

Satellite Image Atmospheric Air Pollution Prediction through Meteorological Graph Convolutional Network with Deep Convolutional LSTM

Pratyush Muthukumar*, Emmanuel Cocom*, Kabir Nagrecha*, Jeanne Holm**, Dawn Comer**, Anthony Lyons**, Irene Burga**, Chisato Fukuda Calvert***, and Mohammad Pourhomayoun*

* Department of Computer Science, California State University Los Angeles, Los Angeles CA, [pmuthuk2, ecocom, knagrec2, mpourho]@calstatela.edu,

** City of Los Angeles, Los Angeles CA, [jeanne.holm, dawn.comer, anthony.lyons, irene.burga]@lacity.org,

*** OpenAQ, Washington D.C., chisato@openaq.org

Abstract—Every five seconds, somebody around the world prematurely dies from the effects of air pollution. Air pollution is one of the world’s leading risk factors for death. To mitigate the deadly effects of air pollution, it is imperative that we understand it, discover the patterns and sources, and predict it in advance. Air pollution prediction in real-time requires extremely powerful models that can solve this spatiotemporal problem in multiple dimensions. We used an advanced graph convolutional network coupled with a deep convolutional LSTM model to learn patterns over the spatial and temporal dimension in real-time. Our model employs a graph convolutional network that models meteorological features and extracts high-level embeddings through unsupervised representation learning. We created a sequential encoder-decoder deep convolutional LSTM that allows for accurate and efficient satellite image based atmospheric Nitrogen Dioxide air pollution prediction over Los Angeles county 10 days into the future using data from 10 days in the past through the use of spatiotemporal satellite imagery and meteorological graph embedding inputs. Our results for predicting spatially continuous atmospheric Nitrogen Dioxide in Los Angeles over various time periods shows improvement in prediction over previous research done on this topic.

Index Terms—spatiotemporal air pollution prediction, deep convolutional LSTM, graph convolutional network, remote-sensing satellite imagery, atmospheric air pollution

I. INTRODUCTION

Air pollution is a pervasive global threat. It is estimated that 92 percent of the world’s population breathes polluted air and the effects of air pollution leads to 7 million deaths annually [1]. Moreover, more than one in every four deaths of children under the age of 5 is directly related to the effects of air pollution [2]. As it is projected that by 2050 more than two thirds of the world’s population or close to seven billion people will live in in urban areas, it is imperative that a solution to mitigate the adverse effects of air pollution must be comprehensive and effective [3].

Our proposed model in this paper seeks to apply deep neural networks and advanced machine learning algorithms to learn patterns in spatiotemporal air pollution and predict spatially continuous atmospheric air pollution multiple days

into the future. Our data of remote-sensing satellite imagery and ground-level meteorological features are highly correlated to past data at and around that geographic location (spatial correlation) and timepoint (temporal correlation).

Previous work on the spatiotemporal problem of predicting weather forecasts or pollutant matter largely focus on either predicting spatial relations or predicting temporal relations, but it is considerably more difficult to create a highly complex model that utilizes measurements that are both spatially and temporally correlated [4] [5] [6] [7]. It is even more challenging to utilize meteorological and air pollution features, employ state-of-the-art deep neural network models to learn and analyze the spatiotemporal patterns and effects on each other, and produce a highly accurate and continuous spatiotemporal prediction.

The Graph Convolutional Network (GCN) is an advanced neural network architecture for machine learning on graphs. The goal of a GCN model is to learn feature embeddings of nodes and edges on a graph through convolutions on neighborhoods of nodes. We can train a neural network with an initial layer embedding $h_v^0 = x_v$, where x_v denotes the node features, to perform convolution on these neighborhoods of nodes. To accomplish this, define the k -th layer embedding of the vertices h_v^k as

$$h_v^k = \sigma \left(W_k \sum_{u \in N(v) \cup v} \frac{h_u^{k-1}}{\sqrt{|N(u)||N(v)|}} \right), \forall k > 0,$$

where σ is some non-linear activation function, h_v^{k-1} is the previous layer embedding of v , W_k is a transformation matrix for self and neighbor embeddings, and $\sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|}$ is the average of a neighbor’s previous layer embeddings. The neural network can be implemented efficiently through sparse batch operations:

$$H^{(k+1)} = \sigma(D^{-\frac{1}{2}} \tilde{A} D^{-\frac{1}{2}} H^{(k)} W_k),$$

where $\tilde{A} = A + I$ and $D_{ii} = \sum_j A_{i,j}$ [8].

