Master's Thesis

Master in Telecommunication Engineering

Forecasting the pollution levels in urban environements

Marcin Malak

Supervisor: Antonio Espinosa Morales

Department

Escola d'Enginyeria
Universitat Autònoma de Barcelona (UAB)

July 2019

El sotasignant, *Nom del Professor*, Professor de l'Escola Tècnica Superior d'Enginyeria (ETSE) de la Universitat Autònoma de Barcelona (UAB),

CERTIFICA:

Que el projecte presentat en aquesta memòria de Treball Final de Master ha estat realitzat sota la seva direcció per l'alumne *Nom de l'Alumne*.

I, perquè consti a tots els efectes, signa el present certificat.

Bellaterra, *data_de_sol.licitud_de_lectura*.

Signatura: *Nom del director del projecte*

Resum:

L'activitat de recerca descrita en aquest treball està centrada en descriure una aproximació al problema de la predicció de la pol·lució en àreas urbanes. En general, aplicat a la concentració incremental de elements a les àrees urbanes. En aquest treball s'analitza la utilització de models deep learning en aquest tipus de problemes de predicció. Es presenta un estudi del problema i un model d'estimació d'un conjunt d'elements presents a l'aire urbà basat en un estudi dels seus coeficients de correlació. Finalment, es presenta una aplicació dels models Recurrent Neural Network (RNN) amb estructures específiques como Long Short Term Memory (LSTM) per analitzar aquest problema.

Resumen:

La actividad de investigación descrita en este trabajo está centrada en describir una aproximación al problema de la predicción de la polución en áreas urbanas. En general, aplicado a la concentración incremental de elementos en las áreas urbanas. En este trabajo se analiza la utilización de modelos deep learning en este tipo de problemas de predicción. Se presenta un estudio del problema y un modelo de estimación de un conjunto de elementos presentes en el aire urbano basado en un estudio de sus coeficientes de correlación. Finalmente, se presenta una aplicación de los modelos Recurrente Neural Network (RNN) con estructuras específicas como Long Short Term Memory (LSTM) para analizar este problema.

Summary:

The research activity described in this paper is focused on describing an approach to the problem of the prediction of pollution in urban areas. In general, applied to the incremental concentration of elements in urban areas. In this paper we analyze the use of deep learning models in this type of prediction problems. A study of the problem and a model of estimation of a set of elements present in urban air based on a study of their correlation coefficients is presented. Finally, an application of the Recurrent Neural Network (RNN) models with specific structures such as Long Short Term Memory (LSTM) is presented to analyze this problem.

Contents

Chapter 1

Abstract

The research activity in this paperwork concern about efficient approach in the problem of forecasting air pollution in urban areas. The problem of increasing air pollution around the word become a great challenge to face with. In general, problem refers to continuously increasing pollutants in urban areas, which mostly affects people living there. Traditional approach into the problem require lots of computing power. To address this issue in this paper we tried to suggest and discover usefulness of deep learning in such a problem. Moreover we want to discover different type of useful structures which might be helpful to discover estimation problem. Thus correlation coefficient is also taking into account in such a difficulty. Furthermore Reccurent Neural Network (RNN) with special structures knows as Long Short Term Memory (LSTM) is proposed to capture this problem. In addition if results would not be accurate enough, we would look forward for another interesting approach.

Chapter 2

Introduction

Currently air pollution is one of the most known problem around the world. It affects every single living being. It chiefly affect people who are living in an urban areas, especially inside big agglomerations. Up to the last several years the environment that surrounds us is becoming more and more polluted due to the human impact. Humanity produces inconceivable amount of pollutants that can not be well controlled anymore. Especially big urban agglomerations stand in front of difficulties related with such a problem. With a concern about citizens commenced on search how to properly detect pollution in the environment. More and more companies start research how to predict impact of pollution in urban areas. European public safety laws rule that the pollution levels in urban areas must not exceed certain average levels. Thus they must apply sort of pollution mitigation actions. This situation is disruptive since they affect to the life of the inhabitants of certain areas and furthermore the output of the economy of the cities. Interestingly it is still an open problem since researchers need to find out solutions to problems such as:

- possibility to measure the pollution levels in a few specific spots of urban areas
- possibility to quantize that the real impact of every pollution mitigation action at the reference stations nor in nearby places
- possibility to measure and collect data for every pollution created by humanity

Everything described above have an influence to rapidly growing interest of this topic. Nonetheless this topic is still open and not fully discovered. That influenced me to start my research on that issue. Thus the main problem that I would try to answer during this work might be described as - is it possible to obtain accurate forecasting at specific

spots for longer period of hours ahead. As a period of hours in this scenario we would considering 1 up to 6 hours ahead. Nevertheless more other elements have an influence to the estimation, e.g lack of well prepared data used for algorithms, which also have an impact to validation of models. Moreover during the work we will take a look at correlation coefficient inside our data which might have a huge influence on this problem, because of highly occur interrelationship between different type of pollutants. These observed relationships has been present in nature for many last years, especially after rapidly growing cars manufacturers industry which also have a huge impact on quickly increase pollution level in the cities. That is the main reason, which push us to have a look, and understand it more clearly. We assume that looking for highly correlated data which we planning to use in our algorithm, might be a best solution in the way of using machine learning, which usually is looking for corresponding paths, visible for machine. During the work it would be a part of our proposal in the way of achieve best possible result for our implementation.

While working on the project we assume to, at the end, based on correlation coefficient between our pollutants type, create and checked best possibly solution for this kind of topic using proposed model (RNN + LSTM) which is useful to predicting pollutants levels few hours ahead. In view of high correspondence with past in algorithm structure. The validation of previously mentioned solution will be checked on dataset from New York Queens neighborhood. This choice is forced because of previously mentioned lack of data structures available for everyone. Since this problem start to become more and more popular, a lot of industries start collecting data which are not available to everyone. Thus produce problems for technical researchers, which in most situation do not have the possibility to collect data on their own. Essentially because of amount of money need to be spent for accurate devices whose have a option to collect data. For the reason that it is popular from several last years, we are also struggling with not enough data, collected in the purpose of deep learning, whose popularity also increase rapidly in the last couple of years.

During the work we also found out that data correspond to traffic data might be very useful. The idea was to work with additional traffic data included inside our dataset structure as a newly corresponded part. We assumed that it might increase accuracy of algorithm in the way of high correlated dependencies between created pollutants on account of cars. Unfortunately we struggle with a lack of data from our chosen neigh-

bourhood, so we decide to use available data from different city, just in way to check if accuracy increase. Our results show different type of errors. Based on assumptions we made, we were considering type of test, which include whole structure of data set, and smaller parts, including traffic data. Table 2.1 is describing all best results achieve for each type of pollutant forecasting 6 hours ahead. It is important, that none of this table is specifically described based on determined type of configuration used to achieve particular results.

METRICS	CO	NO2	SO2	PM 10	OZONE	PRESS
RMSE	0.0863	7.4705	0.6635	26.347	0.0101	12.75
MASE	0.0375	3.438	0.2693	15.29	0.0048	6.745
MAE	0.0597	5.4759	0.429	24.35	0.0077	10.74

Table 2.1: Best pollutants forecasting results

The problem might be still developed to achieve continuously better results. This highly complex problem is not easy to solve. In the way of present everything that was mentioned previously, we will try to fully described it with more details in the next parts of this work.

Chapter 3

Definition of the problem

3.1 Validation of data

At the beginning it is necessary to describe the problem of forecasting with more details. This problem is more complex to solve from practical and theoretical site. Thus it is require to divide it for few smaller specified parts which refers to different aspects. Most important this project aim to investigate is it possible to forecast weather properly in few hours ahead based on the open source data, possible to collect by human in XXI century. In the case of forecasting we need to face with the problem of gain valuable and validate data to work on. As a validate and valuable data in this particular scenario we understand data continuously collect from last several years. The more year of provided data we gain, the better data set will be. Most important mentioned before is word "continuously" in the way of collecting data, which means that data set should not have missing parts, with lack of information. At this moment there is a huge problem with that, because of availability and validation of data we are capable to obtain. We would like to obtain as many years of data from chosen area as it is possible in the way of feed them inside our deep learning network. That is one of the biggest problem we faced during the work. There is no a lot of data available for everyone this days. Even if they are, it usually ends up that data are:

- not properly checked
- not sort or not unified in one particular way
- not fully described

This kind of data are meaningless in the way of use them in our project. On the other hand even well provided, and described data are facing with other problems, such as lack of accuracy in the way of continuity of data. Missing data are usually call "holes". This holes are generating incredible problem in the way of validation, because in simple words, algorithms have no possibility to handle them. Thus, researchers are force to play with data, to make them more accurate for the algorithms. There are several possibilities for that, e.g

- interpolate data
- cut missing holes
- change a structure of data
- delete missing parts.

In order to take care of problems, described above, during this work we used almost every mentioned solution. Moreover different type of solution were based on different part of work, because of distinct usage of data.

Every algorithm force us to provide data in the structure expected from it. Due to this first of all it is necessary to achieve well prospected data from specific area. So basically at the very beginning it was mandatory to contact, and make a research with companies, websites and government industries which collect this kind of data.

Diagram 3.1 is representing process of preparing the data, including research as a very first step. During the work, we were forced to ask ourselves few simple questions, in the middle of doing that we were able to make a significant decision. We were considering that with a problem of forecasting there is a high possibility that a different types of pollution and correlation between them could work well in the way of estimation. We were looking forward to find out most known type of occurring pollutants in the atmosphere, especially above the big city agglomeration. One of the most significant aspect is also a fact, that every kind of data should be collected from the same type of sensors, which increase the probability of same scale, and same valence. To face the preparation of data we used pandas library which include tools, allows us to manipulate data, to submit them in the necessary way.

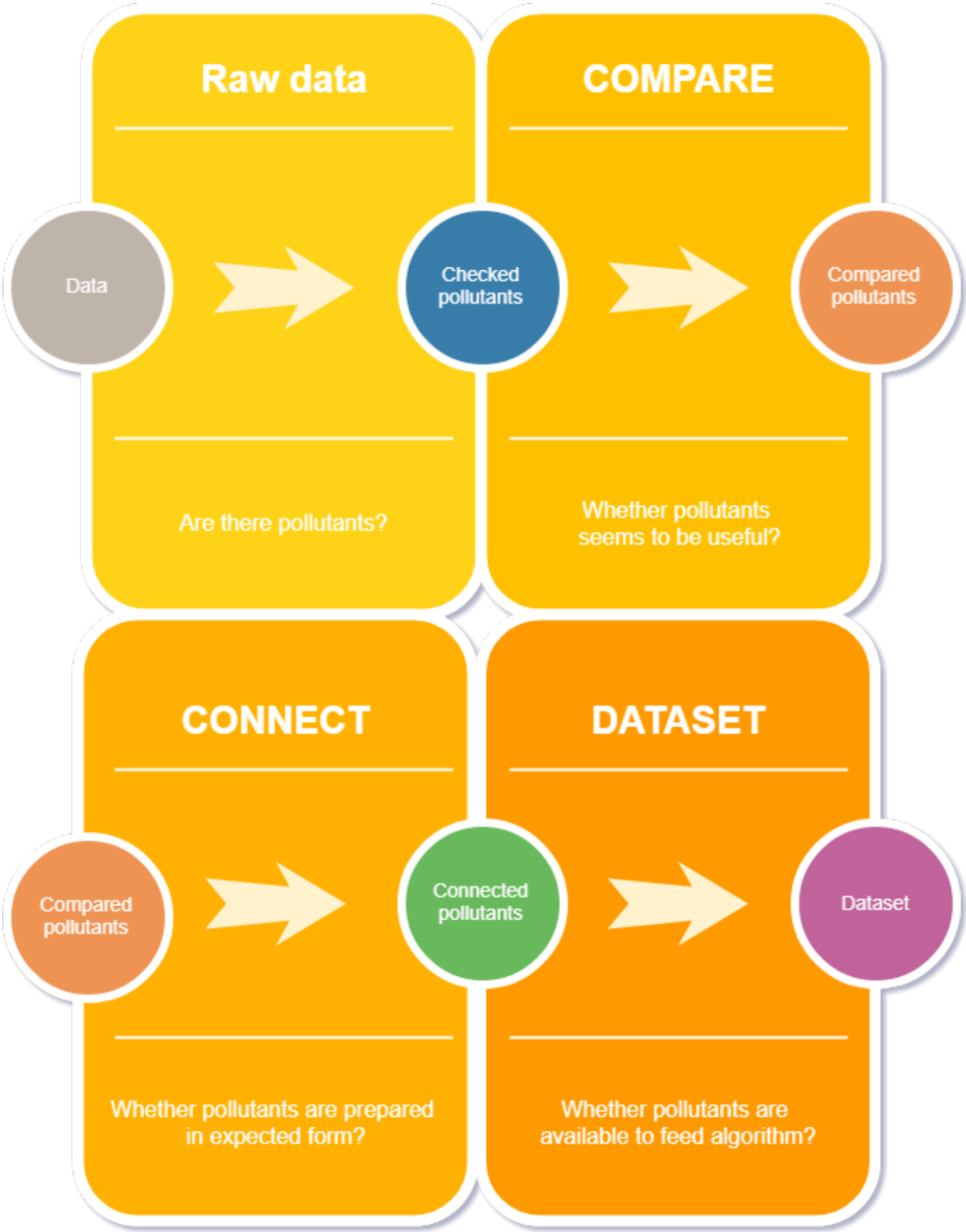

Figure 3.1 Representation of collected data

However all the data were provided separately, which affect that, as it was mentioned before on figure 3.1,- it was necessary to collect them together. The figure 3.2 present representation of collected and prepared data using Pandas Library [7]. It distinguishing part which is corresponding with date time and each of the pollutants level for each unit of time.

