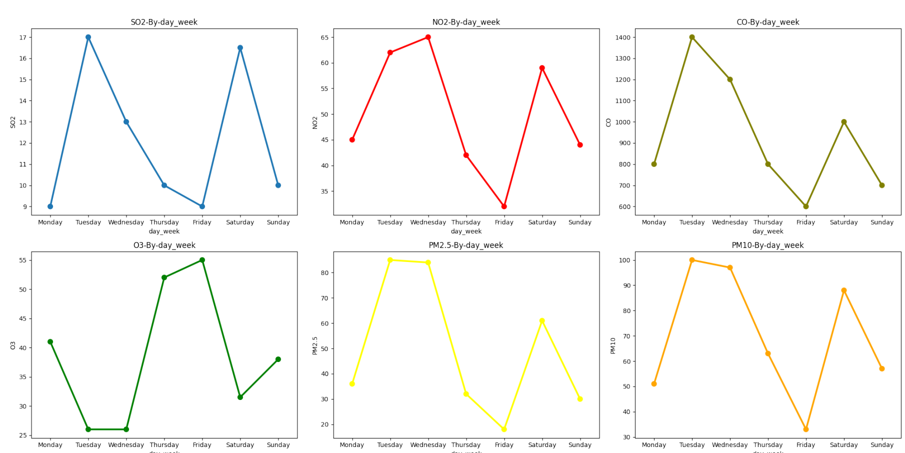


Figure 3: Daily Analysis of Pollutants

Hourly analysis (Figure 4) shows that during business hours we see a huge dip in the concentration of pollutants such as PM2.5, PM10, CO, and NO2. Furthermore, towards the start and end of business hours, when majority of the population is travelling for work, the graphs show higher concentration values of pollutants which is expected.

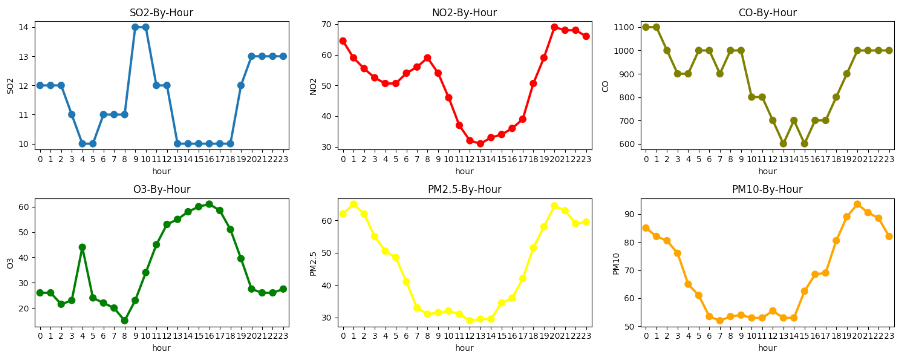


Figure 4: Hourly Analysis of Pollutants

4 Methodology

We use three broad categories of model architectures to train our data and test the performance namely Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and ProphetNet. Each model brings in some improvement over the others and we explore in detail why and how such improvements are coming. The models were trained for a large number of epochs to ensure loss convergence. However, no other hyperparameter fine tuning due to computational limitations.

4.1 Recurrent Neural Network

RNNs are connectionist models that have the ability to selectively pass information across sequence steps, while processing sequential data one element at a time. For practical purposes such as forecasting or natural language processing, RNNs are very useful as they can handle time interdependencies by doing backpropagation through time. RNNs do this by making use of the architecture shown in Fig. 5.

