

Using an Autoencoder Model to Characterize Air Pollution

Nicolas Shannon¹, Vandana Nunna Lakshmi²

¹Department of Computer Science, Loyola University Chicago, Chicago, IL, US

²Department of Computer Science, University North Texas, Denton, TX, US

Abstract

One of the most important ways of combating climate change is to better understand the way that polluting gas levels are geographically distributed. We characterize cities by the amount of pollution present using an autoencoder model. By representing the level of eight toxic gases present in numerous cities across the United States, we explore some of the more prominent features that characterize urban air pollution. We compare two techniques for reducing the dimensionality of our input - Principle Component Analysis (PCA) and a single-layer autoencoder model. We found that the autoencoder was able to explain a larger percentage of the explainable variance for nearly all of the eight gases and particulates. Moreover, we found that the first two hidden dimensions represented 50% of the total explainable variance for Tropospheric Ozone (O_3). Because of this, we compared the clustering of cities along the two axes using several geographic categories, including by region and by population. We found that although there was significant clustering by region, there was no such clustering for the top two hundred most populous U.S. cities.

1 Introduction

For the past several decades, air pollution has become a major concern for human health and the environment. The EPA sets national air quality standards for six major air toxins: ground level Ozone (O_3), Carbon Monoxide (CO), Sulfur Dioxide (SO_2), Lead (Pb), Nitrogen Dioxide (NO_2), and particulate matter (PM). The sources of these emissions vary, but the two largest contributors are industrial processes and motorized vehicles. (EPA 2009).

We wanted to explore how various polluting gases can be characterized with respect to location in an effort to provide researchers and policy makers with the context and tools necessary to make informed decisions. By creating an embedding of time series pollution data for numerous cities throughout the United States, we explore the more interesting features of the model as well as apply it to tangential areas such as the pollution generated by industrial farming and animal husbandry activities.

In this paper we use two different methods for data compression, autoencoders and principle component analysis (PCA). The models are trained using time series data for eight different polluting gases. The trained models are compared in terms of efficiency and data loss. The fluctuations in explainable variance when predicting a target allow us to quantify the discrepancy between the model results and actual data. In this case, the models are trained to predict a single feature out of the time series data, which represents the daily pollutant averages for a single day. Once the training process is complete, the trained models are employed to create embeddings for various encoder dimensions.

Given the relatively short time span of our features along with the fact that pollution levels do not tend to vary drastically day-to-day, it is evident that there is a strong correlation among features. While a healthy correlation among features is encouraged, abnormally strong values reflect redundancy and end up being a burden on the model. (Yu and H. Liu 2003). Ideally there would be high correlation between features and the problem class and a low correlation among the features themselves.

2 Related Work

The idea of generating embeddings for a location is a highly potential expanse for research. The two major forces behind its competency are the fact that large volumes of data is being effectively compressed and that these embeddings, when plugged into any model could effectively represent any given location. A lot of research is being done on location embeddings in recent times.

For instance, location embeddings are being generated based on human-mobility data (Crivellari and Beinart 2019) to understand the behavioral proximity between different locations irrespective of their spatial distances.

The sequence of locations that are frequently travelled were fed to a Skip-gram Word2vec model to generate the location embeddings.

Another approach that was implemented to leverage location embeddings is GPS2Vec (Yin et al. 2019). This application consists of a neural network that would generate embeddings for a location, based on the semantic contexts that are sourced from the user content published on multiple digital platforms. With the integration of these embeddings, a successful geo-tagged image classification task is demonstrated.

In another effort (Yang et al. 2019), the location embeddings derived from user content such as addresses, phone numbers, images, place names and all other details from multiple individual online posts, are compressed to form low dimensional embeddings to solve the place duplication issue. Deduplication of locations, is a paramount task for any digital platform since they provide a place graph to its users, which is the result of merging location data from multiple sources. This merge leads to duplicity since different sources could hold different names for the same place. By applying K-NN search, pair-wise duplication prediction on these generated location embeddings, the application can identify the duplicates and eliminate them to result in an efficient place graph.

In 2018, a new approach, “DeepMove” is proposed (Zhou and Huang 2018). This implementation employs Trip model and OD model to learn latent representations of places. The approach is successfully applied to New York city yellow taxi dataset – “includes pick-up/drop-off times, the number of passengers, pick-up/drop-off locations represented by longitude and latitude, trip distance, rate types, payment type, and itemized fares” (ibid.). Similarities among locations in New York city are derived effectively with “DeepMove”. To resolve the dissonance that arises from the fusion of geographic and semantic features of a location, an unsupervised learning implementation had been demonstrated that would derive location embeddings from human trajectories (Ouyang et al. 2020).

With the ever-growing footprint of digital and social media platforms in everyday life of an individual, a lot of research is happening to generate embeddings based on the content posted in different locations. LeGo-CM is one such effort that would generate embeddings of geotagged tweets (Wei, Anjaria, and Samet 2019). The resulting embeddings could be applied to understand the kind of content each location reflects.

Venue2Vec is another idea that is implemented to enhance the location-aware services in various platforms (Xu et al. 2020). This model is trained on semantic data, temporal-spatial context and sequential relations among locations to generate embeddings. This makes sure that locations with similar data would fall close, when plotted in a multi-dimensional space, based on the input features. Hence, the model is effective in predicting the next and future check-in location.

