Tensorflow Final Project

Konstantin Strömel, Tjark Darius, Johannes Claassen

March 2022

1 Introduction

In 2018 the World Health Organisation discovered that 90% of the world's population is suffering from polluted air and that every year around 7 million people die because of air pollution. The WHO also states that it is a critical factor in many serious health problems such as lung cancer, strokes and heart diseases. Air pollution causes around one quarter of the total adult deaths through these non communicable, chronic diseases (1). And even though poor air quality makes respiratory diseases like COVID-19 more dangerous (2), the current pandemic also showed that during the lockdown in 2020 84% of the countries worldwide experienced better air quality than in the previous year (3). So human related emissions can directly influence the air quality and our project is trying to examine these influential factors as well as implementing a model that can predict air quality based on satellite data, instead of costly and time intensive ground-based sensors. This will potentially enable air quality monitoring and management also for low-income countries in Asia and Africa, where most of the pollution related deaths occur. In figure 1 you get an overview of the distribution of monitoring stations. Especially in South America and on the African continent there are very little ground-based measurements. Satellite data could close this big gap in the global monitoring network.

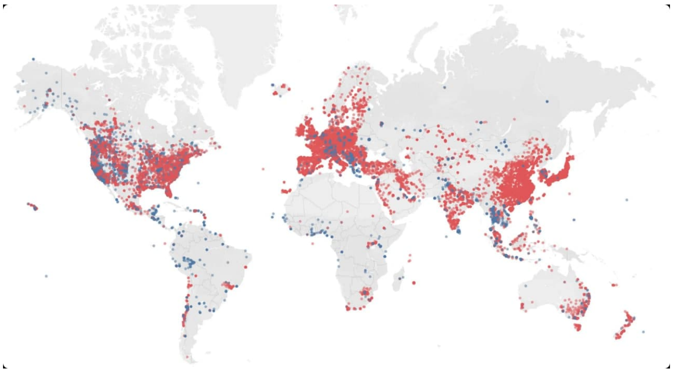


Figure 1: Overview of monitoring stations (2)

2 Task

Our goal is to predict the PM2.5 score for different cities across the globe based on weather and satellite data (4). The PM2.5 score refers to atmospheric particulate matter with a diameter of less than 2.5 micrometers in micrograms per cubic meter air ($\mu g/m^3$). It is one of the most widespread air pollutants, consisting of a mixture of solid and liquid particles suspended in the air. Figure 2 shows how small PM2.5 particles are compared to a human hair or a grain of sand (5).

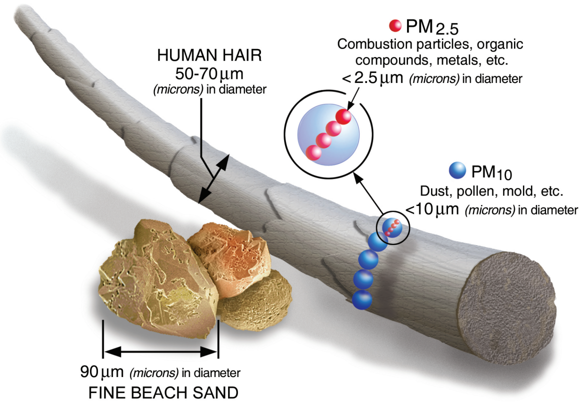
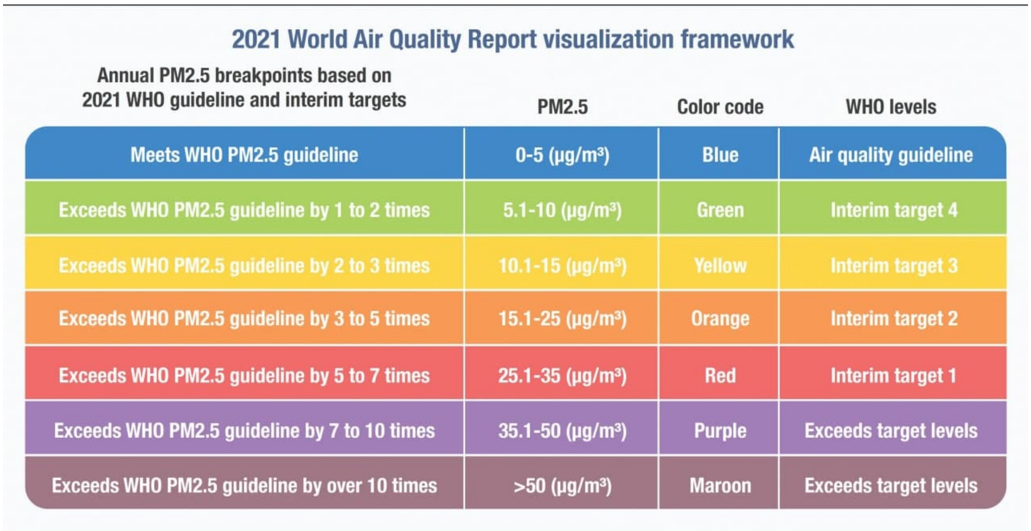