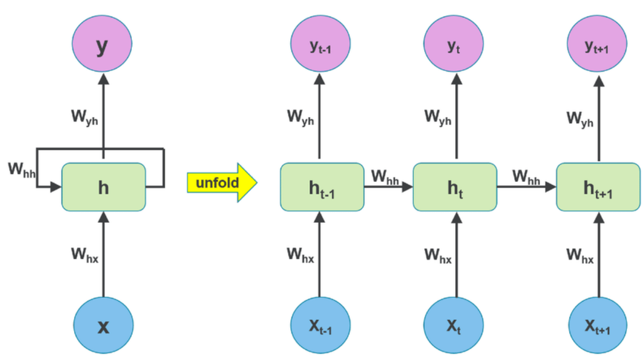


Figure 5: Unpacked RNN

It is an augmented feed-forward neural network which has edges that can span adjacent time steps and each hidden layer sends information to the next subsequent time step and thus the interdependency of the data is maintained. The network can be unpacked to understand this concept better as seen in Fig. 6.

Vanilla RNNs however have some major issues. RNNs often fail to connect long range dependencies of sequential data due to the problem of vanishing/exploding gradients. Gradient descent is the most commonly used technique in optimization for neural networks. It is basically a derivative that is applied on the error between output of the network and the actual expected result. This information is passed on from one hidden layer to the

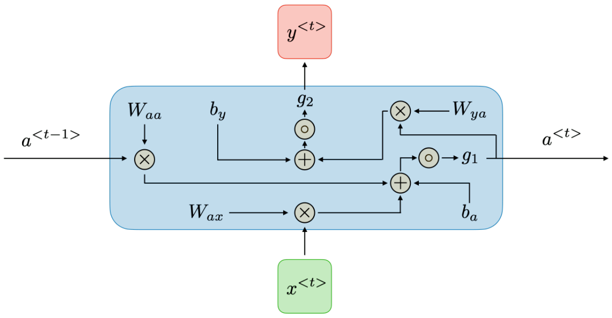


Figure 6: A block of RNN at timestamp t

next in order to perform updates. In RNNs this value has a tendency to quickly approach zero or infinity which would tend to slow the training or cause erroneous updates.

Many modifications were applied to RNNs to mitigate their problems and one of the most successful architectures is the long short term memory(LSTM). They are specifically designed to avoid the problems of vanilla RNN.

4.2 Long Short-Term Memory

The RNN dynamics can be described using deterministic transitions from previous to current hidden states. The deterministic state transition is a function

$$\text{RNN} : h_t^{l-1}, h_{t-1}^l \rightarrow h_t^l$$

For classical RNNs, this function is given by

$$h_t^l = f(T_{n,n} h_t^{l-1} + T_{n,n} h_{t-1}^l), \text{ where } f \in \{\text{sigm}, \tanh\}$$

The LSTM has complicated dynamics that allow it to easily "memorize" information for an extended number of timesteps. The "long term" memory is stored in a vector of *memory cells* $c_t^l \in R^n$. Although many LSTM architectures that differ in their connectivity structure and activation functions, all LSTM architectures have explicit memory cells for storing information for long periods of time. The LSTM can decide to overwrite the memory cell, retrieve it, or keep it for the next time step.

In LSTM we will have 3 gates: Input Gate, Forget Gate, and Output Gate. Gates in LSTM are the sigmoid activation functions i.e they output a value between 0 or 1 and in most of the cases it is either 0 or 1. We use sigmoid function for gates because, we want a gate to give only positive values and should be able to give us a clear cut answer whether, we need to keep a particular feature or we need to discard that feature.

"0" means the gates are blocking everything. "1" means gates are allowing everything to pass through it.

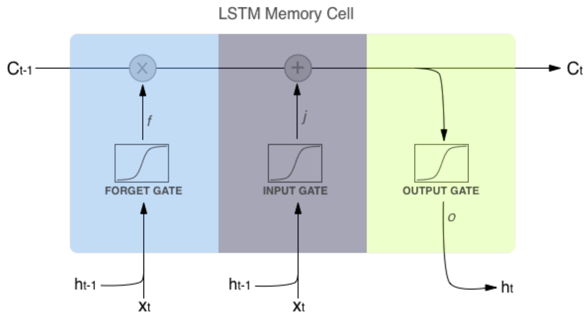


Figure 7: Gates in a LSTM cell

The LSTM architecture used in our experiments is given by the following equations:

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right)$$
$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right)$$
$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$i_t \rightarrow$ represents input gate .
$f_t \rightarrow$ represents forget gate.
$o_t \rightarrow$ represents output gate.
$\sigma \rightarrow$ represents sigmoid function.
$w_x \rightarrow$ weight for the respective gate (x) neurons.
$h_{t-1} \rightarrow$ output of the previous lstm block (at timestamp $t-1$).
$x_t \rightarrow$ input at current timestamp.
$b_x \rightarrow$ biases for the respective gates (x).

$$\tilde{c}_t = \tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$
$$h_t = o_t * \tanh\left(c^t\right)$$

$c_t \rightarrow$ cell state (memory) at timestamp (t).
$\tilde{c}_t \rightarrow$ represents candidate for cell state at timestamp (t).

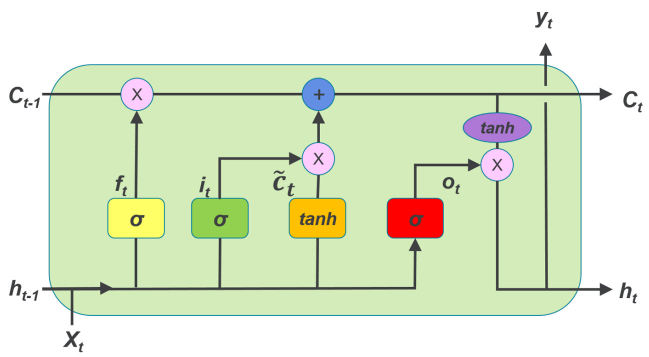


Figure 8: A block of LSTM at timestamp t

To get the memory vector for the current timestamp (c_t) the candidate is calculated. Now, from the above equation we can see that at any timestamp, our cell state knows that what it needs to forget from the previous state(i.e $f_t * c_{t-1}$) and what it needs to consider from the current timestamp (i.e $i_t * c‘_t$).[2]

Lastly, we filter the cell state and then it is passed through the activation function which predicts what portion should appear as the output of current lstm unit at timestamp t. We can pass this h_t the output from current lstm block through the softmax layer to get the predicted output(y_t) from the current block.

4.3 ProphetNet

ProphetNet is a new sequence-to-sequence pre-training model which introduces a novel self-supervised objective named future n-gram prediction and the proposed n-stream self-attention mechanism. Instead of optimizing one-step-ahead prediction in the traditional sequence-to-sequence model, the ProphetNet is optimized by n-step ahead prediction that predicts the next n tokens simultaneously based on previous context tokens at each time step. The future n-gram prediction explicitly encourages the model to plan for the future tokens and prevent overfitting on strong local correlations. ProphetNet is a current state of the art model that is very well suited

[2]Note: * represents the element wise multiplication of the vectors.