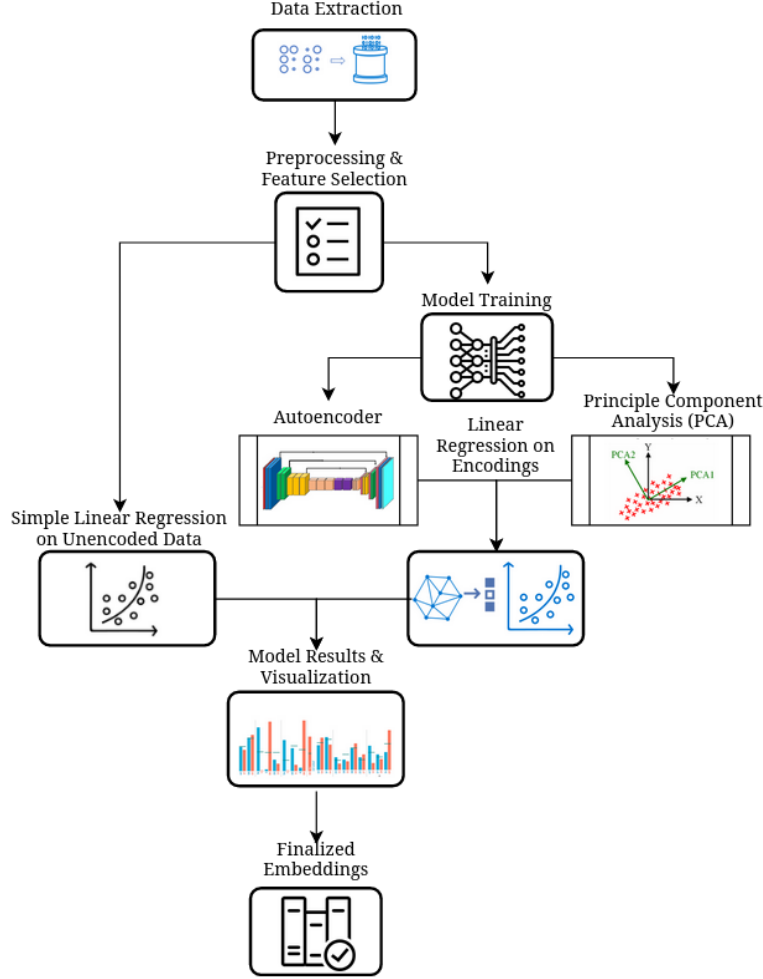
Community Flow prediction is another area of interest where embeddings are a major problem solver. Geo-contextual Multitask Embedding Learner (GMEL), is a model that uses “Graph Attention Network” to encode geographical contexts and spatial correlations to generate location embeddings (Z. Liu et al. 2020). These embeddings are further used to predict the community flow that serves as a vital factor in policy establishment and metro planning.

3 Methodology

In order to execute the proposed idea of pollution data compression, we orchestrated the data through multiple models. We began by applying a linear regression the uncompressed data, which gave us the requisite baseline performance scores to compare the embeddings with. Next, we tuned, trained, and tested eight autoencoder (AE) models and eight principle component analysis (PCA) models. The model metrics of the AE and PCA models are evaluated, and the resulting embeddings are applied to two further linear regressions. Lastly, the scores of the AE and PCA models are compared against the baseline scores to evaluate the performance, efficiency and quality of the embeddings.

As previously mentioned in the introduction, the correlation between features was revealed to be rather high. In particular, days one hundred thirty-five to one hundred-eighty nearly all have a correlation coefficient greater than 0.75. This resulted in some overfitting which can be seen in the autoencoder and pca explainable variance scores, which were greater than the unencoded scores.

Figure 1: Data Pipeline



3.1 Data Collection and Preprocessing

The dataset was created using OpenWeather’s Air Pollution API. The API offers approximately six months of historical pollution data with hourly granularity. The hourly pollution data does not fluctuate drastically over a given day, so we opted to derive the daily average. Doing this made the feature set proportional to the number of cities we had data for; it also reduced the computational load without significantly affecting performance.

In a preemptive effort to explore the dataset, we calculated the Pearson correlation coefficient of the normalized temporal data. The Pearson correlation coefficient allows us to observe the strength of the linear correlation between features.

$$r_{XY} = \frac{\text{cov}(X, Y)}{\sigma_X \cdot \sigma_Y}$$

The correlation matrix of the resultant features illustrates that there are some features with strong correlation bonds, largely toward the end of the feature list. This reflects the fact that the average proportion of the pollutant is similar on some days. The short term consecutive increase in correlation demonstrates that the sources of emission follow a pattern over certain months of the year. Another possibility is that the pollution levels are influenced by some natural phenomenon. Either possibility is difficult to prove given the relatively short time frame of our data. However, there is evidence suggesting that temperature increases have a positive correlation with rising pollution levels, which could explain the stronger correlation present in the summer months of the dataset (Jaya-murugan et al. 2013).

We looked at data for the eight pollutants over a six month time frame: January 27th, 2020 through June 6th, 2021. The final dataset includes just over eighteen thousand cities in the United States and other U.S. territories.

