Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Air Pollution Prediction with Machine Learning Algorithms

Bakalárska práca

2019 Filip Pavlove

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Air Pollution Prediction with Machine Learning Algorithms

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	Mgr. Pavel Petrovič, PhD.
Konzultant:	Mgr. Dušan Štefánik, PhD.

Bratislava, 2019

Filip Pavlove

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Filip Pavlove
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Air Pollution Prediction with Machine Learning Algorithms
Predpoveď znečistenia ovzdušia algoritmami strojového učenia

Anotácia: Znečisťovanie ovzdušia tuhými aerosólmi, oxidmi dusíka a prízemným ozónom patrí medzi najvážnejšie problémy v oblasti životného prostredia na Slovensku. SHMÚ disponuje 38 stanicami vybavenými prístrojmi na meranie koncentrácií znečisťujúcich látok. Pri prekročení limitov, napríklad v prípade prekročenia 12-h kĺzavého priemeru koncentrácií PM10 hodnoty 150 na meracej stanici, je vyhlasovaná výstraha pred závažnou smogovou situáciou. Momentálne sa teda výstrahy pred smogovými situáciami vydávajú na základe vyhodnotenia predošlých meraní. Veľmi žiadúcim je vedieť predpovedať smogové situácie s časovým predstihom. V súčasnosti sa takéto predpovede robia chemicko-transportnými modelmi a štatistickými metódami ako je strojové učenie. Chemicko-transportné modely, ktoré počítajú znečistenie ovzdušia na základe fyzikálno-chemických rovníc majú veľkú neurčitosť, najmä kvôli neurčitosti v meteorologických a emisných vstupoch. Preto sú v súčasnosti na predpovede znečistenia ovzdušia veľmi populárne algoritmy strojového učenia, ktoré predpovedajú s väčšou úspešnosťou.

Cieľ: Primárnym cieľom práce je vytvoriť program na predpovedanie koncentrácií znečisťujúcich látok v ovzduší pre vybrané stanice SHMÚ algoritmami strojového učenia na základe predpovedí meteorologických veličín a histórie meraní znečistenia. Študent by mal v práci porozumieť algoritmom v knižnici Scikit-Learn, prípadne TensorFlow. V práci bude testovať vhodnosť výberu daného algoritmu, ako aj mieru závislosti predpovedaného znečistenia od vstupných prvkov: jednotlivých meteorologických parametrov a časových údajov. Študent by sa mal naučiť štatisticky spracovávať obrovské množstvo dát a graficky ich prezentovať.

Literatúra: Andreas C. Muller and Sarah Guido: Introduction to Machine Learning with Python, Published by O'Reilly Media, Inc., 1005, 2017.
Aurélien Géron: Hands-On Machine Learning with Scikit-Learn and TensorFlow, , Published by O'Reilly Media, Inc., 1005, 2017.
Dixian Zhu 1,*, Changjie Cai 2, Tianbao Yang 1 and Xun Zhou: A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization.
https://medium.com/mongolian-data-stories/ulaanbaatar-air-pollution-part-1-35e17c83f70b

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Kľúčové slová:	predpoveď znečistenia ovzdušia, strojové učenie
Vedúci:	Mgr. Pavel Petrovič, PhD.
Konzultant:	Mgr. Dušan Štefánik, PhD.
Katedra:	FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry:	prof. Ing. Igor Farkaš, Dr.
Dátum zadania:	15.10.2018
Dátum schválenia:	28.05.2019

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

študent

vedúci práce

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname: Filip Pavlove
Study programme: Applied Computer Science (Single degree study, bachelor I. deg., full time form)
Field of Study: Applied Informatics
Type of Thesis: Bachelor´s thesis
Language of Thesis: English
Secondary language: Slovak

Title: Air Pollution Prediction with Machine Learning Algorithms

Annotation: Solid particles, nitrogen oxids, and ground level ozone air pollution are among the most serious environmental problems in Slovakia. Slovak Hydro-Meteorological Institute manages 38 stations equipped with devices for measuring emissions concentrations. Warnings of severe smog situation are declared when the critical thresholds are exceeded. For instance the 12h floating average of PM10 should not exceed a value of 150 at any measuring station. At the moment, the warnings of smog situation are declared based on evaluation of measurements already performed. Of a very high interest is the possibility to predict the smog situation in advance. Such predictions are performed using chemical-transport models and statistical methods of machine learning. The chemical-transport models that calculate the air pollution from physics-chemical equations have large degree of uncertainty, mainly due to the uncertainty in the meteorological and emission inputs. For this reason, the machine learning algorithms are currently more popular having higher prediction success rate.

Aim: The primary goal of the thesis is to develop a program for predicting the concentrations of air pollutants for selected measuring stations of SHMÚ utilizing machine learning algorithms applied to meteorological variables and the history of pollution measurements. Student needs to learn about algorithms in the Scikit-Learn library, and optionally TensorFlow. Suitability of the selected algorithm will be evaluated in the thesis, and the dependence of predicted pollution on input elements: the meteorological parameters and time. The student will learn to statistically evaluate large amount of data and present them in a graphical form.

Literature: Andreas C. Muller and Sarah Guido: Introduction to Machine Learning with Python, Published by O'Reilly Media, Inc., 1005, 2017
Aurélien Géron: Hands-On Machine Learning with Scikit-Learn and TensorFlow, , Published by O'Reilly Media, Inc., 1005, 2017
Dixian Zhu 1,*, Changjie Cai 2, Tianbao Yang 1 and Xun Zhou: A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization
https://medium.com/mongolian-data-stories/ulaanbaatar-air-pollution-part-1-35e17c83f70b

Keywords: air pollution prediction, machine learning

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Supervisor: Mgr. Pavel Petrovič, PhD.
Consultant: Mgr. Dušan Štefánik, PhD.
Department: FMFI.KAI - Department of Applied Informatics
Head of department: prof. Ing. Igor Farkaš, Dr.

Assigned: 15.10.2018

Approved: 28.05.2019 doc. RNDr. Damas Gruska, PhD.
Guarantor of Study Programme

....

Student Supervisor

Poďakovanie: Ďakujem môjmu školiteľovi Mgr. Pavlovi Petrovičovi, PhD. a konzultantovi Mgr. Dušanovi Štefánikovi, PhD. za cenné rady pri vypracovaní bakalárskej práce.

Abstrakt

Vysoké koncentrácie znečisťujúcich látok predstavujú veľkú hrozbu pre dnešnú aj budúcu populáciu. Predložená práca sa zaoberá predpoveďou znečistenia ovzdušia pre látky PM_{10} a NO_2. Celá práca bola vypracovaná v súlade s CRISP-DM procesom, ktorý pozostáva z viacerých krokov ako je porozumenie problému, analýza dát, spracovanie dát a modelovanie. Predpovede boli vytvorené pre stanicu umiestnenú na rušnej dopravnej ulici Bratislava - Trnavské Mýto. Na doplnenie chýbajúcich dát sme použili K-Nearest-Neighbours algoritmus. Na modelovanie predpovede boli použité štyri metódy. Štatistická metóda (Seasonal naive) slúži ako orientačná, od ktorej by mali ďalšie tri modely strojového učenia sa (MLP/ RNN/ LSTM) dávať lepšie výsledky. Pre všetky neurónové siete (NN) sme zvolili architektúru s práve jednou skrytou vrstvou. Predpovede boli vždy vytvorené o 23:00 v hodinových krokoch na ďalšie tri dni. Ako nezávislé premenné do modelov boli použité dáta z predošlých meraní kvality ovzdušia a meteorologických parametrov. Ako sa predpokladalo, výsledky neurónových sietí prekonali jednoduchú štatistickú metódu, ale rozdiely medzi nimi neboli prelomové. V rámci neurónových sietí jediný výrazný rozdiel sme pozorovali medzi sekvenčnými modelmi (RNN/ LSTM) a MLP pri modelovaní predpovedí NO_2, kde výsledky sekvenčných modelov prekonávajú MLP.

Kľúčové slová: strojové učenie, predpovedanie, znečistenie ovzdušia

Abstract

High concentrations of air pollutants are a major threat to current and future populations. This work deals with air pollution predictions for PM_{10} and NO_2. It follows the CRISP-DM process which consists of multiple steps such as problem understanding, data analysis, data pre-processing, and modeling. The forecasts were created for the station located on the busy traffic street Bratislava-Trnavské Mýto. To address missing data K-Nearest-Neighbours algorithm was used. Four methods were used for predictions, one of which was a statistical method (Seasonal naïve) to serve as a benchmark, which was expected to outperform by the other three machine learning models (MLP/ RNN/ LSTM). For all neural networks, we chose architecture with one hidden layer. Predictions were always made at 23:00 in hourly steps for the next three days. Previous measurements of the air quality and meteorological data were used as independent variables. As expected, results of neural networks outperformed simple statistical method. However, differences were not significant among them. We only observed a significant difference for NO_2 forecast modeling when compared sequential models (RNN/ LSTM) with MLP. The results from sequential models outperformed MLP.

Keywords: machine learning, forecasting, air pollution

Contents

List of Figures

List of Tables

Introduction

Nowadays, air pollution is an important problem because of potential harmful effects on human health and the environment. Therefore, air pollution forecasting is an important issue to be improved. Most existing systems for forecasting use chemical transport models (CTM) which are used for modeling of 2D or 3D grid of predicted concentrations. Many studies mentioned in Section 1.5 propose machine learning algorithms to forecast air pollution. In this thesis, we will focus on forecasting of air pollution concentrations for NO_2 and PM_{10} for station Bratislava-Trnavské Mýto. Predictions will be made at 23:00 in hourly intervals 3 days in advance, meaning 72 hours of predictions for NO_2 as well as PM_{10}.

Project methodology

Through the whole thesis, we will be following CRISP-DM (Cross Industry Process for Data Mining) standard process which consists of six phases described in Figure 1 [20]. As shown in the diagram we can see that CRISP-DM process is not linear. CRISP-DM consist of cycle indicated with inner and outer arrows. Wirth & Hipp stated: *The sequence of the phases is not strict. The arrows indicate only the most important and frequent dependencies between phases, but in a particular project, it depends on the outcome of each phase which phase, or which particular task of a phase, has to be performed next* [20].

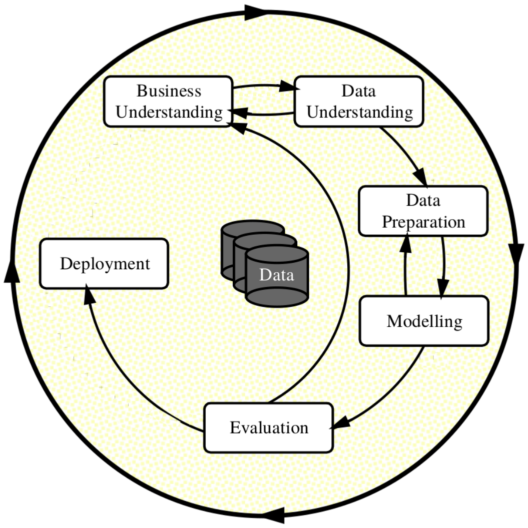


Figure 1: CRISP-DM diagram [20] showing the most important dependencies.

Following the CRISP-DM process, the first phase focused on understanding thesis objectives is presented in Section 1.3. Data understanding phase consisting of understanding initial data and closer insights through statistics is covered in Chapter 2 and Chapter 3, respectively. Data preparation phase covering all activities to construct final dataset for modeling is presented in Chapter 4. The modeling methodologies from Section 1.2 were applied with final architectures and parameters described in Chapter 5. Evaluation phase with performance metrics from Section 1.1 is presented in Chapter 6. It is important to note that deployment is not goal of thesis thus we will not perform the final phase.