	CO	NO2	OZONE	PM10	PRESS	SO2	TEMP
2013-01-01 00:00:00	0.249	21.0	0.019	3.3	1008.0	1.9	40.0
2013-01-01 01:00:00	0.251	20.9	0.020	11.3	1008.0	2.3	40.0
2013-01-01 02:00:00	0.240	20.7	0.020	9.3	1008.0	2.5	40.0
2013-01-01 03:00:00	0.208	15.9	0.024	15.9	1008.0	2.3	40.0
2013-01-01 04:00:00	0.208	15.4	0.025	15.8	1008.0	2.6	40.0

Figure 3.2 Representation of collected data

Sadly there is no provided specification or in-depth information about sensors from which data were received. There is several neighbourhoods where sensors are working, but there is an information that only one agency are collecting all pollutants we were considering. Based on other assumption in the project, we decided that area of New York city is going to be choose, because as one of the very few, is collecting all data we were interested in. Moreover to avoid incompatibility of data, it was decided to decrease area of interest, and choose only Queens neighborhood. It should allow to accomplish best possible results. We assume, that if we take into account bigger area, we might achieve a unsuitable influence on our previous assumptions. Moreover this area contain highest number of different type of collected pollutants from same type of sensors, from last several years. Gathering data went very smoothly, they are provided as an open source data, which means that they are available for everyone who wants to look at them or work on them. It's mandatory to mentioned that at the very end we would like to create data set with different type of data frames, easy to manipulate and change. This structure has to be homogeneous. To achieve previously mentioned compatibility we used pandas library which is one of the most powerful tools to work with data structures. It allow users to manipulate and prepare data in the expected form. Workflow with data such as time series is specific, it requires a compatibility between provided data forms.

```
dataset = pd.HDFStore('/home/marcinmalak/Desktop/Master Thesis/datasets/fin.h5')
```

```
dataset.keys()
```

```
['/New York/Steuben/3',
 '/New York/Queens/124',
 '/New York/Queens/125',
 '/New York/New York/135',
 '/New York/Monroe/1007',
 '/New York/Monroe/15',
 '/New York/Erie/23',
 '/New York/Erie/5',
 '/New York/Bronx/133',
 '/New York/Albany/12']
```

Figure 3.3 Representation of HDFS file

Pandas library has a built-in functionality that allows to create a data structure used to work with Large Data (under 100GB), and it's representation allow us to create files which are also allow to work inside Hadoop Distributed File System(HDFS), which also facilitates working with larger data sets, even larger that ones just mentioned . Figure 3.3 perfectly illustrates the applied solution. Because of this, we are allow to manipulate and work with the data structures inside our file separately. It is a big convenience in the way of comfortable work with data structures.

3.2 Algorithms

Moreover to have stronger understanding of what is going next, we should have a brief look for basic concept behind machine learning.

3.2.1 Basic concept of machine learning

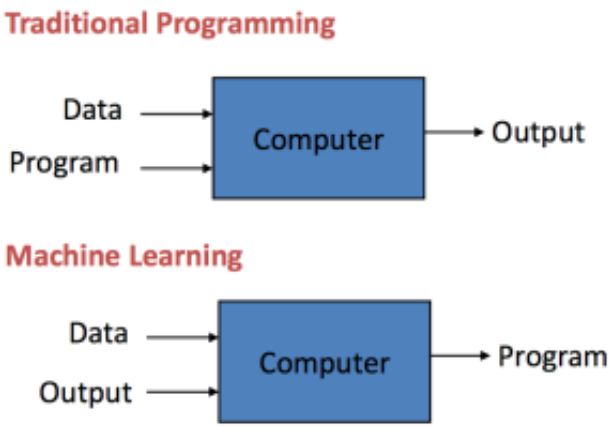

Figure 3.4 Traditional programming vs machine learning

The figure 3.4 [5] is trying to present the representation of the difference between traditional programming and machine learning itself. Essentially machine learning is getting computer to program, and solve the problems themselves. Traditional programming depends on programmers knowledge based on which they are creating a software, whereby are producing an output. This actions are defined by instructions for computer. Comparing this with machine learning we find out completely different way of thinking. In the second scenario, we are using data and outputs create by developers, to produce a program which is able to give as a specific results and calculations.

3.2.2 Workflow under machine learning

The most fundamental thing in machine learning is a fact, that in most cases, at first we are training our machine on special assigned part of dataset, to give an algorithm some understanding of the problem. However there are different approach to machine learning ,such as Supervised and Unsupervised learning. Each carries a wide range of knowledge behind them.

Supervised learning

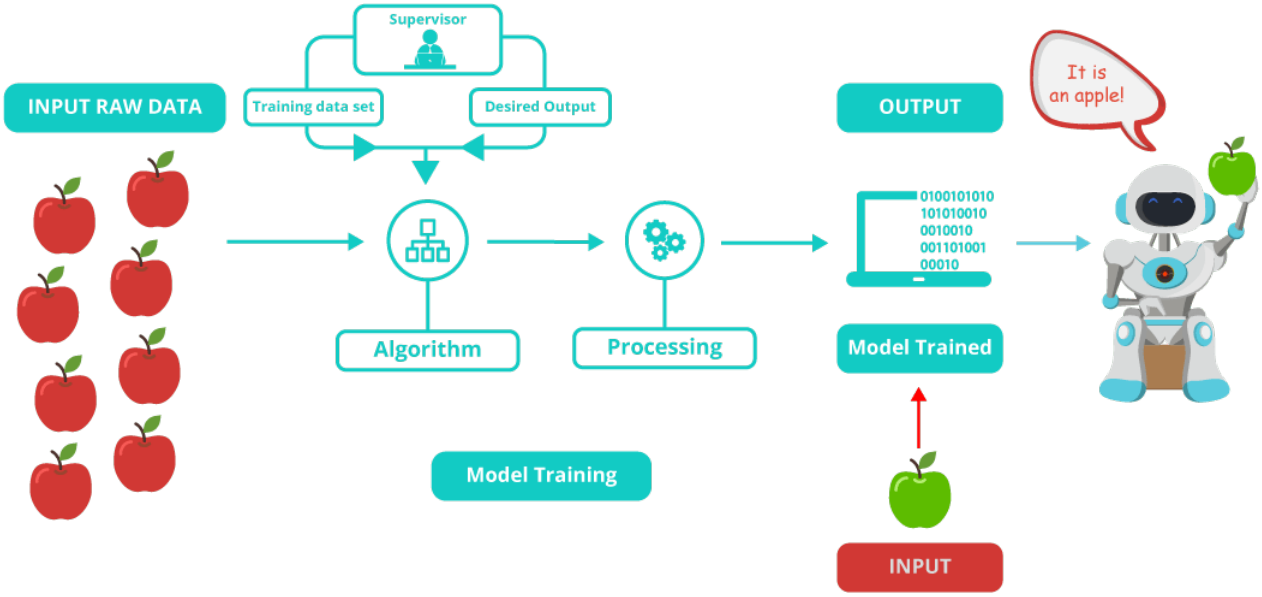

Figure 3.5 Supervised learning

As a first we approximate the functionality of supervised learning, which is present on figure 3.5. It is the one, where we have input variables and output variables. It using algorithm to learn how to predict our input variables, and give us a result as a output variable. The main goal in this particularly scenario is to achieve well suit prediction based on previously training dataset, which actually acting like a teacher teaching his students [5]. There are many different type of inputs that we may assume e.g:

- audio frequency
- pixels of image
- values of database
- biometric data

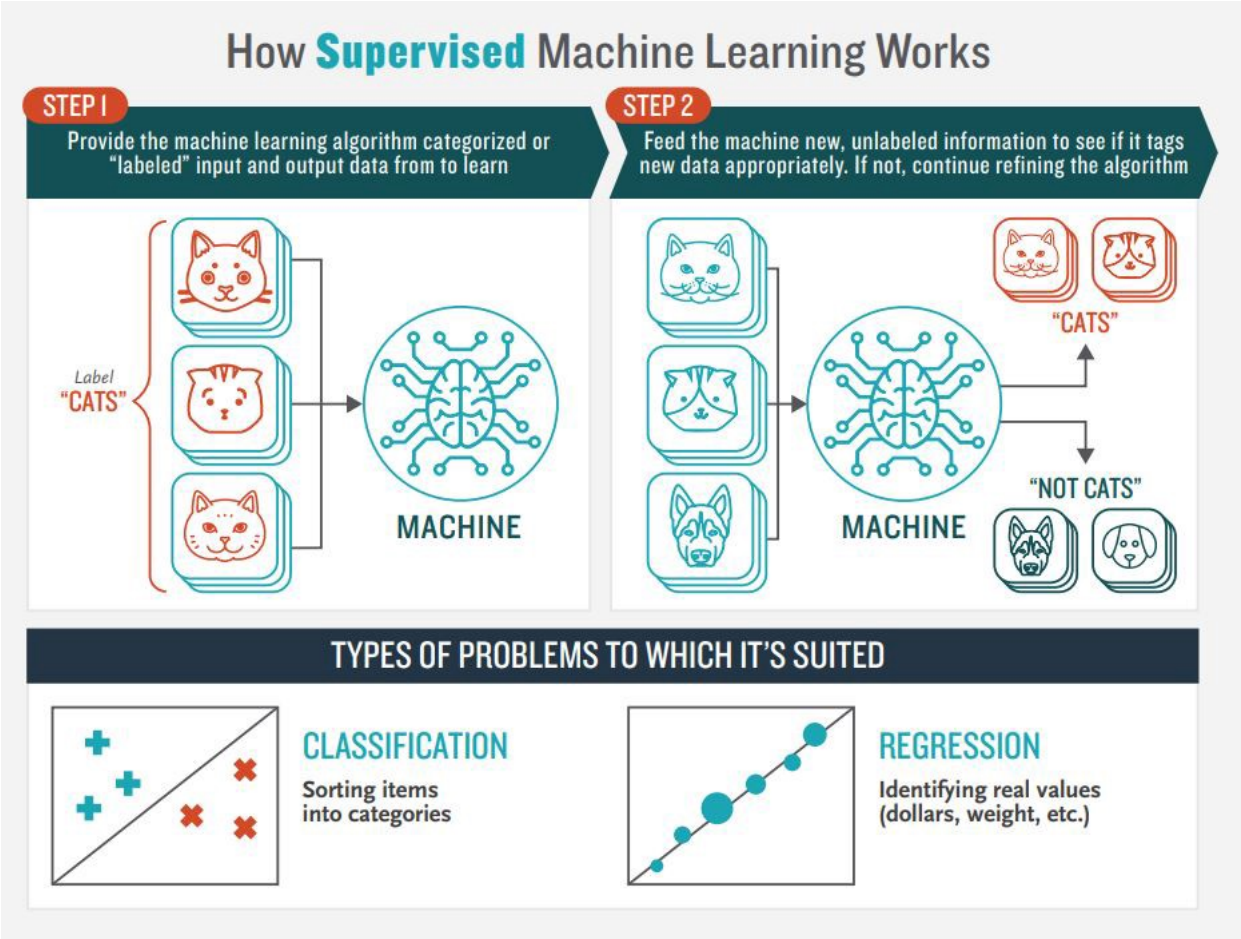

Figure 3.6 Labels in Supervised learning

Since it is supervised learning it is important, that without labels, our algorithm might not work as expect. Lack of labels provided in the training process, affect to misunderstanding of input data, and so on the figure 3.6 [?] is representing steps needed to take ,in the way of working in expected route. In the first step in this special example, collected pictures of cat are "labeled", then based on this, machine is able to recognize cats. In the next step, when we would like to check if our algorithm behaves in expected way, we have to feed him with different type of pictures, which including e.g both cats, and dogs. Based on previously labeled data supervised learning algorithm is able to predict which input represent cats, and which not. It is also recommended to mentioned that this kind of algorithms, are usually face with the problems of classification and regression.

Since not all the problems might be discovered only using supervised learning there is another possibility which is not using labels in the training process, and this process will be described next.

Unsupervised learning

As a second, we will focus on unsupervised learning, which is present on figure 3.7. In this case we have input data, but no corresponding output variables. In this approach the part of training dataset does not have specify output associated with him. In other words, it is working completely opposite than previously mentioned supervised learning, because it doesn't have any type of labeled data. Machine is identifying and analysing the patterns inside the dataset and learn to take decisions based on observations and learning. It is identifying different type of clusters and corresponding to them relation, to be able to decide how to extract the useful information from provided data. And that is a solution we are going to use in our project.

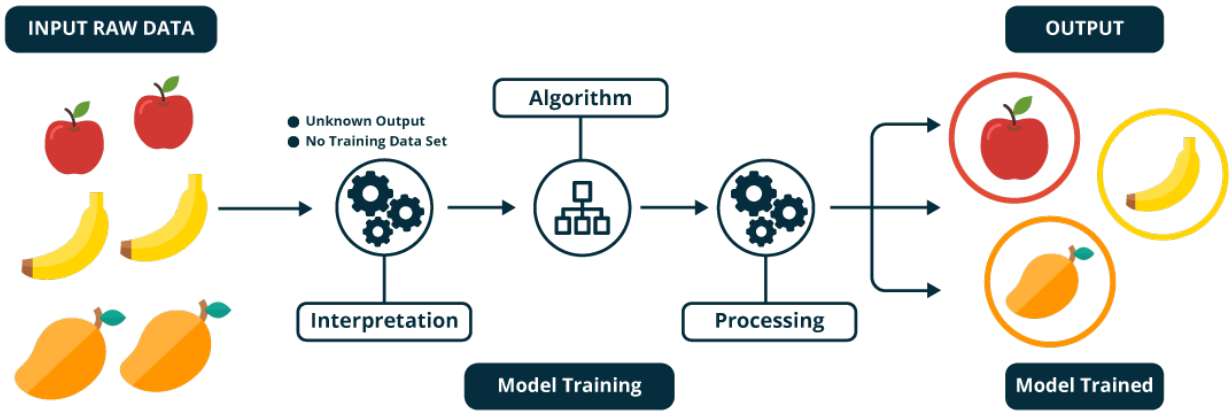

Figure 3.7 Unsupervised learning

After all the information mentioned previously, we assume to validate our algorithm on testing data which should, in the result give a researcher a brief understanding if algorithm is actually heading in the right direction. In case of our problem we considered to divide all possessed data into two main parts, training set and testing set.

During the work we will try to answer what is necessary to achieve best results in this specific problem. Moreover we will try to claim is it meaningful to continuously develop this kind of work. Is the problem likely to continue into the future.