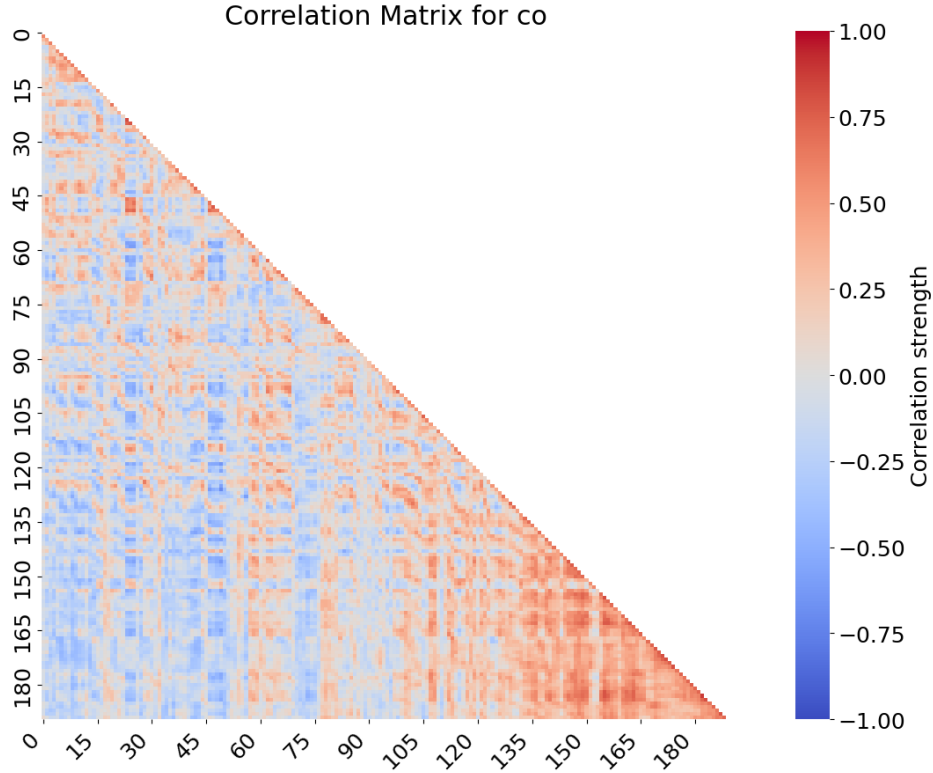


Figure 2: Correlation Matrix for Carbon Monoxide (daily averages for cities throughout the US)

The eight polluting gases and particulates that we selected were CO, NO, NO₂, O₃, SO₂, NH₃, PM_{2.5}, and PM₁₀. The air quality index (aqi) was also included, but we chose not to include it because it is overall less sensitive to change and is unlikely to reveal any long-term patterns over such a short time span. Additionally, we found it easier to focus on a single gas when developing the models, so we chose to concentrate on O₃ during the engineering process. The same process was then applied to the other seven gases and particulates after the development was completed.

Table 1: Dataset Characteristics

	OpenWeather Pollution API
Sampling period	11/27/20 to 06/05/21
Time interval	Daily
Cities	18,526
Features	190
Component gases	8
Dependent variable	Last day of the period (06/05/21)

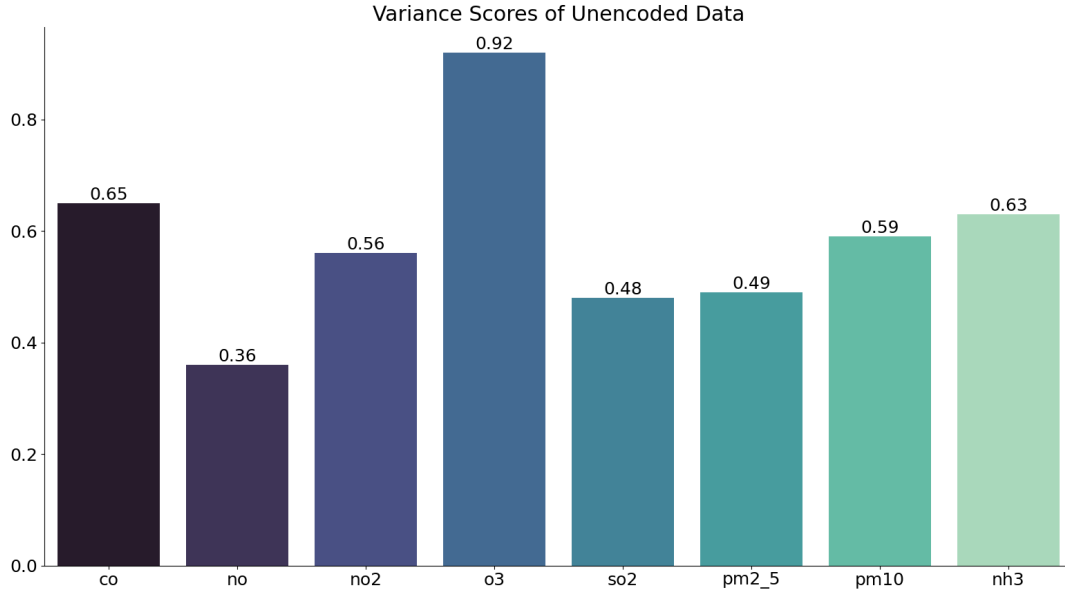
The original data was unit normalized so that the gradient descents could converge more quickly while still preserving the original differences in range. Each step was saved separately to provide expeditious access to both the normalized and non-normalized data.

3.2 Model Validation

To perform the initial linear regression on uncompressed data, we choose the the daily pollution values of one random day from the dataset while accomodating to train the model with the rest of the features. K-fold cross validation is applied on the model after normalizing the training data. This helps with evaluating the minimum, average, and maximum regression scores of the original data that was not condensed. These scores create a baseline that

help us in comparing the scores that were achieved after applying the reduction techniques.

