11/17/2020
CS 230
Jack Seagrist - jkseag@stanford.edu
Karthik Ramesh - kart0197@stanford.edu

Forecasting PM 2.5 Pollution Using Weather Data

Abstract

Air pollution poses serious health hazards, especially for people in urban environments and in developing nations. With the advent of machine learning models and deep learning, it is interesting to explore the potential of using data from existing in-situ monitoring infrastructure to make air pollution predictions. In this paper, we have looked at implementing an RNN with LSTM to make predictions of PM2.5 concentrations in the Bay Area. Our model robustly makes accurate forecasts for different locations spread across the Bay.

Introduction

Air pollution related illnesses are one of the most widespread health hazards responsible for 5 million deaths every year (4th highest risk factor in causes leading to death). A World Bank study also estimates that air pollution illnesses and deaths cost the global economy $225 billion annually. The primary air pollution metric is PM2.5, or particulate matter that is up to 2.5 microns in diameter.

Each 10-g/m3 elevation in long-term average PM2.5 ambient concentrations was associated with approximately 4-8 percent increased the risk of cardiopulmonary and lung cancer mortality[1]. Hence, to grapple with a problem of this size, it is necessary to introduce policies based on real world quality data trends and analysis.

The instruments and sensors used to evaluate air quality levels are usually expensive and hence makes it difficult to deploy them in large numbers. A neural network trained to make predictions of air quality based on meteorological parameters and ground truth values can be a very useful tool to combat this problem.

Related Work

Efficient prediction of the air quality response to different emission and meteorological changes is very challenging because of the nonlinear response of air quality to these changes. Some of the approaches and strategies followed by other researchers have been discussed here.
The most interesting approach we noted was that of the pf-RSM and DeepRSM wherein the network was developed using chemical transport models to create and track emissions in the

[1] arXiv:1804.07891

atmosphere with high spatial and temporal resolution while meteorological fields were based on simulations with the Weather Research and Forecasting (WRF) model. Polynomial functions were used to represent the non-linear responses and a CNN deep neural network was used to analyze satellite images and predict the ambient air quality levels.[2]

The earliest air quality prediction models using machine learning relied on Autoregression approaches but these failed to make good forecasts, especially because it relied on a correlation between the past and the future values.[3]
A study compared the performance of Artificial neural networks and Genetic programming approaches in forecasting the air quality parameters like oxides of nitrogen, oxides of sulphur and particulate matter. Results suggested that the GP worked as well as the ANN approach.[4]

Our model was inspired by a South Korean study implemented LSTM units and Encoder-Decoder model with Adam optimizer to forecast air quality[5] and DeepAir which used LSTM recurrent neural network (RNN) as a framework for forecasting in the future, based on time series data of pollution and meteorological information. These studies observed that their models exhibited appreciable accuracy in forecasting air quality when compared to conventional SVR models.

It was also interesting to study the methodology of using an RNN with LSTM to forecast the complex ozone cycle concentrations in the atmosphere.[6]

Dataset and Features

Our raw data was collected from CIMIS stations[7] and the EPA Outdoor Air Quality Data set[8] . We locate areas which have both CIMIS and EPA stations collecting the desired data we need over a long period of time. Currently, we have collected 5 years worth of daily data for Santa Cruz, San Rafael, and Sebastapol using their respective CIMIS and EPA stations for the years 2015-2019. From the EPA dataset, we use the Daily Mean PM2.5 concentration values. We use the following meteorological parameters from the CIMIS stations: dew point, average air temperature, average vapor pressure, wind run, average wind speed, precipitation, latitude, and longitude. We take these daily values and match them by date to get a complete data matrix for each day. Listed below is a sample figure showing the prepared csv file that is ready to input into the model.

