Faculty of Fundamental Problems of Technology
Wrocław University of Science and Technology

Machine learning approach to air quality prediction

Piotr Kołodziejczyk
Index no: 244750

Master's thesis
written under the supervision of
Marcin Michalski, Ph.D.

Wrocław
University
of Science
and Technology

Wrocław 2022

Contents

Introduction

Air pollution is said to be one of the greatest environmental health risks. In 2016, more than 4 million deaths occurred as a result of exposure to outdoor air pollution [17]. Poor air quality significantly increases the risk of stroke, heart and lung diseases, and cancer. Only in Poland, up to 46 thousand premature deaths in 2017 were estimated to be the result of smog.

Despite efforts by the WHO to improve air quality, 9 out of 10 people worldwide live in places where the guiding limits are exceeded. That is why the awareness of the problem should be raised and people should make a conscious effort to avoid the danger, namely reduce exposure to air pollution.

Poor air quality is a global and constant problem, but the level of air pollutants varies, both annually and daily. It is affected by both season and weather. It is commonly believed that in winter the quality is worse and during windy days it is better. In this paper, these rules of thumb will be used to check whether forecasting the level of certain air pollutants with weather data is possible.

The goals of this thesis are:

- analysis of the air pollution problem,
- collecting and preparing historical weather and smog data,
- creating models that predict the air pollution levels,
- analysis of the results and comparison with the existing models.

The thesis is structured as follows. Chapter 2 introduces basic facts about the air pollution, its dependence on the weather and explains general formula of the regression. Chapter 3 describes the process of collecting and processing weather and air data. In Chapter 4 machine learning models are proposed. All tests, achieved results, and comparison with existing approaches are presented in Chapter 5. Chapter 6 contains the summary of the thesis.

Problem analysis

2.1 Air pollutants

As mentioned in the Introduction, air pollution remains one of the most significant challenges facing the civilization today. Figure 2.1 shows the average concentration of PM_{10} and $PM_{2.5}$ particles around the world over the year. WHO guidelines state that the average concentration of $PM_{2.5}$ should not exceed 5 $\mu g/m^3$, while today those values are much higher in the vast majority of places.

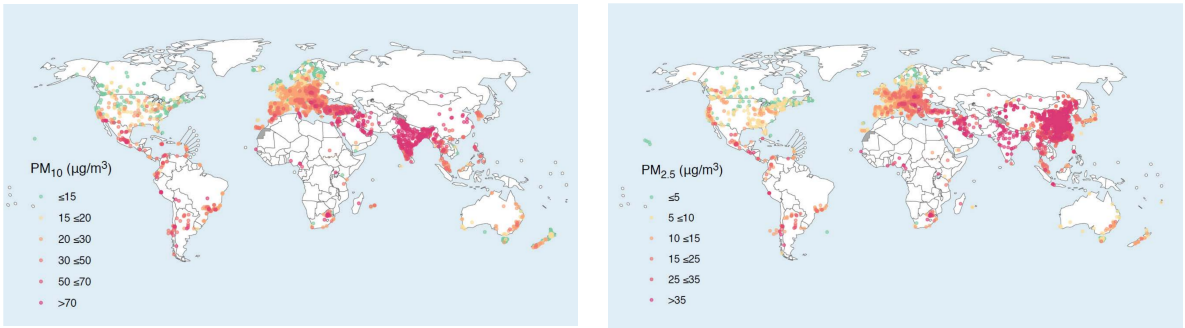
Among all the pollutants, five of them were considered in this thesis due to their harmfulness and accessibility to the measurement data. They are listed below with their potential health impact [11, 16]:

- PM_{10} and $PM_{2.5}$- particles with a diameter of 10 and 2.5 micrometers or less, respectively - they are small enough to enter the lungs through the respiratory system (or even the bloodstream for $PM_{2.5}$); the long-term exposure can lead to a reduced lung function, cardiovascular diseases and a reduction in life expectancy,
- NO_2, nitrogen dioxide - increases the susceptibility to lung infections in the case of asthma and worsens its symptoms,
- SO_2, sulfur dioxide - narrows the airways, leading to the shortness of breath and the chest tightness,
- O_3, ground-level ozone - reduces lung functions and causes chronic respiratory diseases.

Table 2.1 shows the standards for the concentration of air pollutants in Poland established by the state institutions [7]. These categories are determined by the risk of exposure to a given level of the air pollution. When the index is moderate or worse, children, elderly people and pregnant women should be cautious about being outdoors. Below the satisfactory level, all outdoor activities should be limited.

2.1.1 Impact of weather on air pollution

Not only daily observations, but also many studies confirm the influence of weather on air pollution levels [2, 10, 12]. This applies to changes in current weather, as well as changes in seasons throughout the year. It was shown [2], that there is a strong correlation between wind and most pollutant factors considered or between sun radiation and the level of nitrogen dioxide and ozone. On the other hand, a higher concentration of pollutant is visible during the winter [10] or dry seasons [12]. Such studies support the validity of using weather conditions in smog prediction, which is done in this work.



(a) PM_{10} (b) $PM_{2.5}$

Figure 2.1: Annual mean concentration of (a) PM_{10} and (b) $PM_{2.5}$ worldwide [18].

Air quality index	PM_{10}	$PM_{2.5}$	O_3	NO_2	SO_2
Very good	0-20	0-13	0-70	0-40	0-50
Good	20.1-50	13.1-35	70.1-120	40.1-100	50.1-100
Moderate	50.1-80	35.1-55	120.1-150	100.1-150	100.1-200
Satisfactory	80.1-110	55.1-75	150.1-180	150.1-230	200.1-350
Bad	110.1-150	75.1-110	180.1-240	230.1-400	350.1-500
Very bad	> 150	>110	>240	>400	>500

Table 2.1: Polish norms of air quality index for certain factors, all values in $\mu g/m^3$.

2.2 Regression

Regression, along with classification, is a fundamental application of supervised machine learning. Its aim is to predict continuous values by finding the relationship between random variables [5]. Many methods have been developed, from the simplest, such as linear regression, to more complex ones, including random forest regression or support vector regression.