The Convolutional Long Short-Term Memory (ConvLSTM) model is a complex machine learning model that allows video inputs that hold data in spatial and temporal dimensions. The ConvLSTM model is a variation of the traditional Long Short-Term Memory network model, a time series Recurrent Neural Network. In traditional Fully Connected Long Short-Term (FC-LSTM) models, the input data must be a one dimensional vector parameterized by time. To allow for image inputs over time, ConvLSTMs implements convolution over gates and cell/hidden states. There is an alternative approach that other papers have utilized to induce convolution to LSTM models by independently running a Convolutional Neural Network (CNN) in series with a Long Short-Term Memory Network (LSTM) in a modular architecture referred to as a CNN-LSTM model [9].

If we replace the Hadamard products, denoted \circ , of the key equations from the FC-LSTM model with the convolution operation, then the key equations for the ConvLSTM are defined as

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} * c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} * c_{t-1} + b_f) \\ c_t &= f_t * c_{t-1} + i_t * \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} * c_t + b_o) \\ h_t &= o_t * \tanh(c_t), \end{aligned}$$

where $*$ denotes the convolution operation [10].

II. METHODS

A. Dataset

Our meteorological data was sourced from the Iowa State University Environmental Mesonet database which records METeorological Aerodrome Reports (METAR) data of Automated Surface Observing Systems (ASOS) hourly observations in the continental United States [11]. The METAR data utilized in our model recorded 17 different ground-level meteorological features hourly at 24 sensor locations around Los Angeles county. Some of these meteorological attributes include wind speed, wind direction, precipitation, relative humidity, air temperature, and dew point. A map of the meteorological sensor locations is shown in Figure 1.

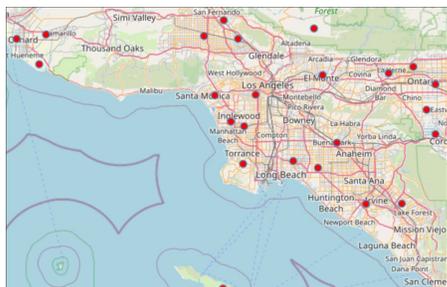


Fig. 1. METAR ASOS observations of Mesonet database: 24 sensor locations in Los Angeles, where each sensor records 17 meteorological attributes hourly

The raw satellite imagery data we used for input was sourced from the U.S. Geological Survey's (USGS) Earth Explorer database records of the Sentinel-2 remote sensing satellite [12]. The Sentinel 2 satellite operates along a 290-km orbital swath and was launched by the European Space Agency in March 2015 to image and record atmospheric and terrain data through 13 spectral bands based on the wavelength of the emitted light [13]. Our model utilized two imaging bands: a 442.7 nm central wavelength spectral band that images coastal aerosol levels of dust, smoke, and general particulate matter, and a finer 945.1 nm central wavelength spectral band that images specifically Nitrogen Dioxide levels in the atmosphere. A sample raw satellite input is shown in Figure 2, where the blue structures correspond to strictly Nitrogen Dioxide air pollution, while the white, cloud-like structures correspond to general particulate matter.

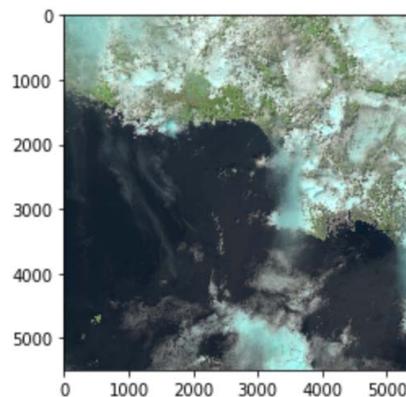


Fig. 2. Sample Raw Data (Source: USGS EarthExplorer database of satellite imagery of Los Angeles taken on April 29, 2019 by ESA's Sentinel 2 satellite)

