Examination of air pollutant concentrations in Smart City Helsinki using data exploration and deep learning methods

Rabbil Bhuiyan

Master’s Thesis
Degree Programme
2021

MASTER’S THESIS	
Arcada	
Degree Programme:	Big Data Analytics
Identification number:	26182
Author:	Rabbil Bhuiyan
Title:	Examination of air pollutant concentrations in Smart City Helsinki using data exploration and deep learning methods
Supervisor (Arcada):	Dr. Amin Majd
Commissioned by:	

Abstract:

Air quality has become a major concern for most of the cities around Europe due to rapid urbanization and industrialization. Smart City is an initiative to solve such problems by integrating information and communication technology with citizens. Smart City, through smart computing technologies, allows capturing of huge data and the real picture of the domain problem. Provided by huge sensor data, air quality can be considered an essential component of the Smart City concept. The current thesis utilized the data from the Horizon 2020 mySMARTLife project, in which pollution detection sensors were deployed on public transport vehicles (trams) for continuous monitoring of pollution concentrations such as NO, NO_2, CO, and O_3 throughout the day. The study applied widely used several deep learning methods such as Convolutional Neural network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) for predicting hourly pollutant concentration based on spatial and meteorological information. The study also proposed an evaluation of features selection with different combinations of features for the model’s performance and showed the accuracy is increased by fusing meteorological variables and temporal feature engineering data. To figure out the best model performance, four evaluation measures such as coefficient of determination (r^2), Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE) along with model parameter optimization were applied. It is observed that all the models performed comparatively well in prediction at 24-hour window horizons. Particularly, LSTM architecture outperforms all the models in prediction quality having lower MAE values of 0.09, 0.056, 0.096, and 0.114 for NO, NO_2, CO, and O_3 respectively. Nevertheless, given the computational efficiency of the CNN algorithm, it can substitute deep feed-backward networks such as RNN, LSTM, and GRU models to predict pollutants rapidly and accurately in case of big data.

Keywords:	Air quality, Smart City, Deep learning algorithms, Time series, Data exploration
Number of pages:	73
Language:	English
Date of acceptance:	27.05.2021

TABLE OF CONTENTS

List of Figures

List of Tables

FOREWARD

First, I would like to express my sincere gratitude to the Big Data Analytics Team at Arcada University of Applied Sciences including my study colleague and the teachers to name a few Dr. Magnus Westerlund and Dr. Leonardo Espinosa Leal, without whom the journey would not be until now. Special thanks to my supervisor Dr. Amin Majd for his scientific and insightful supervision throughout the project.

I would like to thank Timo Ruohomäki, the program director of Forum Virium Helsinki Oy for all sorts of support regarding data acquisition and details. Very special thanks to Matti Irjala, Director of Aeromon, for his kind heart and several tireless efforts in improving the quality of the data hampered by the sensor.

I am also thankful to my reviewer Dr. Magnus Westerlund for his constructive feedback and recommendations throughout the project.

Last but not the least, I would like to thank Almighty God, my family especially my wife – heartful gratitude to her and friends for supporting me both spiritually and mentally throughout this journey. Without them, I would never have accomplished this feat.

Helsinki, May 2021

Rabbil Bhuiyan

1 INTRODUCTION

This chapter provides an overview of the challenges about the air quality around the cities and the motivation for this research, followed by research aims and limitations. Lastly, the outline of this thesis structure is presented.

1.1 Background

Air quality is one of the most important indicators defining an environment to be healthy. Apart from urbanization and industrialization, the increasing population and its automobiles are polluting the air at an alarming rate in major cities around the world. Air pollution refers to the contamination of the air by excessive quantities of harmful substances (pollutants) either in gaseous or particulate matter forms. The common such pollutants emitted into the atmosphere are particulate matter (PM), CO, NO, NO_2 and O_3. The sources of such pollutions are largely man-made such as energy production and utilization (IEA, 2018). The presence of these pollutants in the air deteriorates the quality of air, which eventually affects human health. Particularly children, elderly people, and those who suffer from asthma, cardiovascular problems, and respiratory issues are at high risk of being prone to effects of air pollution (Masih, 2018a). For example, a short-term O_3 exposure could bring acute coronary events in middle-aged adults without heart diseases (Ruidavets et al., 2005). Globally, about 4.2 million deaths were recorded attributed to air pollution during the year 2015 (Cohen et al., 2017). From another source such as International Energy Agency (IEA), about 6.5 million premature deaths were reported globally due to air pollution (World Energy Outlook Special Report, 2016). A regional study in Sweden reported 3000-5000 premature deaths every year because of inhaling pollutant air particles (Frisk & Partiklar, 2015). Therefore, it is important to address the air quality issue with real-time pollution concertation in a city so that people can arrange their activities accordingly, both in time and location.