1. Preliminaries

1.1 Metrics

The predictions accuracy will be evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Bias (MB), Mean Fractional Bias (MFB), Mean Fractional Error (MFE), and Pearson's correlation coefficient (r). The size of the test set consists of N data points, y_i and $\hat{y}_i$ represent real and modeled values, respectively. Metrics are defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}\,, \tag{1.1}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|\,, \tag{1.2}$$

$$MB = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)\,, \tag{1.3}$$

$$MFB = \frac{1}{N}\sum_{i=1}^{N}\frac{(\hat{y}_i - y_i)}{(\hat{y}_i + y_i)/2}\,, \tag{1.4}$$

$$MFE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{(\hat{y}_i + y_i)/2}\,, \tag{1.5}$$

$$r = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(\hat{y}_i - \bar{\hat{y}})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}\,. \tag{1.6}$$

RMSE and MAE are frequently used scale dependent measures of accuracy. Accuracy is dependent on $e = y - \hat{y}$, and is unit dependent. MB is scale dependent average representing systematic error of predictions. Mean Fractional Error and Mean Fractional Bias are a normalized version of Mean Absolute Error and Mean Bias respectively. Since the MFB ranges from -200% to $+200\%$ and MFE ranges from 0% to $+200\%$, these metrics have the advantage of bounding the maximum bias and error and do not allow a few data points to dominate the metric [2]. Pearson's correlation coefficient (r) is often used in statistics to measure the linear relationship between two variables. Value $r = 0$ indicates no relationship, $r = -1$, and $r = 1$ indicates strong negative and positive relationship respectively.

Each model in the experiment was tested by Hold-out method (splitting dataset into training set and test set) with a training set of 3649 samples (2003-2013) and a test set of 1823 samples (2013-2017). The results of experiments will be provided in Table 6.1 with figures of performance metrics for each hour of 72-hour forecast.

1.2 Modeling methodologies

In Section 1.2, we will be using convention for MLP, RNN, and LSTM adopted by Lipton et al. [12].

1.2.1 Seasonal naïve

Seasonal naïve is one of the simplest statistical time series forecasting methods. The method is serving as a benchmark for models presented later in the experiment. Forecasts are produced as follows

$$\hat{y}_{T+h|T} = y_{T+h-m\cdot(k+1)}, \tag{1.7}$$

where m is seasonal period and k is an integer part of $\frac{(h-1)}{m}$. The term $\hat{y}_{T+h|T}$ means the forecast of y_{T+h} taking account of y_1,...,y_T. As seasonal period was chosen $m = 7 \cdot 24$, which produces forecasts as exact copy measured from last week. Since the length of three day forecast is smaller than a seasonal period ($72 < 7 \cdot 24$), the formula can be rewritten as follows

$$\hat{y}_{T+h|T} = y_{T+h-m} \, . \tag{1.8}$$

1.2.2 MLP

The fundamental building block of a multilayer perceptron is a neuron or sometimes called perceptron. A multilayer perceptron is a type of feed forward neural network (FFNN) consisting of input layer, at least one hidden layer and an output layer. Example of a network with single hidden layer excluding bias can be represented as in Figure 1.1.

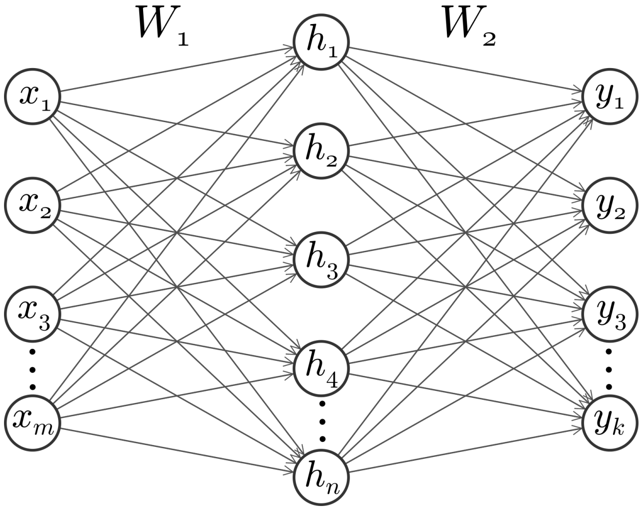


Figure 1.1: Example of MLP with a single hidden layer.

Following the convention from [12] we index neurons with j and j'. $w_{jj'}$ denotes the "to-from" weight corresponding to the directed edge to neuron j from node j' [12]. Each neuron in MLP computes value a_j as a weighted sum of its input in_j, see Equation (1.9).

It then becomes an argument to some differentiable activation function g_j. In the case of sigmoid, the activation is given by Equation (1.10).

$$in_j = \sum_{j'} w_{jj'} a'_j \,, \tag{1.9}$$

$$a_j = g(in_j) = \frac{1}{1 + e^{-\lambda in_j}} \,, \tag{1.10}$$

where λ is the slope of sigmoid. Another common choice for an activation function is Rectified Linear Unit (*ReLU*). The *ReLU* function is defined as follows

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \,. \tag{1.11}$$

Learning MLP

In the process of forward propagating values of neurons in each layer is computed by inputs from its lower layer until output $\hat{y}$ is generated. In supervised learning, learning is accomplished by finding the weights W of the network that minimizes cost function J.

$$W^* = \underset{W}{\text{argmin}} J(W) \tag{1.12}$$

The weights W^* are the weights found by minimizing cost function J. Cost function is computed as mean error of loss given by:

$$J(W) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x^{(i)}; W), y^{(i)}) \tag{1.13}$$

$$= \frac{1}{N} \sum_{i=1}^{N} (f(x^{(i)}; W) - y^{(i)})^2 \,, \tag{1.14}$$

where the size of a dataset consisting of N datapoints. The predicted output of i-th datapoint is $\hat{y}^{(i)} = f(x^{(i)}; W)$ and actual value to be predicted is $y^{(i)}$. The error between predicted and real value is given by loss $\mathcal{L}(\hat{y}, y)$. A common choice for loss is squared error. The minimum loss is found using gradient descent which iteratively computes gradient $\frac{\partial J(W)}{\partial W}$ and updates weights $W \leftarrow W - \eta \frac{\partial J(W)}{\partial W}$ of the network in the opposite direction of that gradient. The term η (eta) refers to the learning rate which is the size of step in the opposite direction of the gradient in the landscape of loss.

The algorithm used for training neural networks with gradient descent is backpropagation which computes derivates of loss with respect to parameters of a network using the chain rule.

1.2.3 RNN

Recurrent neural networks are a type of artificial neural networks. They were designed to work with sequences such as text, handwriting, the spoken word, or numerical times series data. Information from time $(t-1)$ is maintained using hidden state $h^{(t)}$. This behavior allows RNNs to maintain information in $h^{(t)}$ across multiple time-steps. The output $\hat{y}^{(t)}$ and hidden state $h^{(t)}$ at a time (t) in Jordan's architecture where output at a time $(t-1)$ is fed to input is computed as follows

$$\begin{aligned} h^{(t)} &= g(W^{hx}x^{(t)} + W^{hh}y^{(t-1)} + b_h)\,, \\ \hat{y}^{(t)} &= g(W^{yh}h^{(t)} + b_y)\,, \end{aligned}$$

vector $x^{(t)}$ refers to input at time t, weights W^{hx} to weights between input and hidden layer and W^{hh} to recurrent weights between the hidden layer and itself. Vectors b_y and b_h are bias terms. Recurrent neural networks are usually trained using Backpropagation through time algorithm. Learning of RNNs can be especially challenging due to vanishing or exploding gradient problems [12]. The vanishing gradient problem occurs, when gradients of the earlier layers get smaller (vanishes) which results in very slow learning of the weights in the lower layers. On the other hand, the exploding gradient occurs, when gradients in the layer get bigger (explodes) which results in unstable training.

1.2.4 LSTM

Long Short Term Memory (LSTM) is a type of recurrent neural network, developed to tackle vanishing and exploding gradient problem of simple RNN. Each neuron in the hidden layer replaces memory cell. Each memory cell contains a node with a self-connected recurrent edge of fixed weight one, ensuring that the gradient can pass across many time steps without vanishing or exploding [12]. Each memory cell c in step (t) consists of: Input Node $g_c^{(t)}$, Input Gate $i_c^{(t)}$, Internal State $s_c^{(t)}$, Forget Gate $f_c^{(t)}$, Output Gate $o_c^{(t)}$, and produces Output $v_c^{(t)}$. Following the [12] notation, the terms without the subscript c are vectors of values. For example, s is a vector of internal states s_c in a layer. It is important to note, that vectors of outputs v_c is denoted as h. Computations of LSTM in the forward pass at time step (t) is defined as:

$$g^{(t)} = \sigma(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g) \tag{1.15}$$

$$i^{(t)} = \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i) \tag{1.16}$$

$$f^{(t)} = \sigma(W^{fx}x^{(t)} + W^{fh}h^{(t-1)} + b_f) \tag{1.17}$$

$$o^{(t)} = \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o) \tag{1.18}$$

$$s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)} \tag{1.19}$$

$$h^{(t)} = \phi(s^{(t)}) \odot o^{(t)} \tag{1.20}$$

The Matrix W^{ij} refers to edges between units i and j, activations σ and ϕ to sigmoid and tanh, respectively. The operation $\odot$, represent pointwise multiplication. The output of the layer is $h^{(t)}$ and output at the previous time step is $h^{(t-1)}$.

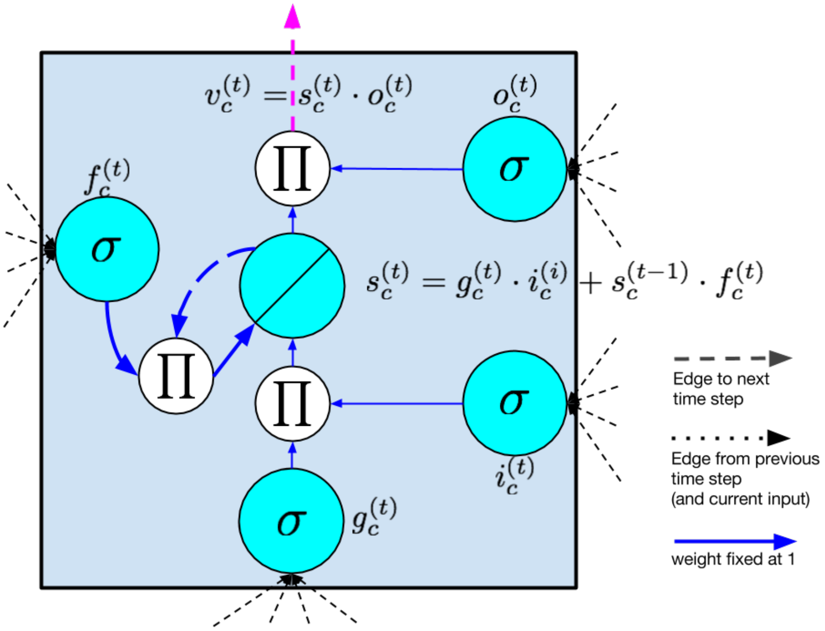

$v_c^{(t)} = s_c^{(t)} \cdot o_c^{(t)}$
$o_c^{(t)}$
$f_c^{(t)}$
$s_c^{(t)} = g_c^{(t)} \cdot i_c^{(i)} + s_c^{(t-1)} \cdot f_c^{(t)}$
$i_c^{(t)}$
$g_c^{(t)}$
Edge to next time step
Edge from previous time step (and current input)
weight fixed at 1

Figure 1.2: Single memory cell [12].