To understand the general idea behind regression, let us examine how the simplest one-variable linear model works. There is an independent variable x - input data and a dependent variable y - outcome. The aim is to find a function

$$f(x) = a \cdot x + b,$$

that best fits the dependent variable. The precision of the fit can be measured by various metrics, e.g., mean squared error, defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2,$$

which should be minimized. Figure 2.2 shows the set of points with the best fit regression line.

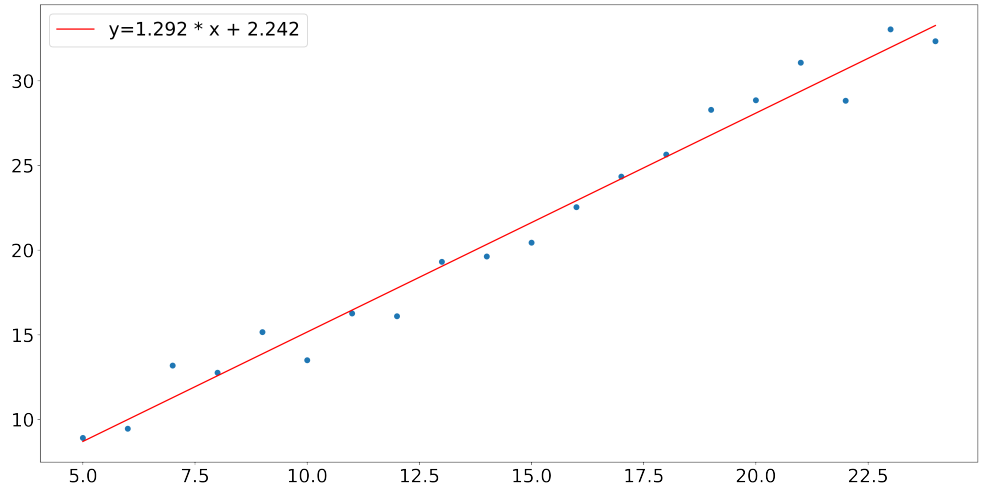


Figure 2.2: Line of best fit model for exemplary linear regression

However, in real applications, more complex models are required to provide satisfactory results. The regression models used in this paper are described in Chapter 4.

Data collecting and processing

3.1 Data source and format

There are two main sources of the data; both are the Polish state institutions. Weather information comes from Instytut Meteorologii i Gospodarki Wodnej [1] - the major Polish weather research institution. It provides historical measurements of weather conditions since 2008. These data are stored in CSV files, one per month of the year and per weather parameter, in the form shown in Table 3.1. Each row consists of 4 columns. The station code is connected to the World Meteorological Organization system of measurement point indices, and the parameter code indicates the measured parameter, e.g., B00202A means the direction of the wind. From all weather parameters, the following will be considered in further calculations: temperature, humidity, wind speed and the sum of precipitation during the last hour. The number of measurements with at least one nonempty parameter was almost 20 millions.

Station code	Parameter code	Date	Measurement
250170110	B00202A	02.03.2009 12:00	194
250170110	B00202A	02.03.2009 12:10	211
250170110	B00202A	02.03.2009 12:20	209
250170110	B00202A	02.03.2009 12:30	202

Table 3.1: Exemplary rows of weather data provided by IMGW.

Information on air pollution comes from Główny Inspektorat Ochrony Środowiska [6]. This institution gathered historical measurements of air pollutants stored since 2000. There are many factors measured, collected in one file per factor per year, distinguishing the type of measurement (daily and hourly). Table 3.2 shows the way records are stored. The size of all the collected files was over 44 GB. The procedure of data preprocessing included analysis of the way the files are stored, their completeness and usability. The Python `pandas` library made it possible to perform this analysis efficiently.

Station code	DsBogChop	DsCzLasMob	DsDzia01	DsDzierPilsA
Factor	NO2	NO2	NO2	NO2
Average time	1g	1g	1g	1g
01.01.2013 01:00	6,41	9,6	7,4	18,3
01.01.2013 02:00	6,58	9,6	7,2	11,4
01.01.2013 03:00	6,59	9,4	7,4	20,8
01.01.2013 04:00	6,15	9,7	6,9	18
01.01.2013 05:00	6,29	7,5	6,6	12
01.01.2013 06:00	5,94	6,4	8,9	13,5

Table 3.2: Exemplary rows of pollutants data provided by GIOŚ.

Of all pollutant factors, the following will be considered: NO_2, SO_2, O_3, PM_{10}, and $PM_{2.5}$, as their harmfulness to the environment or health is significant [17].

3.2 Data processing

3.2.1 Merging weather and smog data

Unfortunately, the weather and air pollution data do not come from the same source and also do not come from the same measurement points. This will not be irrelevant to the accuracy of the results, since the greater the distance, the greater the difference in atmospheric conditions.

To obtain pairs of corresponding weather-smog stations, the distances between all of them were calculated. Information about the stations provided by GIOŚ already contains their geographic coordinates, whereas that provided by IMGW does not. However, Poland is a member of the World Meteorological Organization, and thus some of the stations are included in its system, making it possible to obtain their exact location. Table 3.3 shows the number of pairs within a given distance. To keep the modeling process accurate, stations with a distance greater than 3 km are discarded.

Distance	Number of station pairs
$\leqslant$ 1km	9
$\leqslant$ 2km	26
$\leqslant$ 3km	35
$\leqslant$ 5km	45

Table 3.3: Number of pairs of nearest smog and weather stations within given distance.

3.2.2 Creating complete dataframe

To analyze the entire dataset, it was convenient to store it in a user-friendly format, for which the Python `pandas` library was used. The merging of the weather and smog data consisted of a few steps. First, the weather data were filtered to contain only measures from stations that have a counterpart in the air pollution data. Then, one dataframe per each year was created in a way that each row contains all the weather factors (temperature, humidity, wind speed and precipitation) for each hour and station. Air quality data was filtered in the same way and information about each considered pollutant was merged into one dataframe for all the considered stations. Tables 3.4 and 3.5 show the processed data form.

Date	Station	Temperature	Humidity	Precipitation 1h	Wind speed
2013-01-01 13:00:00	349200660	7.2	3.1	40.0	0.0
2013-01-01 14:00:00	349200660	5.9	2.4	47.0	0.0
2013-01-01 15:00:00	349200660	3.7	1.4	52.0	0.0
2013-01-01 16:00:00	349200660	2.1	0.7	59.0	0.0
2013-01-01 17:00:00	349200660	0.6	1.2	69.0	0.0

Table 3.4: CSV file data format with processed weather data.