Figure 3: Unencoded R2 Scores



3.3 Autoencoder Model

The autoencoder model used consists of a single encoded and decoded layer. The data was split into a train and test set at the beginning of the testing cycle, and was further separated into a validation set for each given dimension. The validation set was essential for updating the model weights after every epoch. After the model was trained and validated, we created the finalized embeddings using the full dataset. We created embeddings for several of the most significant dimensions based off their increase in explained variance scores.

Table 2: Model Splits

Constant Hyperparameters	
Training Size	14,080
Testing Size	3,705
Validation Size	741
Total Size	18,526

A grid search was performed for each model to determine the optimal hyperparameters. The following hyperparameters were tested:

- *Learning Rate*: 0.0001, 0.001, 0.01, 0.1
- *Batch Size*: 32, 64, 128, 256
- *Epochs*: 10, 50, 75, 100

K-folds cross validation was implemented to validate the grid search using five folds. Additionally, the training split for each dimension was separated into a validation set. Each hyperparameter was selected based on the highest coefficient of determination that appeared in each of the five folds for every given dimension. The most frequently occurring combination of hyperparameters was then used to create the finalized embedding. The lowest coefficients of determination were saved and compared with the highest values to give an upper and lower bound of expected values.

After some preliminary testing, it was decided that the activation, optimizer, and loss functions should remain constant. The exponential increase in computing time this would have on the grid search drastically outweighed any minor performance gains that a thorough optimization of these functions may have yielded.

Table 3: Constant hyperparameters used across all autoencoder models

Constant Hyperparameters	
Activation function	tanh
Loss function	Mean Squared Error
Optimizer	Adam

Moreover, grid search was only performed on a select number of key dimensions rather than on all one hundred ninety dimensions. The first ten dimensions, as well as every tenth dimension until one hundred-twenty were tuned. For each intermediary dimension that did not have a set of key hyperparameters, the hyperparameters of the previous key dimension was used. Once again, the tradeoff between minor performance gains and large compute times led us to adjust the scope of the hyperparameter tuning.

Table 4: Grid Search Results for Carbon Monoxide

CO grid search for key dimensions 1-10				
dimension	r^2	learning rate	batch	epochs
1	0.22	0.01	64	10
2	0.44	0.0001	256	50
3	0.49	0.001	64	50
4	0.52	0.0001	32	50
5	0.56	0.0001	32	100
6	0.57	0.01	32	10
7	0.60	0.0001	32	150
8	0.63	0.0001	64	150
9	0.62	0.0001	32	150
10	0.62	0.0001	64	100

CO grid search for key dimensions 20-120				
dimension	r^2	learning rate	batch	epochs
20	0.62	0.01	64	50
30	0.64	0.01	128	75
40	0.67	0.01	256	100
50	0.68	0.01	64	75
60	0.69	0.001	32	150
70	0.70	0.001	64	150
80	0.71	0.001	32	150
90	0.71	0.01	64	100
100	0.72	0.01	64	150
110	0.72	0.01	32	75
120	0.73	0.001	64	100

3.4 Principle Component Analysis

The PCA model was trained with k-fold cross-validation using five folds. The dataset was split into a training and test set. The trained PCA models were used to create embeddings at different scales of dimensionality based on the number of input components. We prioritized the first ten dimensions because they generally represent the largest increase in the percent explained variance.

4 Results

We compared the percent explained variance across all one hundred ninety dimensions for the encoded values generated by the PCA and AE models. While PCA was overall less noisy, we found that the autoencoder model achieved a much higher explainable variance for the first several dimensions.