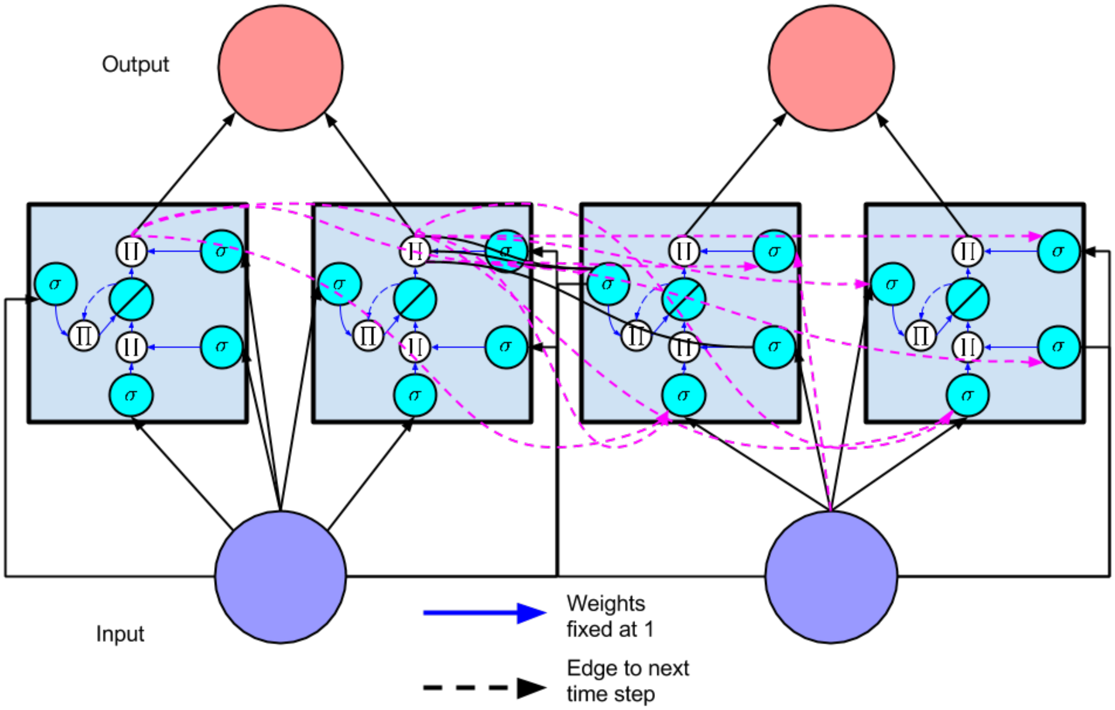
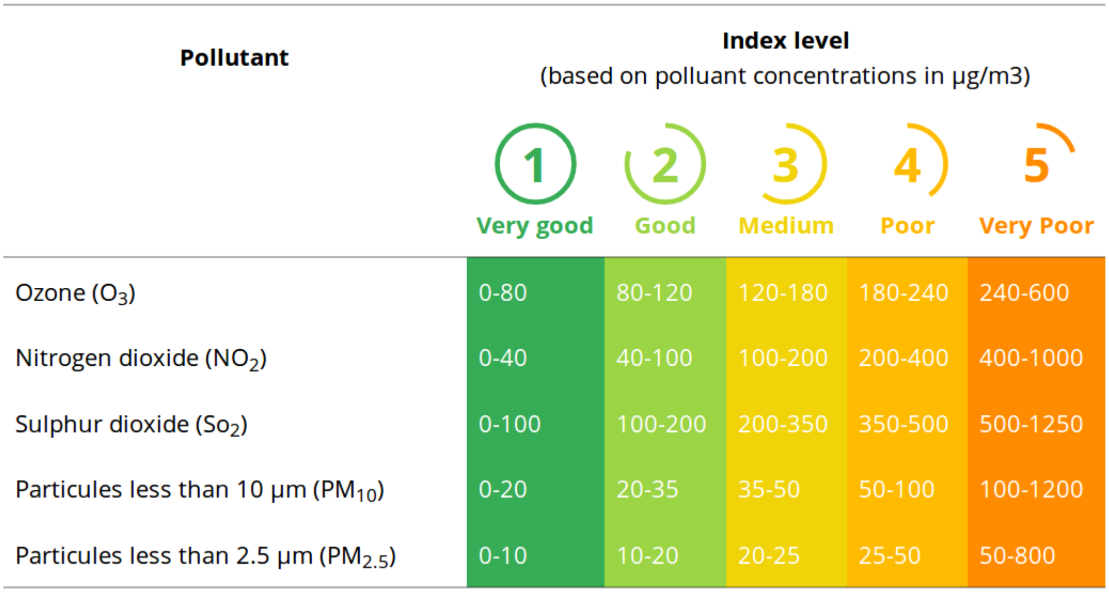


Figure 1.3: Single layer unfolded across two timesteps [12].

1.3 Air quality

1.3.1 Air quality index

Air pollution concentrations are measured in $\mu g/m^3$. However, government agencies use various types of air quality indices (AQI) to inform public about the dangers of measured concentrations. For example in Figure 1.4 [7] is presented European air quality index. AQI values are shown in qualitative categories of air pollution which represents concentrations of pollutants.



Pollutant	Index level (based on polluant concentrations in µg/m3)				
	1 Very good	2 Good	3 Medium	4 Poor	5 Very Poor
Ozone (O_3)	0-80	80-120	120-180	180-240	240-600
Nitrogen dioxide (NO_2)	0-40	40-100	100-200	200-400	400-1000
Sulphur dioxide (So_2)	0-100	100-200	200-350	350-500	500-1250
Particules less than 10 µm (PM_{10})	0-20	20-35	35-50	50-100	100-1200
Particules less than 2.5 µm ($PM_{2.5}$)	0-10	10-20	20-25	25-50	50-800

Figure 1.4: Air quality index (AQI)

1.3.2 Air pollution sources

Air pollution occurs when harmful or excessive quantities of substances including gases, particles, and biological molecules are introduced into Earth's atmosphere. Both human activity and natural processes can generate air pollution [19]. Currently, in Europe there is a problem with pollutants PM_{10}, $PM_{2.5}$, O_3 and NO_2. Particular matter (PM_{10} and $PM_{2.5}$) are mainly produced by fuel in road transport and power generation combustion. Ground level ozone (O_3) unlike stratospheric ozone, which forms naturally is also a source of smog and is the product of an interaction between heat, sunlight, and man-made emissions from sources such as motor vehicles and industry. Ground level ozone is formed in higher concentrations during the summer and reaches its highest concentrations in the afternoon or early evening. Nitrogen dioxide (NO_2) is a highly reactive gas formed by emissions from motor vehicles, industry, and households. High concentrations of NO_2 can be found especially near busy roads. Outdoors, nitrogen dioxide contributes to the formation of ground-level ozone as well as particulate matter pollution [14].

1.3.3 Health effects

The definition of an air pollutant is any substance which may harm humans, animals, vegetation or material [9]. The different composition of air pollutants, the dose and time of exposure and the fact that humans are usually exposed to pollutant mixtures than to single substances, can lead to diverse impacts on human health. Human health effects can range from nausea and difficulty in breathing or skin irritation, to cancer [9].

1.4 Existing systems

The following Section presents some existing systems providing daily forecasts for air pollution.

1.4.1 Copernicus

The Copernicus Atmosphere Monitoring Service (CAMS) provides continuous data and information on atmospheric composition [5]. Copernicus provides hourly predictions of 96 hours interval for many pollutants. It uses ENSEMBLE MODEL which is based on seven state-of-the-art numerical air quality models developed in Europe [5].

1.4.2 THOR

The THOR is integrated weather and air pollution forecast system of models developed by Jørgen Brandt and Jesper H. Christensen at National Environmental Research Institute of Denmark. The system includes several meteorological and air pollution models used by external applications. System produces 72 hours of weather and air pollution forecast four times a day.

1.4.3 Met Office

The Met Office is the United Kingdom's national weather service. Air quality forecasts are produced by model AQUM [17] early in the morning for the current day as well as for the next 4 days.

1.4.4 AirNow

The U.S. EPA (Environmental Protection Agency) AirNow program is the national repository of real-time air quality data and next-day Air Quality Index (AQI) forecasts for United States.

1.5 Scientific studies

In the last decade many studies proposed to apply machine learning algorithms to air pollution predictions. Some researches defined air pollution forecasting as classification problem, however most studies are defined as regression. For instance Corani [6] used feed-forward neural network (FNN), pruned neural network (PNN), and lazy learning (LL) for predictions of O_3 and PM_{10} in Milan. The study is showing, that no significant differences are found between the forecast accuracies of the different models; nevertheless, LL provides the best performances on indicators related to average goodness of the prediction [6]. Lu et al. [13] focused on comparison of prediction abilities between support vector machine (SVM) and radial basis function (RBF). Study demonstrates, that SVM outperforms RBF network.

1.6 Toolkits

1.6.1 Source Code

Source code for a thesis is available at Github public repository [15].

1.6.2 Packages

The programming language of choice for the majority of work was *Python*. In some areas such as timeseries analysis *R* was used, especially library *stlplus*. We used Python since it offers a wide variety of packages from data processing and visualization to modeling. A library like *Pandas* makes it easier to manipulate and analyze data. Most of the descriptive statistics in the thesis were summarized with it. Extension of Pandas, *Geopandas* offers functionalities for manipulation with geospatial data used in the early stages of work. *Numpy* is the fundamental package for scientific computing, adding support for large, multi-dimensional arrays and matrices. As a visualization tool, we used plotting libraries *Matplotlib* and its extention *Seaborn* and *Basemap* for plotting 2D data on a map. Data imputation with KNN and normalization was handled with *sckit-learn*. For modeling with MLP, RNN and LSTM was chosen *Keras*, which is an open source neural network library.

2. Input Data

In this chapter, we explore raw data obtained from SHMU air-quality and meteorological stations. We will describe formats in which data were gathered and a brief description of features containing them. In Section 2.4 we will present selected station for the experiment.

2.1 Data overview

For the purpose of the experiment we received data from 78 SHMU measuring stations shown in Figure 2.1. All air pollution measurements were taken at different spots than meteorological measurements. Both meteorological and air pollution measurements were taken in hourly intervals from 1.1.2003 to 31.12.2017 except $PM_{2.5}$ which holds measurements from 1.1.2005 to 31.12.2017.

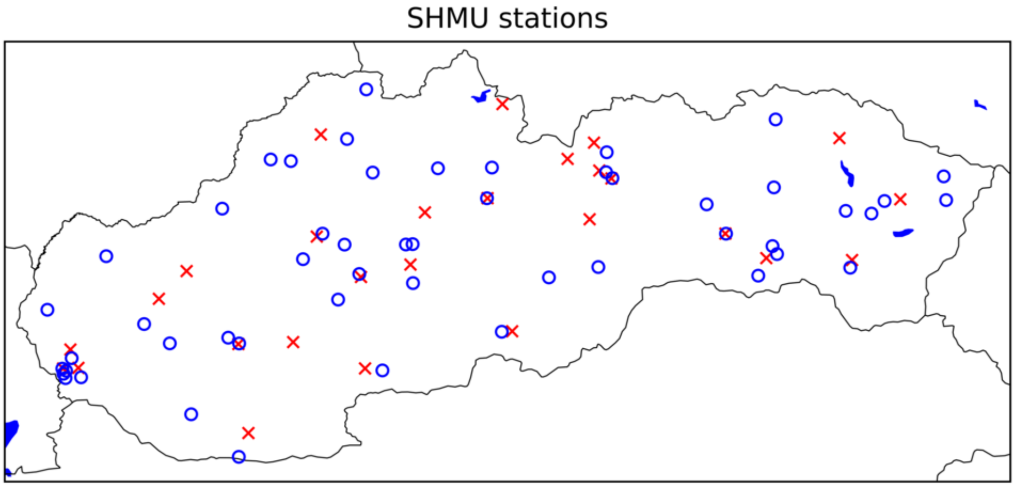


Figure 2.1: Meteorological stations (red cross), Air-pollution stations (blue circle).