HUMAN HAIR
50-70µm
(microns) in diameter
PM2.5
Combustion particles, organic compounds, metals, etc.
< 2.5µm (microns) in diameter
PM10
Dust, pollen, mold, etc.
<10µm (microns) in diameter
90µm (microns) in diameter
FINE BEACH SAND

Figure 2: $PM_{2.5}$ particles size comparison (5)

The score ranges from 0 to 50 $\mu g/m^3$ per 24 hours. The table in figure 3 shows the WHO guidelines for PM2.5 pollution. It also serves as a color code for the global PM2.5 map in figure 4 (2).



2021 World Air Quality Report visualization framework

Annual PM2.5 breakpoints based on 2021 WHO guideline and interim targets	PM2.5	Color code	WHO levels
Meets WHO PM2.5 guideline	0-5 (µg/m³)	Blue	Air quality guideline
Exceeds WHO PM2.5 guideline by 1 to 2 times	5.1-10 (µg/m³)	Green	Interim target 4
Exceeds WHO PM2.5 guideline by 2 to 3 times	10.1-15 (µg/m³)	Yellow	Interim target 3
Exceeds WHO PM2.5 guideline by 3 to 5 times	15.1-25 (µg/m³)	Orange	Interim target 2
Exceeds WHO PM2.5 guideline by 5 to 7 times	25.1-35 (µg/m³)	Red	Interim target 1
Exceeds WHO PM2.5 guideline by 7 to 10 times	35.1-50 (µg/m³)	Purple	Exceeds target levels
Exceeds WHO PM2.5 guideline by over 10 times	>50 (µg/m³)	Maroon	Exceeds target levels

Figure 3: WHO PM2.5 guideline (2)

For this task we had to solve a so-called ”out of distribution” problem, because we had to predict target values for cities that were not part of the training data. In the following section we will describe the available training data and how we used it for our model.