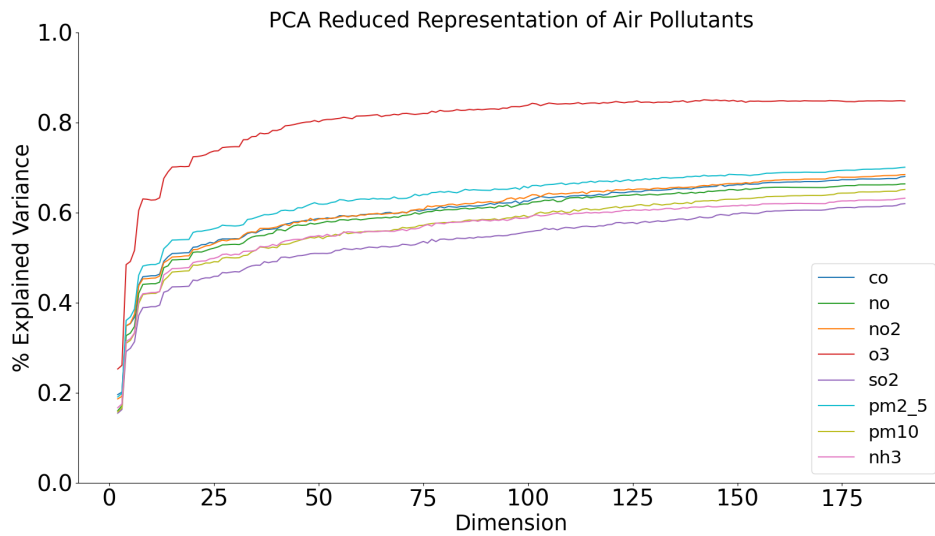


Figure 4: PCA Dimensionality Reduction

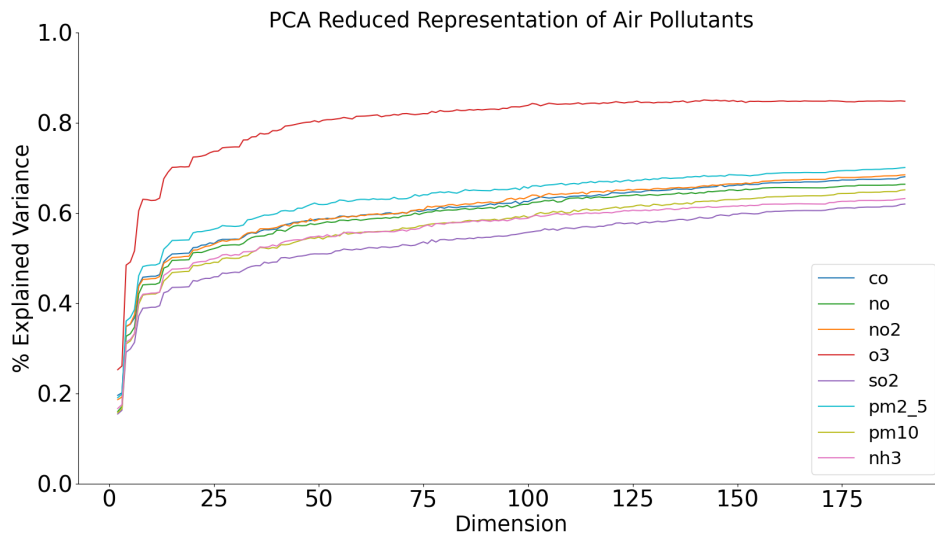
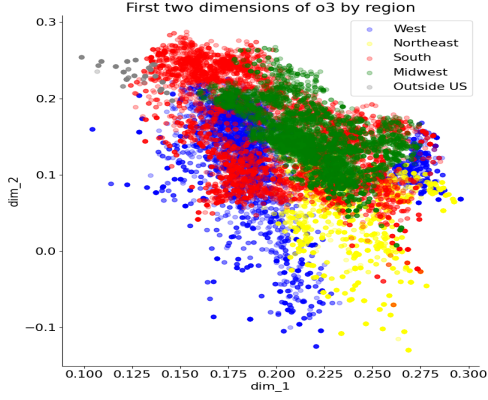


Figure 5: Autoencoder Dimensionality Reduction **Placeholder for AE**

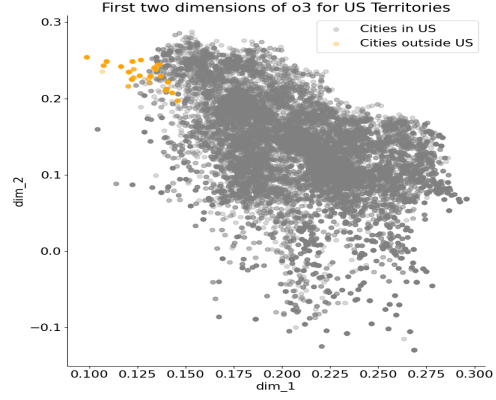
In nearly every case shown in figures 4 and 5, the autoencoder model shows consistently higher explainability than the PCA. Tropospheric Ozone (O3) in particular shows a massive jump of over fifty percentage points ex-

plainable variance in the first two dimensions. For this reason, O3 was the primary model for the subsequent visualizations.

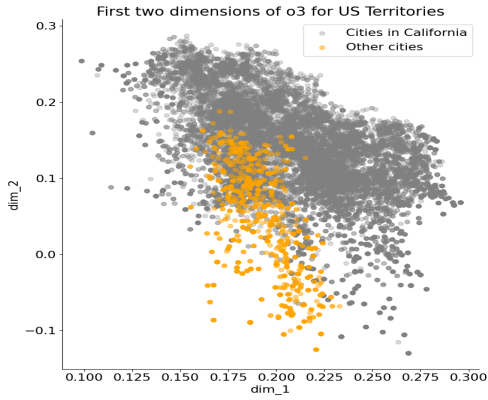
In order to better visualize how the hidden layers distill information, we clustered the cities based on the first two encoded dimensions. Additionally, a number of location heuristics were applied to further understand the distribution of cities. Figure 6 shows how the embeddings are clustered by regions, as well as by individual states and territories.



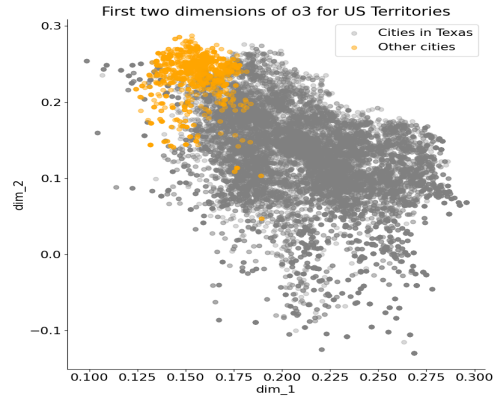
(a) Model characterization grouped by region