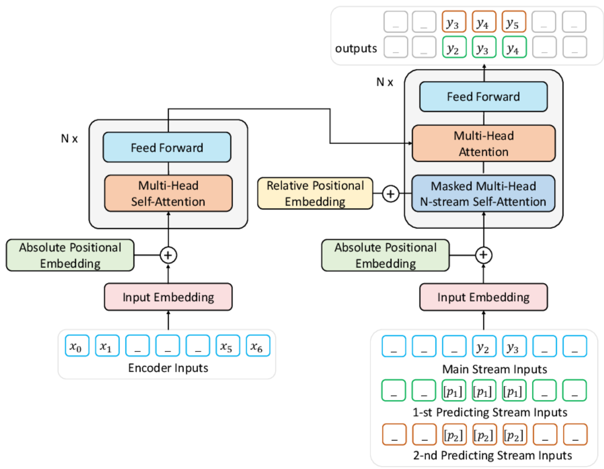


Figure 9: The Architecture of ProphetNet

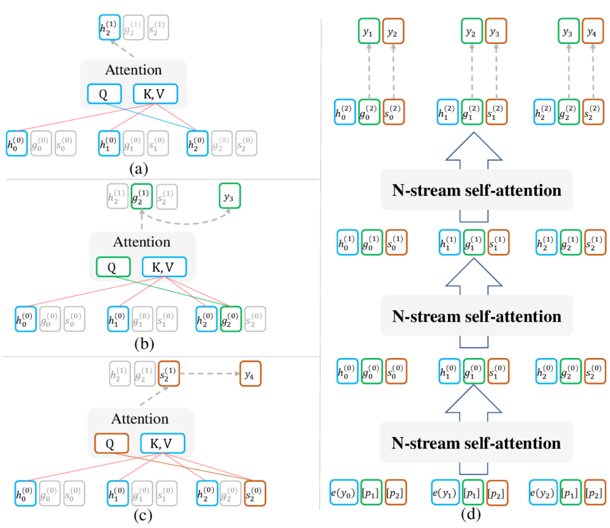


Figure 10: N-stream Self-Attention mechanism

for time-series forecasting tasks due to its ability to learn very long range dependencies using the n-stream self-attention mechanism and the ability to predict multiple time steps in the future.

The equations for the hidden states of the encoders, decoder outputs and loss function are:

$$H_{\text{enc}} = \text{Encoder}\left(x_1, \ldots, x_M\right)$$

$$p\left(y_t \mid y_{<t}, x\right), \ldots, p\left(y_{t+n-1} \mid y_{<t}, x\right) = \text{Decoder}\left(y_{<t}, H_{\text{enc}}\right).$$

$$\mathcal{L} = -\sum_{j=0}^{n-1} \alpha_j \cdot \left(\sum_{t=1}^{T-j} \log p_\theta\left(y_{t+j} \mid y_{<t}, x\right)\right)$$

$$= -\underbrace{\alpha_0 \cdot \left(\sum_{t=1}^{T} \log p_\theta\left(y_t \mid y_{<t}, x\right)\right)}_{\text{language modeling loss}} - \underbrace{\sum_{j=1}^{n-1} \alpha_j \cdot \left(\sum_{t=1}^{T-j} \log p_\theta\left(y_{t+j} \mid y_{<t}, x\right)\right)}_{\text{future n-gram loss}}$$

5 Results and Discussions

We performed experiments on the data of the 3 cities- Guncheng, Nongzhanguan, and Wanshouxigong using all 3 models on all 6 pollutants. Being a regression based problem, we used MAE and RMSE as the evaluation metrics. As is evident from the plots (Figure 14) [3] that ProphetNet outperforms all the non-attention based

[3]The forecast plots of the other models on different cities are added to the "Forecast Images" section of the Github Repository

models ie. RNN and LSTM, in most cased by significant margins and the evaluation metrics of each pollutant follows the same pattern in all the three cities, thus representing the generalizability of the trained models. Moreover we see that the MAE and RMSE scores are the least for SO2, NO2, and O3, relatively higher for PM2.5 and PM10 as they have greater numbers of outliers. The performance on CO is the worst as from the (Figure 1) its evident that CO has a huge range of outliers and the models finds it very difficult to generalize on such out-of-distribution data.