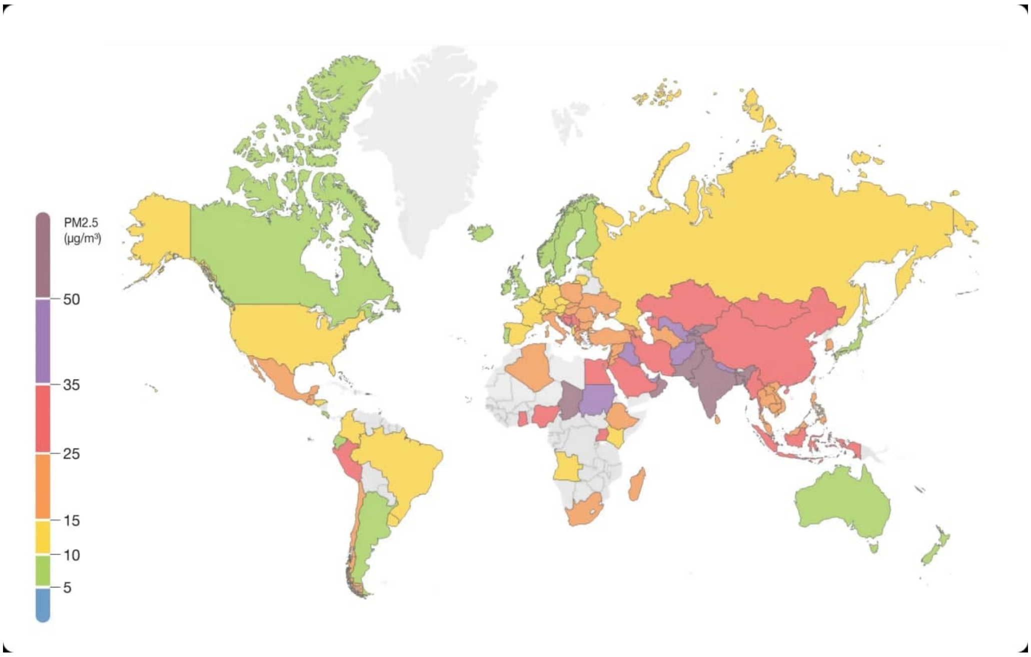


Figure 4: Global PM2.5 map (2)

3 Data

The data provided by *Zindi* comprises of three different main sources of data. First, the target values, which are five columns in total, are obtained from ground-based air quality sensors. Second, there are six columns of general weather data produced by the Global Forecast System, operated by the U.S. National Weather Service. And third, there are in sum 68 columns of satellite data, provided by the Sentinel-5 Precursor project.

3.1 Target data

The column `target` displays the PM$_{2.5}$ particle concentration, which describes the concentration of fine particles of diameter less than 2.5 μm. These are of great public interest due to their health impact (6). The remaining four target columns (namely `target_min`, `target_max`, `target_variance` and `target_count`) are summary statistics of the target variable for each entry.

3.2 Weather data

As mentioned, the weather data is obtained from the *Global Forecast System.* This is no measured data but predicted data produced by the FV3 model developed by the U.S. National Weather Service. Since we use data from January 2020 to April 2020, the model ran in version 15.2 (7). The model output consists of nine variables of which six are used in the data set, `precipitable_water_entire_atmosphere` describes the amount of water that is precipitable for the whole atmosphere in kg/m^2,
`relative_humidity_2m_above_ground` gives the relative air humidity in percent at two meters height, `specific_humidity_2m_above_ground` refers to the mass of water in a specific amount air in kg/kg,
`temperature_2m_above_ground` gives the air temperature at two meters height, `u_component_of_wind_10m_abo` describes the u-component of the wind, i.e., the vector showing the wind direction between east and west and `v_component_of_wind_10m_above_ground` giving the v-component, i.e., the wind direction in terms of north or south (8).

3.3 Satellite data

The satellite data is gathered by the Copernicus Sentinel-5 Precursor mission, conducted by a collaboration of the European Space Agency (ESA), the European Commision and the Netherlands Space Office. The data is collected by the Tropospheric Monitoring Instrument (TROPOMI) carried by the Sentinel-5P satellite which was launched in October 2017. Its main goal is to perform atmospheric measurements in order to monitor parameters such as air quality, UV radiation or climate change (9). It measures the tropospheric concentration of substances associated with air quality, such as carbon monoxide (CO), formaldehyde (HCHO), ozone (O_3), sulphur dioxide (SO_2), nitrogen dioxide (NO_2) and those associated with climate forcing, for instance methane (CH_4) and water vapour (H_2O) (10).

As mentioned, the satellite data consists of 68 columns covering the substances NO_2, O_3, CO, HCHO, H_2O (i.e., clouds), SO_2, CH_4 and produces the UV Aerosol Index.