2.2 Meteorological data

Meteorological data were collected from 29 stations shown in Figure 2.1. Data were obtained in 29 csv files consisting of features such as wind speed, wind direction, temperature, etc. which are closely described in Table 2.1. Locations, names, and elevations corresponding to stations were obtained in a separate file consisting of 27 stations. Missing data appeared minimally with the exception of mean sea-level pressure. According to SHMU documentation, only stations with elevation smaller than 550 m are in a group tagged with code pppp. After closer examination 10 of 27 stations with elevation bigger than 550 m were missing this feature completely.

	code	*units*
air temperature	ttt	°C
dew point temperature	td	°C
wind direction	dd	°
wind speed	ff	m/s
mean sea-level pressure	pppp	hPa

Table 2.1: Meteorological features

In Table 2.1 column code refers to actual names of features listed in SHMU documentation. Code ttt represents an absolute value of air temperature in the time of measuring. Code td represents an absolute value of dew point temperature in the time of measuring. Code dd represents the average wind direction for the last 10 minutes before measuring. The closer description of coding dd is shown in Table 2.2. Code pppp represents values of station air pressure in time of measuring, recalculated on mean sea level with correlation to air temperature.

dd	*Meaning*	*dd*	*Meaning*	*dd*	*Meaning*
00	windless	13	125 - 134	26	255 - 264
01	05°- 14°	14	135°- 144°	27	265°- 274°
02	15°-24°	15	145°- 154°	28	275°- 284°
03	25°- 34°	16	155°- 164°	29	285°- 294°
04	35°- 44°	17	165°- 174°	30	295°- 304°
05	45°- 54°	18	175°- 184°	31	295°- 304°
06	55°- 64°	19	185°- 194°	32	315°- 324°
07	65°- 74°	20	195°- 204°	33	325°- 334°
08	75°- 84°	21	205°- 214°	34	335°- 344°
09	85°- 94°	22	215°- 224°	35	345°- 354°
10	95°- 104°	23	225°- 234°	36	355°- 04°
11	105°- 114°	24	235°- 244°	99	variable wind
12	115 - 124	25	245 - 254		

Table 2.2: Coding of wind direction (dd).

2.3 Air quality data

Air Pollution data were obtained in 4 csv files. Metadata of stations such as location, name, type of station, type of location were stored in a single shp file. Four pollutants files contain measured concentrations of NO_2, O_3, PM_{10} and $PM_{2.5}$. Pollution concentrations were measured in $\mu g/m^3$. Each of them holds records for a different number of stations. Specifically, NO_2 holds records for 50 stations, O_3 for 49, PM_{10} for 50 and $PM_{2.5}$ for 40. There were huge quantities of data missing as shown in Figure 2.2.

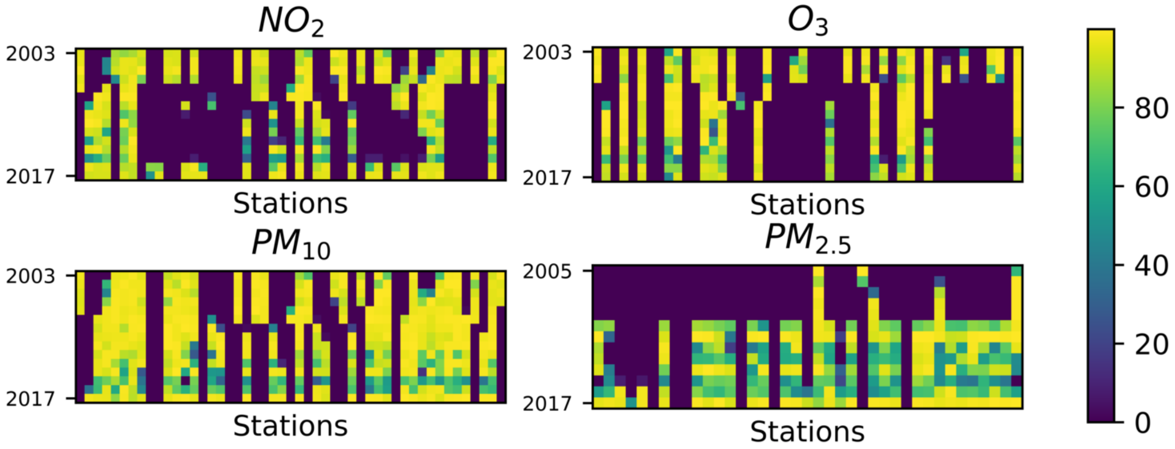


Figure 2.2: Coverage of measurements for different pollutants. The years of measurements are shown on the y-axis and measuring stations on the x-axis. The colour of each square represents the percentage of measured data for a given pollutant, station, and year.

2.4 Station selection

As described in Section 2.1 we need to choose meteorological stations close enough to air-pollution stations to truly represent the impact of meteorological values. In Section 2.2 we talked about missing mean-sea-level pressure at stations with elevation bigger than 550m. Finally, in Section 2.3 we described coverage of pollutant stations within different years. Taking into account all these factors we chose to work with air quality monitoring traffic station Bratislava-Tranvské Mýto and its closest meteorological station Bratislava Koliba. Since Trnavské Mýto did not measure ozone and $PM_{2.5}$ concentrations we decided to perform an experiment with NO_2 with coverage of 93.8% and PM_{10} with coverage of 96%.

3. Data Understanding

3.1 Data quality report

Our data quality report in Table 3.1 shows statistical measurements for the selected station from Section 2.4. Features shown in this report are divided into continuous and categorical features. Both columns for continuous and categorical features include count, percentage of missing values (Miss.%), and cardinality. In addition, continuous features contain calculated columns for minimum, first quartile (Q_1), mean, median, third quartile (Q_3), maximum and standard deviation. Categorical features also contain columns for the two most frequent levels for the feature (Mode and Mode2), frequency and percentage with which these appear. All features are plotted in Figures 3.2 and 3.2.

Trnavské Mýto

Continuous Features

Feature	Count	Miss.%	Card.	Min	Q_1	Mean	Median	Q_3	Max	Std. Dev.
pm_{10}	131496	3.99	92218	0.025	17.31	34.17	28.65	45.08	503.1	23.81
NO_2	131496	6.11	75051	0.088	22.57	41.32	36.55	54.87	208.0	24.84
ttt	131496	0.86	563	-20.0	3.4	10.72	10.8	17.6	38.9	9.2
td	131496	0.87	450	-24.5	-0.3	5.06	5.5	10.9	21.9	7.23
ff	131496	0.86	14	0.0	1.0	2.58	2.0	3.0	13.0	1.62
pppp	131496	0.91	646	976.4	1012.2	1017.11	1016.8	1022.0	1074.5	8.11

Categorical Features

Feature	Count	Miss.%	Card.	Mode	Mode Freq.	Mode%	Mode2	Mode2 Freq.	Mode2%
dd	131496	0.86	38	99.0	65905	50.56	30.0	490	4.45

Table 3.1: A data quality report, summarizing features of the dataset.

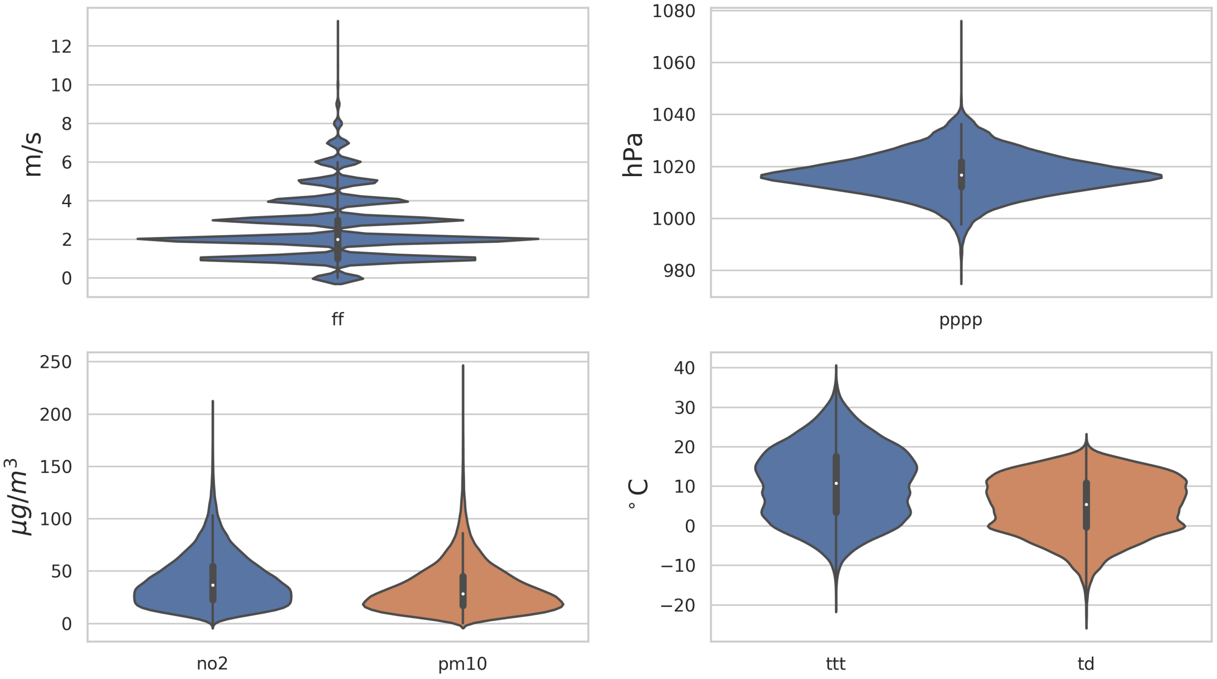


Figure 3.1: Violin plots showing the distribution of each meteorological parameter in Bratislava-Koliba station and $N0_2$ and PM_{10} measurements from Bratislava-Trnavské Mýto station. Inside of each violin is a box plot.

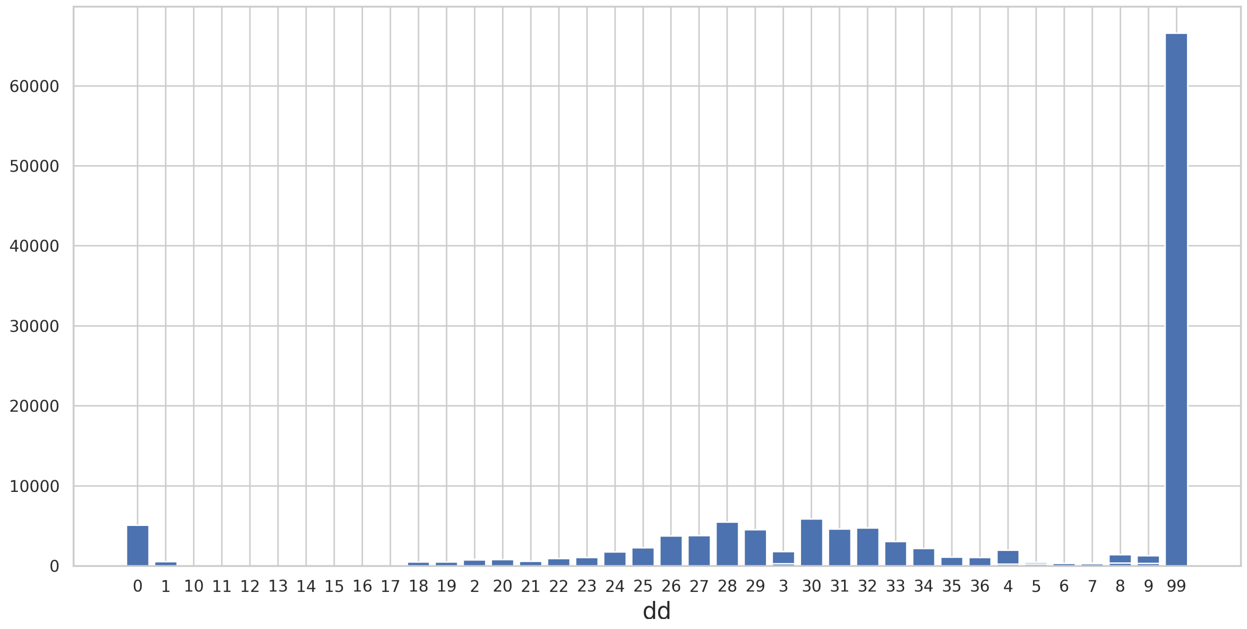


Figure 3.2: Wind direction (dd) bar plot, showing frequency of each category described in Table 2.2.