---

[2] Environ. Sci. Technol.2020, 54, 8589−8600
[3] Journal of the American statistical Association,vol. 65, no. 332, pp. 1509–1526, 1970
[4] IOSR Journal Of Environmental Science, Toxicology And Food Technology(IOSR-JESTFT), vol. 3, no. 5, pp. 01–08, 2013, http://www.iosrjournals.org/iosr-jestft/pages/v3i5.htm
[5] https://arxiv.org/ftp/arxiv/papers/1804/1804.07891.pdf
[6] Brian S. Freeman, Graham Taylor, Bahram Gharabaghi & Jesse Thé (2018) Forecasting air quality time series using deep learning, Journal of the Air & Waste Management Association, 68:8, 866-886, DOI: 10.1080/10962247.2018.1459956
[7] https://cimis.water.ca.gov
[8] https://www.epa.gov/outdoor-air-quality-data/download-daily-data

Date	Daily Mean PM2.5 C	Dew Point (C)	Avg Air Temp (C)	Avg Vap Pres (kPa)	Wind Run (km)	Avg Wind Speed (m/	Precip (mm)	Latitude	Longitude
1/1/2015 0:00	5.7	-3.6	4.7	0.5	121.8	1.4	0	36.997444	-121.99676
1/2/2015 0:00	8.1	0.7	5.2	0.6	126.7	1.5	0	36.997444	-121.99676
1/3/2015 0:00	8.9	0.7	6.5	0.6	119.1	1.4	0	36.997444	-121.99676
1/4/2015 0:00	8.8	3.1	8	0.8	115.7	1.3	0	36.997444	-121.99676
1/5/2015 0:00	8.3	3.3	12	0.8	123	1.4	0	36.997444	-121.99676
1/6/2015 0:00	6.6	4.1	14.1	0.8	133.7	1.5	0	36.997444	-121.99676

Fig 1. - Sample Input Data Combined and Prepared for Input to Model

To evaluate our model predictions, we picked stations in different regions of the bay area which were not in close proximity to our training set stations - Oakland, Napa College and Richmond.

Methods

Our approach follows the work conducted by Vikram Reddy[9] in the paper “Deep Air: Forecasting Air Pollution in Beijing, China”[10] and Sagar Mankari[11]. We first established that we could clone and run their repositories. After we were successful with that endeavor, we began to experiment with running their model using our data as the input.

The model used an LSTM architecture with a mean absolute error loss function.The LSTM network is able to learn long term dependencies through its feedback connections and cell memory unit. This makes it especially useful when dealing with sequential data, such as time series air pollution information. For our model we ultimately decided to use a mean squared error loss function, described further in the experiment section below.

Experiments/Results/Discussion

After successfully running our first prototype for the project milestone, we were able to expand upon our work and develop a more accurate model.

Based on the initial results from training version 1, we decided to keep the batch size (72) and epochs (50) hyperparameters the same as the model appeared to be converging relatively quickly. We also decided to keep the train/test split at 70/30 for our moderate amount of data.

Since we are working on a regression rather than classification problem, we chose the root mean squared error to be our primary metric to determine model success. We adjusted the network architecture (layer and neuron count), loss function, and dropout rate to adjust model performance. A summary of the model versions and the hyperparameters and architecture that were tuned can be seen in figure 3 below.

[9] https://github.com/vikmreddy/deep-air
[10] https://www.ischool.berkeley.edu/sites/default/files/sproject_attachments/deep-air-forecasting_final.pdf
[11] https://github.com/sagarmk/Forecasting-on-Air-pollution-with-RNN-LSTM

Model Version	Key Features	RMSE
Version 1	1 layer; 50 neurons; loss=mean absolute error; dropout=.3	6.990
Version 2	1 layer; 100 neurons; loss=mean absolute error; dropout=.4	6.060
Version 3	2 layers; 50 neurons/layer; loss=mean squared error; dropout=.4	5.380
Version 4	3 layers; 100 neurons/layer; loss=mean squared error; dropout=.4	3.696

Fig 3 - Summary of Model Performances

After each model was trained, we tested it by predicting PM2.5 values for Oakland. Figure 4 shows the predictions for Oakland with model version 2. We found the model to have a high bias. This prompted us to create a deeper network architecture in the future versions to address this problem.