B. Data Preprocessing

To apply our raw satellite images as input to the ConvLSTM model, we had to transform our data into a 5D tensor. Our data included 882 GeoTIFF high resolution images which corresponded to 1642 days of data. This is because from the USGS EarthExplorer dataset, we selected Nitrogen Dioxide satellite imagery from August 3, 2015 to March 19, 2020, or 882 images covering a 1642 day timespan, where each image taken is 46 hours apart from the next. For compatibility with the outputs of the meteorological GCN and for ease of use, we reduced the resolution of all our satellite images in the dataset to 40px by 40px. Since we focused on predicting nitrogen dioxide, we filtered out the general particulate matter shown as a white, cloud-like structure, the ocean imagery, and the land imagery. We were solely focused on predicting the blue cloud-like structures that correspond to the Nitrogen Dioxide imaging band, so we used a color mask from the OpenCV Python library to retain all light blue hues of the satellite images, but turn any other hue to black.

The goal of our model is to use the data of 10 days prior to predict spatiotemporal nitrogen dioxide in Los Angeles 10

days in the future. Since each image is spaced roughly 2 days apart from the next, we singly staggered and bundled 5 frames into a sample. For example, sample 1 contained frames 1-5, sample 2 contained frames 2-6, etc. Not only does this allow for manageable temporal chunks of the data to be trained and predicted on, but it allows us to create a 5D tensor to input to the ConvLSTM model, where the dimensions of each sample are: (sample dimension, frame dimension, x-dimension of image, y-dimension of image, channel dimension). Without including the outputs of the meteorological GCN, the channel dimension refers to the red, green, and blue channels of the images.

For effective predictions, the distinction and size of the features, labels, and train-test split are key to the performance of the model. We denoted our label dataset to be 5 frames ahead of the feature dataset. In this way, the feature dataset contains samples of images 5 frames prior to a certain date (for example, frames 1-5), while the label dataset contains samples of images 5 frames in the future from a certain date (frames 6-10). Thus, we can ensure that the model has enough information to learn the patterns while not encountering data leakage, as there is no overlap between the label and features, but there is a staggered overlap among samples within the label dataset or samples within the feature dataset. Finally, we split 80 percent of our data into the training set and 20 percent of our data into the test set.

For the meteorological ground-level sensor data to be used as an input for the GCN model of our implementation, we preprocessed the raw CSV data into time parameterized multidimensional gridded graphs. The data source provided hourly measurements of all meteorological features, however we used the same time span from March 19 2015 to August 3 2020 and frequency of 46 hours in order to later combine with the satellite imagery data. The data source also provided the various meteorological values in percentile units, as there were various readings that use different units, so a percentile measure was implemented to standardize units. We first defined a 40 by 40 grid with the values of the four boundary points of the grid corresponding to geographical latitude and longitude of the boundaries of the satellite imagery from the Sentinel-2 satellite. Thus, this geographically bounded grid covers the same area as any satellite image from the USGS EarthExplorer dataset. We then mapped each of the 17 ground-level meteorological sensors to the grid by using their respective longitudes and latitudes to set onto a specific element of the grid with the closest relative latitude and longitude value.

We then used the StellarGraph Python package to map a directed weighted graph where a node of the graph must lie within an element of the grid. We then created multiple grids for multiple node attributes. We defined a node attribute of this “meteorological graph” as one that is constant in reference to the sensor location but varies through time. Some of the node attributes of our “meteorological graph” include precipitation, Air Quality Index, relative humidity, and air temperature.