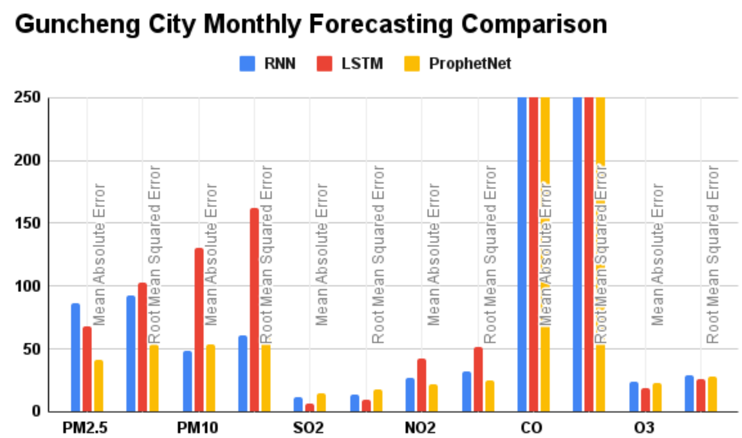


Figure 11: Guncheng City Monthly Forecast Evaluation Results Comparison

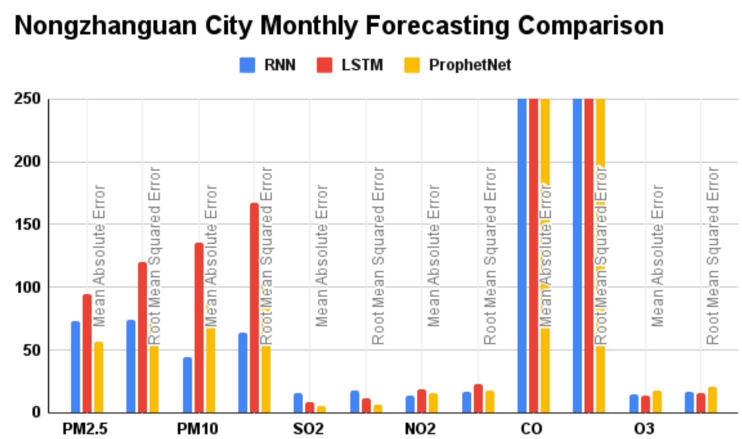


Figure 12: Nongzhanguan City Monthly Forecast Evaluation Results Comparison

6 Conclusion

In this work, we carried out a comparative study between different neural network based forecasting methods such as RNN, LSTM, and ProphetNet. We trained the model for roughly for a period of three years and evaluated its performance on the last one year. We observed that ProphetNet outperforms the other models in most of cases. These observations could be clearly seen from the error plots of Section 5. We further try to investigate the reason of poor performance of ProphetNet on pollutants such as "CO". It would be a good future direction to include the spatial correlation between the different cities and use those features to train the models to make better predictions. Furthermore, hyperparameter tuning on these models should also be done to achieve their best performances. We weren't able to perform these due to computational and time limitations.

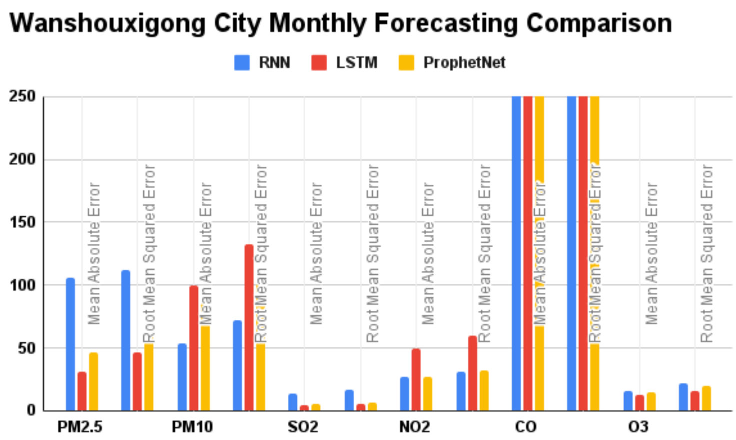


Figure 13: Wanshouxigong City Monthly Forecast Evaluation Results Comparison

7 Acknowledgements

We would like to thank Prof. Adway Mitra for teaching the Course - Machine Learning for Earth System Sciences and for evaluating this term project. It was a great learning experience for us. Finally, we would like to extend a big thanks to makers and maintainers of the exemplary TensorFlow (Abadi et al., 2015) repository, without which most of our research would have been impossible.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Shivang Agarwal, Sumit Sharma, Suresh R., Md H. Rahman, Stijn Vranckx, Bino Maiheu, Lisa Blyth, Stijn Janssen, Prashant Gargava, V.K. Shukla, and Sakshi Batra. 2020. Air quality forecasting using artificial neural networks with real time dynamic error correction in highly polluted regions. *Science of The Total Environment*, 735:139454.

Anastasios Alimissis, Kostas Philippopoulos, Chris Tzanis, and Despina Deligiorgi. 2018. Spatial estimation of urban air pollution with the use of artificial neural network models. *Atmospheric Environment*, 191, 08.