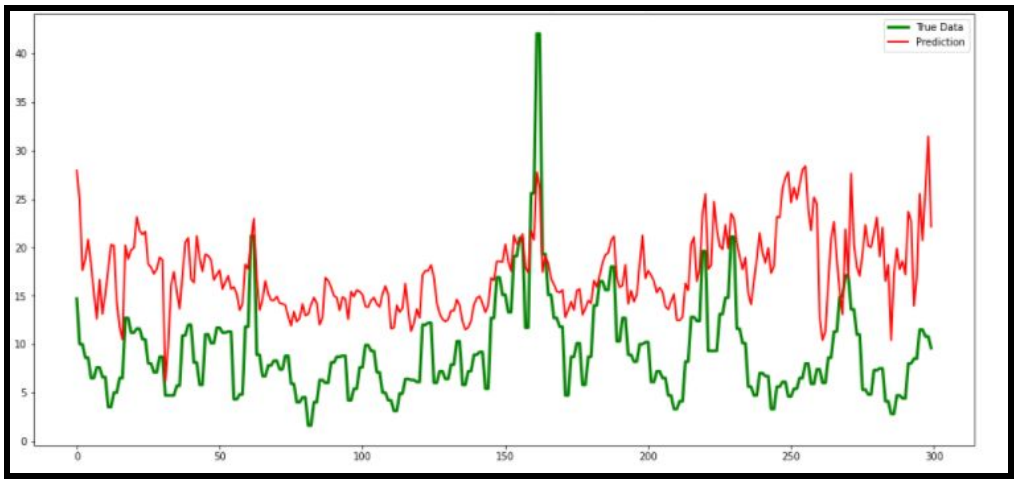

Fig 4 - Oakland Prediction PM2.5 with Model Version 2

After training model version 4, we saw better performance with less bias overall as shown in Figure 5. Future improvements could be made by using an even larger data set or even deeper network architecture.

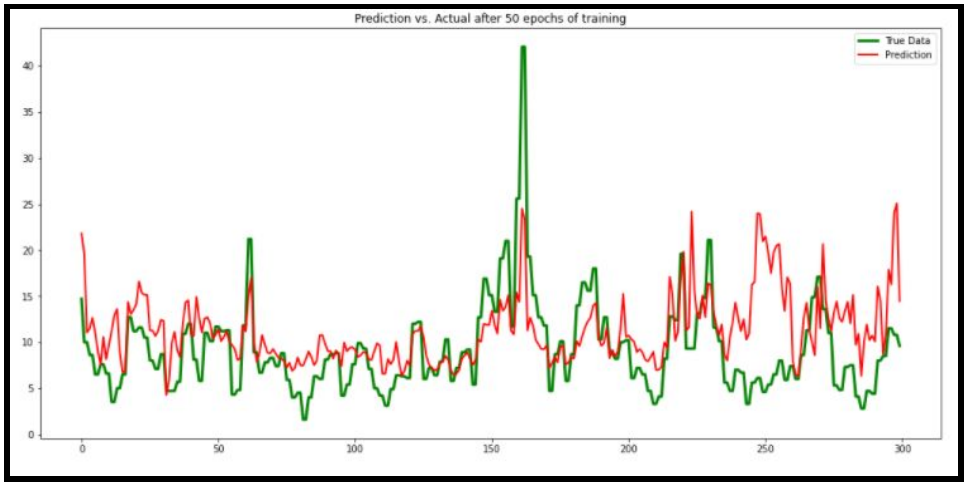

Fig 5 - Oakland Prediction PM2.5 with Model Version 4

We also tested the final model, version 4, on the data from Napa and Richmond to compare and the model performed relatively well. The figures can be found in the appendix.

Our model appears to have identified the general trend in PM2.5 pollution as it correlates to weather conditions. The predictions may not be 100% accurate but they follow the general trend lines in both rural and urban areas. We also see less spikes in the data relative to the actual meter readings. This could be evidence that the model is eliminating some of the noise that the equipment inherently experiences as a result of the monitoring technique.

