Spatio-temporal air quality forecasting in the Western states of USA using Graph Neural Network

Abrar Akhyer Abir

May 4, 2023

Abstract

Monitoring outdoor air quality is important for qualitatively measuring pollutants and making policies to reduce the adverse effect caused on human and nature. Recently, massive wildfires in the Western regions of the United States rendered air quality to be very unhealthy for outdoor activities. Forecasting air quality for the future can be applied to know about the possible air quality to some extent. To address these issues in this project, several shallow and deep neural network-based architectures have been used to predict the particulate matter 2.5. Using these models as baseline, I have implemented Graph Neural Network based model, compared the results with the baseline models and find out that GNN based model outperform traditional neural network models and showed almost double improvement in accuracy.

Introduction

Air quality drew attention worldwide after the Covid-19 breakout when we observed a drastic reduction in air pollution [pan]. In the United States, industrialized and urbanized areas have traditionally had poor air quality, especially for children and the elderly. In recent years, the number of studies and inquiries into source apportionment and forecasting for air quality monitoring has expanded. However, very few studies have looked at the effects of pollution on health and how to reduce exposure to it in the regions under discussion. By investigating the air quality in these areas and creating a forecasting model, I hope to close this gap.

In this project, I want to track and predict air pollution, especially in the most wildfire-prone areas of the Western United States. The air quality will be measured using particulate matter 2.5 (PM2.5) values together with other atmospheric factors like temperature, humidity, and pressure. I will use wunderground data for meteorological variables [wun] and Berkeley Earth air quality data for hourly readings of particulate matter 2.5 [ber]. Along with the geographic location, both data are available on an hourly basis. The data will first be gathered and cleaned. I will then get model data ready for testing. Our model for capturing relationships between related parameters will now begin to take shape. Then, as a means of sequence prediction, I will look at the temporal properties of the time series data. The spatial relationship of the time

series data will next be addressed in order to deduce the pollutant transportation and make an attempt at forecasting utilizing geographic features. In particular, I have done the following implementation and analysis:

- Implement Linear and convolutional models as baselines models.These models accounted for all the variables that are available to the dataset for the last 14 days.
- Implement Graph convolutional Network using our collected data to detect spatial relations between different places and predict PM 2.5.

After implementing the above models, I have found out that Graph Neural Network gave us best performance because it captures the spatial relation quite well. In result analysis section, I will discuss our finding more elaborately.

Problem Definition

Numerous health issues, including cardiovascular and respiratory illnesses, can be brought on by poor air quality and can be made worse by pre-existing conditions. Accurate air quality predictions might encourage people to take precautions, like remaining inside or donning a mask, to lessen their exposure to pollutants and safeguard their health. Millions of people around the world suffer from the health and well being effects of air pollution, which is a serious environmental issue. When making decisions to safeguard the public's health and lessen the effects of air pollution, individuals, communities, and governments can benefit from accurate forecasting of the quality of the air. The complicated interactions between air pollution and numerous environmental conditions cannot be fully captured by traditional air quality forecasting methods.

By simulating the intricate geographical interactions between air quality measurements and environmental factors like meteorological information, land use, and emissions sources, graph neural networks (GNNs) offer a potential method for forecasting air quality. The goal of this project is to create and assess an air quality forecasting model using machine learning that can predict future urban air quality levels with accuracy. In order to produce more precise and dependable air quality forecasts, I have used a GNN-based technique to capture the intricate relationships between many environmental elements and air quality levels. This project is intriguing because it investigates how GNNs might be used to forecast air quality, which could have a big impact on environmental and public health regulations.

Models

I have implemented two linear models (Linear Regression and Bayesian Ridge) for predicting the PM 2.5. After implementing the linear models, I extend our work to deep learning models where I have used Conventional Neural Network and Long Short-Term Memory Networks. From literature review, I have find out that time series forecasting is frequently done using both long short-term memory (LSTM) networks and conventional neural networks (CNNs). Forecasts that are more precise and dependable can

be produced by combining the advantages of these two models. A hybrid model can generate forecasts that are more accurate than either model alone by combining the advantages of the CNN and LSTM models. While the LSTM can identify long-term dependencies in the data, the CNN can assist in locating important features in the input data. This can result in projections that are more precise and trustworthy for a variety of time series forecasting applications. The architecture of Conventional Neural Networks are shown 1