Date	LuZielKrotka	MpZakopaSien	SlKatoPlebA4	WpPilaKusoci
2020-12-30 12:00:00	17.8852	14.8207	98.0266	29.8511
2020-12-30 13:00:00	18.1937	12.8678	95.0317	26.6779
2020-12-30 14:00:00	19.8262	13.2863	93.5674	27.0993
2020-12-30 15:00:00	27.2694	11.3413	98.9168	31.7514

Table 3.5: CSV file data format with processed air pollution data (here, NO_2 factor)

The final step (shown in listing 3.1 was to combine all the smog and weather data into one dataframe, so that each row contains both weather conditions and air pollution measures per station per hour. Rows that are incomplete, e.g. some weather factors are missing or for certain station there are no air quality data, are discarded. Finally, the number of records with complete weather data and at least one pollution factor exceeds 1.6 million. Table 3.6 shows how the final data set looks.

```
factors = ['NO2', 'O3', 'PM10', 'PM25', 'SO2']
for factor in factors:
```

```
    pollutant_df = pd.read_csv(f'smog/_{factor}_1g.csv', parse_dates=
   True, index_col=0)
    pollutant_df.index = pollutant_df.index.round('H')
    pollutant_df = pd.DataFrame(columns=['date', 'station', factor])
    pollutant_df.set_index(['date', 'station'], inplace=True)
    for column in pollutant_df.columns:
      temp_df = pd.DataFrame(pollutant_df[column])
      temp_df = temp_df.assign(station=smog_weather_stations_mapping[
   column])
      temp_df.set_index(['station'], append=True, inplace=True)
      temp_df.columns = [factor]
      pollutant_df = pd.concat([pollutant_df, temp_df])
    pollutant_df.dropna(inplace=True)
    weather_df = pd.merge(weather_df, pollutant_df, how='left', on=['
   date', 'station'])
weather_df.to_csv('weather_smog.csv')
```

Listing 3.1: Python code for merging weather and smog dataframes

Date	Station	Temp.	···	Wind speed	NO_2	O_3	PM_{10}	$PM_{2.5}$
2018-07-06 13:00	349190625	18.3	···	2.1	20,6225	94,9072		14,9219
2018-07-06 14:00	349190625	17.9	···	1.3	19,4068	104,02		18,4585
2018-07-06 15:00	349190625	19.0	···	1.1	28,04	88,9606		27,2642
2018-07-06 16:00	349190625	18.7	···	1.3	27,886	93,9117		20,3386

Table 3.6: Dataframe containing information about weather and air pollution

3.2.3 Appending historical data

To forecast the level of air pollution factors, which is the main goal of the work, historical data are required. To train certain models, it is convenient to store in each row not only actual measures but previous ones as well, e.g., data about pollutant and weather from the previous 2 and 4 hours. This functionality is obtained by a method shown in the listing 3.2.

```
def createDataFrameWithHistoricalDataForFactors(df: pd.DataFrame,
   df_factors: list, factors_to_shift: list, history: list):
    frame = df[[*df_factors]].copy(deep=True)
    for factor in factors_to_shift:
        for hour in history:
            frame[f'{factor}-{hour}'] = df.groupby(level=0)[factor].
   shift(hour)
    frame.dropna(inplace=True)
    return frame
```

Listing 3.2: Python code for adding data from previous hours to a dataframe.

3.2.4 Extracting additional parameters

Not only the weather and pollution information can be used in prediction. As time dependence is observable, various time data may be added to the parameters of the models. In case of this work, hour, weekday and month are considered. All are transformed into numbers and scaled to the 0-1 range using `MinMaxScaler` from the `scikit-learn` [13] package.

Implementation

4.1 Models

Models used for prediction are described in this section. Some of them belong to a group of ensemble learning methods. It means that they combine multiple algorithms to achieve better results.

4.1.1 Random Forest

Random forests are a method for both classification and regression problems. They are sets of decision trees that grow on random subsets of data. The final result is obtained either as a majority vote for a certain category in classification problem, or as a mean of all results (in regression). Pseudocode 4.1 shows a general regression schema using random forests.

Pseudocode 4.1: General algorithm for the random forest regression [8]. Steps 4-7 repeated recursively are responsible for growing the tree T_k.

```
1 for k=1 to K do
2     Draw a bootstrap sample L of size N from the training set
3     foreach node in tree do
4         while the minimum size of node m is not reached do
5             Select F variables at random from the n variables
6             Pick the best variable/split-point among the F
7             Split the node into two child nodes
8 Output the ensemble of trees {T_k}
9 To make a prediction at a new point x:
```

$$f(x) = \frac{1}{K}\sum_{k=1}^{K} T_k(x)$$

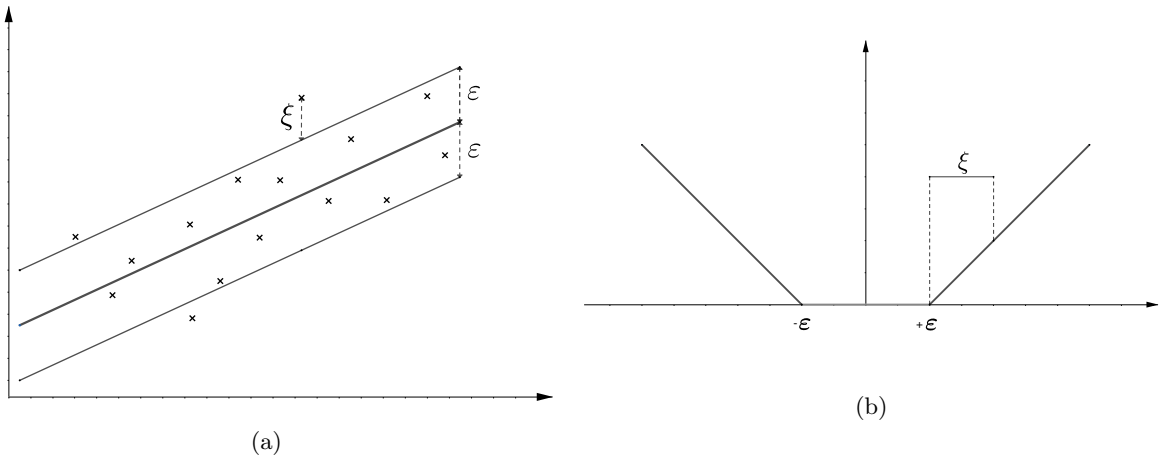
While individual decision trees in RFR tend to overfit learning irregular patterns, the forest aims to reduce the prediction variance by averaging subresults.