Richard T. Burnett, C. Arden Pope, Majid Ezzati, Casey Olives, Stephen S. Lim, Sumi Mehta, Hwashin H. Shin, Gitanjali Singh, Bryan Hubbell, Michael Brauer, H. Ross Anderson, Kirk R. Smith, John R. Balmes, Nigel G. Bruce, Haidong Kan, Francine Laden, Annette Prüss-Ustün, Michelle C. Turner, Susan M. Gapstur, W. Ryan Diver, and Aaron Cohen. 2014. An integrated risk function for estimating the global burden of disease attributable to ambient fine particulate matter exposure. *Environmental Health Perspectives*, 122(4):397–403, April.

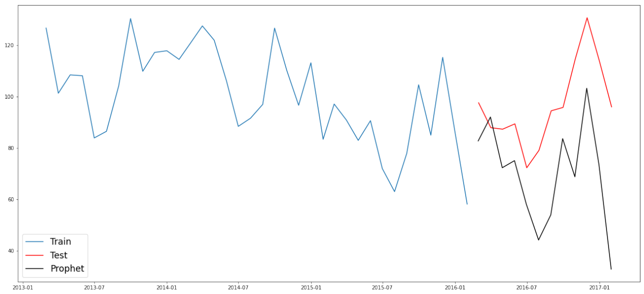
Chardin Hoyos Cordova, Manuel Niño Lopez Portocarrero, Rodrigo Salas, Romina Torres, Paulo Canas Rodrigues, and Javier Linkolk López-Gonzales. 2021. Air quality assessment and pollution forecasting using artificial neural networks in metropolitan lima-peru. *Scientific Reports*, 11(1), December.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov.

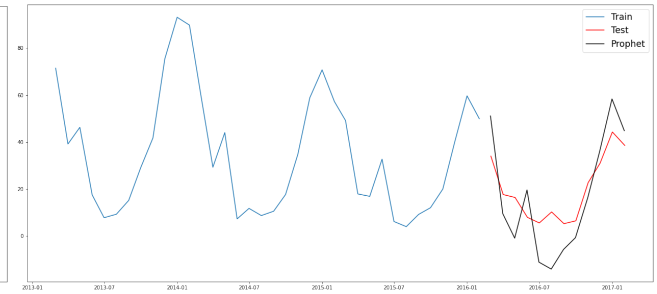
Jef Hooyberghs, Clemens Mensink, Gerwin Dumont, Frans Fierens, and Olivier Brasseur. 2005. A neural network forecast for daily average pm10 concentrations in belgium. *Atmospheric Environment*, 39:3279–3289, 06.

Suresh Jain and Mukesh Khare. 2010. Adaptive neuro-fuzzy modeling for prediction of ambient CO concentration at urban intersections and roadways. *Air Quality, Atmosphere &amp Health*, 3(4):203–212, May.

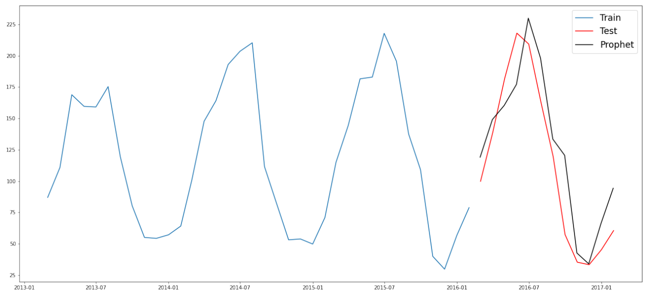
Atakan Kurt, Betul Gulbagci, Ferhat Karaca, and Omar Alagha. 2008. An online air pollution forecasting system using neural networks. *Environment international*, 34:592–8, 08.