For instance, the data set contains twelve columns for NO_2 (starting with `L3_NO2_`). Four of these variables are actual measurements. `column_number_density` displays the vertical column density[1] of NO_2 divided by the total air mass factor for the whole atmosphere. `tropospheric_NO2_column_number_density` and `stratospheric_NO2_column_number_density` show the same just for the troposphere and stratosphere, respectively.

(11)

3.4 Data preprocessing

The provided training data described above has 30557 entries in total, spanning over 82 features. Only 12 of these features have no missing values while the other features range between missing a couple hundred values to over 24000 missing values.

Simply removing each entry containing a missing value would result in a data set with only 3915 entries left. So to solve this problem of missing values we came up with two possible solutions:

3.4.1 Remove features with most missing values

The first approach is based on the fact that most features contain more than 23000 complete entries and only a couple features have more than 80 % of their data missing. So the idea is to exclude these mainly incomplete features and afterwards remove the entries with missing data. This results in a data set where the columns detailing methane (CH_4) concentration and respective satellite meta data are removed entirely. After removing individual entries with missing data, the data set then has 18219 entries in total spanning over 75 features.

3.4.2 Fill in all missing values

The second approach is trying to use all the data we are provided with as best as possible. So we calculated the mean of each feature and then replaced missing values with the respective mean value. This method is somewhat questionable, since some features consist of mainly missing data, but we were interested how our models would perform on each of these two data sets.

The results were that all models performed marginally better on the data set with the removed methane columns than on the data set with filled in mean-values.

3.4.3 Adding and removing features

Independent of the approach we used for the missing values, we also added some features relating to dates and times. Since the dates were provided in the data set, we then used this information to add a feature that tells us what weekday the date is, whether the given date is on a weekend or not, which month and season it belongs to and what day of the year the provided date is. With these new features we then could remove the features that we could not use based on the object types, which were the Place_ID's and Dates.

After normalizing the data set using the MinMaxScaler from the sklearn library (**?**) we used Principal Component Analysis (PCA) (**?**) to determine which features are considered important

[1] For more information, refer to https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-5p/level-2/doas-method

for the variance of the data set and thus useful to add previous time steps to the data set. In total six features were found to be responsible for more than 75% of the data sets variance. These six features were then added to the data set with a shift of one, such that each data point would have the information for these six features of the previous day as well. It would have been possible to add the data of more than one day, for example a whole week previous to the given data points date, but since the data set has many different locations, these shifts would end up adding data of previous days to locations that are possibly independent of each other. So to not mix up different locations with each other, while still getting the benefit of these useful previous features, we decided to only shift by one day.

4 Related work

In this section we will review previous studies on air quality prediction by researchers, who employed different deep learning techniques. Ong, Sugiura and Zettsu proposed a deep recurrent neural network (DRNN) specifically designed for PM2.5 prediction. Their network is improved by a novel dynamical pre-training method. They use stacked autoencoders to build up their RNN. In this concatenation of autoencoders the output of the model of the layer below serves as input for the next autoencoder. Due to that each hidden layer is a higher-level abstraction of the previous layer, therefore the last hidden layer contains the high-level structure and representative information of the input. This results in the big advantage that the network can select relevant sensors for its predictions. They achieve this through a regularized regression technique called elastic net (EN). The training data is often quite sparse and incomplete. Through the EN they were able to filter out sensors, that did not improve the predictions significantly. This reduces the overall computational costs and results in a more interpretable response–predictor relationship (12).

In 2019 Wen et al. developed a novel neural network arcitecture for air pollution prediction. It is a combination of a convolutional neural network (CNN) and a long short-term memory (LSTM) network. In figure 5 their C-LSTM model is sketched out. They argue that this combination is beneficial due to the combination of temporal and spatial features in the training data(13).