4.1.2 Support Vector Regressor

Support Vector algorithms were created as a generalization of the Generalized Portrait by Vapnik in 1963 in Russia [14]. Since then, it has been developed by Vapnik and ATT Bell Laboratories and extended to work for binary and multiclass classification and regression.

In the regression problem, the goal is to find a function $f(x)$ that has at most ε deviation between $f(x_i)$ and y_i for the training data $\{(x_1, y_1), ...,(x_n, y_n)\} \subset X \times \mathbb{R}$. Figure 4.1 shows a simple two-dimensional case that can be extended to any dimension (the line will be replaced by the hyperplane). After rewriting the function as $f(x) = \langle w, x\rangle + b$, where $w \in X, b \in \mathbb{R}$, the problem can then be described as an optimization problem:

$$\text{minimize } \frac{1}{2} \parallel w \parallel^2$$



(a) (b)

Figure 4.1: (a) sample function f with upper und lower ε error boundary, (b) linear loss function with ε toleration - actual error value is decreased by the ε parameter.

$$\text{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \leqslant \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leqslant \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geqslant 0 \end{cases}$$

4.1.3 AdaBoost Reggresor

In AdaBoost models, weaker algorithms are trained and then fitted on repeatedly modified dataset. At each iteration, the weights are changed - the weights of inputs that were not predicted correctly are increased, and those that were predicted correctly are decreased. At the beginning, the samples for each learner are distributed uniformly. Then, the error and the value of the loss function are calculated, and the new distribution is computed - all the weights are updated based on the exponential loss function and normalized to make their set a distribution [15]. The final result is the weighted median.

4.2 Models validation

4.2.1 Metrics

In order to compare the efficiency of implemented models, certain metrics will be used.

$\mathbf{R^2}$ score

It is a measure representing the proportion of the variance in the dependent variable that can be explained by the independent variable. It can be calculated using the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - f_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

where $y = [y_1, ..., y_n]$ is the vector of real values and $f = [f_1, ..., f_n]$ - of predicted ones. In general, the closer the score is to 1, the better the prediction. It is worth mentioning that this is not a metric in the mathematical sense, since in some cases the its value can be lower than 0.

Mean absolute error

The mean absolute error is nothing more than the average difference between the actual and predicted values.

$$\text{MAE} = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| y_i - f_i \right|$$

Mean squared error

MSE is a metric similar to MAE, but it averages squared differences of the actual and predicted values.

$$\text{MSE} = \frac{1}{n} \cdot \sum_{i=1}^{n} \Big(y_i - f_i \Big)^2$$

Air quality index accuracy

Let the accuracy function for the prediction of the pollution level be defined as follows: if the predicted and real value are in the same category from Table 2.1, then the accuracy is equal to 1, 0 otherwise. With this metric, the real usability of the prediction can be checked, as the exact forecast is not as important as the range to which it belongs, indicating the harmfulness.

Results

Tests of each model involved predicting pollutant level 2, 6 and 12 hours ahead. It was also checked whether extending the time window would improve the results. The models were trained with input that included only current weather and pollution data and with hourly data for the last 6 and 12 hours. For each pollutant, a five-fold cross-validation was performed on 80% of the collected data, and the rest was used as a validation set. Due to the extensive size of the data, the models were trained on the Google Colab platform, which provides large memory resources. Where possible, the training was performed using the GPU memory to speed up the process.

5.1 Random Forest Regressor

To obtain the best results, the hyperparameters were tuned using `GridSearchCV` from `scikit` package. The variables tested were the number of estimators, maximum depth of tree and the maximum number of features to consider looking for the best split.

The plots 5.1 and 5.2 show the mean absolute error for the NO_2 pollutant, for the training and the validation set, respectively. It should be compared with the AQI thresholds (for NO_2 the upper limit for 'very good' level is 40 $\mu g/m^3$). The plot 5.3 presents the precision in classifying air quality indices for the validation set. This metric is the most useful from the perspective of people. For all pollutants, the AQI accuracy for the train set was above 90% and with a tolerance of one level (e.g., when the actual level is 'good', but the prediction is either 'very good' or 'moderate'), the accuracy for all factors was above 99%.

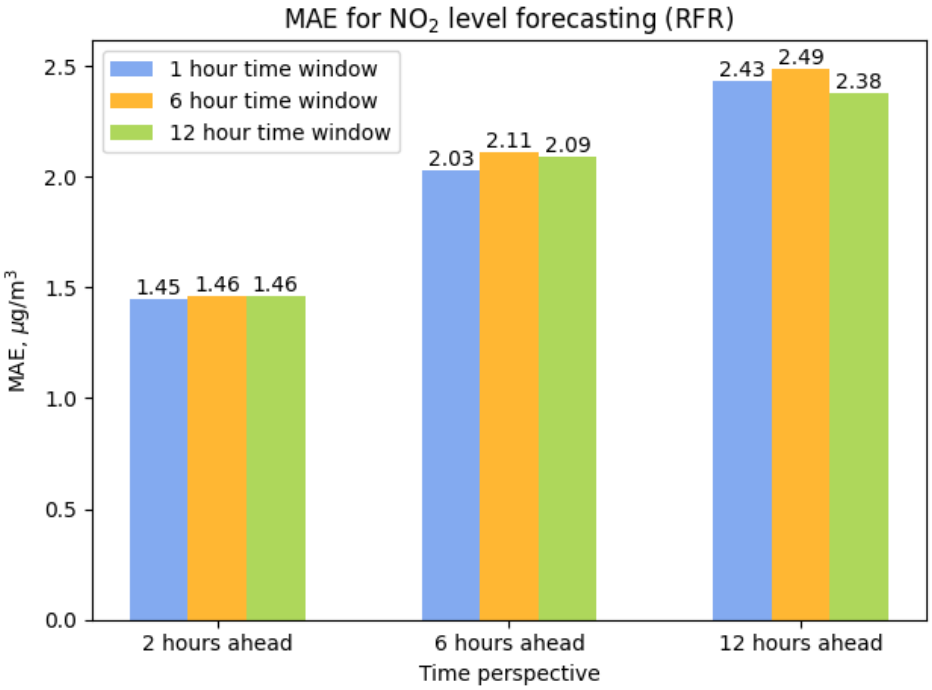


Figure 5.1: Mean absolute error in NO_2 forecasting (on train set) using RFR algorithm.