Chapter 4

State of art and methodology

4.1 Previous discoveries

There are different approach to the problem of forecasting. Problem is highly known, and there is a lot of companies, researchers and scientist who are facing this problem. All of the previously mentioned are working with different type of tools, to achieve satisfied results. Xiaosong Zhao and Rui Zhang in their work "A Deep Recurrent Neural Network for Air Quality Classification" [2] are considering to predict Air Quality Classification (AQC) on three different industrial cities in United States, using different models such as Recurrent Neural Network(RNN), Support Vector Machine(SVM), and Random Forest. Based on their comparision it show up that the best possible results were achieved used RNN solution. The important thing is that they also use some similar type of pollutants to work with. The other work belongs to Srinivasa Rao, Dr. G. Lavanya Devi, N. Ramesh called "Air Quality Prediction in Visakhapatnam with LSTM based Recurrent Neural Networks" [3] is also considering use of different real time dataset of the city Visakhapatnam, having a record of 12 pollutants. They also achieve the best possible results using RNN comparing with different models. Based on that we assume that is highly indicated to apply machine learning solution to such a problem of forecasting. To predict feature we need to have some base previous information, through the years. This approach might allow us to learn the patterns, which we use for forecasting. Recurrent neural networks have high inherent ability to learn sequentiality, because of that, they are one of the mostly used solution applied with problems such as forecasting, especially with terms of time series. In one of the academic work called "Recurrent Neural Networks for Time Series Forecasting" Gábor

Petneházi assume that this type of solution is very successful in terms of prediction time-dependent targets. Going through all previously mentioned assumptions and conclusions in different type of research work, we are considering that using RNN might be one of the best choices in an attempt to predict pollutants. Moreover last several years provide us a lot of different type of works with use of RNN, and based on quality and performance achieved there, we assume to use this solution.

4.2 Data representation

In this case it end up that only well provided data with a lot of available different pollution types, were provided by United States Environmental Protection Agency[1]. On their website we can easily check history which said that "this agency was born in the wake of elevated concern about environmental pollution, to consolidate in one agency a variety of federal research, monitoring, standard-setting and enforcement activities to ensure environmental protection. Since its inception, EPA has been working for a cleaner, healthier environment for the American people"[6]. This government agency site released data from several years, contain pollutants such as:

- Carbon Monoxide (CO) - parts per million
- Nitrogen Dioxide (NO2) - parts per billion
- Sulfur Dioxide(SO2) - parts per billion
- Oxygen 3 (O3) - parts per million
- Particulate matter (PM10) - micrograms
- Pressure (P) - millibars
- Temperature (T) - degrees fahrenheit

and also traffic data which include:

- speed
- travel time

In the way of training our algorithm we assume to use scalar function implementation to change all the data on the scale between 0 and 1, which is highly reccommended in the way of forecasting using RNN with LSTM.

4.3 Correlation coefficient

It is important to explain, that the main objective of obtain well prospected, and prepared data is, that deep learning neural networks works more effective with huge amount of data. Based on this it is recommended to consider also different aspects. One of them is correlation coefficients between different pollutants. This different approach is probably capable to increase accuracy of our algorithm, but first we need to realize what kind of information correlation coefficient is providing for us.

In statistic, correlation coefficient is used to measure how strong a relationship is between two variables. The range of values are between -1 and 1. A calculated number greater than 1.0 or less than -1.0 means that there was an error in the correlation measurement. In other hand correlation equal to -1 shows a perfect negative correlation, and on the other hand correlation equal to 1 shows a perfect positive correlation. Which is also important, correlation equal to 0 shows that there is no relationship between the movement of the two variables. There is few representation of correlation coefficient, but in our particular situation we used "Pearson Product-Moment Correlation"(or Pearson correlation coefficient, for short).

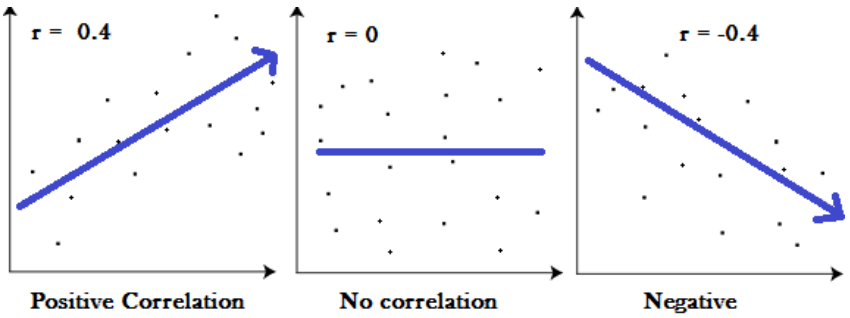

Figure 4.1 Example of correlation coefficient representation

4.4 Plot - graphic

The next tool helpful during the work were matplotlib library, which is mostly used to generate visual representation of data, which in this situation were extremely helpful. This kind of representation is called plot, and in other words that is a graphical technique to symbolize data sets. It is usually a visualization of relationships between two or more samples. It is very useful, especially for humans, who can quickly find out corresponding dependencies between data. This kind of representation might not be easily seen comparing different type of numerical values. There is a various number of uncommon plots, such as:

- line plots
- image plots
- histogram plots
- path plots
- 3-dimensional plots
- scatter plots

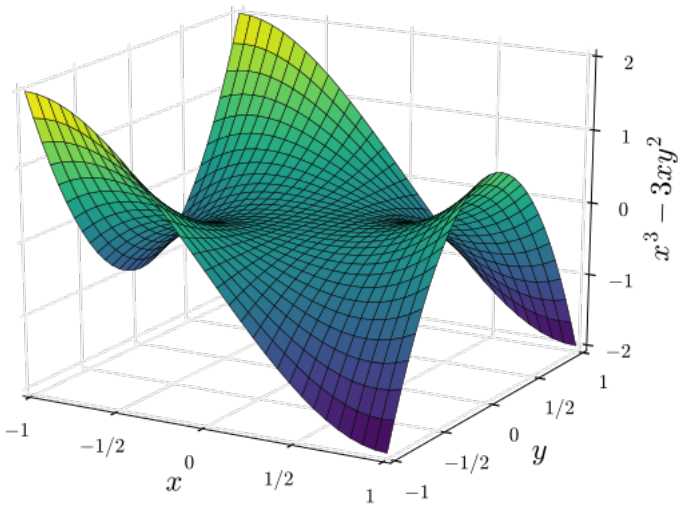

Figure 4.2 3-dimensional plot[9]

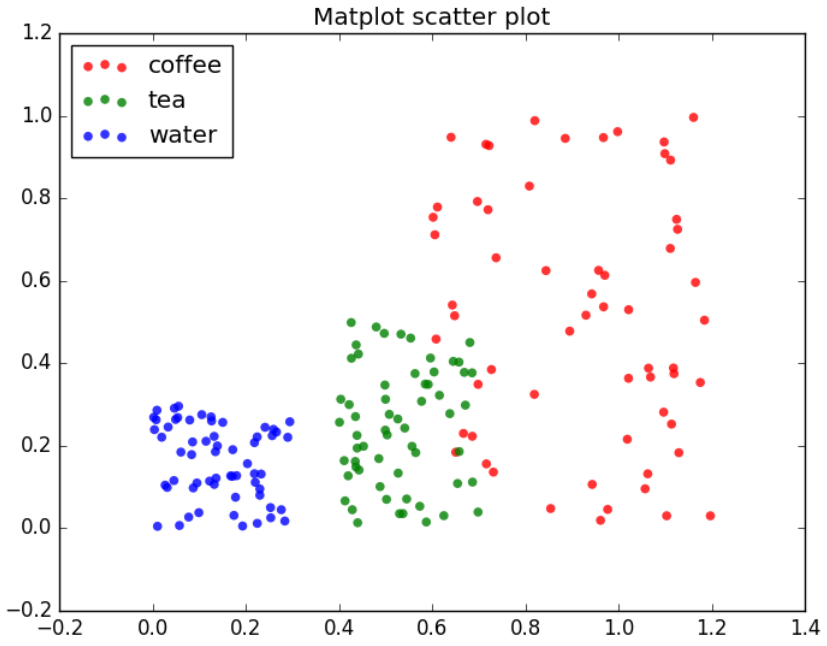

Figure 4.3 scatter plot[11]

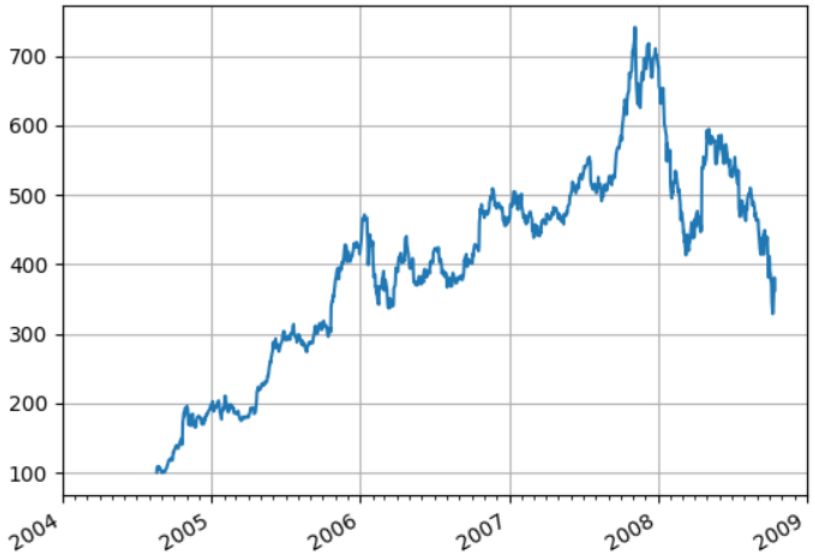

Figure 4.4 timeseries plot[12]

Each of them is used to prepare representation of data sets, used during the work. It is also common, that every plot is working special, with data adapted to him. Above there is a few images - 4.5 - 4.7 representing different type of plots, which might be used, using matplotlib library. It is also known that each problem might be represent using different

type of plots. In our work we are considering that best visual representation might occur using date handling plots, which is show on pictures 4.7. It should also be noted that that this kind of representation, while working simultaneously with numerical data is one of the best possible solution to adequately present the results through time.

Moving forward there are various possibility to compare results of data collected by one specific spot, such as few mentioned above. Most important is fact, that this kind of representation reacts smoothly to changes. Which means that if we plan to add other type of data such as wind speed, season or traffic data during the work, it allow us to easily represent new results, compare to previous one. If there is any possibility that some type of data might improve even more, expected results, this kind of solutions make it easier for researchers to present them.

4.5 Reccursive Neural Network

During the work with problems such as forecasting it is mandatory to choose best possible available solution. After make a research around already prepared works about this topic, it show up that probably the best way to achieve good results is to work with Reccurent Neural Networks. However, it is mandatory to look at these networks more closely. Recurrent neural network is build to deal with problem such as memory. Architecture of this networks are mostly based on loops inside them, allowing information to persist inside.

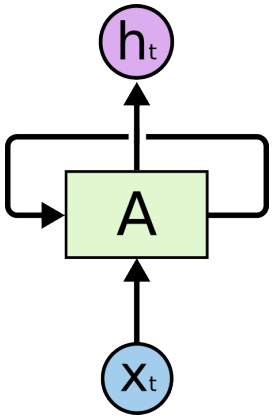

Figure 4.5 Recurrent Neural Network rolled

It is very important that somehow this networks are working same as a normal Neural Networks, with the difference that they look like multiple copies of the same network, each passing the information to the next successor. This is show in the image below.

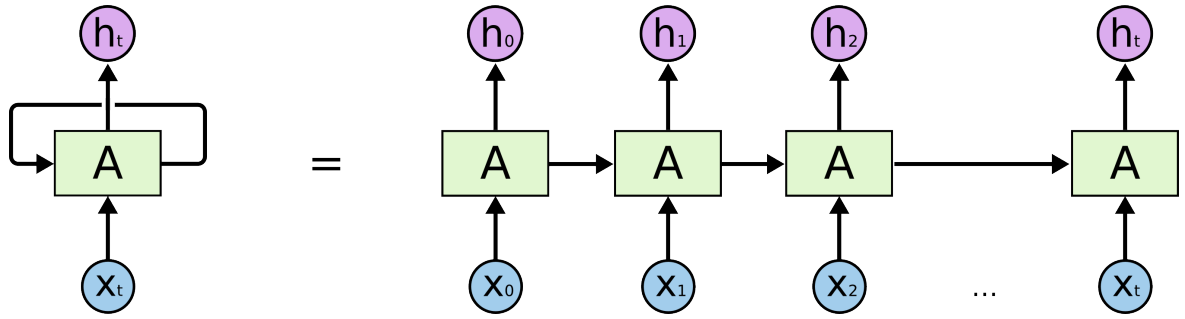

Figure 4.6 Recurrent Neural Network unrolled

The most known appeals of this network is the idea that network would be able to connect and "remember" previous information with present. This solution is not working with all the problems. If we are going to consider a problem such as predicting word, based on previous sentence then standard version of neural network works fine. To make it more clear, below we described RNN way of working on a simple example.

If we take into account that there is a small representation of cooking schedule, then this representation might be present as below, where each day is corresponding to different type of food[13].

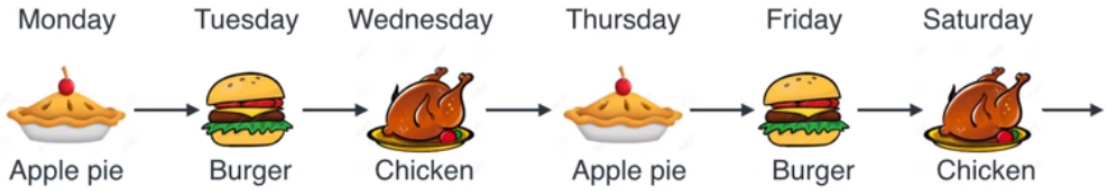

Figure 4.7 Cooking schedule

If we would like to predict based on the schedule, what is going to be our food in the next days, we would like to use RNN.

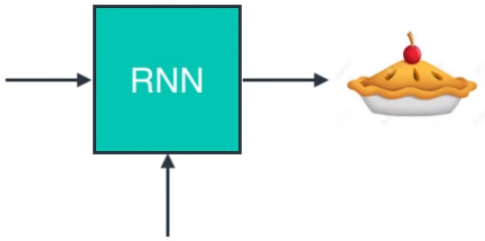

Figure 4.8 Prediction steps

At the beginning in Recursive Neural Network it is very significant that our output is going back and taking as a new input in network. This transition is represent betwen figure 4.11 and figure 4.12.

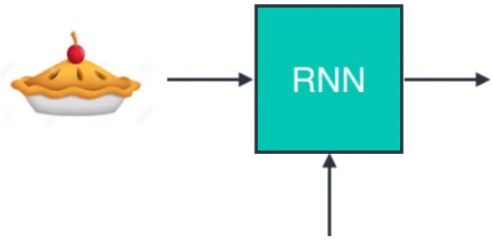

Figure 4.9 Prediction steps

Based on that we can assume, previously checked representation of a week, that we are capable to define next food. So on, next picture has a possibility of previously mentioned prediction future/next food, which in this situation should be a burger represent on figure 4.13.