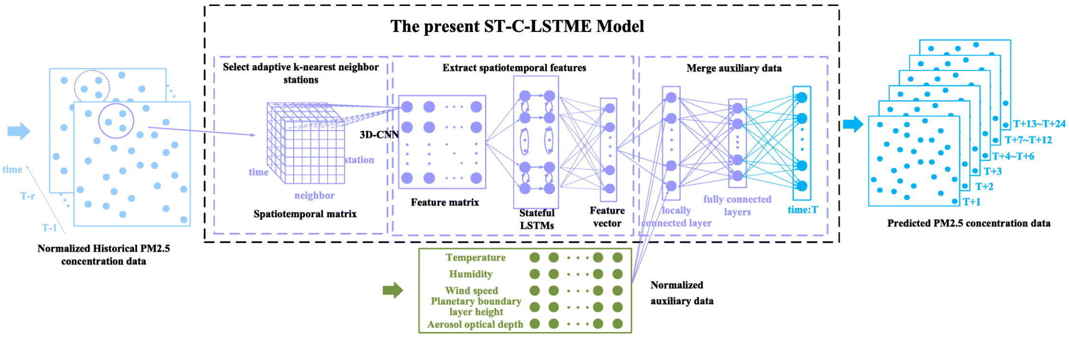


Figure 5: C-LSTM model (13)

Chae et al. used an interpolated convolutional neural network (ICNN) for their predictions of air pollution in South Korea. CNNs work best with an evenly spaced grid like data. To achieve this spatially balanced structure they made use of interpolation. The pollution monitoring stations are concentrated in bigger cities with unequal distances between the stations. They created an equally spaced empty grid and filled in data points as if a virtual measuring station was located at every grid point through interpolation from the existing measurements (14). Figure 6 visualizes their network architecture.

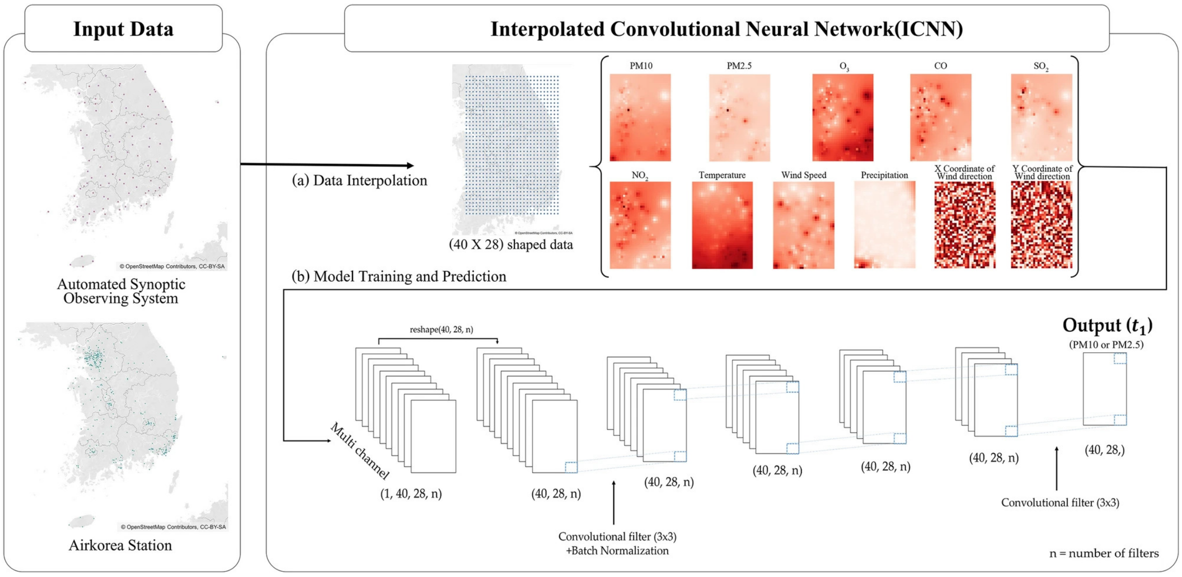

Input Data
© OpenStreetMap Contributors, CC-BY-SA
Automated Synoptic Observing System
© OpenStreetMap Contributors, CC-BY-SA
Airkorea Station
Interpolated Convolutional Neural Network(ICNN)
(a) Data Interpolation
(40 X 28) shaped data
PM10
PM2.5
O_3
CO
SO_2
NO_2
Temperature
Wind Speed
Precipitation
X Coordinate of Wind direction
Y Coordinate of Wind direction
(b) Model Training and Prediction
reshape(40, 28, n)
Multi channel
(1, 40, 28, n)
(40, 28, n)
(40, 28, n)
(40, 28, n)
(40, 28, n)
(40, 28, n)
Convolutional filter (3x3) +Batch Normalization
Output (t_1)
(PM10 or PM2.5)
(40, 28,)
Convolutional filter (3x3)
n = number of filters