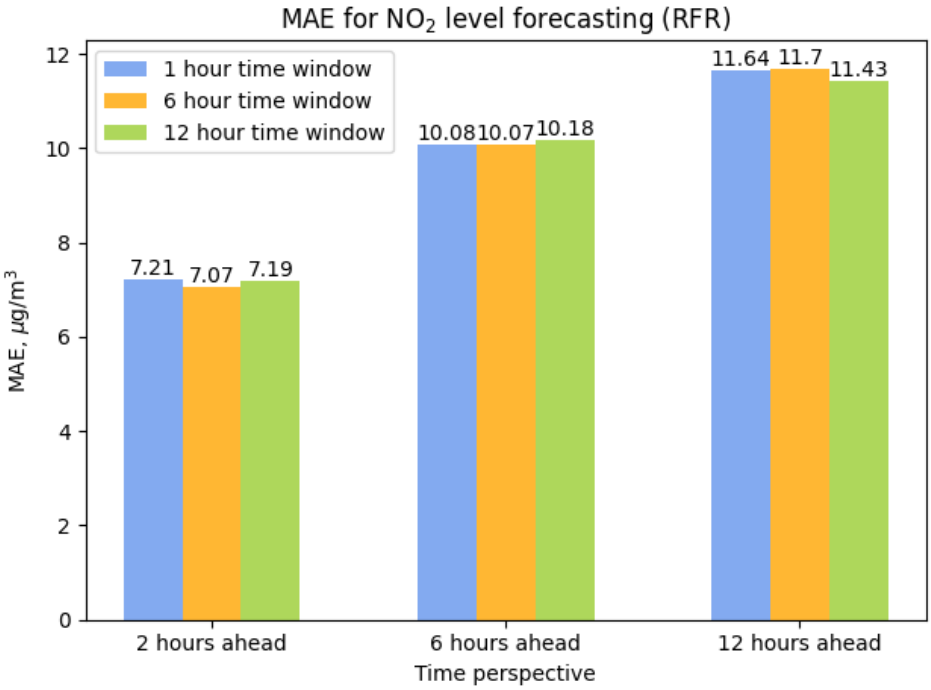


Figure 5.2: Mean absolute error in NO_2 forecasting (on validation set) using RFR algorithm.

5.2 AdaBoost Regressor

AdaBoost model was used with decision trees as sublearners. The parameters tuned by grid search were: number of estimators (number of trees), loss function (choice between linear, square and exponential functions) and the learning rate.

Plot 5.4 presents the MAE values for the NO_2 forecast. Comparing the error to the range of thresholds values for the different AQI classes (the smallest is 40 $\mu g/m^3$), provides an overview of the effectiveness of the model. AQI accuracy for the validation set in most cases is above 70% (lower values for $PM_{2.5}$ and PM_{10}. Index accuracy with tolerance of one level in each case is greater than 95%.

5.3 Support Vector Regressor

Support vector algorithms are a group of algorithms where the measure function is based on the distance between points. In order to avoid exaggerated influence of high levels of the air pollutants, the input values are scaled using `StandardScaler` from `sklearn.preprocessing` package. Because of this, error values cannot be directly compared with the results of previous models. AQI accuracy for this model is shown in the plot 5.5. It is noticeable that increasing the time window worsens the results. It may be caused by a large number of features (high dimension); however, the reduction of dimensionality by the PCA algorithm did not significantly improve the results. A similar problem was observed in [4].

5.4 Models comparison and discussion

Of the three models considered, the random forest was found to be the best. For most cases, it had the lowest mean absolute and mean squared error and the highest AQI accuracy. However, in most cases, all models have an index accuracy of more than 70-75%, and when considering the tolerance of 1 level of AQI, then all of them are accurate in over 99% - Table 5.1.

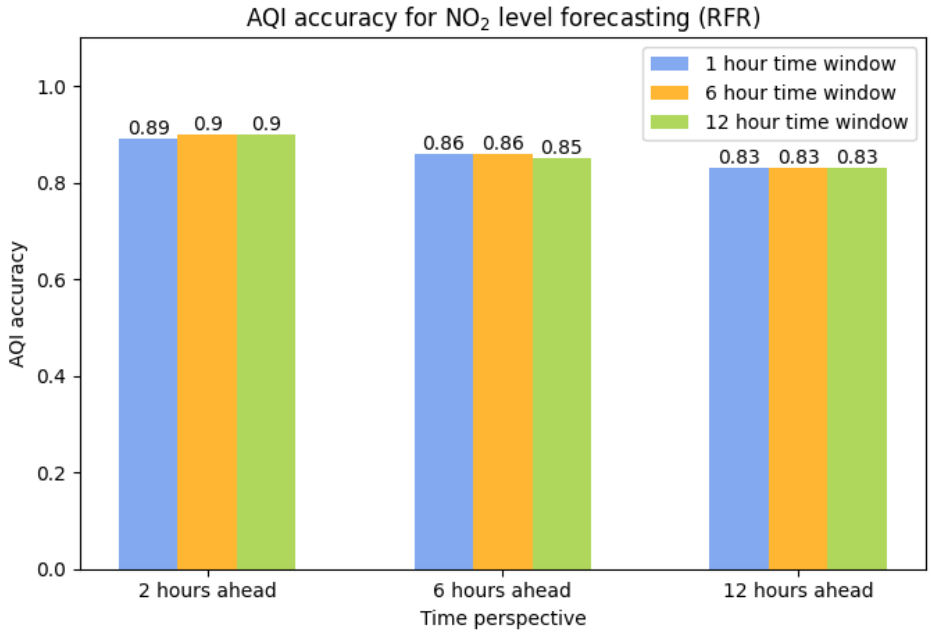


Figure 5.3: AQI accuracy in NO_2 forecasting (on validation set) using RFR algorithm.