3.2 Time Series Analysis

A time series is a set of observations x_t, each one being recorded at a specific time t [3]. In this thesis we are dealing with a discrete-time time series. Time series often shows patterns such as trend, seasonality, and cyclic. A trend exists when there is a long-term increase or decrease in the data [8]. The trend does not have to be linear. A seasonal pattern occurs when a time series is affected by seasonal factors and is always of a fixed and known frequency[8]. We are expecting three seasonal patterns hour of the day, day of a week and month of a year. A cycle occurs when the data exhibit rises and falls that are not of a fixed frequency [8].

3.2.1 Time Series Decomposition

As we discussed time series tend to exhibit various patterns. In the process of decomposition, we decompose time series into components of trend component, seasonal component and remainder (sometimes called error or residual). We usually combine trend and cycle into a single trend-cycle component (sometimes called the trend for simplicity) [8]. Since multi-seasonal time series decomposition is beyond the scope of this thesis we will take a different approach to visualize and measure the strength of different seasonalities. We will be resampling time series observations using the average downsampling method from hourly frequencies observations to daily and monthly frequencies. The method used for decomposition called STL (Seasonal and Trend decomposition using Loess) developed by Cleveland et al. [4] returns components of time series as an additive decomposition defined as

$$y_t = S_t + T_t + R_t \, , \tag{3.1}$$

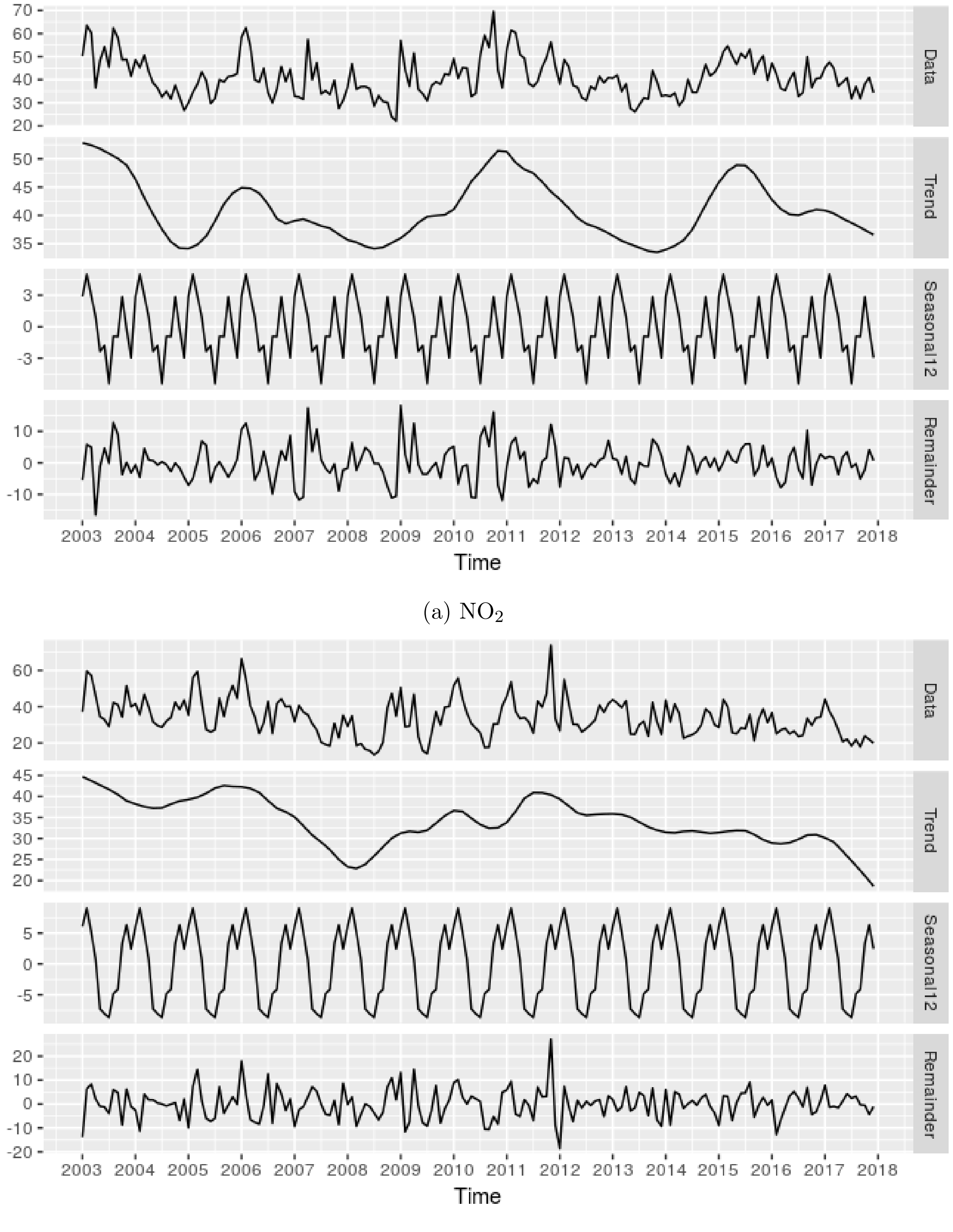
where y_t is time series, S_t is seasonal, T_t is a trend, and R_t is a remainder component. We are presenting STL decomposition of monthly measurements of PM_{10} and NO_2 in Figure 3.3. The strength of seasonality is defined as

$$F_s = max\left(0, 1 - \frac{Var(R_t)}{Var(S_t + R_t)}\right) , \tag{3.2}$$

where $Var()$ is variance. A series with seasonal strength F_s close to 0 exhibits almost no seasonality, while a series with strong seasonality will have F_s close to 1 [8].

Season	**Pollutant**	F_s
Daily	NO_2	0.275
	PM_{10}	0.085
Weekly	NO_2	0.293
	PM_{10}	0.077
Annual	NO_2	0.204
	PM_{10}	0.478

Table 3.2: Seasonal strength of PM_{10} and NO_2.



(a) NO_2

(b) PM_{10}

Figure 3.3: STL decomposition of monthly downsampled time series of PM_{10} and NO_2.

3.2.2 Autocorrelation

Autocorrelation is used in time series analysis, to measure the linear relationship between lagged values of series. Just as correlation shows how much two features are similar, autocorrelation shows how similar the time series feature is with itself. The value of r_k (k-th lag) can be written as

$$r_k = \frac{\sum\limits_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum\limits_{i=1}^{T} (y_t - \bar{y})^2} . \tag{3.3}$$

In Figure 3.4 are shown autocorrelation plots of 72 lags (3 days) for NO_2 as well as PM_{10}.

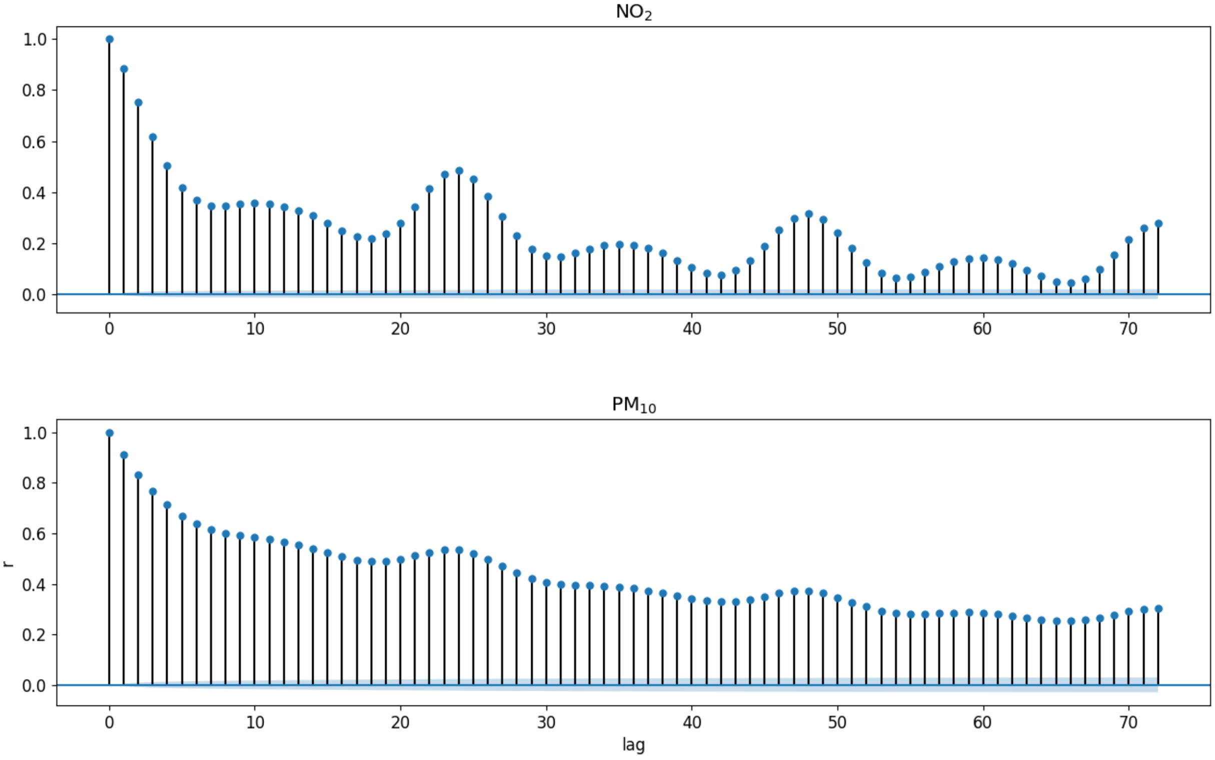


Figure 3.4: Autocorrelation plot is showing the autocorrelation coefficient on y-axis and different lags of the series on x-axis. For example, lag 10 is the correlation between time t and $t - 10$.

The periodicity shown in the Figure 3.4 is due to daily seasonality. The higher peaks in NO_2 part in contrast with PM_{10} part of the figure confirms the difference between the strength of daily seasonality in Table 3.2.

4. Data Preparation

4.1 Data quality issues

After getting to know the data it is important to take a closer look and identify some of the data quality issues. The most common are missing values, irregular cardinality, and outliers [10]. The main focus of this section is identifying these issues if they are present and correct them.

4.1.1 Missing values

In Table 3.1 column Miss.% highlight the percentage of missing values for each feature and Table 4.1 shows coverage of measured values within years. There are several methods used for dealing with missing data depending on specific circumstances. First, we are going to determine what type (also referred to as missingness mechanism) of missing values are we dealing with. In the literature [16, 1, 18] missingness mechanisms are generally referred to as: Missing Completely At Random (MCAR), Missing At Random (MAR) and Not Missing At Random (NMAR). In our case, we are assuming that MCAR is the most appropriate mechanism since missing data depends on technical errors. Missing data handling techniques can be classified into two classes Traditional (Conventional) Methods and Imputation Methods. Some of the conventional methods are Ignoring, Deletion, or Mean/Median/Mode Imputation [16]. Imputation methods could be Regression Imputation, Hot and Cold Deck Imputation, K-Nearest Neighbor (KNN) Imputation and many more. Nowadays, there are many studies [1, 18] comparing effectiveness of different methods depending on the nature of missing data.