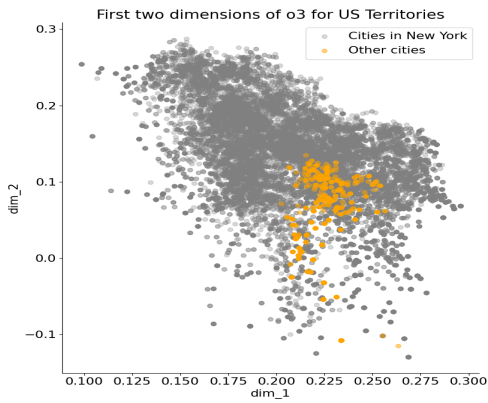
(b) Model characterization of U.S. Territories



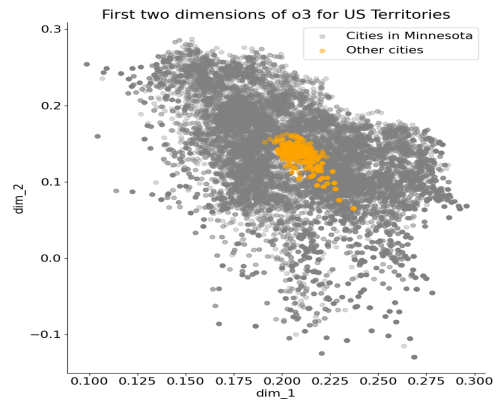
(c) Model characterization of California



(d) Model characterization of Texas



(e) Model characterization of New York

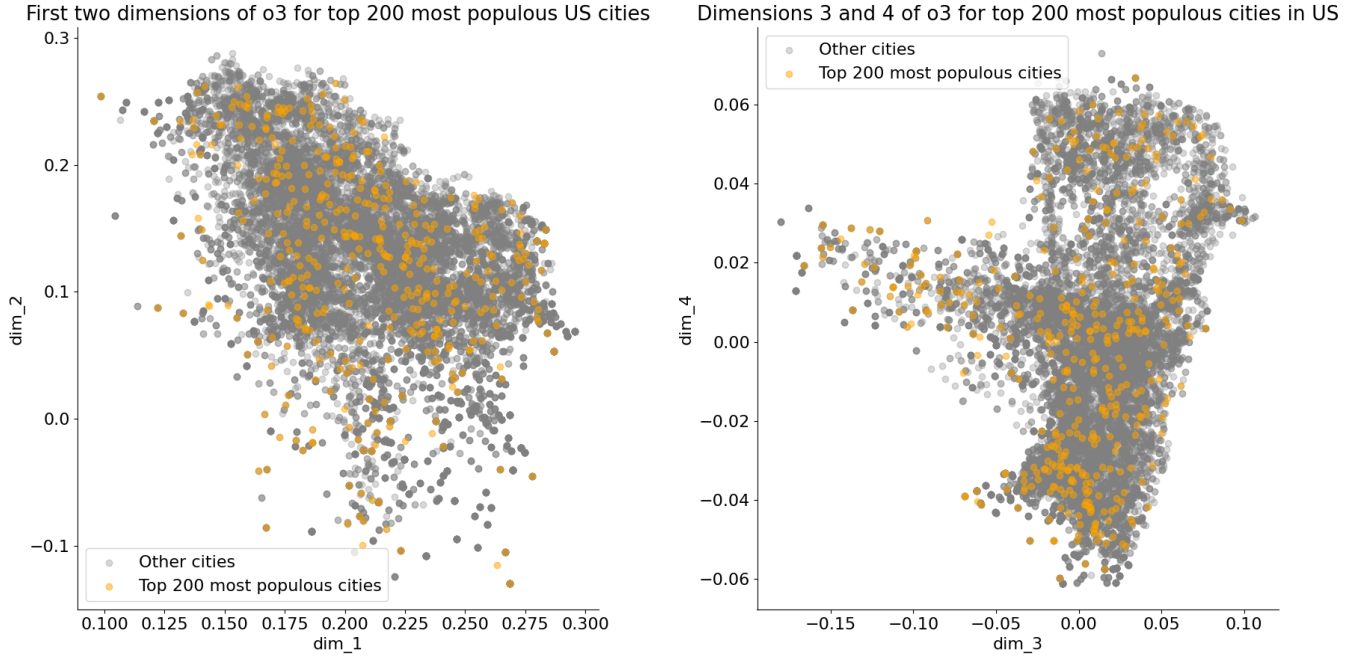


(f) Model characterization of Minnesota

Figure 6: Scatter plots of outlier and highly-populated cities over first two dimensions

The model’s characterization appears to be heavily dependent on geographic location. When grouped by region, one can see that there is a fairly tight distribution, especially among the Midwest and Northeast regions. Puerto Rico is also tightly clustered, but the territory contains a limited 211 representative data points.

Curiously enough, population size does not seem to be one of the most succinct features of the model. Figure 7 shows the top two hundred most populous cities in the US for the first four dimensions of the model. The first four latent dimensions of O3 account for nearly 60% of the total explainable variance. However, the top 200 cities appear to be randomly dispersed throughout, suggesting that population alone has less of an effect on the model. It is possible that population is represented indirectly