5.4.1 Comparison with related works

Since air pollution is a popular problem in the world today, many works discuss a similar issue with different approaches, including ML-based. Kumar and Pande in [9] use simple models including Gaussian Naive Bayes and SVM in classifying problem. Bekkar et al. in [3] compare various deep learning algorithms (such as LSTM and CNN layers) in one hour ahead $PM_{2.5}$ prediction. Similar work was done in [4] to predict AQI and $PM_{2.5}$ in California.

Despite different approaches to model validation and different time perspectives, some results can be compared with those obtained in this work. In [3] 1 and 7-day lags were used in the $PM_{2.5}$ prediction. The smallest mean squared error (obtained in CNN-LSTM model) was 6.742, while the RFR and SVR models here have MAE equal to 3.63 and 4.50 in the 2 hour perspective. At the same time, their model specifies a better R^2 score, reaching 0.989 (0.82 for the RFR model proposed here). Castelli et al. in [4] used a similar approach in predicting AQI (with respect to pollution standards in California). Their accuracy was up to 90.02% on the training set and 94.1% on the validation set, while RFR obtained 96% on the training set and 82% on the validation set in similar case.

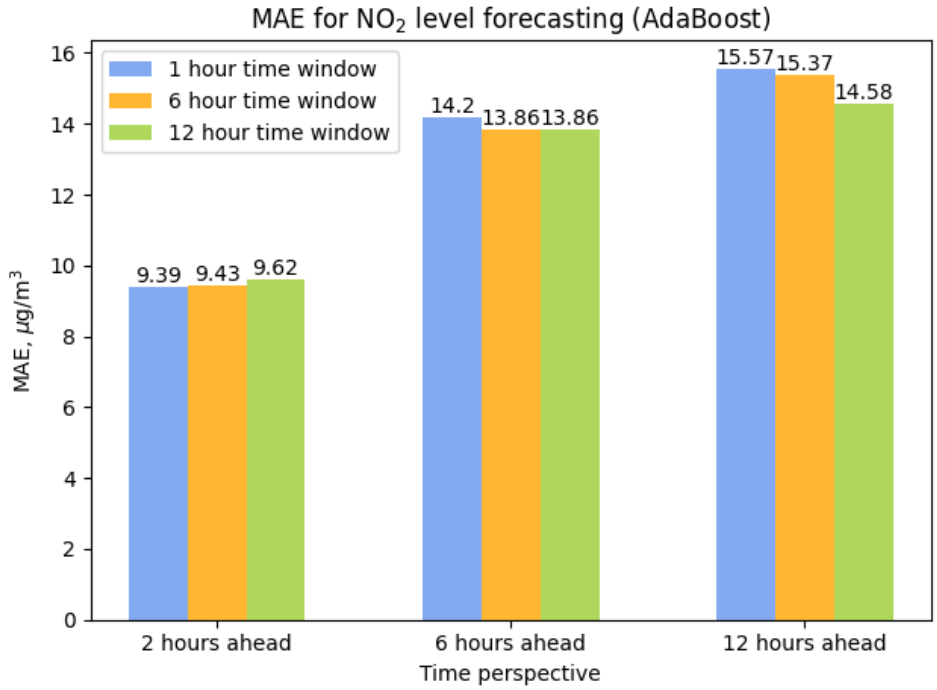


Figure 5.4: Mean absolute error in NO_2 forecasting (on validation set) using AdaBoost algorithm.