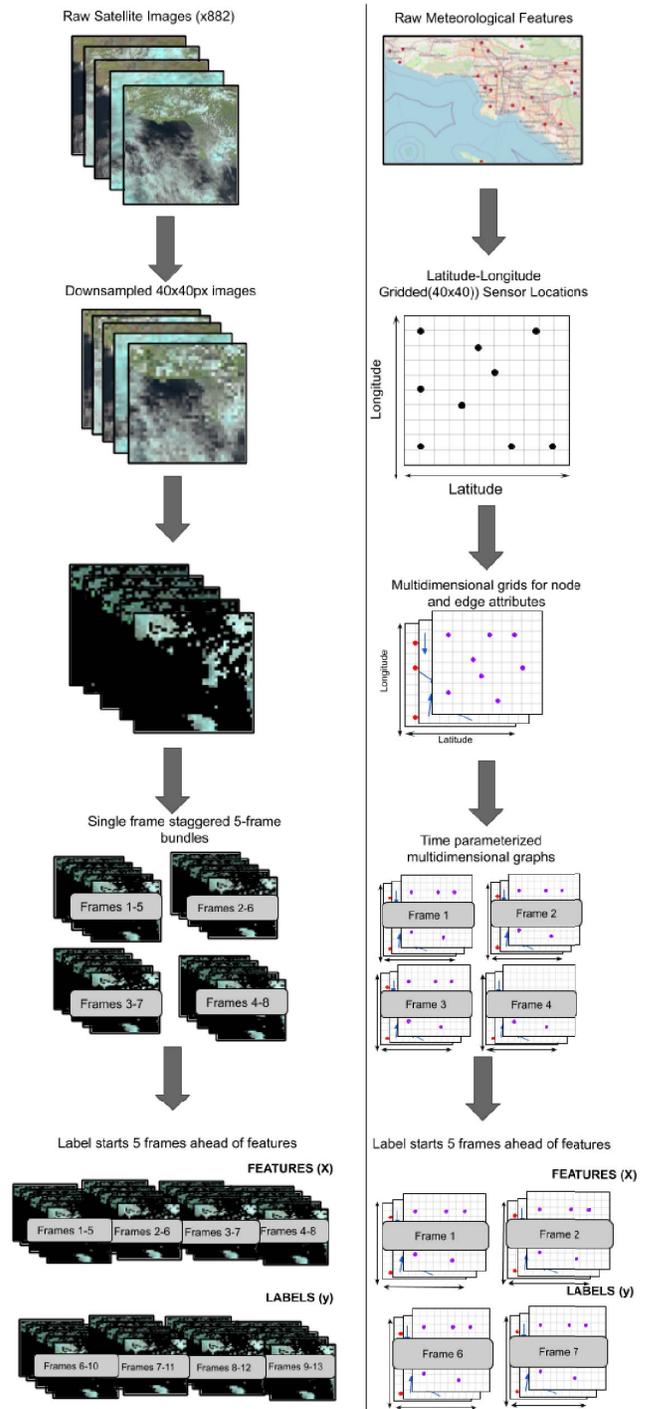


Fig. 3. High-level overview of data preprocessing for satellite image and ground-based meteorological input data.

The edge attributes of our “meteorological graph” were wind speed, wind direction, max wind gust, and average wind gust. We used simple calculations to obtain the magnitude and direction of a wind vector’s u and v components from wind speed and direction and used these values as directed edge weights to create a set of graph attributes that each were on a separate grid so that all node and edge attributes combined would make a weighted directed graph with a bounded multi-dimensional grid structure. These structures were created for each of the 882 timepoints that we have set to parameterize our satellite image and meteorological ground-based data.

Similarly to the satellite image data, we set our labels for the multidimensional grid meteorological graphs to be 5 frames ahead of our features. Thus, a sample of the feature dataset contains “meteorological graphs” for 5 frames prior to a certain date, while a sample of the label dataset contains “meteorological graphs” for 5 frames in the future from a certain date. Figure 3 describes a high-level overview of the data preprocessing process.

C. Model Architecture and Implementation

Our implemented model can be broken into three parts. The first part or the meteorological Graph Convolutional Network (GCN) uses the preprocessed meteorological data as input. We adapted previous work on using Graph Convolutional Networks to perform spatiotemporal interpolation of nodes and edges in a graph [14]. We used the StellarGraph GCN layer to create denser, more complex weighted directed “meteorological graphs” created with the bounded multidimensional grid structure.

Our deep neural network interpolation of the sparse time-series graphs trains by hiding a small percentage of the nodes and respective edge attributes of the label graph corresponding to the input feature graph. Recall that the label graph is the ground truth value of the feature graph but 5 frames ahead or nearly 10 days in the future. These hidden nodes and edges are used as the label for the GCN model to learn the interpolation, through the use of the non-hidden nodes and edges of the feature graph.