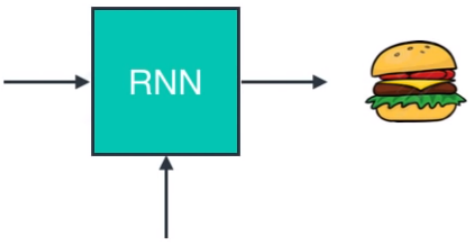

Figure 4.10 Prediction steps

At the end question arises, whether in any case the network will function as well as

in this mentioned situation? In such cases, where the gap between predicting part and used information is small, then RNN can learn to use this past information to give us some good predicting results, but what if we need more context in the way to predict something?

4.5.1 Long Short Term Memory

Thankfully LSTM do not have this problem. Long Short-Term Memory networks are special kind of Recurrent Neural Network, capable to hold and learning long-term dependencies. This solution is designed to avoid any long-term problems. Their default behaviour is to work with such a problems. Standrad RNN is design in order to have simple structure as a single tanh layer.

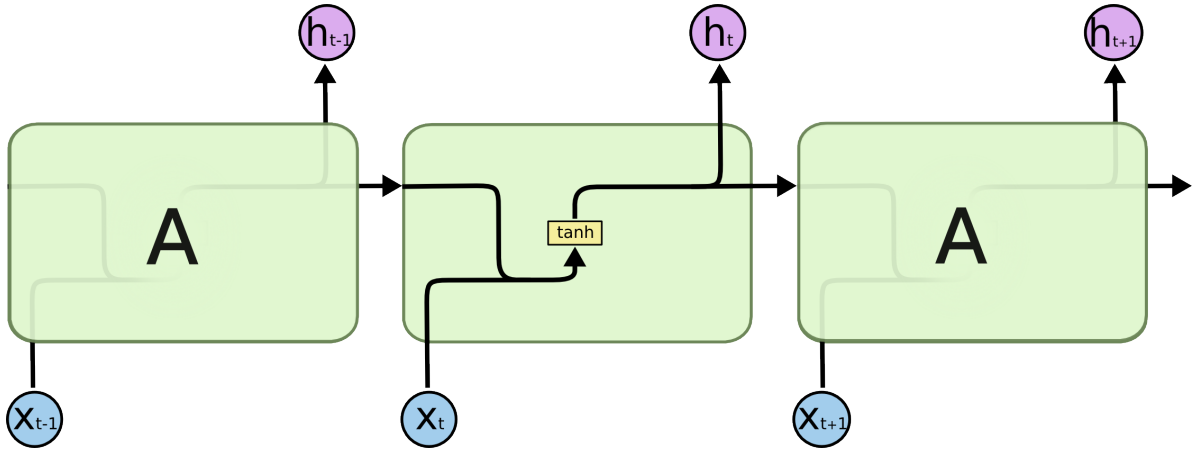

Figure 4.11 Recurrent Neural Network contain single layer

LSTM also have structure as chain but the repeating module structure is quite different. There are four different neural network layer, which are interacting in very special way.

The core idea in the diagram include below (4.15) is that in this situation our network is containing four interacting layers. The main idea behind, is that in this particular network our gates inside are available to "decide" which information are relevant in the way of learning. Input gate is deciding which value is going to be updated based on provided information. In other way it's deciding which information are important (1) and which are not (0). The cell state is getting information multiplied by the forget vector. This forget vector is really important, because it's allow our network to decide, which information are meaningless in the way of further calculations and learning. Then cell

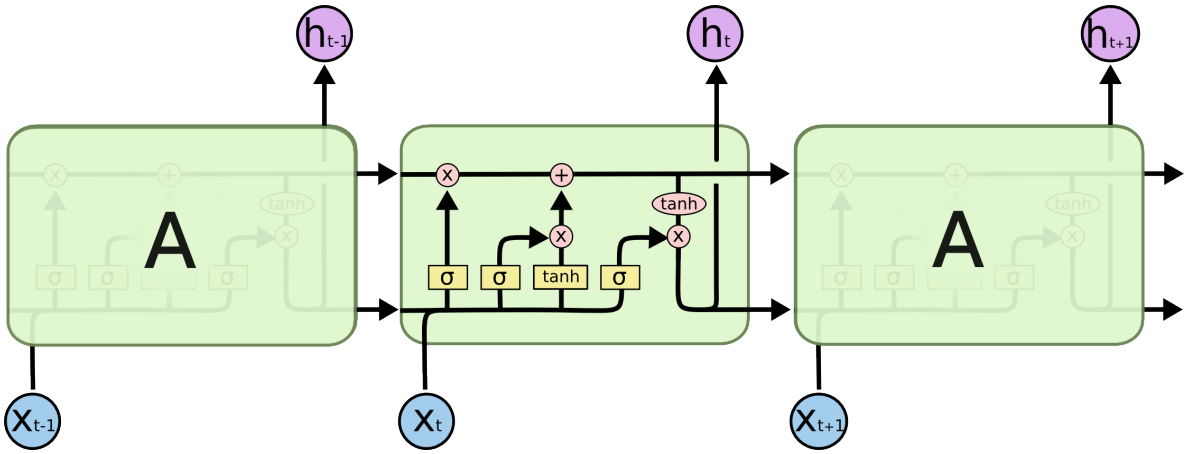

Figure 4.12 Recurrent Neural Network contain four layers

state is being updated to the new values that our network finds relevant.

The last one is Output Gate which decide about next hidden state, which contains information about the previous inputs.

The main concept above described the main functionality of LSTM network. This is just main concept of LSTM described in simple words. There are a lot of different implementation of this work such as:

- Depth Gated RNNs by Yao, et al. (2015)[14]
- Clockwork RNNs by Koutnik, et al. (2014)[15]
- Recurrent nets that time and count Gers and Schmidhuber (2000)[16]

But there is a lot more of them. If we would like to answer the question "Which of these variants is best?" or "Do the differences matter?", then Klauf Greff, in his work "A Search Space Odyssey"[17] do a nice comparison of popular variants, finding that they're all about the same[18].

As mentioned previously based on information already prepared in different research work, it's advisable to use RNN with LSTM, and so on we choose Keras - Python Deep learning library. This high level neural network API is written in Python, and it's available to run on top of such known frameworks like TensorFlow, CNTK, or Theano[19]. Moreover next part of the work is going to describe every step we take in our work to settle previously defined question, that we would like to answer.

4.6 Training and testing

In the way of machine learning it is very important to mentioned our approach to training and testing preparation of data before feeding it into algorithm. There is also interesting assumption that we made. In the way of working with deep learning it is very important to include large value of data to train. However, data should be included from specific part. Different area around the word, even around the cities may show different scale of pollutants, based on geographical location, influence of citizens, or factories which are working around, or inside interesting us city. Due to this we assume, that in the way of creating our model, it is mandatory to focus on specific area, in this scenario - node, around which we want to include our algorithm. Especially in the way of preparing forecasting we would like to achieve well a thriving succession of implementations, which at a later stage could lead to wider development, around different plane.

In context of training with full available data set, we separate 90% of our samples, as a training part, on which our algorithm is actually learning. This kind of data separation usually depends on problem for which algorithms are implemented. The reason that we choose this solution is that our data set is not very large, it could be classified as small. We have a collection of data about the size of 50 000 rows, each including 7 different type of column with pollutants. This division of data, allow us to include information about corresponding data, classified for each month. As an example, our data are collected hourly, so for each average month (30 days) falls around 720 rows of data. Going forward it gives us around 8640 rows of data for each corresponding year.

Therefore, for our testing set falls 10% of our data. It correspond to around 4500 rows of data to our prediction part, which we easily compare with separately prepared data.

Next we need to have a look on preparation and implementation of algorithm. In our RNN LSTM algorithm, we could specify, four most important parameters in the way of learning:

- batch size
- epoch number
- steps per epoch
- sequence length

The first parameter called batch size is determine for us the number of samples in each mini batch. It means that it is defining number of samples from our data set that are used during the learning. It might be 1 or even whole size of data set. The next parameter called epoch number is informing us about the number of times that model is trained over the entire dataset. Steps per epoch determine for us number of batch iterations before a training epoch is considered finished. Our sequence length is corresponding with information of how far in the way of past data our algorithm is taking into account. Algorithm is prepared in the way to avoid overfitting cases, and that's the reason and main functionality of included batch generator, and dropout layers. The first functionality is to create for each training iteration on our data, new part of training set, on which our algorithm is learning. It keep away from situation where we always use same parts of data representation in the way of learning, it might affect, preparation algorithm for specially tailored data. That is what we would like to avoid most. The second one - dropout is a technique where randomly selected neurons are ignored during training. They are "dropped-out" randomly, at each update of the training phase[10].

In the purpose of testing performance and accuracy of our algorithm, we use different type of metric and representation of data, which are described in next chapter.

4.6.1 Metrics

In the whole problem that is being developed, it is very important to include metrics that allow us to determine whether our algorithm is actually heading towards the correct learning.

One of the first that we are taking into account is Root Mean Squared Error (RMSE). It is the standard deviation of the residuals (prediction errors). Mentioned residuals are the measure which inform of how far from regression line data points are. In other words it is informing how concentrated the data is around the line of best fit [20]. Lower values of RMSE is indicating better fit. Moreover this type of error is commonly used with problems of forecasting, climatology, and regression analysis. To calculate RMSE the below equation is needed:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2}$$

Figure 4.13 Root Mean Squared Error formula

Where:

- n: number of samples
- f: forecasts
- o: observed values

The second metric that might be used is Mean Absolute Error(MAE). It is based on absolute error, which in simple words is the amount of error in the measurements we analysing. Our MAE is the average of all absolute errors. The formula is described as:

$$\mathbf{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{x}_i - \boldsymbol{x}|$$

Figure 4.14 Mean Absolute Error formula

Where:

- n = the number of errors,
- $\sum$ = summation symbol (which means "add them all up"),
- $|xi–x|$ = the absolute errors.

This steps described above might be introduce in other way, e.g:

- Find all of your absolute errors, $|xi–x|$.
- Add them all up
- Divide by the number of errors. For example, if there is 10 measurements, divide by 10.

The last type of metrics we are considering to use during this work is Mean Absolute Scaled Error. It is a statistical measure of how accurate our forecast system is. The output of this metric implies that the actual forecast does worse or better, depends if our result is higher than 1 or smaller than 1 . It is the most common metric used to forecast error, and works best if there is no missing value data. The formula representation is described below:

$$MASE = \frac{MAE}{MAE_{in-sample,}}$$

Figure 4.15 Mean Absolute Scaled Error formula

Where MAE is the mean absolute error produced by the actual forecast; while MAE insample is mean absolute error from actual values.

All of the metrics we are considering to use in the way of check accuracy and performance of our model will be present normalized. Since our forecasting is including different type of pollutants, only normalize errors are compatible to present and compare different type of pollutants together.

Chapter 5

Results

In this chapter we are going to show the best occurred results, with different approach inside our implementation. As a first approach it is necessary to check the relevant of data. We decide to use two different way of interpretation data such as correlation plots, and standard plot implementation.

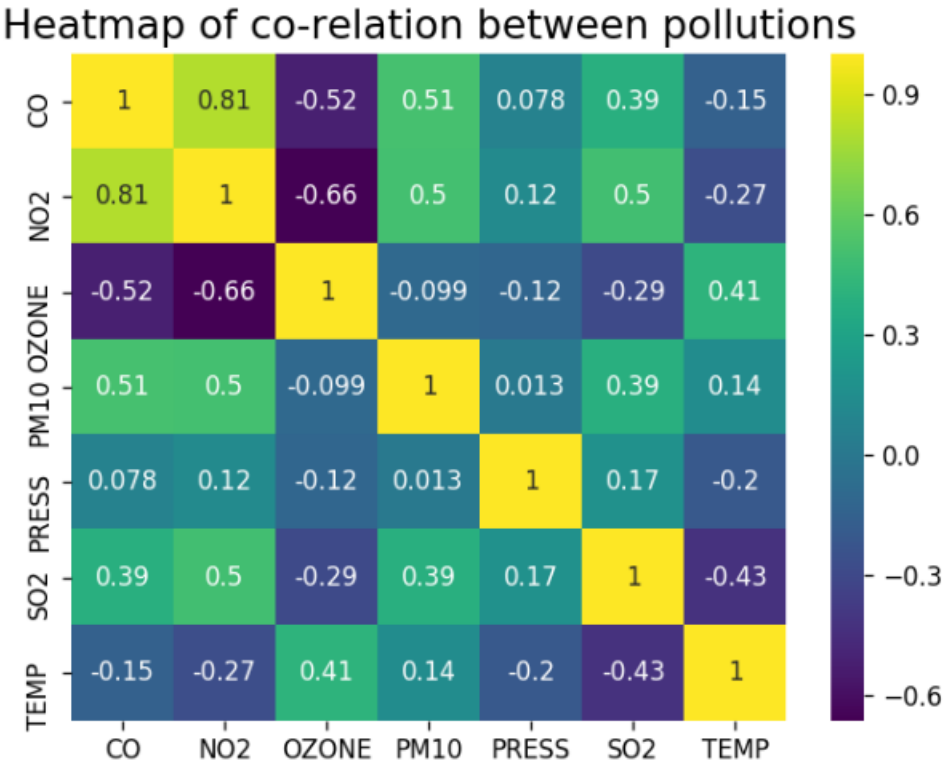

Figure 5.1 Correlation coefficient between different pollutants

As a first representation we took into account during the work is described above. The figure 5.1 is representing correlation coefficient between different pollutants. It give

a graphical representation of correlation between data we are including in our algorithm. It allow to assume that e.g NO2 is highly correlated with CO, and so others. That makes a prove that use of this data inside an algorithm could have an influence for calculations, and final accuracy.