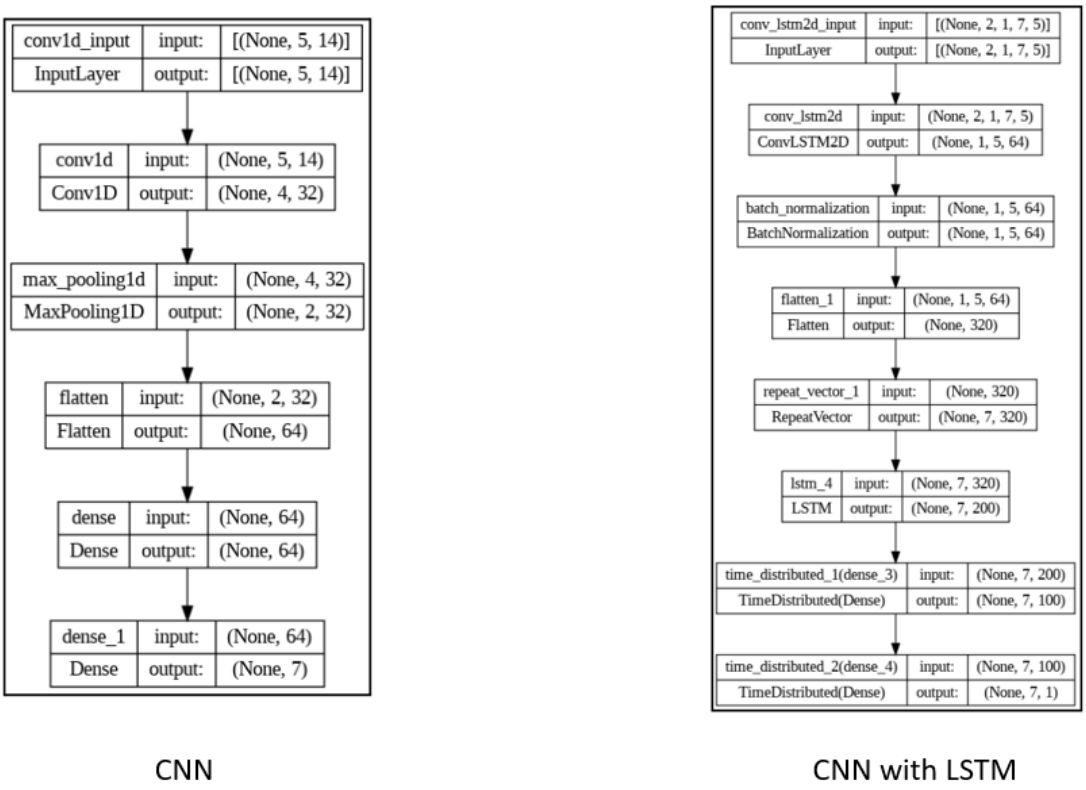

Figure 1: Conventional Neural Network Architectures

After implementing the linear and deep learning models, I start implementing graph neural network based which incorporate a graph convolutional network and a recurrent network. The spatial relationships between various regions and the temporal dependencies throughout time are also considered in this model. The architecture of the graph neural network is shown on Figure 2.

Implementation

My process is broken down into multiple stages. First, I perform candidate zone selection in our dataset and gather them using techniques for crawling and crabbing. Then I have cleaned the data and prepared it for the model. I used hourly data and daily average data to forecast the reading for the following day as well as the following few days for the forecasting challenge. The features of the model were the historical PM reading data as well as the meteorological variables such as temperature, wind speed,

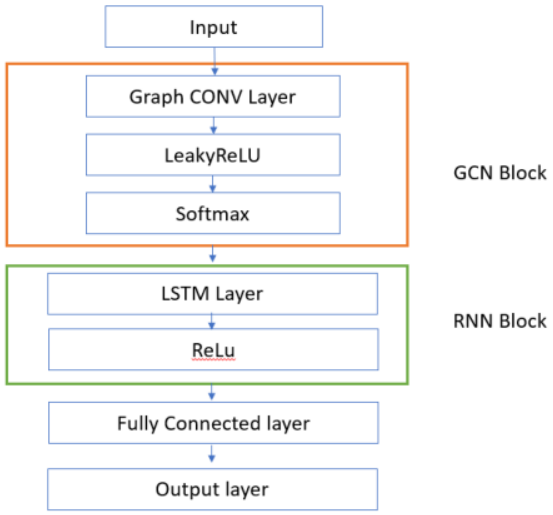

Figure 2: Architecture of Spatial-Temporal Graph Convolution Networks

humidity, precipitation, etc. I started with linear models and deep learning models then extend it to graph neural network.