Conclusion/Future Work

Our final model was able to learn the general trends and changes in the levels of PM 2.5 with respect to changes in the meteorological parameters. We were able to accurately forecast the air quality of locations in the Bay Area appreciably far apart from our training set coordinates, such as Oakland (the train set did not contain stations from the East Bay).

However, with a relatively restrictive training dataset, this model might not be as accurate in trying to forecast the ambient air quality of climatic types different from the Bay Area Microclimates. We would require to significantly increase the size of the training set to make the model more generalizable. Additionally, a framework to keep track of the sources of emissions (absent from our work) will add greater accuracy to the predictions.

Contributions

The team consisted of two members, Jack Seagrist and Karthik Ramesh. Jack was responsible for gathering data from Santa Cruz, Sebastopol, and Richmond. Karthik was responsible for gathering the data from San Rafael, Oakland, and Napa College. Both team members participated in background literature review, testing and tuning the model in google colab, and publishing the final results.

Appendix

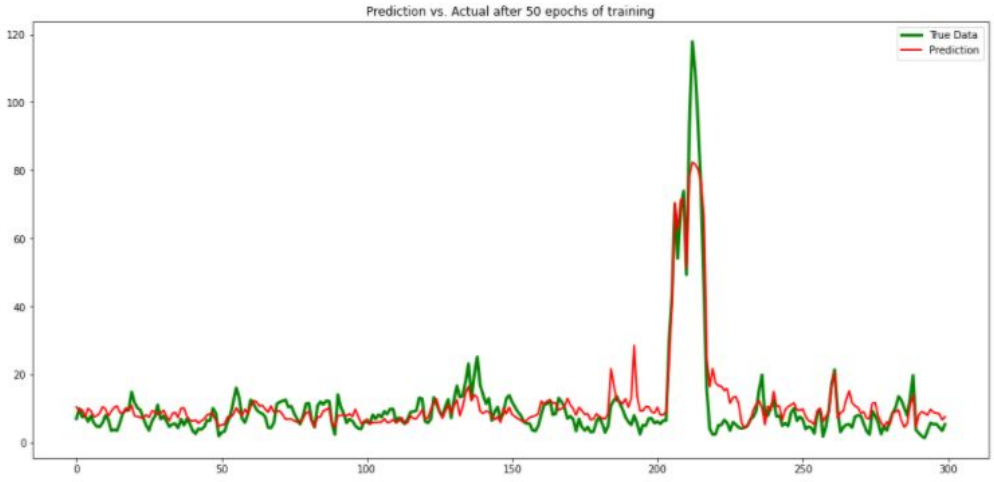

Fig A1 - Napa Prediction Final Model

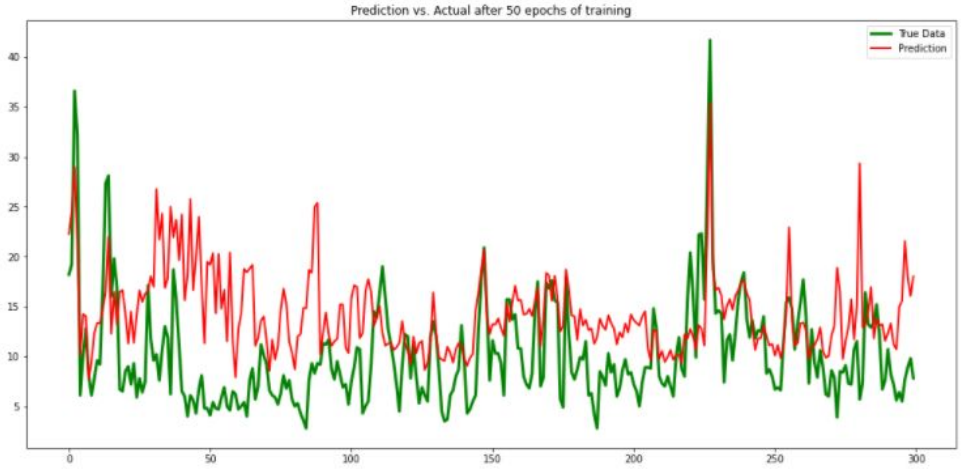

Fig A2 - Richmond Prediction Final Model