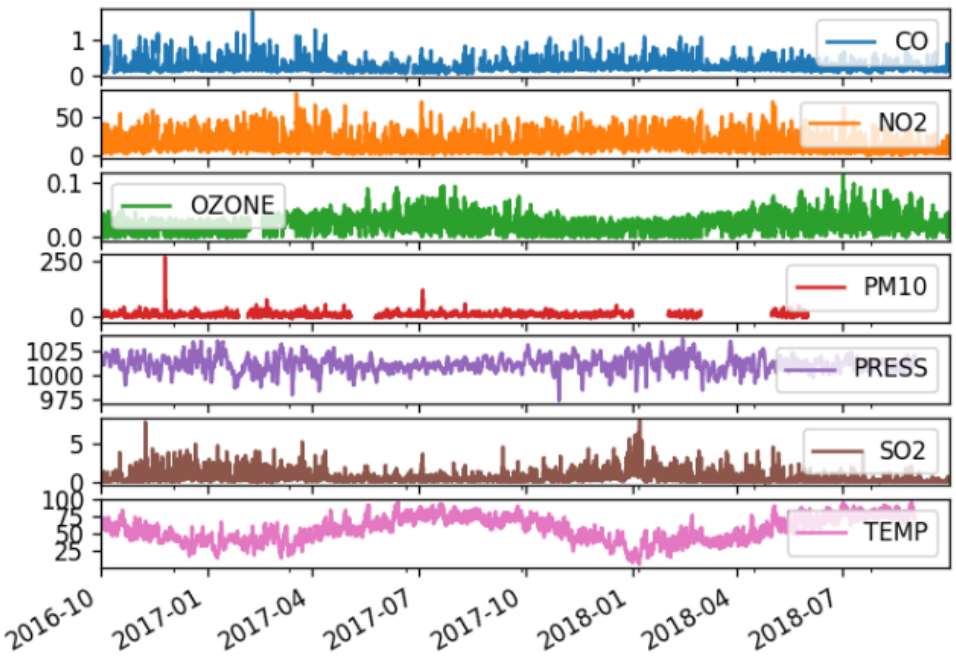

Figure 5.2 Correlation coefficient between different pollutants - plot representation through years

As a confirmation of previously represented co-relation other representation of the plot is used. Figure 5.2 describe our data behaviour through the years. So on, humans could easily deduce that there is a huge compatibility between figure 5.1 and figure 5.2 which represents correlation. Moreover during the analyse of figure 5.2 which represents our pollutants, occurred that there are, previously mentioned missing data (holes) .

As a next point it is mandatory to mentioned, that all the best results, will be presented in separate tables. This solution is taken into account because of different configuration of presented before algorithm. Table 5.1 described below is representing different type of configuration used for tests, in the way of achieve best possible forecast of different pollutants up to 6 hours ahead.

Algorithm configuration			
batch size	sequence length	optimizer	number of epoch
1	1,3,6	adam	20
50	1,3,6	adam	20
100	1,3,6	adam	20
200	1,3,6	adam	20

Table 5.1: Algorithm configuration

It is also required to delve into "sequence length" meaning, once again. This specific data is corresponding with a equation which is used to described number of past weeks, which algorithm is taking into account in way of learning.

Tables below is representing best occurred results for different chosen pollutants. Each of them is representing achieved results based on previously prepared data set, which include data since 2013 up to testing part which ends in the middle of 2018. It is also important to mention that we providing few different tables representing each best results, because of different configuration at which we achieve them. It show up that because of different type of correlation between our pollutants, some of them need different configuration to achieve best results than the others. It also prove that correlation between our data has a huge influence of results. Based on the knowledge about pressure, temperature and ozone, we decide to not include this in our predictions, because of different type of dependence which have bigger influence on results in prediction their level. They are used in the way of check if they actually have an impact on our prediction, if they correlation with other data might improve accuracy of prediction pollutants. Below we are trying to represent predictions occurred in the way of forecasting through different period of time ahead, since 1, 3 and 6 hours ahead. For each of predicted pollutants we are including separately changes during the prediction to show how our algorithm is behave in different configurations. At first we include estimation of CO through changed time and configurations. Then we propose the same way of presenting data through other pollutants which we try to predict, in this scenario it would be NO2, SO2. Based on our MASE metric we would achieve results, which inform us if our implementation of LSTM algorithm is actually performing well.

Pollution predictions					
Normalized Root Mean Squared Error results					
Representation of best results with configuration batch size 1-50 and corresponding sequence length 1,3,6					
Time ahead CO: 6h , 6h, 3h, 6h, 6h Time ahead NO2: 3h , 6h, 6h, 6h, 6h Time ahead SO2: 6h , 3h, 3h, 3h, 6h					
CO	0.047	0.052	0.053	0.062	0.068
NO2	0.062	0.67	0.07	0.073	0.076
SO2	0.138	0.154	0.169	0.182	0.183

Table 5.2: Best results for NRMSE

Pollution predictions					
Normalized Root Mean Squared Error results					
Configuration batch size 1-50, sequence length 1,3,6					
Time ahead CO: 6h , 6h, 3h, 3h, 3h Time ahead NO2: 3h , 6h, 6h, 6h, 6h Time ahead SO2: 6h , 3h, 3h, 3h, 6h					
CO	0.032	0.036	0.037	0.072	0.08
NO2	0.052	0.055	0.062	0.074	0.078
SO2	0.106	0.113	0.125	0.14	0.143

Table 5.3: Best results for NMAE

Pollution predictions					
Mean Absolute Scaled Error results					
Configuration batch size 1-50, sequence length 1,3,6					
Time ahead CO: 3h , 6h, 6h, 3h, 6h Time ahead NO2: 6h , 3h, 3h, 3h, 3h Time ahead SO2: 6h , 6h, 3h, 3h, 3h					
CO	1.06	1.088	1.217	1.263	1.679
NO2	1.183	1.445	2.045	2.291	2.662
SO2	5.03	5.058	5.783	6.002	6.421

Table 5.4: Best results for MASE

But that is not only possible results we were considered to include. There is also a visual representation of estimation, which clearly show the representation of considered estimation. This representation is introduce below, for some best occurred result. All of the figure below correspond to the best results presented earlier