First, I have used web scraping and crawling to gather the data for the data selection phase. I started by looking into every feasible zone for the 12 states in the Western part of the United States using Berkeleyearth. The selected states are: New Mexico, Texas, Montana, Idaho, Wyoming, Colorado, Nevada, Arizona, Oregon, Utah, California, and Washington. I have searched Wunderground for weather factors that match to the zones before gathering all the data points [wun]. I have observed that the location of the airports only had historical data accessible. Therefore, I took into account the most populated, industrialized cities in each state. Therefore, for the time being, I have only gathered weather data by scraping websites and PM reading data for those 12 zones.

I read the text files containing the raw data as the next step in the data preparation procedure. It was transformed into hourly time-stamped information. I also gathered meta data on the population and location. I have used hourly temperature, dew points, wind speed, humidity, and pressure as meteorological variables. I have used identical cleaning techniques for both meteorological data and fine particles. By averaging the comparable hours on the adjacent days, I was able to compensate for missing hourly measurements of fine particles and climatic factors. Because fine particles exhibit diurnal patterns, I have taken hours from the previous and subsequent three days, for a total of six days, and took the mean of the readings. The entire dataset was then resampled using the daily average. To obtain hourly readings, I simply took the average for duplicate data or downsampled values. Impractical or aberrant values were deemed as invalid and were also imputed. The daily average was then used to resample the full dataset. I averaged duplicate data or downsampled numbers to get hourly measurements. Values that were declared invalid or abnormal were also imputed.

In order to feed our dataset into machine learning models, we prepped it accordingly. I have used the daily average of the past 14 days, or the last two weeks, to anticipate the values for the following day and the following six days. For the purpose of training a linear model, a deep learning model, and a graph-based neural network model, I have used data for meteorological variables from 2004 to 2020 as well as data for PM 2.5.

All twelve of the zone data have been used. I have used the sliding window on PM 2.5 and weather factors to prepare the data and create a feature and target dataset for a machine-learning model. Only the PM 2.5 reading was taken into account for the target variable. For graph neural network, I have used approximately 10 years data for training, 3 years data for validation and 3 years data for testing.

The network is shown as a graph, with the nodes being the states and the edges denoting the distance between them. I have used geo desic distance to represent edge values. Each data point in the time series of the PM 2.5 data indicates the PM 2.5 value at a certain time interval. The spatial and temporal representations are combined to create a spatio-temporal graph. The time series data is attached to the nodes in the graph, which also contains edges that are connected by their respective attributes. I have used the formula from the paper "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting" to calculate the adjacency matrix from the distances matrix [YYZ17]. When the distance between two nodes in the graph is less than a predetermined threshold, the paper presume that there is an edge between them. I have selected 12 states as mentioned earlier and these zones acted like nodes in our dataset. I have an almost fully connected graph which has 128 edges. Some of the edges have been removed because of the threshold. I calculated the square of the distances between the nodes after dividing the distance matrix by 10,000. It then applies a Gaussian kernel with a width set by the value of sigma to the distances matrix. The epsilon value, which serves as a threshold to assess if there is an edge between two nodes, is compared to the kernel values.

Using Glorot uniform initialization, I first initialize the weight matrix with random values. The representations of the nodes in the graph convolution are calculated using this weight matrix. Then, I have an aggregation function that returns the aggregated representation for each node for each of the neighbors' representations. I multiplied the input features tensor with the weight matrix to calculate the representation for each node. A method I've written calculates the messages each node gets from its neighbors. This is accomplished by obtaining neighbor representations using edge information from the graph and then using the aggregate method to apply the provided aggregation function. I have used **mean** as the aggregation type and **concatenation** as the combination type. To get the aggregated message for each node, the resulting tensor is then multiplied by the weight matrix. The GCN layer has since undergone a forward pass. It computes the aggregated messages, updates the node representations with the aggregated messages, and then computes the nodes' representations. The layer's output is the tensor that was produced. Finally, the output of GCN layer is passed to LSTM layer. The activation function used in the LSTM layer is ReLU.The LSTM layer is followed by a dense layer and I have finally get the output.