1.2 Motivation

Air pollution forecasting- the prediction of the air pollutant concentrations for a given time and location is a hot spot of a current research topic. With an accurate air quality forecast and better knowledge, one can arrange his/her activities considering air pollution health effects. Such as one can choose the right choice for the cleanest routes for the commute, the best time for outdoor activities, and other daily activities. On a national level, accurate forecasting will assist the government for planning and establishing procedures to reduce the effect of air pollution.

The air quality issue is of major concern for many European citizens and one of the areas in which the EU has been focused seriously, to take preventive and regulatory measures. Followingly, Clean Air Force Europe (CAFE) initiative has been formed and set out the strategies and objectives among the member states. This initiative underlines a common framework of methods and criteria for direct comparison of air quality in different member states, as well as forecasting and management of air quality. Moreover, the CAFE Directive (2008/50/EC) includes mandates for the provision of air pollutant information and their predictions for the next days to the public (EC Directive, 2008). Thus, air quality information and their prediction are in demand by citizens, the EU, and the government alike. Foremostly, in Finland, air quality information and its predictions are urgently needed for improving the health of the citizen under CAFE guidelines.

Smart city initiative is on the rise in most cities around Europe, given the importance of increased attention on air quality from environment managers and citizens. One such smart city initiative is to mitigate the effect of air pollution by creating awareness with the emerge of new tools. The aim of the smart city initiative is to create a smarter environment and improve the quality of its citizen’s lives. This is done with the help of smart technologies available using the Internet of Things (IoT). In other words, Smart City is urban computing which is a process of acquisition, integration, and analysis of a large amount of heterogenous data generated by diverse sources in urban spaces (Yu et al., 2014). In the process, several sensors can be installed for example on a running vehicle in the city, to effectively monitor and forecast the air pollutant such as NO, CO and O_3 in smart cities. Next, IoTs can be used to transfer the information and data to the monitoring servers for data monitoring and tracking. However, for a smart city, merely monitoring the data is not enough, rather necessitates producing information from the data after analysis. Therefore, big data analysis techniques can be used such as machine learning algorithms for effectively monitoring, managing, and providing weather and air pollutant concentration information to the citizen. In this paper, we will apply artificial intelligence to accurately forecast various pollutant concentrations in the short future.

1.3 Aims of the study

The overall aim of this project is to create deep learning algorithms that will be able to predict the hourly pollutant concentration in the atmosphere. The specific research objectives of this project are:

1. A critique review of existing literature on forecasting air quality using deep learning algorithms
2. To discover the correlation between dependent variables particularly meteorological components and concentration level of pollutants (NO, NO_2, O_3, CO) and the hotspot of different air pollutants in the city
3. To investigate which features/variables have the highest impact on the machine learning algorithm’s ability to accurately perform prediction
4. To implement and evaluate the deep neural networks such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) in forecasting air pollutant concentrations

1.4 Limitations

The thesis project has been completed according to its requirement, however, there were some limitations particularly the data measurement. The dataset period of this project is only from February through August of 2019. The sensor used in collecting the data was an experimental project of Horizon 2020 mySMARTLife. The continuation of data collection for the whole year was not possible due to the costly calibration of the sensor. In addition, the quality of the data was degraded due to the inability to auto-calibrate the sensor. The dataset used for building the model was from February through June rather than through August because of the discontinuity of data which was caused by sensor degradation.