For example, a typical training iteration could utilize the input feature and label graph data of 12 meteorological sensors, corresponding to 12 nodes in the “meteorological graph”. To train, the GCN could set 5 of the 12 nodes and respective edges of the label graph to hidden and learn the interpolation by using the data of the 7 non-hidden nodes and edges from the feature graph. After the GCN has trained on the feature and label graph datasets, we can scale up the prediction to interpolate for many nodes and edges: for example, an additional 40 nodes and respective edges. The output of this model is a set of time-series weighted directed graphs where the temporal difference of the meteorological features described in each graph output is 46 hours.

A visualization of the training predictions of AQI node attributes from 5 hidden nodes, denoted with stars, using seven non-hidden nodes, denoted with circles, of the graph 5 frames prior is shown in Figure 4.

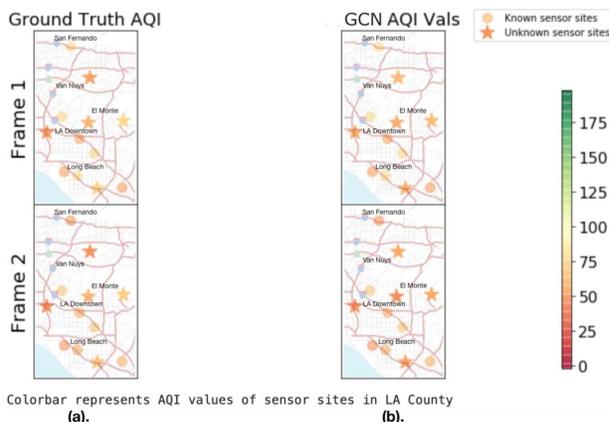


Fig. 4. AQI Node Attribute Training Prediction Visualization. (a). shows the Ground Truth AQI node attribute values over 2 frames separated by 46 hours, (b). shows the GCN Predicted AQI node attribute values over 2 frames separated by 46 hours.

The second part of the model allows the output of the meteorological GCN to be used as an input to the ConvLSTM model along with the satellite image data to ultimately predict spatiotemporal atmospheric nitrogen dioxide in Los Angeles. This unsupervised graph representation learning section of the model allows us to translate the weighted directed graph outputs of the GCN to high-level time-series indexed sets of 40px by 40px images. We used the StellarGraph Unsupervised Graph Representation Learning layer to extract the information on the interactions and effects of various meteorological features on each other along geographic location and temporal location.

We trained the unsupervised graph representation learning model using the directed graph outputs of the meteorological GCN for 100 epochs. Figure 5 shows a high-level graph embedding visualized with Matplotlib.

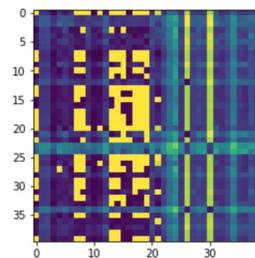


Fig. 5. High-level meteorological graph embedding visualization (not human-interpretable).

The final part of our model architecture is the ConvLSTM which takes in inputs of both the Meteorological Graph Embedding “Images”. The image arrays of the meteorological graph embeddings are packaged into a 5D tensor in a similar way to the raw satellite images, where each sample includes the singly staggered bundle of 5 frames. Since the graph

embeddings are not RGB, the images of the meteorological embeddings use a single channel dimension, thus the shape of the entire meteorological dataset without splitting to the training and testing set is (880, 5, 40, 40, 1). The shape of the raw satellite images is (880, 5, 40, 40, 3). We concatenate two datasets through the channel dimension so that the input dataset to the ConvLSTM model is of shape (880, 5, 40, 40, 4). The output of the ConvLSTM model that uses data of five frames or roughly 10 days in the past is the predicted image of atmospheric nitrogen dioxide every 46 hours for roughly 10 days into the future.

III. RESULTS

Our model predicted spatiotemporal atmospheric nitrogen dioxide continuously in Los Angeles county 10 days in the future using data of remote-sensing satellite imagery and meteorological ground-level sensors from 10 days in the past. Figures 6-8 show the visualization of our predictions of the first two and fifth frame in the testing set, corresponding to predictions of roughly 2 days in the future, 4 days in the future, and 10 days in the future.