I used multiple types of performance indicators for time series forecasting after creating the models. I chose to utilize Root Mean Square Error (RMSE) since it will penalize predictions with significant variance from the truth more severely. The percentage deviation is calculated using the mean absolute percentage inaccuracy. In order to assess the accuracy of forecasts while ignoring outlier errors, I have also utilized median absolute error.

Results and Discussion

The results of our experiments are shown in Table 1. I have predicted three years PM 2.5 value using the previous 12 years of data. When I have used a deep learning model, I could not consider the weather data for other stations. I have only considered the Seattle area data to predict the PM 2.5 value. I found out that using other states data does not give us a better model. So, I have only considered Seattle data to predict the PM 2.5 value in case of linear and deep learning models. The result is shown in figure 3 and 4.

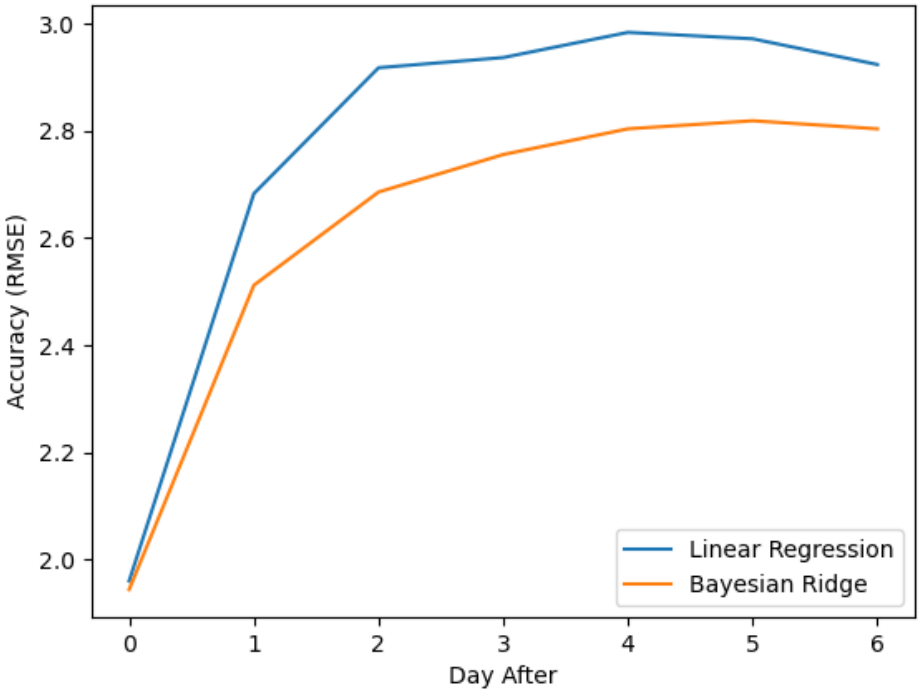

Figure 3: Performance of Linear models

From 3, I can see that bayesian ridge gave us better performance because it has lower RMSE. In case of deep learning model, the CNN model gave us best result. Using the linear model and deep learning model, I saw a common pattern in which the immediate day prediction is more accurate than the rest of the six days. Conventional deep learning approaches for forecasting do not explicitly model the spatial dependencies between different sites in the network, which may limit their ability to capture the complex interrelationships between different network components. Furthermore, conventional deep learning techniques may be incapable of identifying the long-term temporal correlations in the data, which could lead to inaccurate forecasts.

Figure 5 shows the result of GCN with LSTM model result along with the best model from linear and deep learning model. we can see that graph based neural networks gave us the best model. In our GCN along with the LSTM model, I have considered all the 12 regions data and predicted the value of PM 2.5 in Seattle for the last twelve years. The Spatial-Temporal Graph Convolution Networks method beats other approaches in