Figure 6: ICNN model (14)

5 Model

Due to the chronologically ordered data we believe a LSTM is the best choice for predicting the desired targets. Since these Long Short-Term Memory models are also being actively used for time series forecasting like the weather or stock markets. This special kind of Recurrent Neural Network would be a good choice, if our data did not include over 300 different locations and the target locations we want to predict were part of the training data, but the locations for training and predicting are entirely different from each other. This problem led us to try multiple models that could disregard the chronological order and different locations by only using the weather and satellite data itself. As mentioned in the Data preprocessing section, we have included the data of the six most important features of the previous data points date and thus managed to include this information without the data being necessarily chronologically ordered.

We always use the same hyperparameter, error function and optimizer to be able to compare the different model structures better. We use a training loop with 100 epochs, the Mean Squared Error function and the Adam optimizer with a learning rate of 0.01 for all models. The four different models we used are:

5.1 Linear Regression

The first and most simple idea is to use a linear regression model to predict the target values. To our surprise the linear regression model performed rather well with a Root Mean Squared Error (RMSE) of 37.6 for both data sets.

As the name may suggest this model only has one layer and one unit with a linear activation function, but this simple structure was only marginally worse than the next model we used, which was a Multi-Layer Perceptron (MLP).

5.2 MLP

The MLP is a better fit for our problem, since the complex data is most likely not best described through a single linear function, but rather multiple (non-)linear functions. So the input layer has as many units as our data features, which is either 78 for the first data set without the methane columns or 85 features for the data set with the mean filled missing values. These input layer units have a linear activation function, while the next three hidden layers use a rectified linear activation function. These hidden layers double each time in unit size, so the first on has 160, then 320 and then 160 again. The output layer has again only one unit, since we only want to predict one value, with a linear activation function. We got a RMSE of 30.6 for the first data set and a RMSE of 33.3

for the second one. As mentioned in the data preprocessing section, all models performed slightly better on the first data set. The next model we try is a Convolutional Neural Network (CNN).

5.3 CNN

The CNN is the first unconventional model for this kind of task. To be able to use the CNN we need to add an extra empty dimension to our data, so the shape is then `(number of entries, number of features, 1)` . The CNN itself consists of five 1-dimensional convolutional layers (also called temporal convolution) starting with 32 filters and then doubling for each layer up to 128 filters in the third layer and then halving again back to 32 filters in the fifth layer. The kernel size 2 and the ReLU activation function stay the same across each layer. Then we flatten the input and use simple dense layers for the output. We are able to get a RMSE of 31.9 for the first data set and a RMSE of 33.8 for the second data set.

5.4 LSTM

Lastly we implement the LSTM which we deemed the most promising model for this problem and data. Due to time constraints we were not able to implement the LSTM the same as the other models, but rather we were using the functional API of TensorFlow, but we do not consider this a problem in the comparison to the other models. The model achieved a RMSE of 30.6, equal to the MLP, for the first data set and the best so far for the second data set with a RMSE of 31.4 with the same parameters as the other models.

But as we think that this is the best approach to our problem, we then tried to get it perform the best we could. We implemented a model with a LSTM with 200 units and to counter the overfitting a l2 regularizer of 0.001. Afterwards we used a dropout layer with a value of 0.5, which means that half the neural connections would be dropped. Also in response to the overfitting was an early-stopping function used, and at the end a linear dense output layer with 1 unit. We trained for 200 epochs and were using the Nadam optimizer with a learning rate of 0.01. To be able to use the `PlaceID` we encoded the different locations, but to our surprise the LSTM performed worse with unshuffled data that also has locations, even though the LSTM should in theory profit from chronologically ordered data that can be distinguished by different locations.