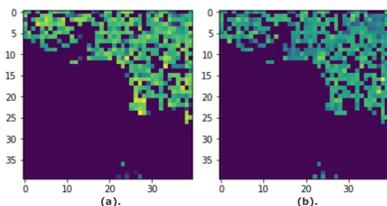


Fig. 6. Frame 1 Prediction: Roughly 2nd day (46 hrs) in the future prediction of Nitrogen Dioxide air pollution in Los Angeles County from previous 10 days of data, (a) Prediction, (b) Ground Truth

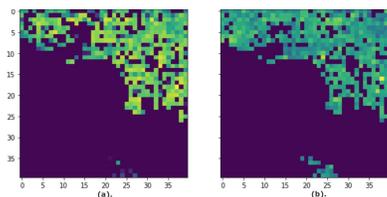


Fig. 7. Frame 2 Prediction: Roughly 4th day (92 hrs) in the future prediction of Nitrogen Dioxide air pollution in Los Angeles County from previous 10 days of data, (a) Prediction, (b) Ground Truth

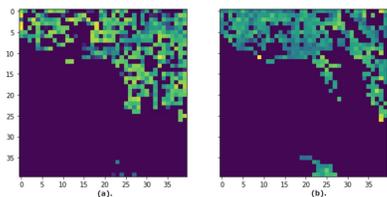


Fig. 8. Last Frame Prediction: Roughly 10th (230 hrs) day in the future prediction of Nitrogen Dioxide air pollution in Los Angeles County from previous 10 days of data, (a) Prediction, (b) Ground Truth

We can see from the visualizations of the 1st, 2nd, and 5th frames that the predictions closer to the specified date are more similar to the ground truth than predictions on further dates. This is expected as the correlation between real world nitrogen dioxide 10 days in the past is less correlated than nitrogen dioxide 10+ days in the future, so we see a drop in accuracy over time in our predictions accordingly.

IV. ERROR ANALYSIS

Since the output of our model is a series of images of atmospheric nitrogen dioxide in Los Angeles over time, traditional error methods do not truly display the accuracy of our model. Pixel by pixel RMSE error metrics between the ground truth and predicted images would not reflect the similarities in the overall structures of nitrogen dioxide in the images. Instead, we utilized the Structural Similarity Index Measure (SSIM) metric commonly used in computer vision and image related domains [15]. The SSIM error metric measures the difference in overall structures pictures in two images through a normalization of traditional pixel-by-pixel analysis by considering pixel averages and standard deviations. The SSIM metric is defined as

$$SSIM(I, \hat{I}) = \sum_{p \in I} \frac{2\mu_p\mu_{\hat{p}} + c_1}{(\mu_p^2 + \mu_{\hat{p}}^2 + c_1)\sigma_p^2 + \sigma_{\hat{p}}^2 + c_2},$$

where I and \hat{I} denotes the ground truth and predicted images respectively, p and \hat{p} denote the pixels of the ground truth and predicted images respectively, μ denotes the average pixel value of each image, σ denotes the standard deviation of pixel values of each image, and c_1 and c_2 are constants relating to the relative noise of the images [16]. Table 1 shows the SSIM error values between the predicted outputs and the ground truth for the first five frames of the testing set which span approximately 10 days in the future.

Our model shows a 69.5% decrease in average error from our previous model which solely utilized a ConvLSTM structure [17]. The incorporation of the meteorological GCN and unsupervised graph representation learning structures to the model architecture greatly improved our model predictions. The information and effects on atmospheric nitrogen dioxide flow from the meteorological features increases the accuracy of the model by 20%.

TABLE I
SSIM VALUES FOR FIRST SAMPLE (FIRST 5 FRAMES): STRUCTURAL SIMILARITY PERCENTAGES OF TEN DAYS IN THE FUTURE NITROGEN DIOXIDE PREDICTIONS IN LA COUNTY FROM PREVIOUS TEN DAYS OF DATA.

SSIM Values over Sample 1		
	MeteoGCN-ConvLSTM Model	ConvLSTM Model
Frame 1	0.93	0.77
Frame 2	0.88	0.70
Frame 3	0.84	0.63
Frame 4	0.79	0.56
Frame 5	0.71	0.51

V. CONCLUSION

In this paper, we used complex deep learning models to accurately predict spatiotemporal atmospheric nitrogen dioxide air pollution in Los Angeles over time. In designing the models, we accounted for the spatial and temporal correlations of air pollution so that we could learn, analyze, and predict the patterns of nitrogen dioxide.