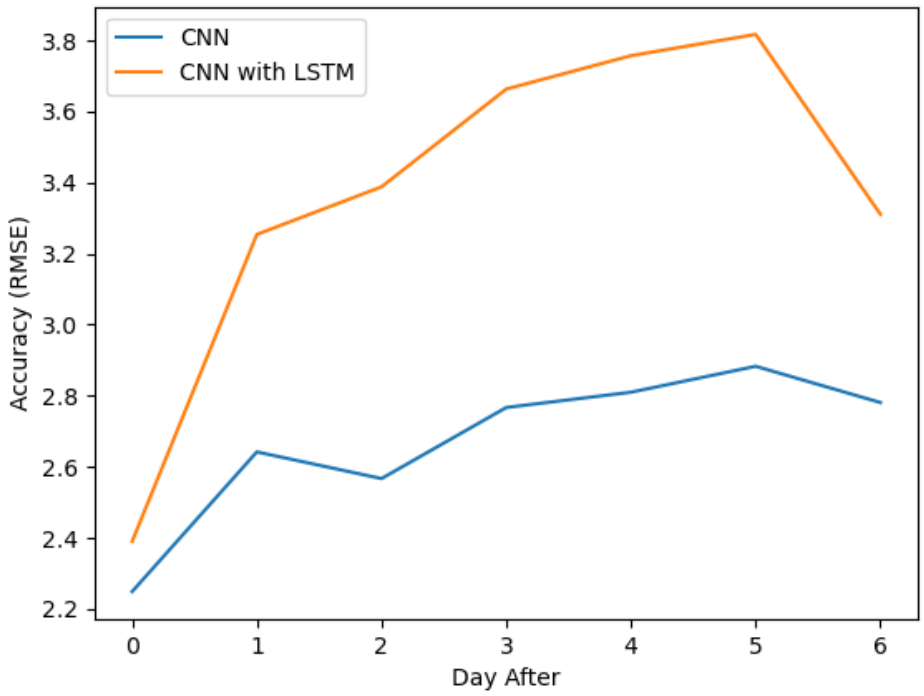

Figure 4: Performance of CNN models

Models	**0** days after	**1**	**2**	**3**	**4**	**5**	**6**
LinearRegression	1.96	2.683	2.918	2.937	2.984	2.972	2.924
BayesianRidge	1.944	2.512	2.686	2.756	2.804	2.819	2.804
CNN	2.249	2.642	2.567	2.767	2.81	2.883	2.781
Encode Decoder	2.176	2.909	3.218	3.363	3.524	3.595	3.62
Conv LSTM	2.39	3.254	3.388	3.663	3.757	3.817	3.311
GCN with LSTM	**0.338**	**0.361**	**0.362**	**0.433**	**0.433**	**0.538**	**0.413**

Table 1: RMSE for linear, CNN and GNN models

weather forecasting for a number of reasons. The ST- GCN model can first capture the spatial links between distinct sites in the network. The model develops the capacity to assess the relative importance of various locations, improving the accuracy of weather predictions. Secondly, the graph-based model may capture the temporal interdependence throughout time by using a temporal attention mechanism. The model acquires the capacity to assess the relative importance of different time steps, which helps to capture long-term interdependence and improves PM 2.5 forecast accuracy. Finally, the graph-based model unifies both spatial and temporal attention mechanisms into a single framework, in contrast to earlier approaches that only take into account either spatial or temporal dependencies, allowing it to better capture the spatio temporal relationships in the PM 2.5 network.