(a) Model characterization grouped by region

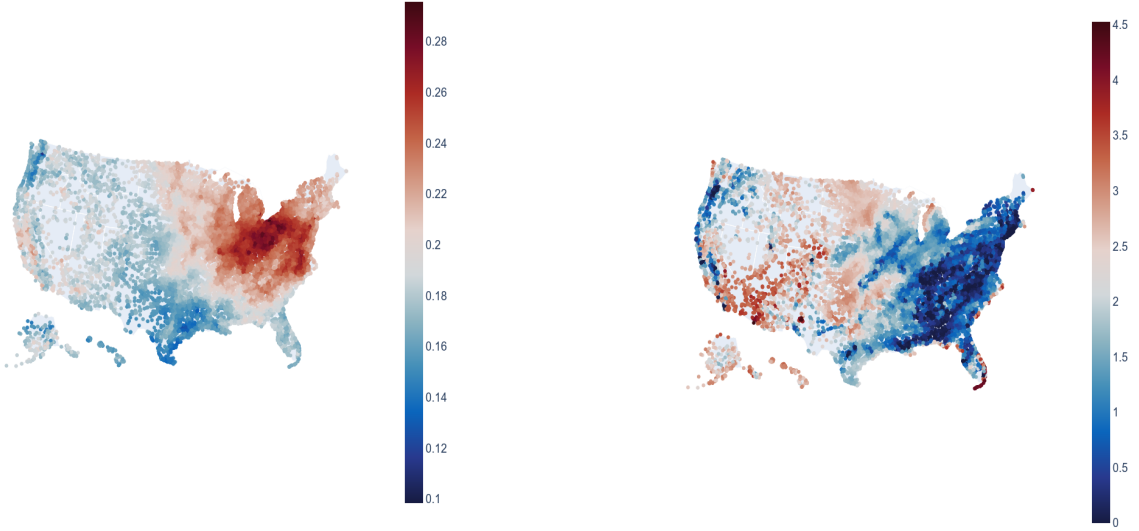
(b) Model characterization of U.S. Territories

Figure 7: Scatter plots of outlier and highly-populated cities over first two dimensions

It is difficult to determine why the model does not seem to consider population size. One reason for this may be that temperature and climate "overpowers" any influence a city’s size may have. Any number of secondary effects related to climate and location could pull the model away from considering population. Moreover, The time frame of our data spans from winter 2020 to early summer of 2021, which is not long enough to fully capture seasonality. Given a larger data set, it is likely that both population and seasonality would play much larger roles.

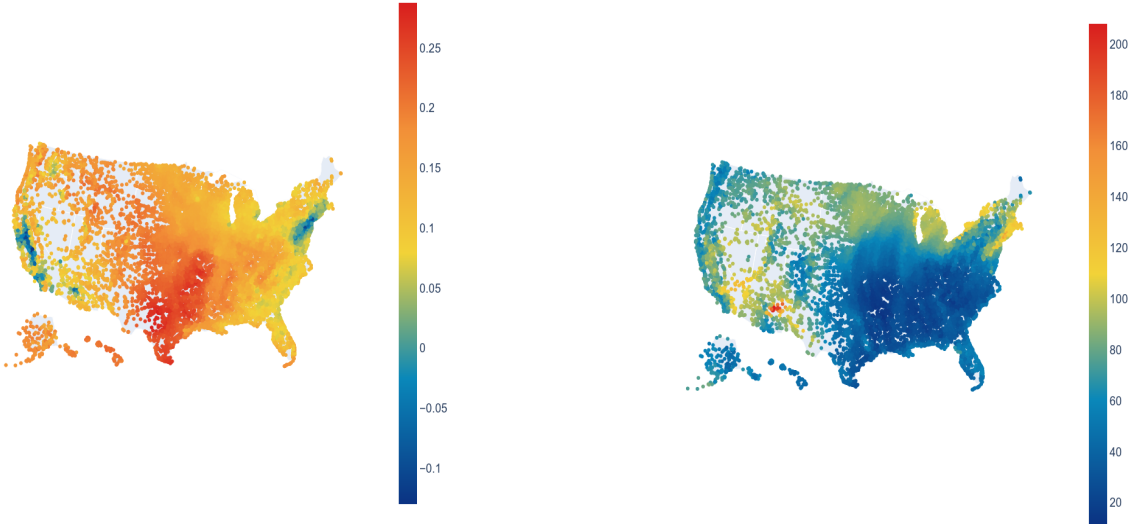
Due to the importance of geography in the embedding, we created several geographic scatter plots to visualize both the embedding values and pollution levels with respect to location.

Figure 8: Geographic comparison of model embedding values versus daily pollution levels



(a) O3 values of the first encoded dimension (units of compression)

(b) O3 pollution values for the first day of data set ($\mu\text{g}/\text{m}^3$)



(c) O3 values of the second encoded dimension (units of compression)

(d) O3 pollution values for the last day of data set ($\mu\text{g}/\text{m}^3$)

5 Discussion

The proposed model provides us with many insights on the most succinct features that characterize urban air pollution in US cities. Based on location identifiers such as city name, state, country, and the geographic coordinates, we are able to infer much of the explainability of the model. However, more specific external data is needed to fully uncover these features.

One study (Bai et al. 2019) uses a stacked autoencoder approach and focuses on the seasonality of pm2.5 levels. The model considers both the seasonality and temporality of pm2.5 levels and supporting meteorological data. The research group was able to accurately predict the pm2.5 levels to the hour for a given season. The pollution autoencoder models we created only consider half a year of data, so we were unable to fully account for seasonal variations in the dataset.

Another study focusing on pm2.5 (Doreswamy et al. 2021) uses autoencoders for spatio-temporal clustering analysis. The group looks at Taiwan for its massive air pollution problem, coupled with the fact that the area’s unique topography makes it difficult to map the distribution of pollution. The model uses approximately six years of sensor data from Eastern Taiwan to analyze the spatio-temporal qualities of pm2.5, as well as the socioeconomic effects of its presence.