1.5 Thesis structure

The thesis is structured as follows: Chapter 2 introduces the knowledge necessary to understand the domain of the problem such as air pollution and quality followed by the Artificial Neural Network methods investigated in this study. Chapter 3 presents the existing and related works similar to the data model and prediction of this study. Chapter 4 describes the implementation of the models, along with data preprocessing methods used in this study as well as the evaluation of the models. Chapter 5 presents the results, validation, and discussion of the results followed by the conclusions and future works in chapter 6.

2 BACKGROUND THEORY

This chapter introduces the relevant theory of air pollution and the related work of air quality prediction with deep learning methods for the reader to pick up on key issues and concepts on the research topics. Section 2.1 defines air pollution and its main causes and air quality index in the context of Europe. Section 2.2 presents a brief introduction of the deep learning methods considered for prediction study.

2.1 Air pollution

Air pollution refers to the contamination of the air, irrespective of indoors or outdoors, by the excessive amounts of harmful substances (pollutants). The pollutants could be either in the form of gases, particulates matter, or biological molecules. Air pollution occurs when the pollutants enter the atmosphere which is a combination of gases and after mixing up makes the air dirty and difficult for plants, animals, and humans to survive (Akimoto, 2003; Seinfeld et al., 2012).

2.1.1 Air quality

Air quality refers to the condition of the air within particular surroundings. Good air quality describes the condition when the air is clean, clear, and free from pollutants as mentioned in Table 1. Thus, clean air is considered to be a basic requirement for human and well-being. The most common type of pollutants along with their sources and short descriptions are listed in the table. On the other hand, the air with impurities implies poor air quality which causes the risk to the lives on earth e.g for humans, plants, animals, and natural resources, and deterioration of the environment from acid rain to global warming.

Table 1. The main types of pollutants. Nox, CO, and O3 are the pollutants of interest in this thesis

Pollutants	Sources and descriptions
CO_2	Carbon dioxide from various types of combustion in industrial and domestic environments, power plants, and transport.
CO	Carbon monoxide primarily from motor vehicle exhaust and machinery that burns fossil fuel. Naturally present in the air and transformed into CO_2 over time.
SO_X	Sulfur oxides (SO_2 and SO_3) from the combustion of coal or oil. The emission of gases by the industry is high.
NO_X	Nitrogen oxides (NO, NO_2 and N_2O) from vehicle, industry, and agricultural, and livestock activities. The presence of this element in the atmosphere destroys ozone layer.
O_3	Ozone is not emitted directly but is formed from exposure to sunlight. It generates chemical reactions with other components and becomes into hazardous element.
PM	Particulate matter highly dependent on local conditions such as climate, traffic, and pollution. Two diameter classes (in microns) of $PM_{2.5}$ and PM_{10} are most common.
VOC	Volatile organic compounds like methane, hydrocarbons, chlorofluorocarbons mostly come from industrial production.

2.1.2 Sources of air pollutants

Various emission sources can reduce air quality. The sources of air pollutants could be primary and secondary pollutants sources. The primary pollutants are emitted from the source either natural or man-made to the atmosphere directly. Natural sources could be e.g sandstorms, volcanic eruptions, forest fires, and biological decay where human-made sources could be industrial emissions, vehicle emissions, burning wood or coal, power plant, etc. The secondary air pollutants result from the chemical or physical interactions between primary pollutants such as photochemical oxidants and particulate matter.

2.1.3 Air quality index

As most of the air pollutants are known to hazardous to health, the US Environmental Protection Agency (EPA) measures the levels of these pollutions in order to control the overall air quality. EPA thus has set standards for the level of these pollutants for providing allowable guidelines. Similar standards and guidelines are also set by the EU and other national environmental agencies which are expressed as air quality index. The air quality index (AQI) is an indicator created to report air quality, describe how clean or unhealthy air is, and potential health risks especially to vulnerable groups. The AQI focuses on the various health effect that people might experience within a few hours or days after being exposed to polluted air. The AQI values differ from country to country and higher AQI indicates the greater risk to the health of people. Table 2 below describes the different classifications of AQI for Europe (EEA, 2017).