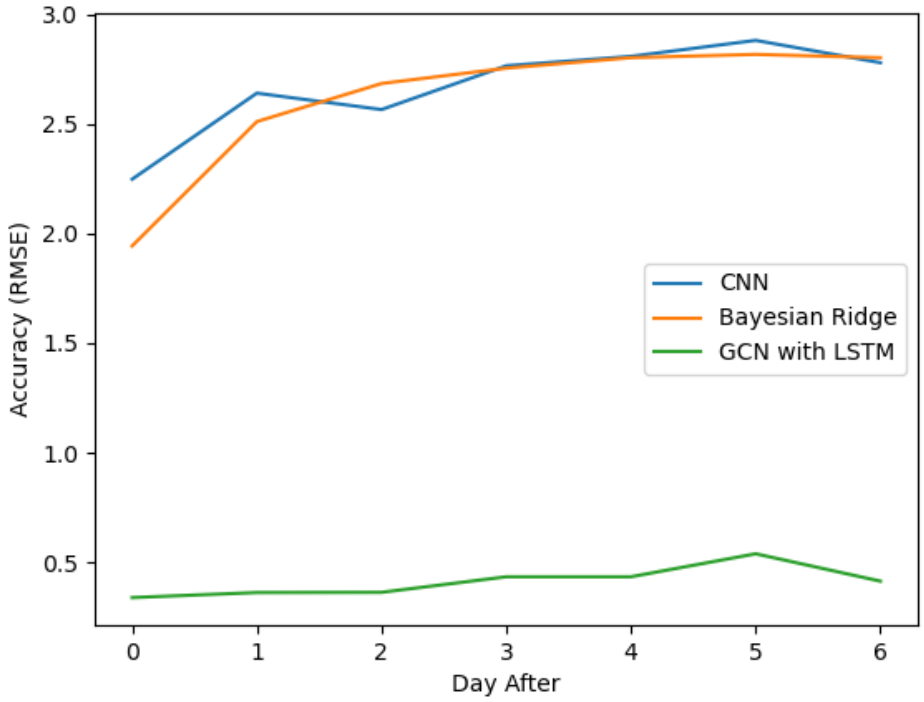

Figure 5: Performance of CNN models

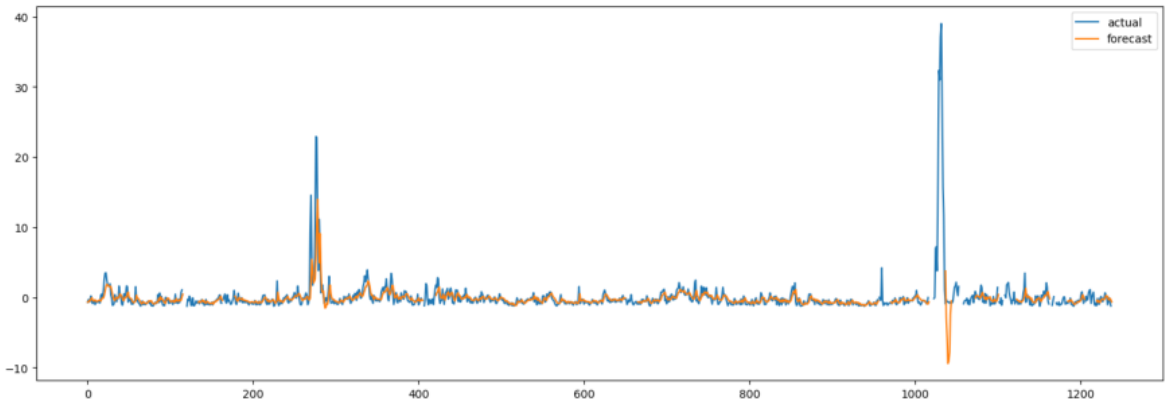

Figure 6: Comparison of time series data and prediction by GCN and LSTM for the Seattle area

Literature Review

Air pollution is a growing concern in developing countries due to rapid urbanization[Aki04]. To address this issue, Xiuwen Yi et al. propose a deep neural network called DeepAir, which includes a spatial transformation component and a distributed fusion network. The former converts sparse air quality data into a consistent input by accounting for spatial correlations between air contaminants [YDL+22]. The latter uses a neural architecture to combine diverse metropolitan data and capture factors that affect air quality,

including meteorological conditions. Another approach to air quality forecasting is the ST-DNN model proposed by Ping-Wei Soh et al., which uses data-driven models to estimate PM 2.5 levels over a 48-hour period [SCH18]. This model can also be applied to other pollutants. The ST-DNN model incorporates an LSTM module to improve first-hour predictions and a CNN module to extract temporal delay factors from nearby target features for longer-term predictions. In another study, Xiang Li et al. propose a new air quality prediction technique based on spatio temporal deep learning (STDL), which accounts for both geographical and temporal relationships.

Graph Neural network also has been using for forecasting purpose like traffic forecasting and building disease models. In order to predict COVID-19 cases in Germany, the study " Combining Graph Neural Networks and Spatio-temporal Disease Models to Predict COVID-19 Cases in Germany" [FDR22] suggests a unique framework that combines graph neural networks (GNNs) and spatio-temporal illness models. A spatio-temporal disease model that depicts the temporal dependencies in the spread of COVID-19 is included in the proposed framework along with a GNN that models the spatial relationships across various regions in Germany. The evaluation of the framework using actual data reveals that it performs better than conventional models for COVID-19 forecasting. The "Graph Neural Networks for Traffic Forecasting"[RBO21] paper, on the other hand, suggests a method for forecasting traffic using graph neural networks (GNNs). The suggested method visualizes the road network as a graph and applies GNNs to understand the intricate spatial relationships between various road segments. On real-world traffic datasets, the technique is assessed and found to perform better than conventional traffic forecasting methods.