Below are the 4 different LSTM configurations we have implemented. On the top left etc ..

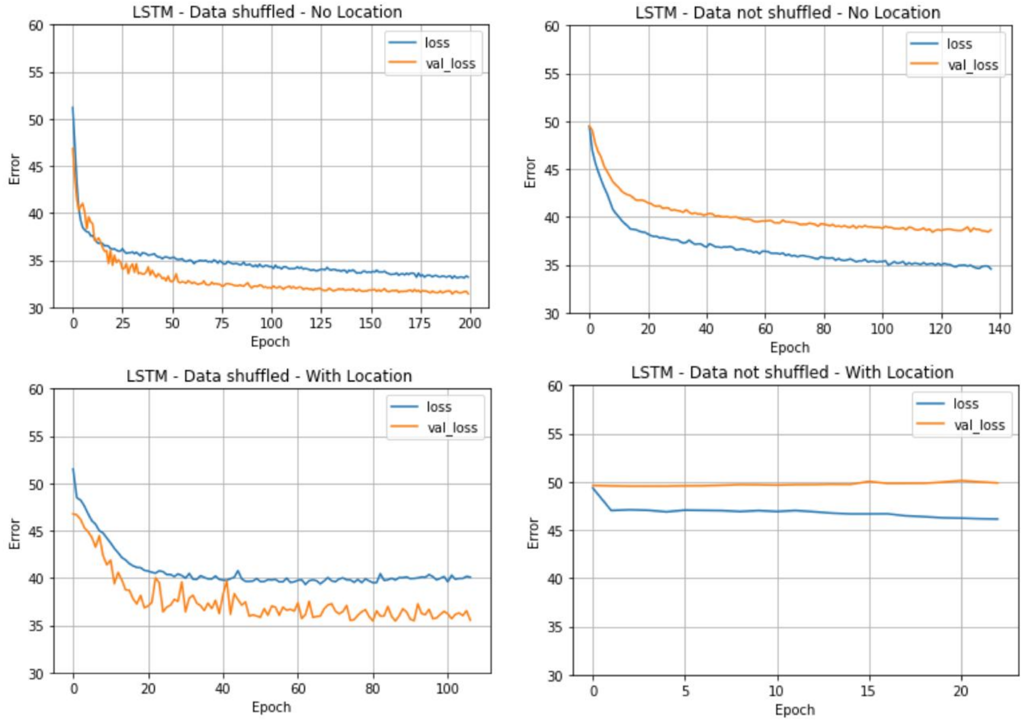


Figure 7: Model comparison

6 Results

Throughout the paper we already mentioned multiple interesting results. For once we determined that the first data set with the removed methane columns was better suited for training and predictions than the second data set that used the mean to fill in missing values. This can most likely be explained by the fact that filling in too many missing values actually changes the data in a way that does not reflect the original data distribution anymore. Another observation was that the MLP and LSTM without locations were the best performing models. The linear regression model assumes a way too simple data distribution and the CNN model structure is not necessarily build for this regression task with (originally) chronological structured data. Another interesting point is that the LSTM actually performed worse when we added the location data and did not shuffle the data to keep the chronological structure of the data.
Regarding the data set we found out that certain features are more important in explaining the distribution than others. For example the feature `L3-SO2-sensor-azimuth-angle` explains 29% of the variance of the data set or the fourth most important feature `weekend` was added artificially by us and explains 8 % of the data sets variance. And more than $\frac{3}{4}$ of the data sets variance can be explained by only 6 of the around 80 different features.