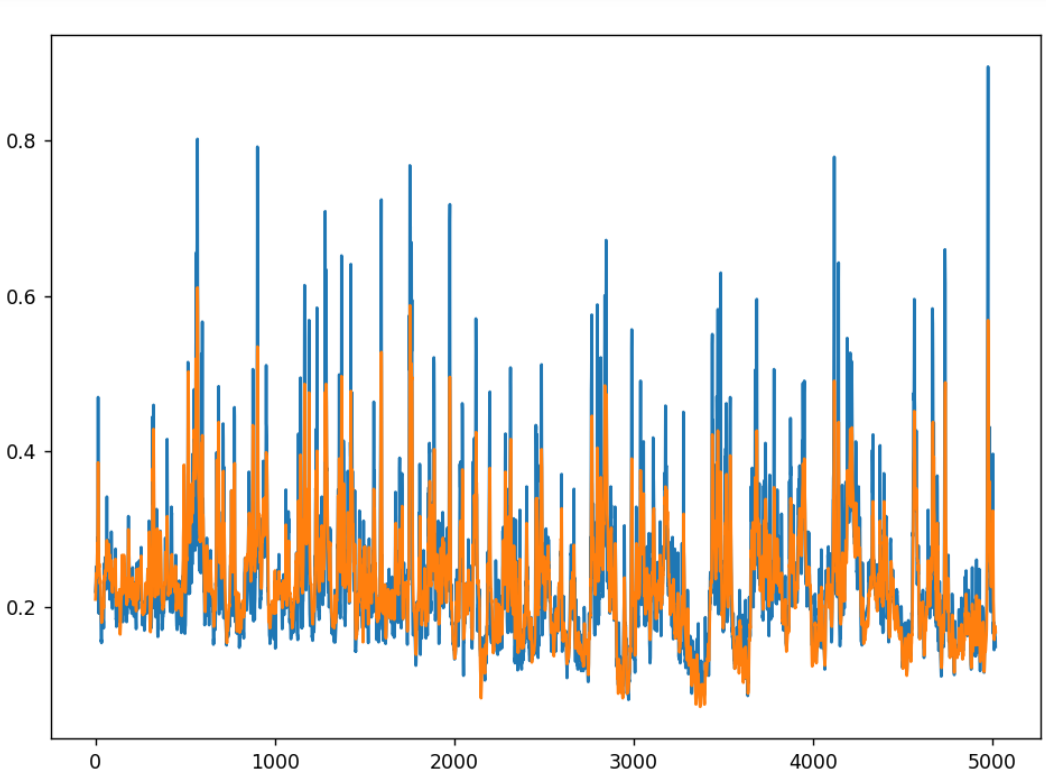

Figure 5.3 Visual representation of CO - best occur result

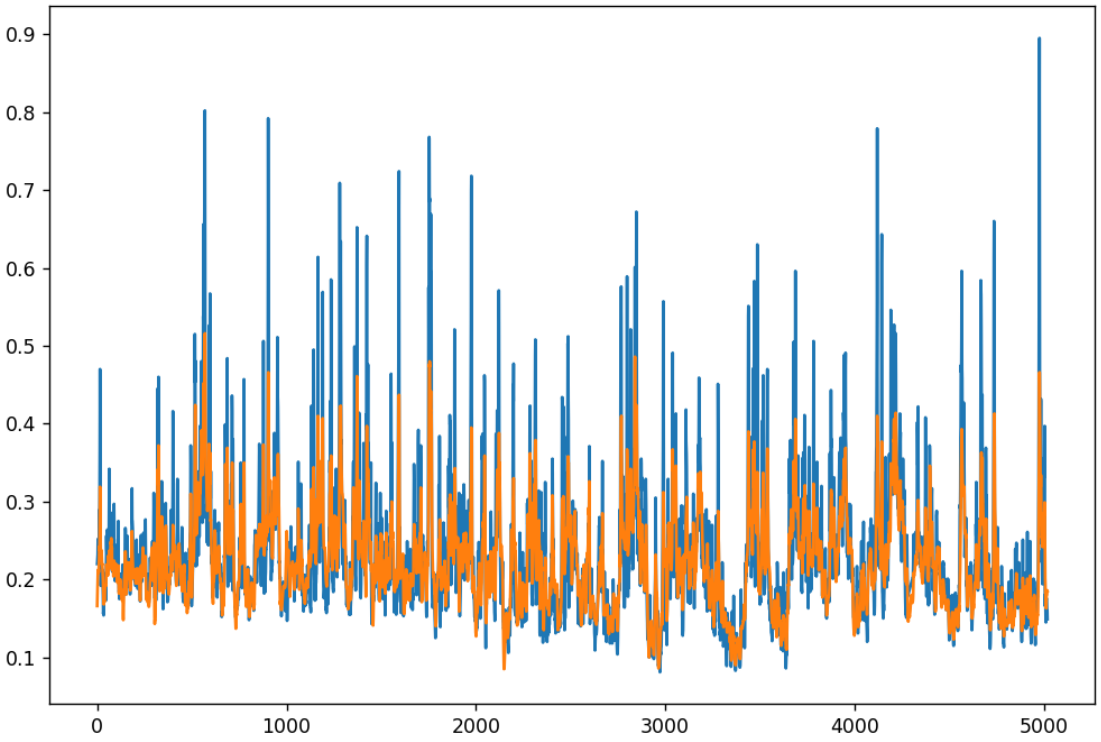

Figure 5.4 Visual representation of SO2

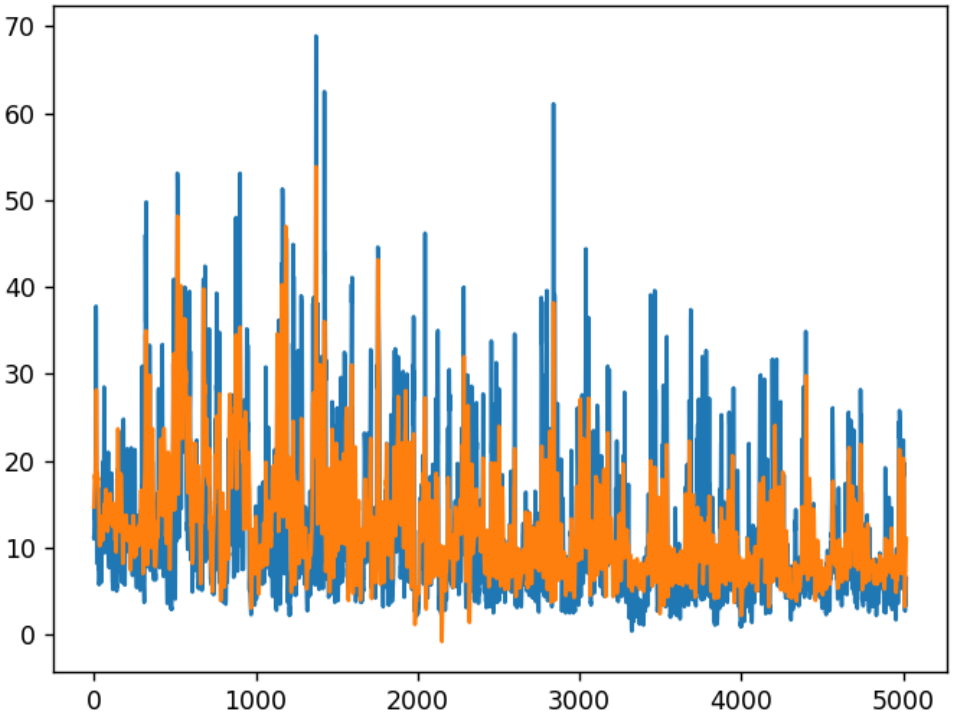

Figure 5.5 Visual representation of NO2

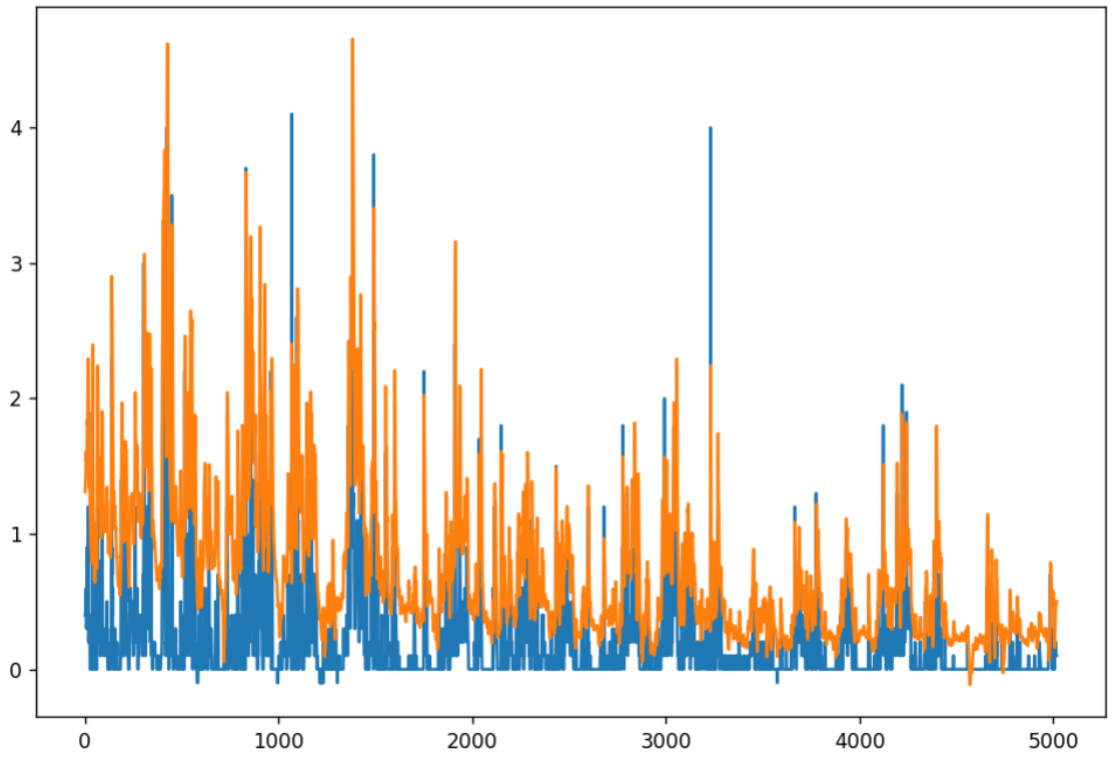

Figure 5.6 Visual representation of weak estimation of SO2

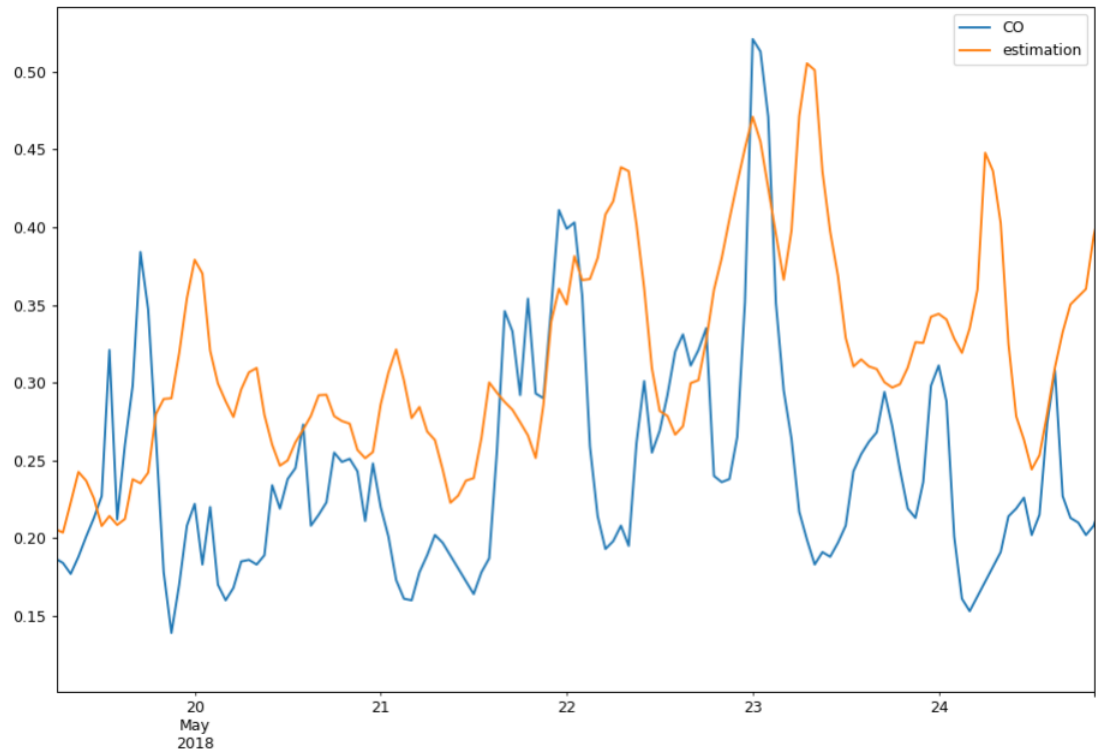

Figure 5.7 Visual representation of CO

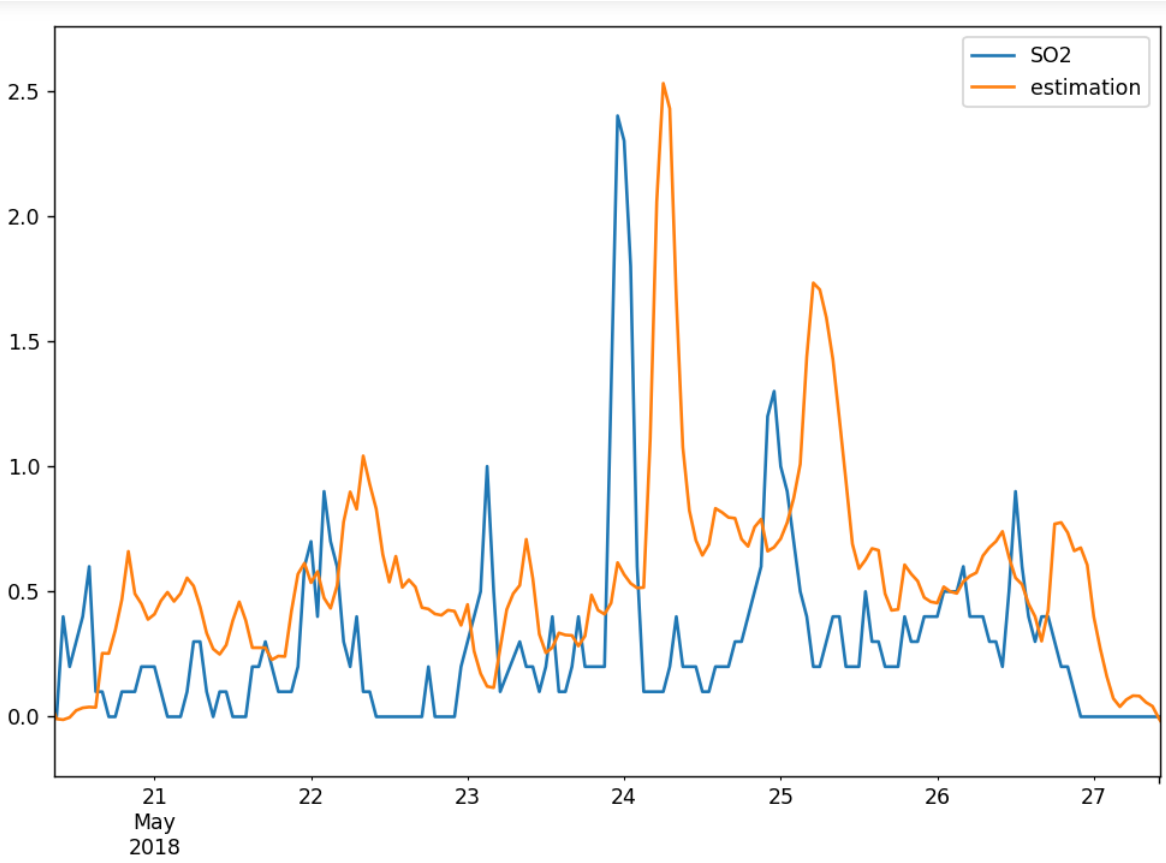

Figure 5.8 Visual representation of SO2

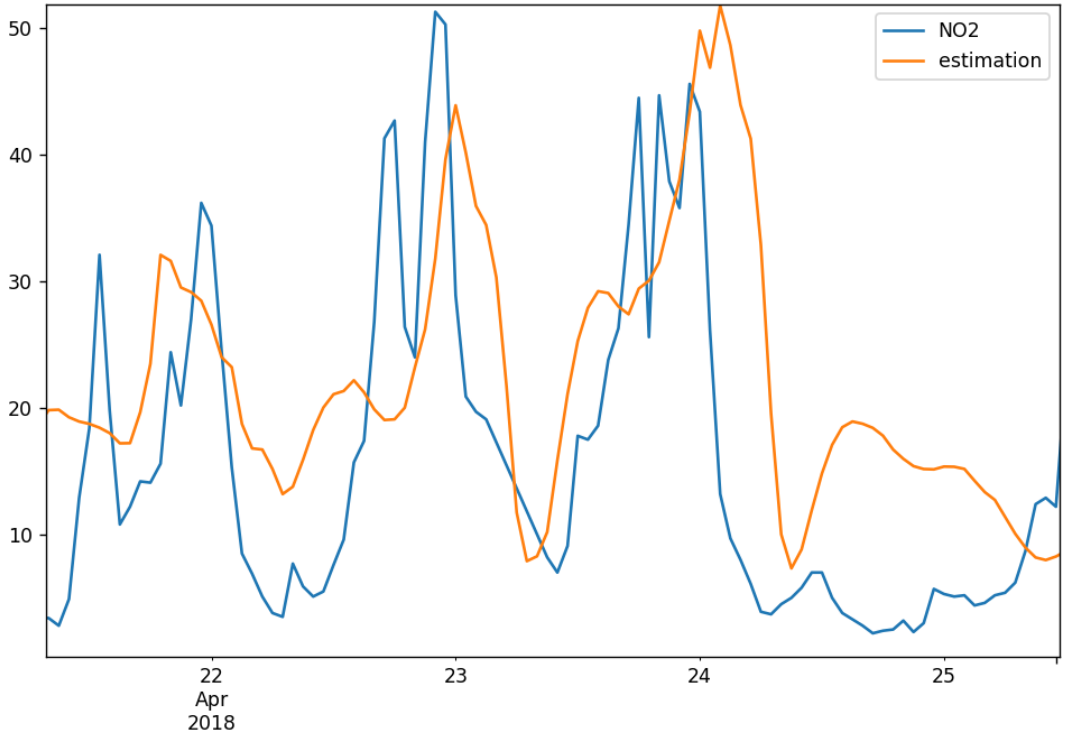

Figure 5.9 Visual representation of NO2

Now we would like to present results which include our dataset with implemented inside traffic data, which we believed that might improve our results. The way of providing this data inside our previously prepared dataset it is carried out in the same way as preparing data set from the beginning. This data are represented as a new added column. Sadly there is no huge amount of data, which correspond to the previously mentioned period of time inside results provided above. Yet, we were taking into consideration that even decrease number of training set, might allow us to check if our assumption is correct. To prove that, we use smaller period of time (1 year) to train our algorithm again, with same configurations as before, then we test it on half-yearly data. Below there are all result we achieved.

Pollution predictions				
Batch size: 1, sequence length: 2, epoch: 20				
METRICS	**CO**	NO2	SO2	**PM 10**
RMSE	**0.1135**	9.344	0.8735	24.737
MAE	**0.0712**	7.341	0.542	22.562

Table 5.5: Best results of CO - small data set

Pollution predictions				
Batch size: 1, sequence length: 2, epoch: 20				
METRICS	CO	**NO2**	SO2	PM 10
RMSE	0.1058	**9.120**	0.8677	25.146
MAE	0.0712	**7.201**	0.551	23.277

Table 5.6: Best results of NO - small data set

Pollution predictions				
Batch size: 1, sequence length: 2, epoch: 20				
METRICS	CO	NO2	**SO2**	PM 10
RMSE	0.1172	9.834	**0.852**	25.615
MAE	0.0733	7.710	**0.528**	23.599