The paper "Spatial-Temporal Graph Attention Networks: A Deep Learning Approach for Traffic Forecasting" makes its main contribution in the form of a new deep learning framework called Spatial-Temporal Graph Attention Networks (STGAT), which can model the spatial and temporal dependencies in traffic data by using a graph attention mechanism [ZJL19]. They offer a cutting-edge method for traffic forecasting that considers the interconnections between various traffic sites in a traffic network, allowing for more precise estimates of traffic flow.

A novel deep learning framework for traffic forecasting is proposed in the study "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting" that takes into account the spatio-temporal correlations in traffic data [YYZ17]. The method employs a graph convolutional neural network (GCN) that works with traffic graphs, where the nodes stand for traffic locations and the edges for the movement of traffic between those sites. By performing a convolutional operation on the traffic graph at various time steps, the research offers a spatio-temporal GCN that combines both spatial and temporal information into the model. The model additionally makes use of a residual connection to speed up training and an attention mechanism to determine the relative relevance of various nodes and edges in the traffic graph.

Conclusion

I have predicted the PM 2.5 value using the last twelve years data from 12 different state. From the result analysis, we have seen that a graph-based neural network performs best when we need to take into account both spatial and temporal interactions. We have

used linear and deep learning models as base line models and saw that Graph based neural network outperform these model significantly. One of the new future work from this project would be graph based node prediction. While collecting our data, I have observed that we have data only for a particular station. However, we do not have the data of surrounding places of the station. Graph based prediction can help us to predict nearby places weather information. We can introduce new nodes in our graph and the edge will represent the distance between the station to our desired location where we want to predict the weather information. It will give us a better model for predicting the weather information of the parks or school areas that are situated a certain distance from the weather data collection station. For graph based model, we did not consider the meteorological variable. However, it gave us better models than linear and deep learning model which considered these meteorological variable. We can also incorporate these meteorological variable and analysis how the graph convolutional network performs.

References

[Aki04] Hajime Akimoto. Global air quality and pollution. *Science (New York, N.Y.)*, 302:1716–9, 01 2004.

[ber] Air quality real-time map. https://berkeleyearth.org/archive/air-quality-real-time-map/.

[FDR22] Cornelius Fritz, Emilio Dorigatti, and David Rügamer. Combining graph neural networks and spatio-temporal disease models to improve the prediction of weekly covid-19 cases in germany. *Scientific Reports*, 12:3930, 03 2022.

[pan] Air pollution dropped during pandemic lockdowns. https://uh.edu/news-events/stories/2022-news-articles/may-2022/05092242022-uh-covid-air-pollutant-pollution-ghahremanloo-choi.php.

[RBO21] João Rico, José Barateiro, and Arlindo Oliveira. Graph neural networks for traffic forecasting. 04 2021.

[SCH18] Ping-Wei Soh, Jia-Wei Chang, and Jen-Wei Huang. Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations. *IEEE Access*, PP:1–1, 06 2018.

[wun] Weather underground. https://www.wunderground.com/.

[YDL+22] Xiuwen Yi, Zhewen Duan, Ruiyuan Li, Junbo Zhang, Tianrui Li, and Yu Zheng. Predicting fine-grained air quality based on deep neural networks. *IEEE Transactions on Big Data*, 8(5):1326–1339, 2022.

[YYZ17] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.

[ZJL19] Chenhan Zhang, JQ James, and Yi Liu. Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting. *IEEE Access*, 7:166246–166256, 2019.

Appendix

Code base

https://emailwsu-my.sharepoint.com/:f:/g/personal/abrarakhyer_abir_wsu_edu/EieaNwzS5E9PkFEm68Sy3ZkBOWYxxwfWQE8zc3jbfmgxcQ?e=MAahPX

GNN Detailed Results

Days	model	mae	mse	rmse	mape	mdae
0	GNN	0.264	0.114	0.338	63.131	0.254
1	GNN	0.335	0.13	0.361	87.615	0.331
2	GNN	0.32	0.131	0.362	79.837	0.287
3	GNN	0.37	0.188	0.433	86.128	0.267
4	GNN	0.384	0.187	0.433	74.438	0.336
5	GNN	0.501	0.289	0.538	69.559	0.502
6	GNN	0.37	0.171	0.413	50.886	0.295