7 Outlook

Due to the limited realm of our project we were not able to employ hyperparameter optimization and test further regularization techniques. In the future another interesting approach would be to use the location (Place-ID) with an improved encoding, so the LSTM and maybe the other models as well, could profit from this, instead of worsening the performance. We consciously decide against this, since we were dealing with an out of distribution problem, in the sense that the location id would have been useless for the prediction, since they were entirely different from each other, but maybe a model might profit from the location, even though it did not work for our simple encoding scheme. - also: avoid overfitting in CNN All in all we are satisfied with our work but improvement possible

8 Conclusion

We did not outperform the winning model of the Zindi challenge, but are confident to be at least in the top ten ;) Satisfied etc. Important topic (predicting pm2.5) proud to be part of that etc. Give attention to this topic, broaden our minds and hopefully yours as wells. like to see limitation/outlook section be improved in the future.

References

[1] "9 out of 10 people worldwide breathe polluted air, but more countries are taking action." https://www.who.int/news/item/02-05-2018-9-out-of-10-people-worldwide-breathe-polluted-air-but-more-countries-are-taking-action. accessed 15.03.2022.

[2] "Empowering the world to breathe cleaner air | IQAir." https://www.iqair.com/world-air-quality-report. accessed 15.03.2022.

[3] J. Y. CNN, "Pandemic lockdowns improved air quality in 84% of countries worldwide, report finds." https://www.cnn.com/2021/03/16/health/world-air-quality-report-intl-hnk-scn/index.html. accessed 15.03.2022.

[4] "#ZindiWeekendz Learning: Urban Air Pollution Challenge." https://zindi.africa/competitions/zindiweekendz-learning-urban-air-pollution-challenge/data. accessed 26.03.2022.

[5] US EPA, "Particulate matter (PM) basics." https://www.epa.gov/pm-pollution/particulate-matter-pm-basics. accessed 15.03.2022.

[6] A. P. K. Tai, L. J. Mickley, D. J. Jacob, E. M. Leibensperger, L. Zhang, J. A. Fisher, and H. O. T. Pye, "Meteorological modes of variability for fine particulate matter ($PM_{2.5}$) air quality in the United States: implications for $PM_{2.5}$ sensitivity to climate change," *Atmospheric Chemistry and Physics*, vol. 12, pp. 3131–3145, Mar. 2012.

[7] "GFS Documentation at National Centers for Environmental Prediction." https://www.emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/gfs/documentation.php. accessed 17.03.2022.

[8] "GFS: Global Forecast System 384-Hour Predicted Atmosphere Data | Earth Engine Data Catalog." https://developers.google.com/earth-engine/datasets/catalog/NOAA_GFS0P25. accessed 17.03.2022.

[9] "Sentinel-5P - Missions - Sentinel Online." https://sentinel.esa.int/web/sentinel/missions/sentinel-5p. accessed 20.03.2022.

[10] "Thematic areas and services - Sentinel-5P Mission - Sentinel Online." https://sentinel.esa.int/web/sentinel/missions/sentinel-5p/thematic-areas-services. accessed 24.03.2022.

[11] "Sentinel-5P OFFL NO2: Offline Nitrogen Dioxide | Earth Engine Data Catalog | Google Developers." https://developers.google.com/earth-engine/datasets/catalog/COPERNICUS_S5P_OFFL_L3_NO2. accessed 27.03.2022.

[12] B. T. Ong, K. Sugiura, and K. Zettsu, "Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5," *Neural Computing and Applications*, vol. 27, pp. 1553–1566, Aug. 2016.

[13] C. Wen, S. Liu, X. Yao, L. Peng, X. Li, Y. Hu, and T. Chi, "A novel spatiotemporal convolutional long short-term neural network for air pollution prediction," *Science of The Total Environment*, vol. 654, pp. 1091–1099, Mar. 2019.

[14] S. Chae, J. Shin, S. Kwon, S. Lee, S. Kang, and D. Lee, "PM10 and PM2.5 real-time prediction models using an interpolated convolutional neural network," *Scientific Reports*, vol. 11, p. 11952, June 2021. Number: 1 Publisher: Nature Publishing Group.