Table 5.7: Best results of SO2 - small data set

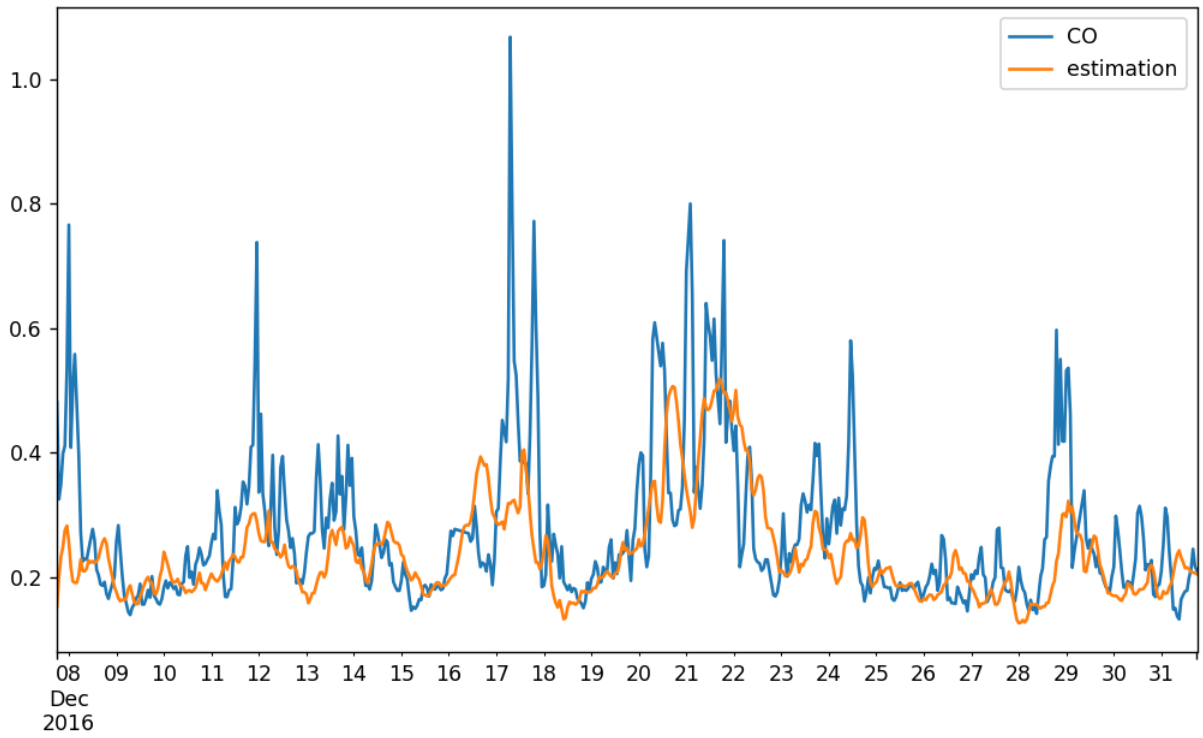

Figure 5.10 Visual representation of CO - small data set

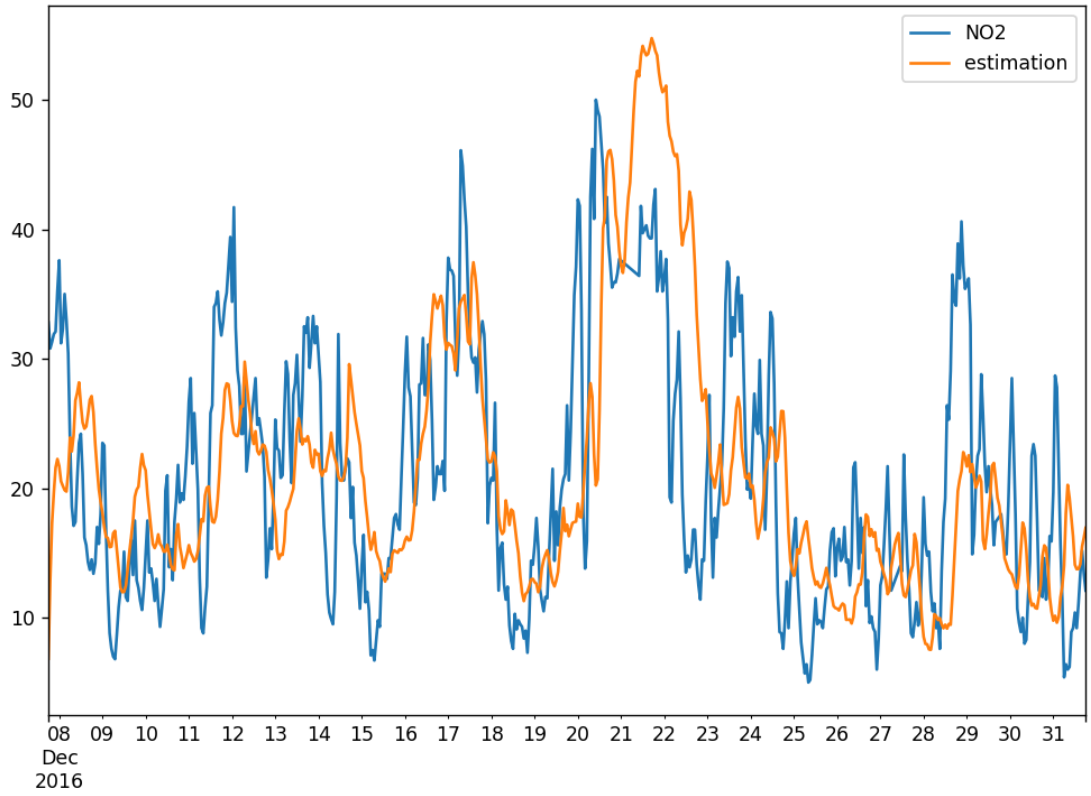

Figure 5.11 Visual representation of NO2 - small data set

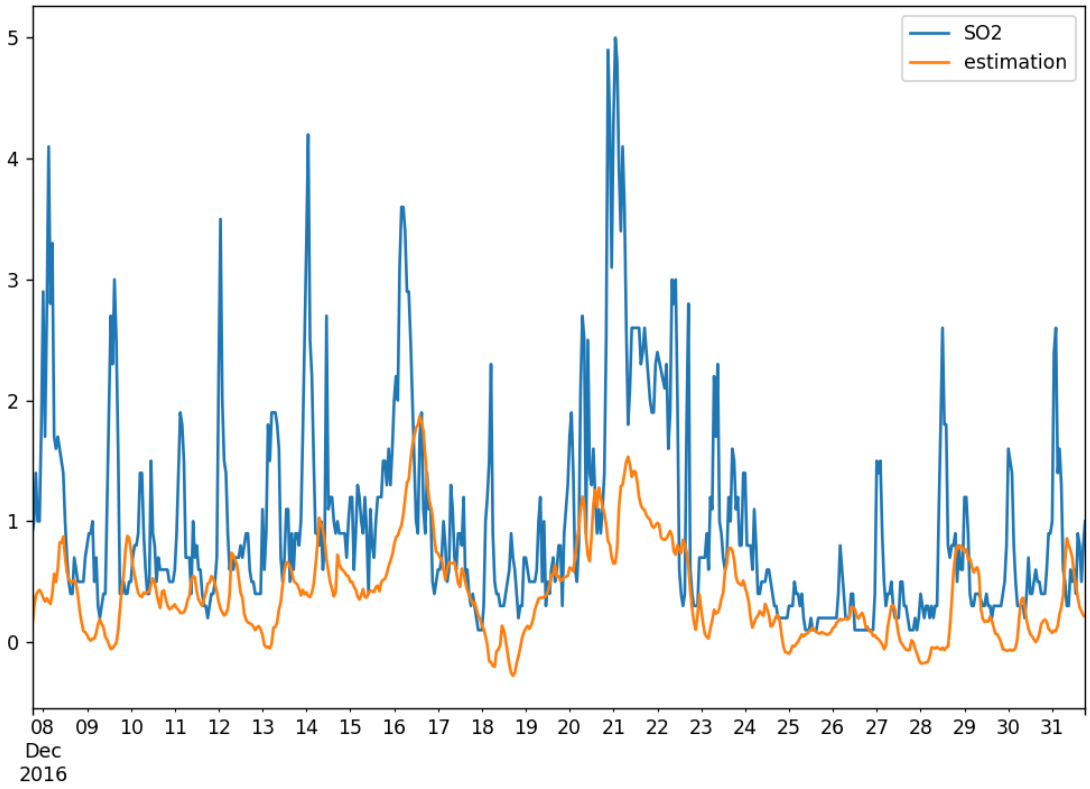

Figure 5.12 Visual representation of SO2 - small data set

As it was mentioned previously, now we are including traffic data inside our dataset, which we assume - should improve the accuracy of our algorithm. All achieved results are presented below.

Pollution predictions				
Batch size: 1, sequence length: 2, epoch: 20				
METRICS	**CO**	NO2	SO2	PM 10
RMSE	**0.1104**	9.6713	0.9233	22.332
MASE	**0.0253**	2.772	0.2159	7.401
MAE	**0.0695**	7.595	0.591	20.27

Table 5.8: Best results of CO - small data set including traffic data

Pollution predictions				
Batch size: 50, sequence length: 6, epoch: 20				
METRICS	CO	**NO2**	SO2	PM 10
RMSE	0.1361	**9.3338**	0.9654	29.4129
MASE	0.0352	**2.636**	0.2439	9.656
MAE	0.0966	**7.223**	0.6683	26.455

Table 5.9: Best results of NO2 - small data set including traffic data

Pollution predictions				
Batch size: 500, sequence length: 2, epoch: 20				
METRICS	CO	NO2	**SO2**	PM 10
RMSE	0.1434	16.910	**0.8100**	30.6794
MASE	0.0364	5.233	**0.1896**	10.159
MAE	0.0998	14.339	**0.5196**	27.834

Table 5.10: Best results of SO2 - small data set including traffic data

Pollution predictions				
Batch size: 1, sequence length: 1, epoch: 20				
METRICS	CO	NO2	SO2	**PM 10**
RMSE	0.1168	9.7351	0.8602	**13.3942**
MASE	0.0269	2.746	0.2005	**4.385**
MAE	0.0737	7.525	0.5494	**12.015**