A third study (Ma et al. 2019) concentrates on recovering lost pollution mapping data with mobile sensing networks. The proposed framework separates the process of pollution generation and data sampling to allow the model to deal with irregular time and space intervals. To aid in this, a deep learning approach using autoencoders was chosen to handle the irregular sampling. Additionally, a convolutional long short-term memory (ConvoLSTM) structure was utilized because it allows for the capture of both temporal and spatial dependencies. The autoencoders were used to describe the conditional probability of complete data given partial observations, whereas the ConvLSTM was used as a sampling imitator to model the pollution generation.

6 Conclusion

This paper presents the advantages of using a deep learning approach to characterize urban pollution data. The bottleneck architecture of the autoencoder model allows the network to extract the most representative features of air pollution in US cities. Furthermore, the autoencoder model is shown to be more effective at representing lower dimensional latent spaces than principle component analysis. The model may be expanded to apply to many different types of polluting gasses, beyond the eight gasses and particulates that we have already analyzed. The generalized model allows us to codify the influences on urban air pollution, which can be useful for further predictive modeling and policy making.

References

- Bai, Yun et al. (2019). “Hourly PM2.5 concentration forecast using stacked autoencoder model with emphasis on seasonality”. In: *Journal of Cleaner Production* 224, pp. 739–750. ISSN: 0959-6526. DOI: <https://doi.org/10.1016/j.jclepro.2019.03.253>. URL: <https://www.sciencedirect.com/science/article/pii/S095965261930962X>.
- Crivellari, Alessandro and Euro Beinat (Mar. 2019). “From Motion Activity to Geo-Embeddings: Generating and Exploring Vector Representations of Locations, Traces and Visitors through Large-Scale Mobility Data”. In: *ISPRS International Journal of Geo-Information* 8, p. 134. DOI: 10.3390/ijgi8030134.
- Doreswamy et al. (2021). “Spatio-Temporal Clustering Analysis for Air Pollution Particulate Matter (PM2.5) Using a Deep Learning Model”. In: *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 529–535. DOI: 10.1109/ICCCIS51004.2021.9397129.
- EPA (2009). “Carbon Monoxide NAAQS: Scope and Methods Plan for Health Risk and Exposure Assessment”. In: https://www.epa.gov/sites/production/files/2020-07/documents/2009_04_coscopeandmethodsplan.pdf.
- Jayamurugan, Ramasamy et al. (2013). “Influence of temperature, relative humidity and seasonal variability on ambient air quality in a coastal urban area”. In: *International Journal of Atmospheric Sciences* 2013, pp. 1–7. DOI: 10.1155/2013/264046.
- Liu, Zhicheng et al. (Apr. 2020). “Learning Geo-Contextual Embeddings for Commuting Flow Prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01, pp. 808–816. DOI: 10.1609/aaai.v34i01.5425. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/5425>.
- Ma, Rui et al. (2019). “A Deep Autoencoder Model for Pollution Map Recovery with Mobile Sensing Networks”. In: *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*. New York, NY, USA: Association for Computing Machinery, pp. 577–583. ISBN: 9781450368698. DOI: 10.1145/3341162.3349327. URL: <https://doi.org/10.1145/3341162.3349327>.
- Ouyang, Kun et al. (2020). “Unsupervised Learning of Disentangled Location Embeddings”. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN48605.2020.9207324.
- Wei, Hong, Janit Anjaria, and Hanan Samet (2019). “Learning Embeddings of Spatial, Textual and Temporal Entities in Geotagged Tweets”. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL ’19. Chicago, IL, USA: Association for Computing Machinery, pp. 484–487. ISBN: 9781450369091. DOI: 10.1145/3347146.3359108. URL: <https://doi-org.libproxy.library.unt.edu/10.1145/3347146.3359108>.

- Xu, Shuai et al. (2020). “Venue2Vec: An Efficient Embedding Model for Fine-Grained User Location Prediction in Geo-Social Networks”. In: *IEEE Systems Journal* 14.2, pp. 1740–1751. DOI: 10.1109/JSYST.2019.2913080.
- Yang, Carl et al. (2019). “Place Deduplication with Embeddings”. In: *The World Wide Web Conference*. WWW ’19. San Francisco, CA, USA: Association for Computing Machinery, pp. 3420–3426. ISBN: 9781450366748. DOI: 10.1145/3308558.3313456. URL: <https://doi-org.libproxy.library.unt.edu/10.1145/3308558.3313456>.
- Yin, Yifang et al. (2019). “GPS2Vec: Towards Generating Worldwide GPS Embeddings”. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. SIGSPATIAL ’19. Chicago, IL, USA: Association for Computing Machinery, pp. 416–419. ISBN: 9781450369091. DOI: 10.1145/3347146.3359067. URL: <https://doi-org.libproxy.library.unt.edu/10.1145/3347146.3359067>.
- Yu, Lei and Huan Liu (2003). “Efficiently Handling Feature Redundancy in High-Dimensional Data”. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’03. Washington, D.C.: Association for Computing Machinery, pp. 685–690. ISBN: 1581137370. DOI: 10.1145/956750.956840. URL: <https://doi.org/10.1145/956750.956840>.
- Zhou, Yang and Yan Huang (2018). “DeepMove: Learning Place Representations through Large Scale Movement Data”. In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2403–2412. DOI: 10.1109/BigData.2018.8622444.