Handling missing values

As we can see in Table 4.1 meteorological values were missing in much smaller quantities except for the year 2010 in which data were missing in one continuous gap. A solution for the 2010 gap we decided to fill in data with observations from last year. For the rest of meteorological data we performed mean/mode imputation with respect to the nearest observed values. For air pollution data we decided to perform K-Nearest Neighbor (KNN) imputation which is a non-parametric instance-based learning method used for classification or regression. The metric used for the algorithm was Euclidean distance. The Euclidean distance between two points in N-dimensional space is given by Equation (4.1). Target was predicted as a weighted mean of k nearest neighbors in the training set.

	pm10	no2	ttt	td	ff	pppp	dd
2003	99.1	87.2	99.97	99.97	99.97	99.97	99.97
2004	97.91	96.32	99.97	99.97	99.97	99.97	99.97
2005	98.76	95.27	100.0	99.85	100.0	100.0	100.0
2006	98.74	98.0	99.92	99.92	99.92	99.92	99.92
2007	97.69	98.14	100.0	100.0	100.0	100.0	100.0
2008	99.36	99.53	99.92	99.85	99.92	99.92	99.92
2009	98.93	96.62	100.0	100.0	100.0	100.0	100.0
2010	98.47	82.91	88.36	88.36	88.37	88.36	88.37
2011	96.54	83.61	99.71	99.71	99.71	99.71	99.71
2012	88.13	94.24	99.69	99.69	99.62	99.67	99.62
2013	95.53	99.63	99.61	99.61	99.61	99.61	99.61
2014	94.77	98.26	100.0	100.0	100.0	100.0	100.0
2015	96.77	97.09	99.98	99.98	99.98	99.29	99.98
2016	89.83	90.66	100.0	100.0	100.0	100.0	100.0
2017	89.58	90.79	99.99	99.99	99.99	99.99	99.99

Table 4.1: Coverage of variables.

$$d(p,q) = \sqrt{\sum_{i=1}^{N}(q_i - p_i)^2}\,. \tag{4.1}$$

Overall, we found 7492 instances of missing NO_2, 4570 instances of PM_{10} and 1920 instances of missing both NO_2 and PM_{10} at the same time. For each of the next cases: predicting PM_{10} with and without NO_2, and predicting NO_2 with and without PM_{10}; we trained different models tested by Hold-out method in which we partitioned training set to test set in ratio 75:25. A number of neighbors (k) used in the models was chosen by best MAE given in Figure 4.1. Predictors of all models were normalized using range normalization to scale from 0 to 1.

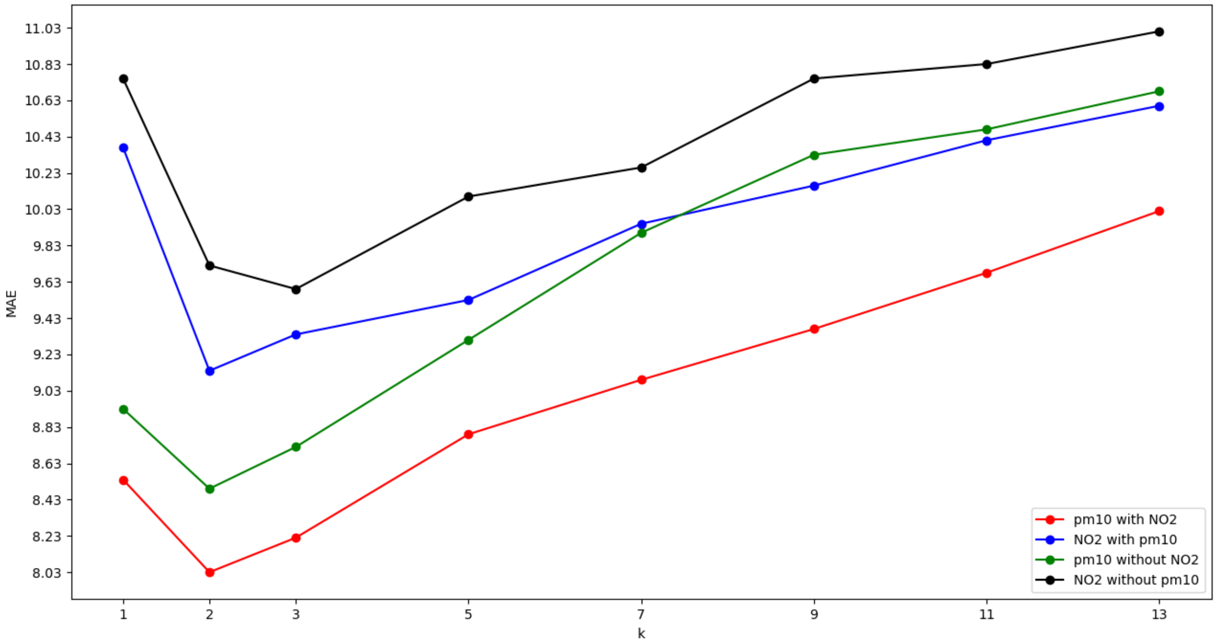


Figure 4.1: Mean absolute errors of four models (predicting PM_{10} with and without NO_2, and predicting NO_2 with and without PM_{10}) are shown on y-axis. Parameters k of the KNN-imputation models are shown on x-axis.

4.1.2 Irregular cardinality

In the process of examining possible irregular cardinality, we mainly focus on the column Card. in Table 3.1. Only irregularity found is ff (wind speed) with the cardinality of 14 which is unusual for continuous features. After closer examination of the data we can see that values of wind speed were rounded on the whole number and measured in m/s thus we can assume that nothing is wrong.

4.1.3 Outliers

Outliers are values that lie far away from the central tendency of a feature. There are two kinds of outliers that might occur; valid outliers and invalid outliers [10]. We did not find any invalid outliers in the whole dataset, however valid outliers were found in PM_{10} and NO_2. Since all outliers lie in the range of possible concentrations in AQI Table 1.4 we chose to not handle them.

4.2 Normalizations

4.2.1 Range Normalization

Normalization is a technique often used in machine learning to transform features within different ranges to a common scale. The simplest approach is range normalization which is given by

$$a_i' = \frac{a_i - min(a)}{max(a) - min(a)} \times (high - low) + low\,, \tag{4.2}$$

where a_i' is a normalized feature and a_i is original value. The term *high* and *low* refers to range in which we want to scale the data.

4.2.2 Standard Score Normalization

Another approach to normalizing the data is to standardize them into standard scores, as follows

$$a_i' = \frac{a_i - \bar{a}}{sd(a)}\,, \tag{4.3}$$

where $\bar{a}$ is mean and $sd(a)$ is the standard deviation of the feature. This way is accomplished that mean of feature is zero and standard deviation is one. Standard score normalization was applied in the experimental phase of the thesis.

4.2.3 Handling Circular Features

Numerical values of circular features such as hours, weekday or wind direction are not fully representative of its meaning. For example, sunday is close to monday however their numerical values are not. One way to deal with this problem is to transform this feature into two features given by

$$\sin\_a_i = \sin\left(\frac{2\pi a_i}{\max(a)}\right)\,, \tag{4.4}$$

$$\cos\_a_i = \cos\left(\frac{2\pi a_i}{\max(a)}\right)\,. \tag{4.5}$$

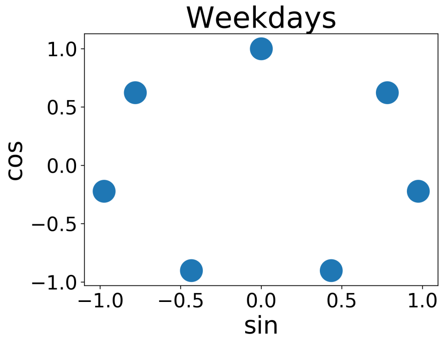


Figure 4.2: Example of weekday transformation (Sunday on top).

5. Proposed Solution

We started with MLP as a benchmark to other neural network models. Next, we used vanilla RNN which is more suitable for sequenced data modeling. The last method we chose was the improved recurrent neural network LSTM, to be compared with vanilla RNN.

5.1 Experimental Setup

Parameters of all models were picked with trial and error procedure. To avoid overfitting, we chose to use *early stopping*, which refers to stopping the training process before an error on the validation set starts to increase.

5.1.1 NN Architecture

All models described below consist of a single hidden layer and an output layer of 72 outputs. The output vector with linear activation function represents 72 hours step forecast. *Adam* [11] optimization algorithm was used for the training process. Instead of a single stationary learning rate as in Stochastic Gradient Descent, Adam maintains learning rates per weights, which are adapted through the training process. Last 24 observations of predicted pollutant were used as inputs to MLP. The hidden layer consists of 150 neurons with *ReLU* activation function. Inputs used for RNN and LSTM were multiple sequences of the last 24 observations shown in Figure 5.1. For a hidden layer, we chose 180 neurons and 200 memory cells for RNN and LSTM, respectively. Both used *tanh* activation function within a hidden layer.

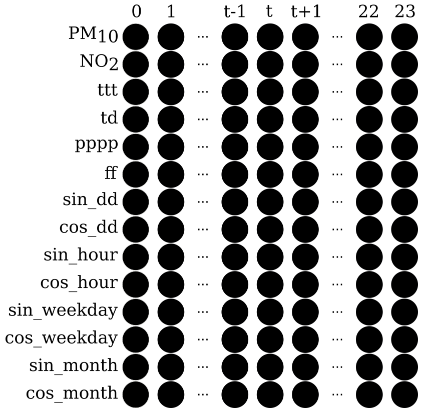


Figure 5.1: Input sequences for LSTM and RNN.

6. Results

In this chapter, we are presenting the final results of the experimental phase. Table 6.1 shows the preformance of each model. *SN* refers to the Seasonal naïve method. Column *Overall* summarize the mean performance of models for 72 hours of the forecast, columns *Day*1, *Day*2, *Day*3 mean of 24 hours of each day separately.

The Figures 6.1-6.6 show performance of models for each hour of the forecast. From figures and Table 6.1 it can be concluded that machine learning models outperform simple Seasonal naïve model. Difference between MLP, RNN, and LSTM is not as significant for PM_{10}, however NO_2 predictions of recurrent networks are better than simple MLP. The occurring daily periodicity shown in figures can be explained with the location of the measuring station on a busy traffic street. The bigger variance of measured air pollution can occur due to population traveling to the work and back home; consequently, errors of models peak approximetly at 8:00 and 20:00.

In the work [2], authors introduced goals and criteria of PM modeling for three-dimensional air quality models. Nevertheless, authors established criteria and goals of three-dimensional modeling, we adopted them in the work as well for both NO_2 and PM_{10}. Boyland and Rusall lay goals and criteria as follows: *Goal has been met when both the mean fractional error (MFE) and the mean fractional bias (MFB) are less than or equal to* +50% *and* ±30%*, respectively. Additionally, the model performance criteria has been met when both the* $MFE \leq +75\%$ *and* $MFB \leq \pm 60\%$ [2]. From the column *Overall* for NO_2 we can see that all models passed both criteria and goals, however in Figure 6.3, is shown, that Seasonal naïve method did not met goal of MFE $\leq +50\%$ for several hours of the forecast so we did not consider the model to met the goals. Column *Overall* for PM_{10} clearly shows that neural networks outperform the Seasosonal Naive method and all of them met performance goals.