Table 5.11: Best results of PM 10 - small data set including traffic data

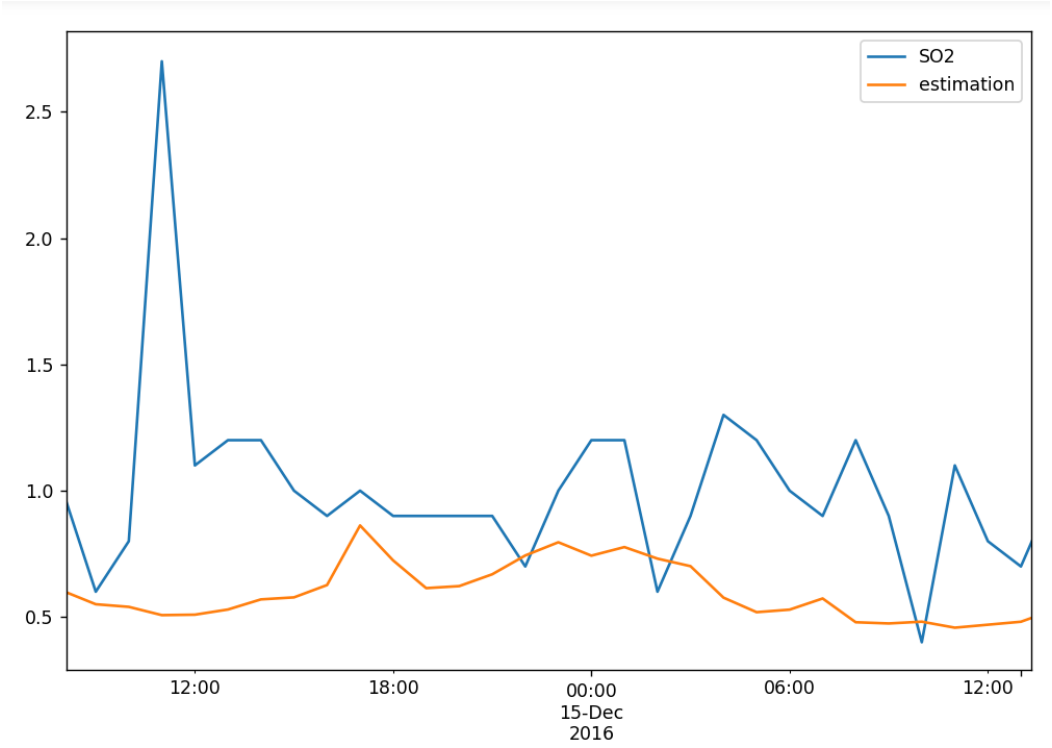

Figure 5.13 Visual representation of SO2 - small data set with traffic data zoom

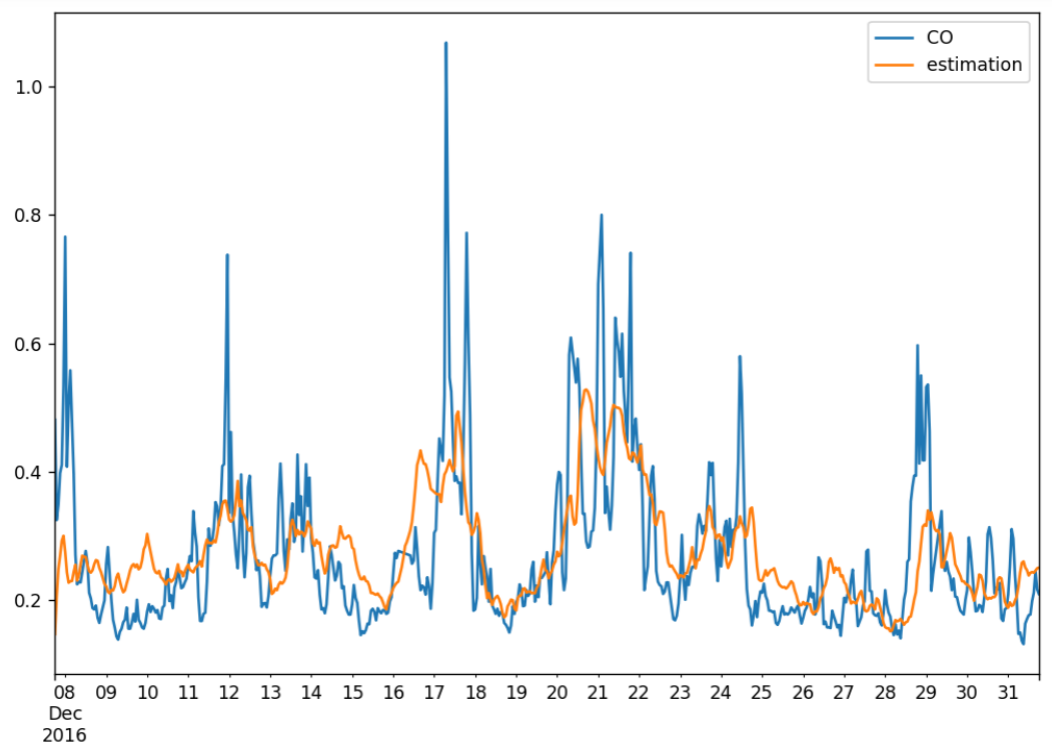

Figure 5.14 Visual representation of CO - small data set with traffic data

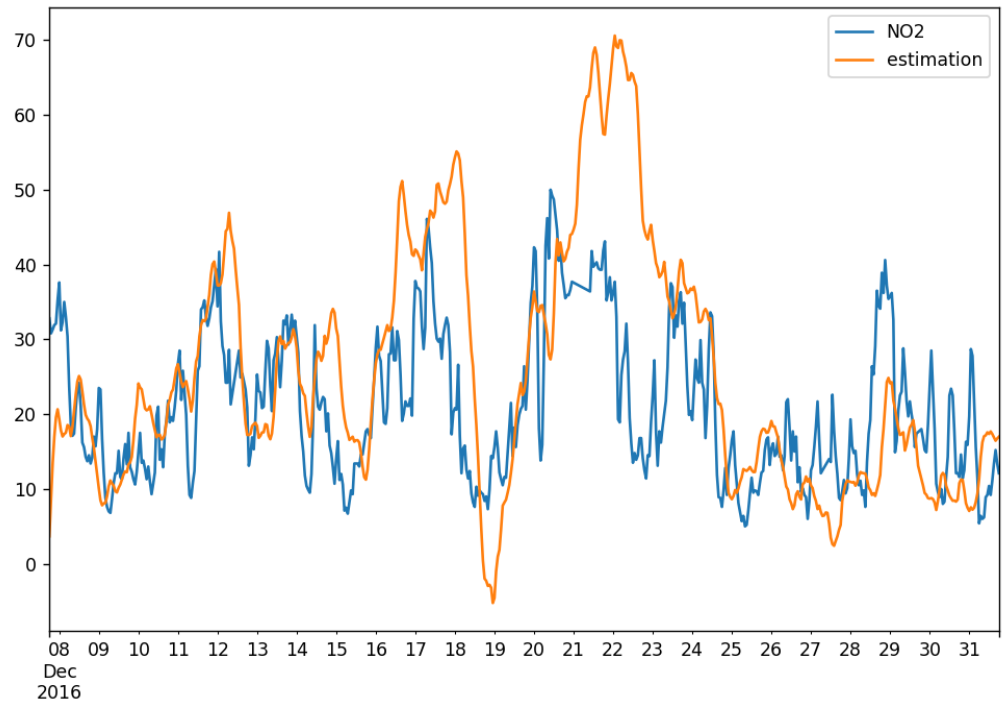

Figure 5.15 Visual representation of NO2 - small data set with traffic data

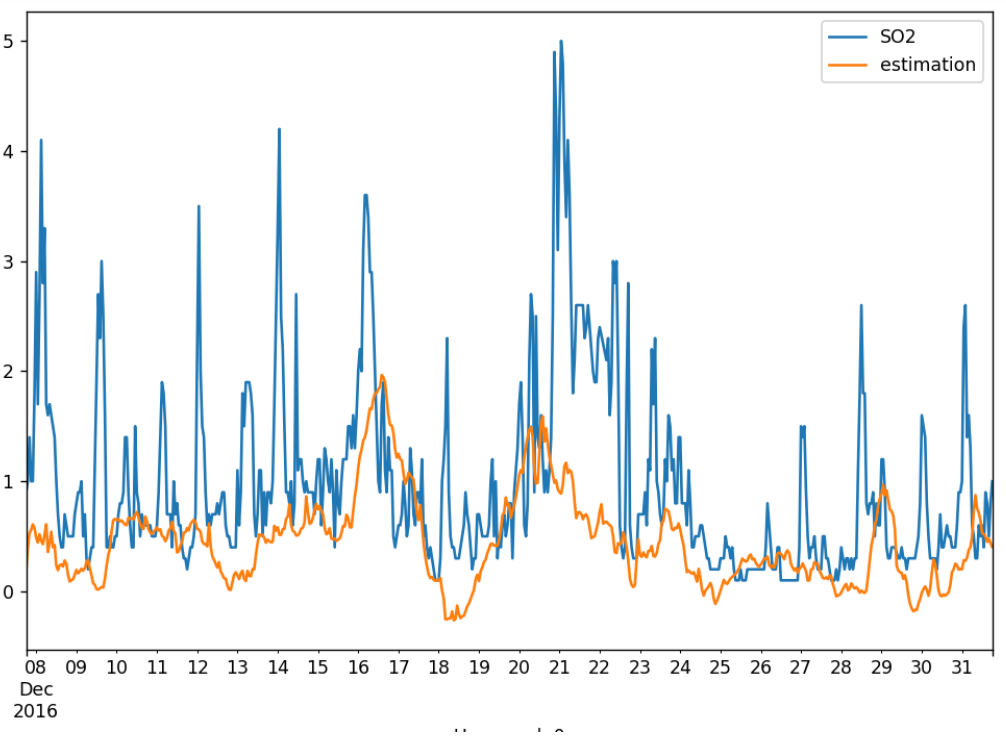

Figure 5.16 Visual representation of SO2 - small data set with traffic data

Chapter 6

Proposal

The short-term exposure to contact with contaminated air is now almost for every human being. The effects of such influences are more or less described. This is related to both respiratory mortality and morbidity, which can be caused by continuous staying in contaminated areas. During the work on such a problem it starts to be more clear, on what a large scale this problem occur during the last couple of years. Continuously growing interest in this topic, might have a huge influence on solution.

Considering our assumption at the beginning of this work, we might get into a conclusion that it is not that easy to estimate future pollutants, not only around urban areas, but everywhere. There is still huge challenge for researchers to be able to predict such a difficult problem, especially considering long period of time. During this work we were taking into account future estimation based on couple of hours, more specifically up to 6 hours ahead. Even if the period of time, and results show that algorithm is not estimate properly pollutants level in this type of configuration it still detect some patterns in way of prediction, but unfortunately it is still not enough.

We are sure that not every type of pollutants are predictable, because of complexity of the problem. A large number of variables have an influence for pollutants, so it is not a trivial to prepared algorithm with every necessary variable. We are also sure, that huge influence of this situation is fact, that the data were not entirely well prepared, there were many holes in them, and imperfections that even improved could have a negative impact on the algorithm.

There is one assumption that we made during the work, more precisely we checked if traffic data might have an influence for problem like forecasting. We found out that results

changed after adding this type of data. Even though, training period of time was very inadequate, for this type of problem, it ends up that the accuracy of the model change. Based on that there is a high possibility that for problem such as forecasting is affected by an unimaginable number of variables which people are not available to collect correctly during this years. Moreover, with usage of deep learning there is also another assumption, that we need incredible amount of data, in order to feed it inside our algorithm.

What is more interesting, during my research about problems of forecasting I did not come across of any work that included "traffic data" in research. It is important to mention that this problem is very complex. Even though with such a small amount of data used during this experiment, we were able to achieve some assumptions of our algorithm which include, e.g continuous follow-up in relatively corresponding results, which might be clearly seen in standard plot representation include below.

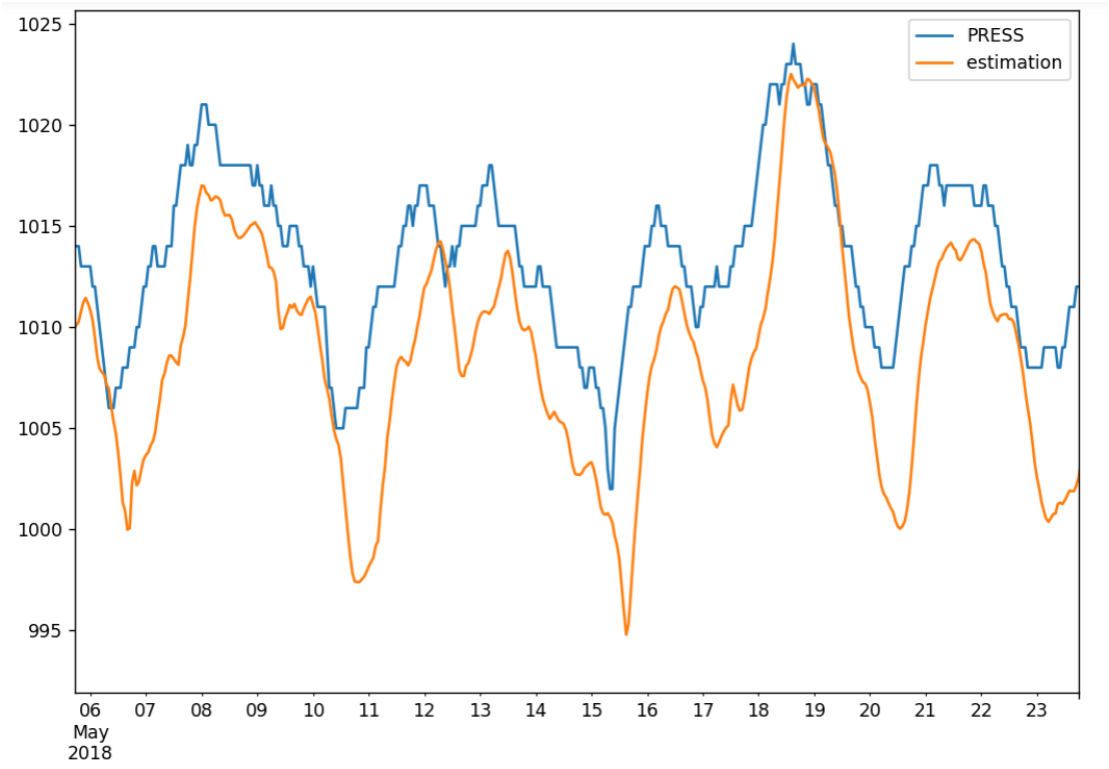

Figure 6.1 Visual representation of SO2 - small data set with traffic data

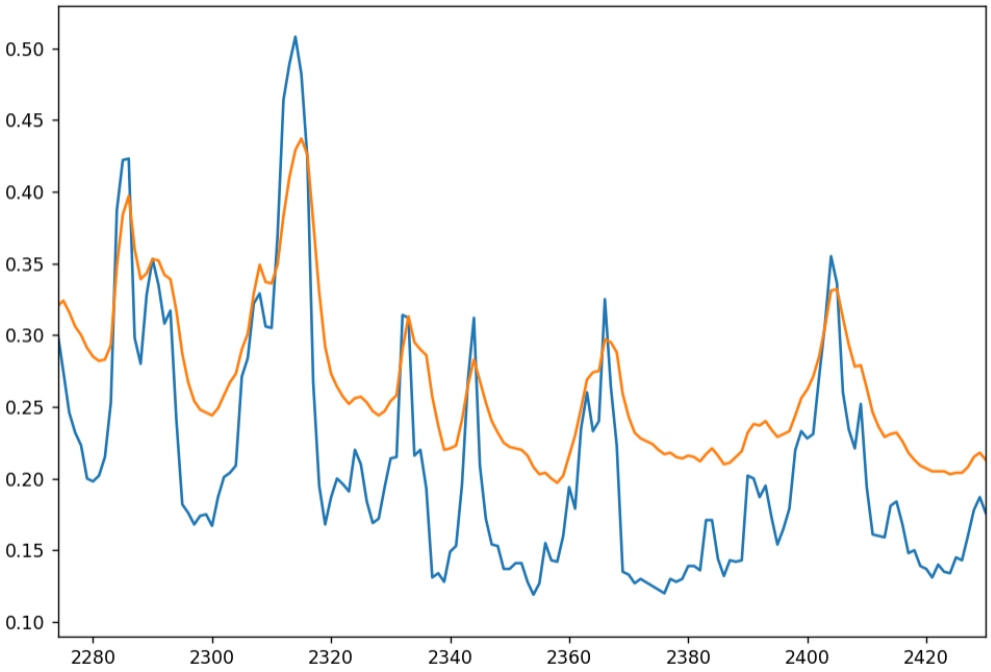

Figure 6.2 Visual representation of CO - best result

As we can see, the line which corresponding with estimation is following-up the true values in both cases, received during dissimilar period of time. This behavior of results may allow us to draw conclusions that algorithm, might improve results, if we include more specifically prepared data. As it was mentioned before, amount of data has a huge influence for accuracy prediction, with deep learning algorithm, in our scenario few years of data, with different type of problematic holes inside, might have undesired effect for work.

We also found out that extending the time that we intend to predict also has a negative effect on the results, with increasing batch size. The more we extend the time, and batch size, the less accuracy we got.

Yet we discover that our results, even if far from perfect, are still improving. There is also a chance that the impact on such a large discrepancy may have peaks departing from the normal behavior which are shown in the figure below. A large discrepancy may affect the results of the metrics, which also makes it impossible to obtain meaningful results.

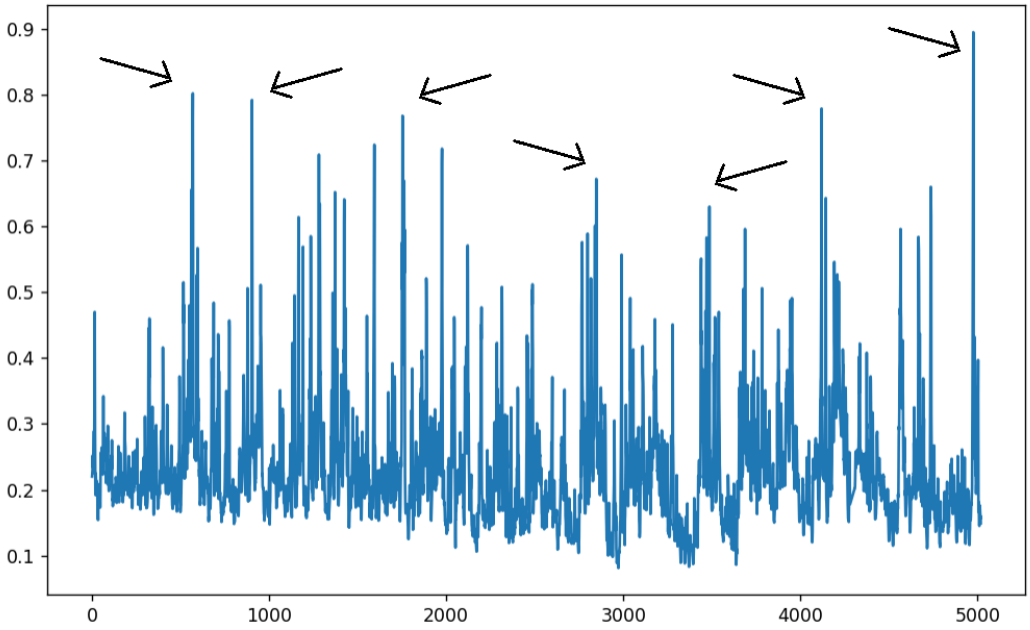

Figure 6.3 Visual representation of data imperfections

Summarizing everything, we were not able to predict even with the average result the future level of pollution in the given region, but we have discovered that the increasing amount of input data, which actually affect the level of pollution, also has a significant impact on the algorithm. To improve the correctness of the algorithm, it would be necessary to take into account very detailed data, free of errors and holes. The data taken into account should be collected in a size many times larger than the data used in this work. Moreover algorithm should be tested in the way of learning with huge amount of data. Moreover is appears that our implementation of Keras algorithm is working regardless of the assumptions imposed on it, especially because prediction of different pollutants have a completely different results depending on configuration. Due to this, it is very possible that algorithm should be prepared for a single detailed pollutant.

List of Figures

List of Tables

Bibliography

[1] United States Environmental Protection Agency
https://www.epa.gov/outdoor-air-quality-data

[2] A Deep Recurrent Neural Network for Air Quality Classifcation
http://bit.kuas.edu.tw/ jihmsp/2018/vol9/JIH-MSP-2018-02-009.pdf

[3] Air Quality Prediction in Visakhapatnam with LSTM based Recurrent Neural Networks
http://www.mecs-press.org/ijisa/ijisa-v11-n2/IJISA-V11-N2-3.pdf

[4] https://www.edureka.co/blog/machine-learning-tutorial/

[5] https://medium.com/datadriveninvestor/supervised-and-unsupervised-learning-7281050

[6] https://www.epa.gov/history

[7] https://pandas.pydata.org/

[8] https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/corre

[9] https://medium.com/@sebastiannorena/3d-plotting-in-python-b0dc1c2e5e38

[10] https://machinelearningmastery.com/dropout-regularization-deep-learning-models-ker

[11] https://pythonspot.com/matplotlib-scatterplot/

[12] https://matplotlib.org/3.1.0/gallery/text$_{l}abels_{a}nd_{a}nnotations/date.html$

[13] https://www.youtube.com/watch?v=UNmqTiOnRfg

[14] https://arxiv.org/pdf/1508.03790.pdf

[15] https://arxiv.org/pdf/1402.3511.pdf

[16] https://www.researchgate.net/publication/3857862$_{R}ecurrent_{n}ets_{t}hat_{t}ime_{a}nd_{c}ount$

[17] https://arxiv.org/pdf/1503.04069.pdf

[18] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[19] https://keras.io/

[20] https://www.hatarilabs.com/ih-en/how-to-calculate-the-root-mean-square-error-rmse-