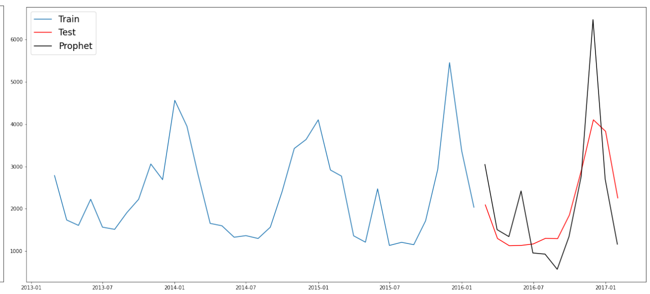
(a) NO2



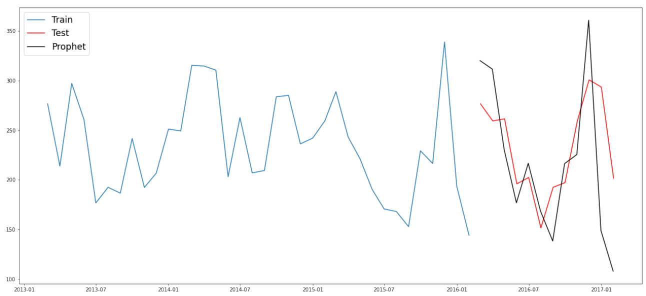
(b) SO2



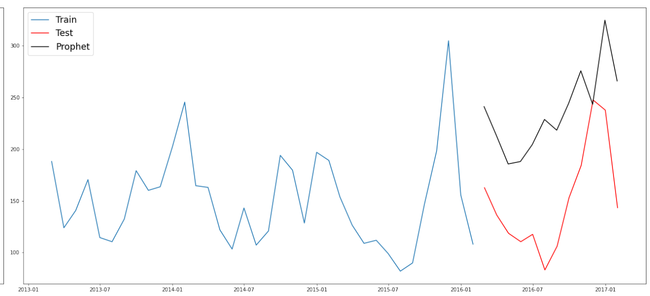
(c) PM10



(d) PM25



(e) O3



(f) CO

Figure 14: Guncheng City ProphetNet Forecast

Amita Mahapatra. 2009. Prediction of daily ground-level ozone concentration maxima over new delhi. *Environmental Monitoring and Assessment*, 170(1-4):159–170, October.

Chinmay Mallik, Parth Sarathi Mahapatra, Prashant Kumar, Subhasmita Panda, R. Boopathy, Trupti Das, and Shyam Lal. 2019. Influence of regional emissions on SO2 concentrations over bhubaneswar, a capital city in eastern india downwind of the indian SO2 hotspots. *Atmospheric Environment*, 209:220–232, July.

Amit Prakash, Ujjwal Kumar, Krishan Kumar, and V. K. Jain. 2011. A wavelet-based neural network model to predict ambient air pollutants' concentration. *Environmental Modeling &amp Assessment*, 16(5):503–517, June.

P A Rahman, A A Panchenko, and A M Safarov. 2017. Using neural networks for prediction of air pollution index in industrial city. *IOP Conference Series: Earth and Environmental Science*, 87(4):042016, oct.

Ahmad Sakhrieh, Mohammad Hamdan, and Mohammad Bani Ata. 2021. Air quality assessment and forecasting using neural network model. *Journal of Ecological Engineering*, 22(6):1–11, June.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sumit Sharma, Prateek Sharma, Mukesh Khare, and Swati Kwatra. 2016. Statistical behavior of ozone in urban environment. *Sustainable Environment Research*, 26(3):142–148.

Kunwar P. Singh, Shikha Gupta, Atulesh Kumar, and Sheo Prasad Shukla. 2012. Linear and nonlinear modeling approaches for urban air quality prediction. *Science of The Total Environment*, 426:244–255, June.

Sean J Taylor and Benjamin Letham. 2017. Forecasting at scale. September.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. 2017. Cautionary tales on air-quality improvement in beijing. *Proc. Math. Phys. Eng. Sci.*, 473(2205):20170457, September.

Chen Zhao, Mark van Heeswijk, and Juha Karhunen. 2016. Air quality forecasting using neural networks. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, December.