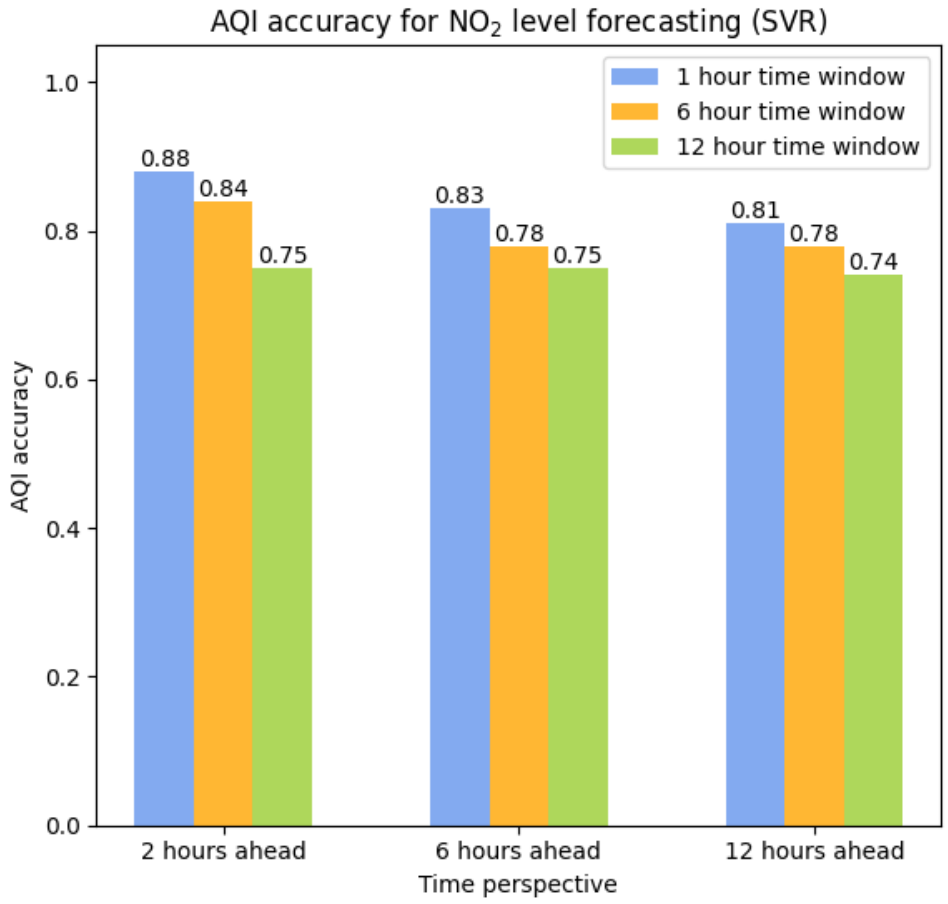
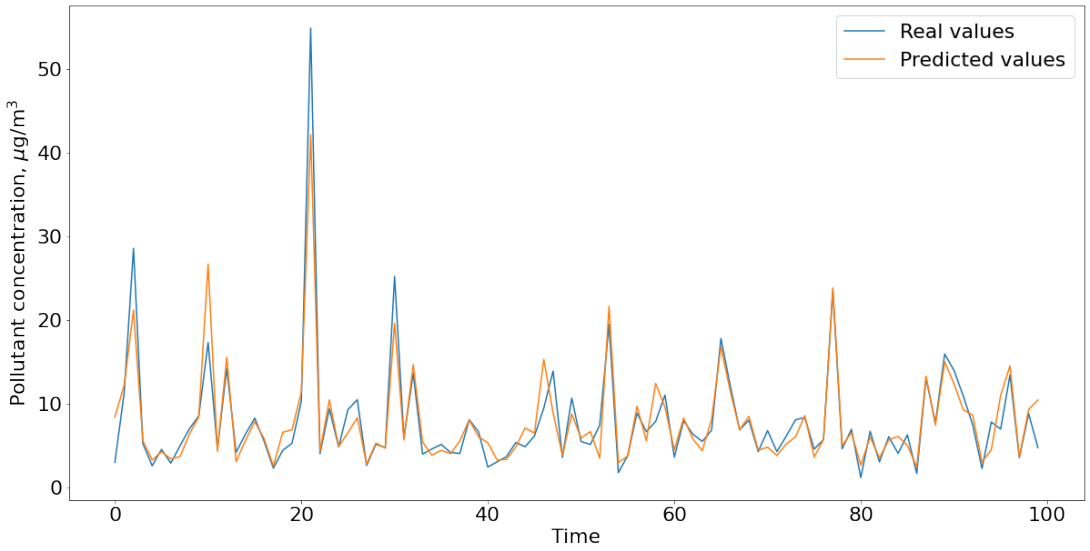
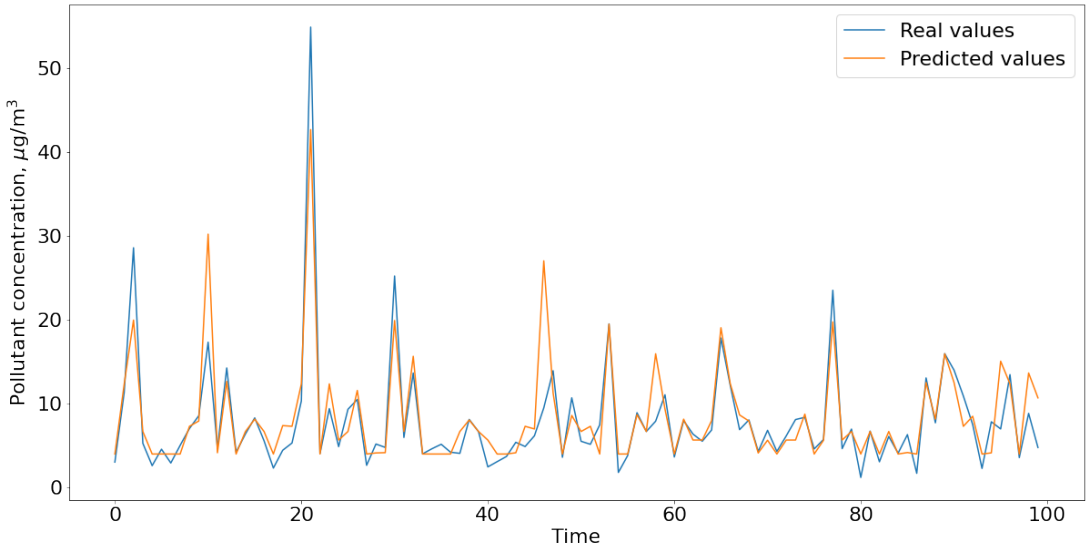


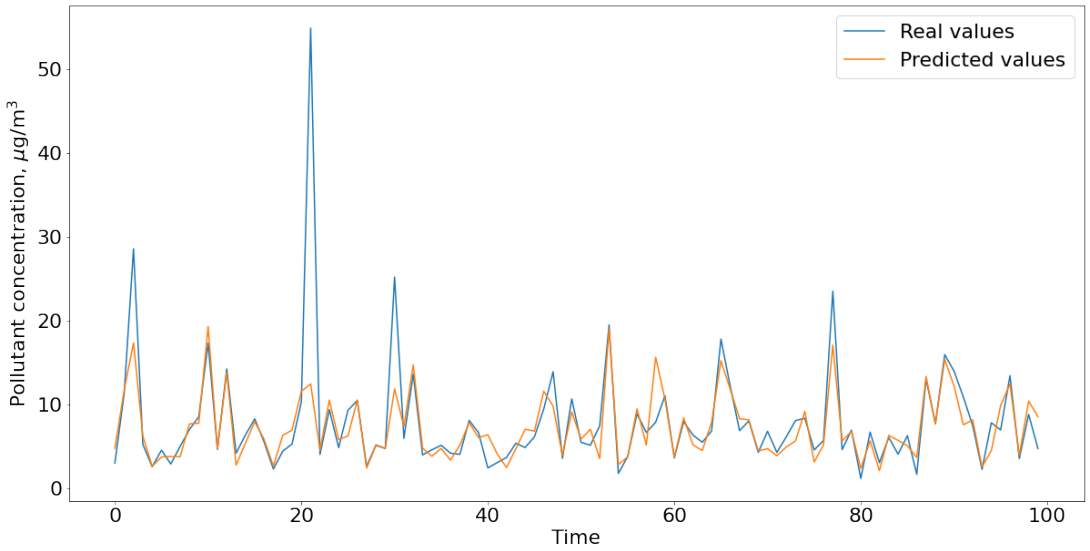
Figure 5.5: AQI accuracy in NO_2 forecasting (on validation set) using SVR algorithm.