		NO_2				PM_{10}			
		Overall	Day1	Day2	Day3	Overall	Day1	Day2	Day3
SN	RMSE	24.55	24.54	24.55	24.55	23.9	23.9	23.9	23.89
	MAE	18.12	18.11	18.12	18.13	17.52	17.52	17.52	17.51
	MB	**-0.04**	**-0.06**	**-0.04**	**-0.03**	**-0.11**	**-0.12**	**-0.11**	**-0.09**
	MFB	**-0.17**	**-0.24**	**-0.18**	**-0.11**	**-0.38**	**-0.43**	**-0.39**	**-0.32**
	MFE	48.1	48.06	48.1	48.15	58.57	58.56	58.57	58.58
	r	0.44	0.44	0.44	0.44	0.18	0.18	0.18	0.18
MLP	RMSE	20.43	18.82	21.07	21.3	16.63	14.08	17.25	18.27
	MAE	15.54	14.0	16.18	16.44	12.38	10.14	13.02	13.97
	MB	2.17	1.85	2.31	2.35	-2.08	-1.91	-1.84	-2.5
	MFB	-6.23	-5.46	-6.53	-6.7	-18.15	-16.07	-17.98	-20.39
	MFE	41.54	37.09	43.42	44.11	42.46	35.83	44.49	47.06
	r	0.48	0.59	0.43	0.41	0.48	0.66	0.41	0.31
RNN	RMSE	18.05	16.71	18.53	18.82	16.16	13.78	16.74	17.71
	MAE	13.6	12.36	14.08	14.37	11.99	9.96	12.56	13.47
	MB	-0.63	-0.84	-0.48	-0.58	-1.25	-0.95	-1.22	-1.59
	MFB	-10.6	-10.24	-10.54	-11.0	-11.61	-8.25	-12.12	-14.48
	MFE	36.99	**33.27**	38.52	39.17	41.97	36.31	43.61	45.98
	r	0.63	0.69	0.6	0.59	0.53	0.68	0.49	0.4
LSTM	RMSE	**18.01**	**16.68**	**18.46**	**18.81**	**15.78**	**13.72**	**16.19**	**17.22**
	MAE	**13.56**	**12.34**	**13.99**	**14.34**	**11.78**	**9.91**	**12.26**	**13.16**
	MB	-0.77	-0.59	-0.74	-0.96	-1.82	-1.08	-1.92	-2.47
	MFB	-11.21	-9.17	-11.73	-12.73	-16.23	-12.11	-17.19	-19.39
	MFE	**36.81**	33.6	**37.99**	**38.85**	**40.68**	**35.05**	**42.27**	**44.73**
	r	**0.63**	**0.69**	**0.61**	**0.59**	**0.55**	**0.68**	**0.51**	**0.42**

Table 6.1: Daily and Overall performance of models. RMSE, MAE, and MB are in units $\mu g/m^3$. The best results of models for given metric and time period for both pollutants are highlighted in bold.

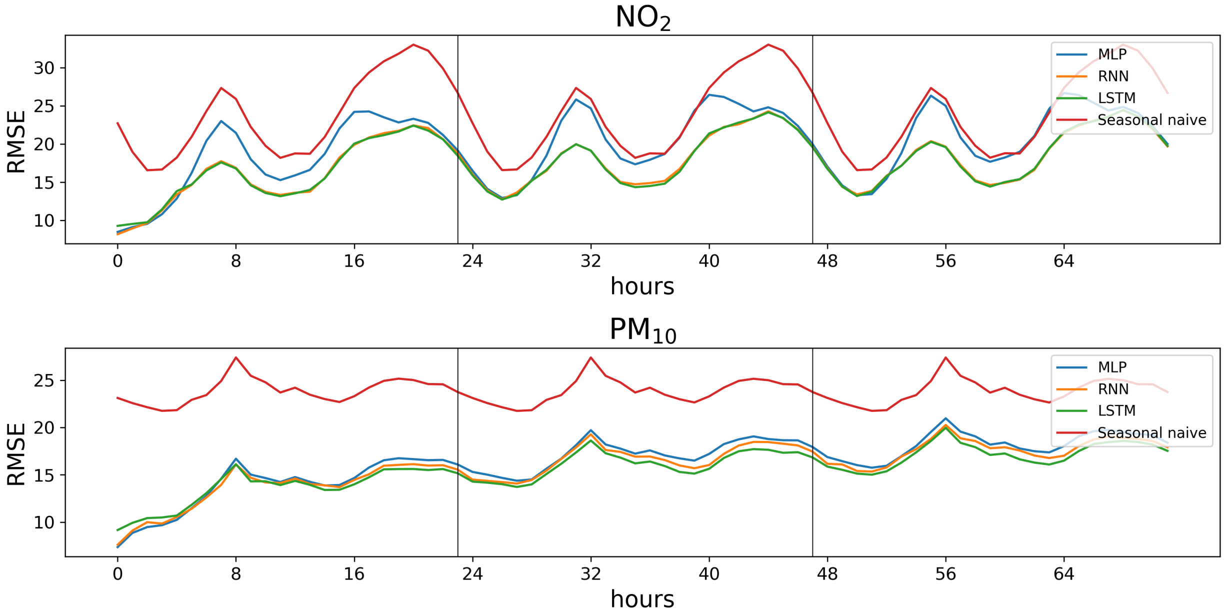


Figure 6.1: Figure shows RMSE for PM$_{10}$ and NO$_2$ predictions of test set. On x-axis are hours of 72-step forecast. On y-axis is RMSE.

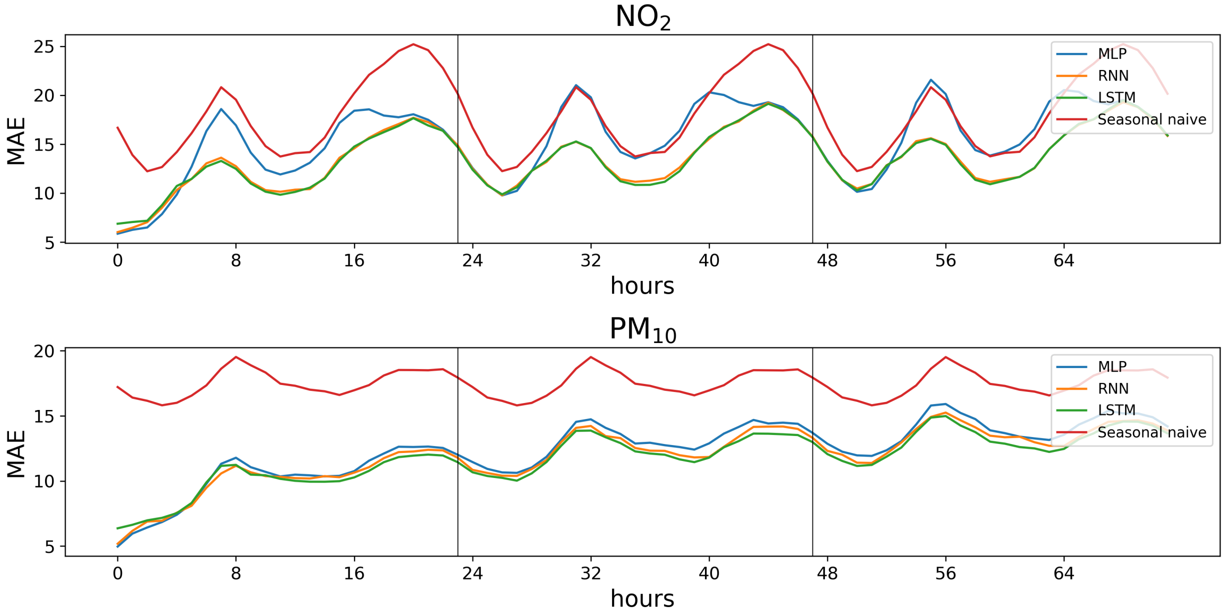


Figure 6.2: Figure shows MAE for PM$_{10}$ and NO$_2$ predictions of test set. On x-axis are hours of 72-step forecast. On y-axis is MAE.

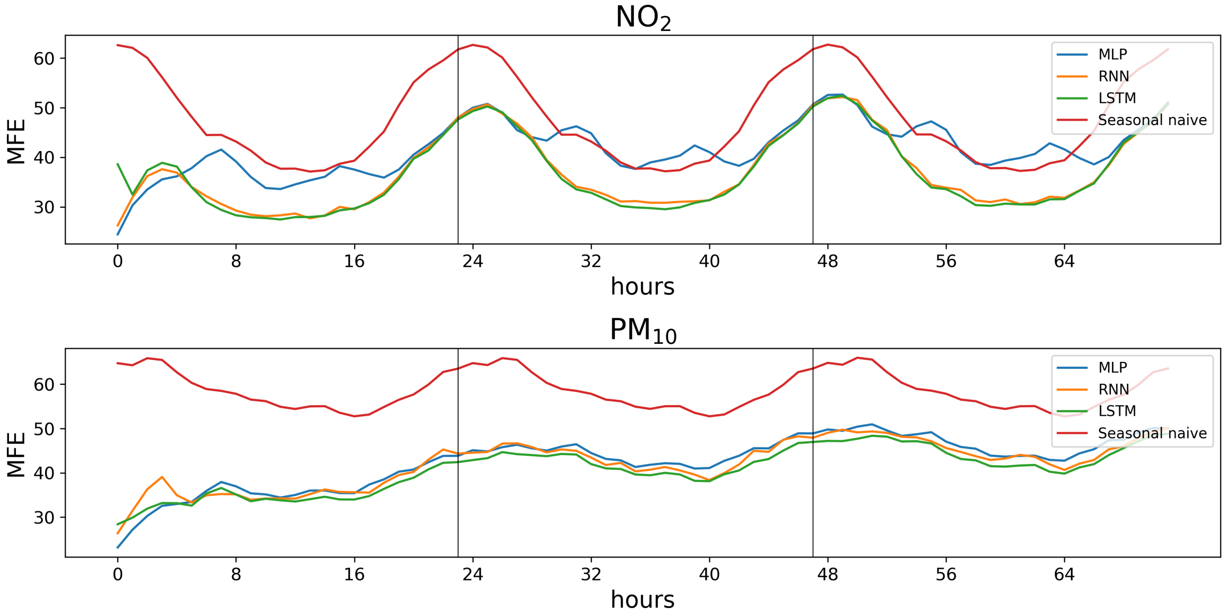


Figure 6.3: Figure shows MFE for PM_{10} and NO_2 predictions of test set. On x-axis are hours of 72-step forecast. On y-axis is MFE.

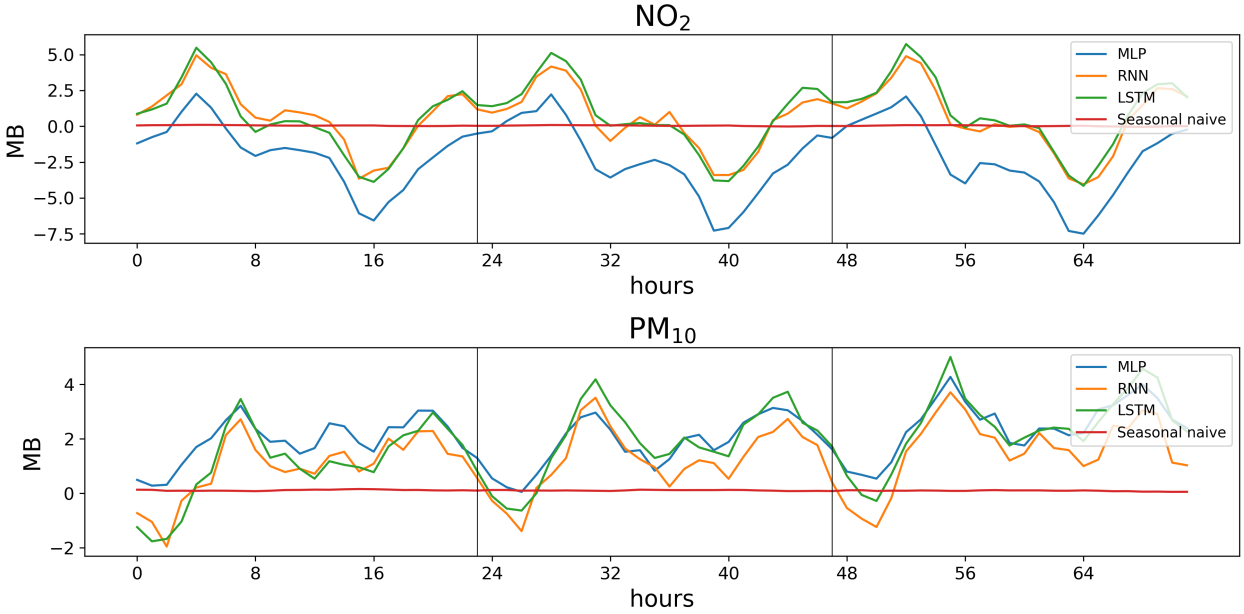


Figure 6.4: Figure shows MB for PM_{10} and NO_2 predictions of test set. On x-axis are hours of 72-step forecast. On y-axis is MB.