We utilized various ground-level meteorological features by constructing a weighted directed graph with a geographically bounded multidimensional grid. We then used advanced Graph Convolution Networks (GCN) to analyze the interactions and patterns of meteorological data and its effects on the flow of nitrogen dioxide. We used unsupervised graph representation learning structures to extract high level embeddings of these time-series meteorological graphs that inherently contains details on the spatial and temporal correlations of these features. Finally, we used a Convolutional Long Short-Term Memory (ConvLSTM) model to utilize preprocessed satellite image data along with the meteorological graph embeddings over time to predict spatially continuous images of atmospheric air pollution in Los Angeles. We used satellite image and meteorological features of data roughly 10 days (230 hrs) prior to predict atmospheric nitrogen dioxide 10 days in the future. Each frame prediction was spaced roughly 2 days or 46 hours apart.

Our results showed a 69.5% reduction in error from our previous model which solely used a ConvLSTM model to predict nitrogen dioxide in Los Angeles using data from roughly 10 days in the past to predict roughly 10 days in the future. We used the Structural Similarity Index Measure (SSIM) error metric to accurately understand the differences of general structures of nitrogen dioxide displayed in ground truth and predicted images.

This work can be used to alert researchers in various fields on the flow and patterns of Nitrogen Dioxide for any given time period at any location in Los Angeles for the next five years.

VI. FUTURE WORK

In the future, we hope to utilize both ground-based and remote sensing satellite image data of Nitrogen Dioxide and other pollutant matters to improve our model. We hope to directly predict the raw values of ground-level Nitrogen Dioxide in parts per million (ppm) through an improved model.

This study can also extend further than Los Angeles county, predict for larger time ranges, and predict an array of pollutants including PM2.5, Carbon Monoxide, Ozone, and Sulfur Dioxide.

REFERENCES

- [1] United Nations. With a premature death every five seconds, air pollution is violation of human rights, 2019.
- [2] World Health Organization et al. Air pollution and child health: prescribing clean air: summary. Technical report, World Health Organization, 2018.
- [3] Hannah Ritchie and Max Roser. Urbanization. *Our world in data*, 2018.

- [4] Benjamin Klein, Lior Wolf, and Yehuda Afek. A dynamic convolutional layer for short range weather prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4840–4848, 2015.
- [5] Sanam Narejo and Eros Pasero. Meteorocasting using deep learning architecture. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 8(8), 2017.
- [6] Aditya Grover, Ashish Kapoor, and Eric Horvitz. A deep hybrid model for weather forecasting. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 379–386, 2015.
- [7] Erik Boye Abrahamsen, Ole Magnus Brastein, and Bernt Lie. Machine learning in python for weather forecast based on freely available weather data. 2018.
- [8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [9] Chiou-Jye Huang and Ping-Huan Kuo. A deep cnn-lstm model for particulate matter (pm2. 5) forecasting in smart cities. *Sensors*, 18(7):2220, 2018.
- [10] Yipeng Liu, Haifeng Zheng, Xinxin Feng, and Zhonghui Chen. Short-term traffic flow prediction with conv-lstm. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2017.
- [11] Dennis P Todey, DE Herzmann, and ES Takle. The iowa environmental mesonet—combining observing systems into a single network. In *Sixth Symposium on Integrated Observing Systems*, 2002.
- [12] USGS. Usgs earthexplorer satellite imagery database. Website. <https://earthexplorer.usgs.gov/>.
- [13] Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: Esa’s optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012.
- [14] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. Inductive graph neural networks for spatiotemporal kriging. *arXiv preprint arXiv:2006.07527*, 2020.
- [15] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [16] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011.
- [17] Pratyush Muthukumar, Emmanuel Cocom, Jeanne Holm, Dawn Comer, Anthony Lyons, Irene Burga, Christa Hasenkopf, and Mohammad Pourhomayoun. Real-time spatiotemporal air pollution prediction with deep convolutional lstm through satellite image analysis. In *Proceedings of the 16th International Conference on Data Science (ICDATA)*, 2020.