(a) Random Forest Regression



(b) AdaBoost



(c) Support Vector Regression

Figure 5.6: Comparison of the prediction efficiency of each model on the same interval of 100 consecutive hours, SO_2 forecasting two hours ahead.

	Time perspective	Metric	NO_2	SO_2	O_3	PM_{10}	$PM_{2.5}$
RFR	2 hours ahead	AQI	0.90	1.0	0.92	0.76	0.82
		AQI ±1	1.0	1.0	1.0	0.99	1.0
	6 hours ahead	AQI	0.86	1.0	0.87	0.68	0.76
		AQI ±1	1.0	1.0	1.0	0.98	1.0
	12 hours ahead	AQI	0.83	1.0	0.85	0.64	0.74
		AQI ±1	1.0	1.0	1.0	0.97	1.0
AdaBoost	2 hours ahead	AQI	0.88	1.0	0.89	0.74	0.79
		AQI ±1	1.0	1.0	1.0	0.98	0.99
	6 hours ahead	AQI	0.82	1.0	0.79	0.62	0.70
		AQI ±1	1.0	1.0	1.0	0.96	0.99
	12 hours ahead	AQI	0.79	1.0	0.80	0.58	0.67
		AQI ±1	1.0	1.0	1.0	0.96	0.98
SVR	2 hours ahead	AQI	0.88	1.0	0.90	0.80	0.75
		AQI ±1	0.99	1.0	1.0	0.99	0.98
	6 hours ahead	AQI	0.83	1.0	0.80	0.65	0.72
		AQI ±1	1.0	1.0	1.0	0.98	0.99
	12 hours ahead	AQI	0.80	1.0	0.77	0.62	0.69
		AQI ±1	1.0	1.0	1.0	0.99	0.98

Table 5.1: Comparison of the air quality index (AQI) accuracy of trained models. For each time perspective, the best model of the three trained was chosen.

Summary

This thesis had two main goals. The first one was to find, collect and process weather and air pollution data. Historical information from Polish state institutions - Instytut Meteorologii i Gospodarki Wodnej and Główny Inspektorat Ochrony Środowiska, that consisted of meteorological measurements gathered for 13 years, was used. The second goal was to propose, train and test models forecasting air pollution level. Five air pollutants were considered: NO_2, SO_2, O_3, PM_{10}, and $PM_{2.5}$ due to their harmfulness and measurement data. The weather factors used, were: temperature, the sum of precipitation over last hours, humidity, and wind speed.

Three machine learning algorithms were proposed, tuned and tested: random forest regression, support vector regression and AdaBoost regression. The tests consisted of trying to predict the level of each pollutant 2, 6 and 12 hours ahead with historical knowledge from the last hour, last 6 or last 12 hours. The best results for each air parameter were obtained using the random forest regressor. R^2 score was between 0.89 and 0.98 for the train set and between 0.5 and 0.91 for the validation set. AdaBoost had slightly worse results, with R^2 score from 0.45 to 0.81 for the validation set. Mean absolute error obtained by the RFR is comparable (and even smaller) to the CNN-LSTM model proposed in [3].

Forecasting the level of air pollution is a challenging task, but it is not always necessary. The exact measure of the pollutant is not as important as the harm caused by the exposure to it. That is why the air quality index metric was introduced. The prediction was said to be correct when the predicted level was in the same category as the real value. When considering this simplified function, the accuracy of most models exceeded 80%, which can be considered as a satisfactory result and, with a dose of caution, used in real life applications.

6.1 Possible improvements and future works

There are many possible improvements that can be considered for future works. In many papers, deep learning methods give better results, such as LSTM (long short-term memory) and deep layers. It could also be checked whether training model on data only from a certain station results in better performance.

Bibliography

[1] Instytut Meteorologii i Gospodarki Wodnej, dane publiczne. https://danepubliczne.imgw.pl/.

[2] G. Battista and R. de Lieto Vollaro. Correlation between air pollution and weather data in urban areas: Assessment of the city of rome (italy) as spatially and temporally independent regarding pollutants. *Atmospheric Environment*, 165:240–247, 2017.

[3] A. Bekkar, B. Hssina, S. Douzi, and K. Douzi. Air-pollution prediction in smart city, deep learning approach. *Journal of Big Data*, 8(1), 2021.

[4] M. Castelli, F. M. Clemente, A. Popovič, S. Silva, and L. Vanneschi. A machine learning approach to predict air quality in california. *Complexity*, 2020:1–23, 2020.

[5] I. D. Dinov. Data science and predictive analytics.

[6] Główny Inspektorat Ochrony Środowiska. Bank danych pomiarowych. https://powietrze.gios.gov.pl/pjp/archives.

[7] Główny Inspektorat Ochrony Środowiska. Informacje zdrowotne - indeks jakości powietrza. https://powietrze.gios.gov.pl/pjp/content/health˙informations.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer series in statistics. Springer, New York, NY, 2 edition, Feb. 2009.

[9] K. Kumar and B. P. Pande. Air pollution prediction with machine learning: A case study of indian cities. *International Journal of Environmental Science and Technology*, 2022.

[10] S. Kumar and A. Dash. Seasonal variation of air quality index and assessment. *Global Journal of Environmental Science and Management*, 4(4):483–492, 2018.

[11] NSW Health. Common air pollutants and their health effects. https://www.health.nsw.gov.au/environment/air/Pages/common-air-pollutants.aspx.

[12] S. Oji and H. Adamu. Correlation between air pollutants concentration and meteorological factors on seasonal air quality variation. *Journal of Air Pollution and Health*, 5(1):11–32, May 2020.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[14] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[15] D. Solomatine and D. Shrestha. Adaboost.rt: a boosting algorithm for regression problems. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 2, pages 1163–1168 vol.2, July 2004.

[16] UNEP. 5 dangerous pollutants you're breathing in every day. https://www.unep.org/news-and-stories/story/5-dangerous-pollutants-youre-breathing-every-day.

[17] WHO. Air pollution website. https://www.who.int/health-topics/air-pollution.

[18] WHO. Ambient air quality database. https://www.who.int/data/gho/data/themes/air-pollution/who-air-quality-database.

Contents of CD disc

- A digital copy of the thesis
- Python notebook files with data processing and models training
- Training results in .xlsx file