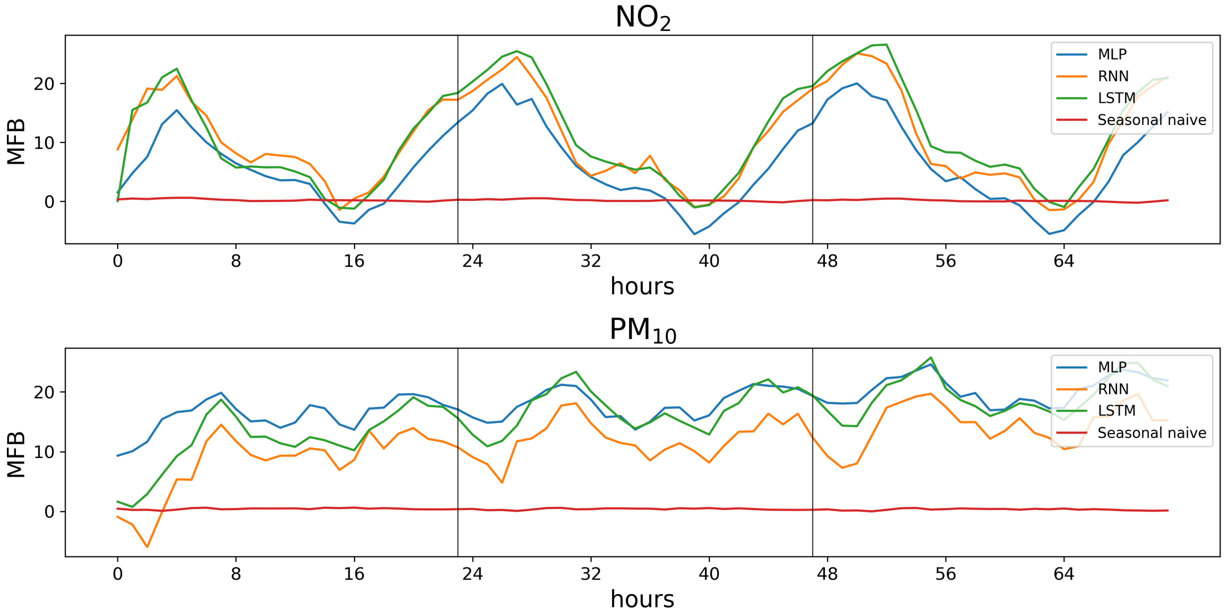


Figure 6.5: Figure shows MFB for PM$_{10}$ and NO$_2$ predictions of test set. On x-axis are hours of 72-step forecast. On y-axis is MFB.

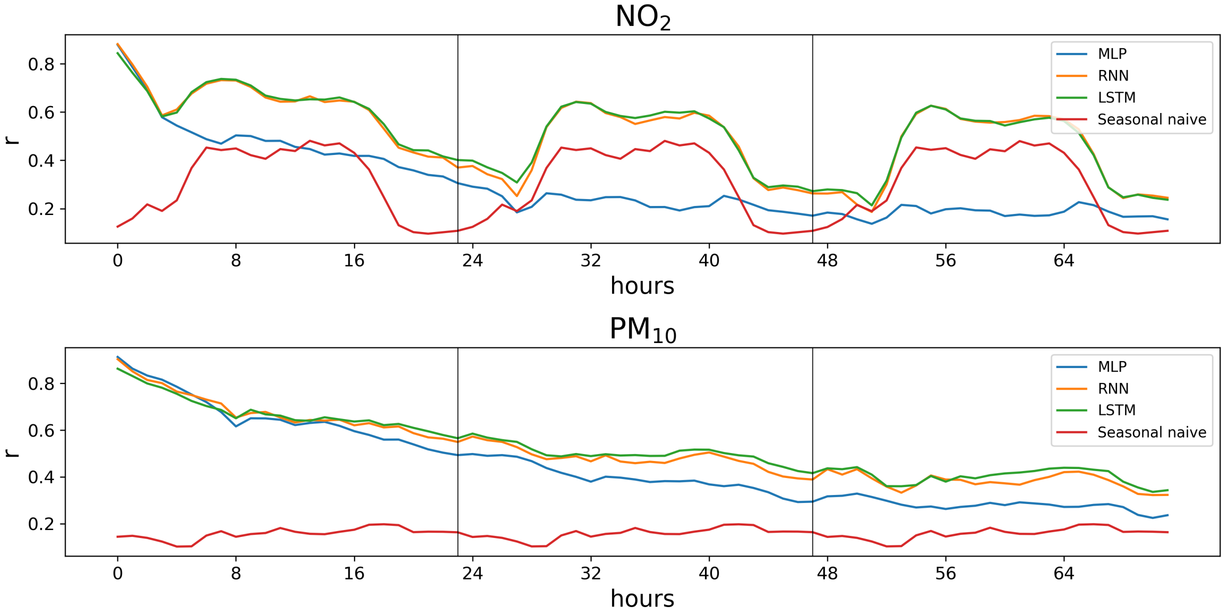


Figure 6.6: Figure shows correlation coefficient r for PM$_{10}$ and NO$_2$ predictions of test set. On x-axis are hours of 72-step forecast. On y-axis is correlation coefficient r.

Conclusion and Future Work

Future Work

In the future, it would be interesting to compare other advanced timeseries statistical methods such as Autoregressive integrated moving average (ARIMA) or Vector autoregression (VAR) with machine learning models presented in the thesis. Future studies should also target on different formulation of problems. We propose that further research should be undertaken in the following direction:

- SHMU official website informs the public about smog situations which are declared when the mean of last 12 hourly measurements crossed the threshold of concentrations dangerous to human health. This way the output of models can be formulated as binary classification (smog situation/ not smog situation) rather than to forecast the expected concentrations.
- Another way to transform regression to classification is to use Air Quality Index (AQI) categories; models could be trained to output one of six categories (very good/ good/ medium/ poor/ very poor) for a given pollutant.
- In this work we predicted the concentrations of the pollutants based on the previous measured data only. In the future, it will be interesting to include the predicted meteorological values obtained from physical models. For this purpose the SHMUs model Aladin provides multivariate forecasts of the weather. We hope, that future work will include data obtained from model Aladin and compare the performance with models presented in the thesis. There is also possibility to use these models to forecast the air pollution on the daily basis.

Conclusion

In this thesis, we decided to structure the work similarly to the CRISP-DM process. We found out that most of the air quality measurements stations at Slovakia exhibit big drop-out gaps leaving datasets with many missing values. The gaps at Bratislava-Trnavké Mýto were not as severe. Nevertheless, we had to perform imputation using the K-Nearest Neighbor algorithm. Many stations are also not suitable for modeling with meteorological variables due to the long distance between meteorological and air quality measurements stations.

We decided to divide the data understanding phase into the classical descriptive statistics part, which enables us to understand and summarize features, and time series analysis part.

Time series analysis focused mainly on exploring the strength of seasonal component using STL decomposition algorithm. We found out that annual seasons tend to exhibit the biggest PM_{10} seasonal strength and almost none daily and weekly. On the other hand, NO_2 shows moderate strength for all (annual/ weekly/ daily) seasonalities.

To prepare the dataset for modeling we described techniques like standard score normalization and range normalization used in KNN-imputation and neural networks, respectively. We also used the transformation of circular features (hours/ weekday/ month/ windspeed), which preserves the nature of such features.

We theoretically described selected methods for modeling and compared the performance of each method. The machine learning models outperformed simple statistical method however, neural network results behaved likewise. The only significant difference occurred in predicting NO_2, where RNNs (vanilla RNN/ LSTM) gave better results than MLP in some parts of the forecast. Otherwise, we surprisingly found minimal differences between vanilla RNN networks and LSTM.

Bibliography

[1] S Asian, C Yozgatligil, C Iyigaun, İ Batmaz, M Tiirkes, and H Tatli. Comparison of missing value imputation methods for turkish monthly total precipitation data. *Middle East Technical University Ankara*, 2014.

[2] James W Boylan and Armistead G Russell. Pm and light extinction model performance metrics, goals, and criteria for three-dimensional air quality models. *Atmospheric environment*, 40(26):4946–4959, 2006.

[3] Peter J Brockwell, Richard A Davis, and Matthew V Calder. *Introduction to time series and forecasting*, volume 2. Springer, 2002.

[4] Robert B Cleveland, William S Cleveland, Jean E McRae, and Irma Terpenning. Stl: A seasonal-trend decomposition. *Journal of official statistics*, 6(1):3–73, 1990.

[5] Copernicus. European air quality — copernicus atmosphere monitoring service. `www.regional.atmosphere.copernicus.eu/`. (Accessed on 02/13/2019).

[6] Giorgio Corani. Air quality prediction in milan: feed-forward neural networks, pruned neural networks and lazy learning. *Ecological Modelling*, 185(2-4):513–529, 2005.

[7] Euronews. Copernicus Air quality index — euronews. `www.euronews.com/weather/copernicus-air-quality-index`. (Accessed on 02/06/2019).

[8] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice, 2nd edition.* OTexts, 2018.

[9] Marilena Kampa and Elias Castanas. Human health effects of air pollution. *Environmental pollution*, 151(2):362–367, 2008.

[10] John D Kelleher, Brian Mac Namee, and Aoife D'Arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.* MIT Press, 2015.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[13] Weizhen Lu, Wenjian Wang, Andrew YT Leung, Siu-Ming Lo, Richard KK Yuen, Zongben Xu, and Huiyuan Fan. Air pollutant parameter forecasting using support vector machines. In *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, volume 1, pages 630–635. IEEE, 2002.

[14] NSW. Air pollution: an overview - air quality. `www.health.nsw.gov.au/environment/air/Pages/air-pollution-overview.aspx`. (Accessed on 02/14/2019).

[15] Filip Pavlove. Bp/src at master · pavlovator/bp · github. `https://github.com/pavlovator/BP/tree/master/src`. (Accessed on 05/28/2019).

[16] Irfan Pratama, Adhistya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. A review of missing values handling methods on time-series data. In *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–6. IEEE, 2016.

[17] NH Savage, P Agnew, LS Davis, C Ordóñez, R Thorpe, CE Johnson, FM O'Connor, and M Dalvi. Air quality modelling using the met office unified model (aqum os24-26): model description and initial evaluation. *Geoscientific Model Development*, 6(2):353–372, 2013.

[18] B Shylaja and R Saravana Kumar. Traditional versus modern missing data handling techniques: An overview. *International Journal of Pure and Applied Mathematics*, 118(14):77–84, 2018.

[19] Wikipedia. Air pollution — Wikipedia, the free encyclopedia. `en.wikipedia.org/w/index.php?title=Air\%20pollution&oldid=881901314`, 2019. [Online; accessed 07-February-2019].

[20] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer, 2000.