Table 2. AQI level classification for Europe

Index level	Pollutant (pollutant concentration in μg/m3)				
	PM $_{2.5}$	PM_{10}	NO_2	O_3	SO_2
Good	0-10	0-20	0-40	0-50	0-100
Fair	10-20	20-40	40-90	50-100	100-200
Moderate	20-25	40-50	90-120	100-130	200-350
Poor	25-50	50-100	120-230	130-240	350-500
Very Poor	50-75	100-150	230-340	240-380	500-750
Extremely poor	75-800	150-1200	340-1000	380-800	750-1250

2.2 Artificial Neural Network Methods

The objective of this project is to develop machine learning models in forecasting air quality concentration for a short period time. And to implement the research objective, ANN architecture is applied. ANN is a mathematical model that imitates the function of biological neurons. An earlier ANN architecture such as Multilayer Perceptron (MLP) has already shown good performance and applied widely. However, MLP which consists of neural networks of fully connected architecture is a kind of shallow neural network and fails to perform effectively when data complexity is high. At present, many deep neural network architectures have been developed for ANN. In this project, the main deep neural network architecture is Convolutional Neural Network (CNN). In addition, we will apply Recurrent Neural Network (RNN) based Long Short-Term Memory (LSTM) and Gated Recurrent Network (GRU) methods for a comparative performance evaluation. We will apply RNN and its variants because they can forecast with sequential information as for time series data by connecting with previous events.

2.2.1 Convolutional Neural Network (CNN)

Although CNN is particularly well known for its success in image analysis, it can be effectively applied to time series analysis because of its weight sharing and sparse connectivity concept. CNN algorithms comprise of multiple-layers and the layers can be grouped into three parts.

Part one mainly deals with convolution. The input or the output of the previous layer is convolved with the kernel, which is a sliding window, to extract the sequential features of time series data.

Part two is concerned with the pooling layer. The pooling layer provides translational invariance, aiming to preserve the features in a smaller representation by discarding less significant data. Max pooling operates by sliding the window sequentially on the input vector and then takes the maximum value of the window region and gets rid of the other values.

Part three concatenates all vectors together to form a long vector which is called Fully Connected Layer. It is usually a Multilayer perceptron (MLP) which is a kind of feedforward neural network and consists of three layers of nodes such as input layer, an output layer, and a hidden layer. In any Neural Network (NN), the hidden layers play a critical role in the back-propagation algorithm process. For example, a network with a single hidden layer can approximate any function between input and output vectors by selecting a suitable set of connecting weights and transfer/activation functions (Hornik et al., 1989).

2.2.1.1 *Feed Forward propagation*

In the multilayer perceptron neural network, the input is passed through the hidden layer to the output layer in a feed-forward manner. The feed-forward maps the MLP layers to that of the output layers/values using an activation function (ʃ), which is also known as the transfer function. The activation function aims to introduce nonlinearity to the neural network and the function is applied to every layer of the network except the input nodes. In literature, there are two kinds of activation functions available such as the tangent function whose output ranges from -1 to 1 and the sigmoid function with range from 0 to 1 (Karlik et al., 2015).

In CNN the training procedure is similar to that for a standard NN using the back-propagation algorithms and associates with biases (b) and errors for the training. The purpose of using biases is to preserve the universal approximation of the neural network (Hornik et al., 1989; Cai et al.2009).

2.2.1.2 *Weight update*

The weight share and update concept are the main parameter of CNN that differs it from MLP. The back-propagation process calculates the error function and thus updates the synaptic weight (W) of the input nodes to reduce the loss function (Cai et al.2009; Battiti, 1992). The error functions are backpropagated through the network and adjust the weight to that of error until the desired output is achieved in the output layers. In the back-propagation training method, the weight is updated using the delta rule which is given by gradient descent on the square error (Battiti, 1992). Delta rule determines the iterative process to achieve the reduced error between the desired output and network output.

In this algorithm, a parameter called learning rate is used to determines the weight for each updating step and thus to achieve gradient descent in error functions. A smaller learning rate takes a longer time to achieve gradient descent, while a bigger learning rate requires a larger modification of weight to achieve gradient descent. A diagram of deep CNN architecture is presented in figure 1 below, where k = input number, ln=Input, Out=Output, continuous lines = weights, and bias (Bassam et al., 2010).

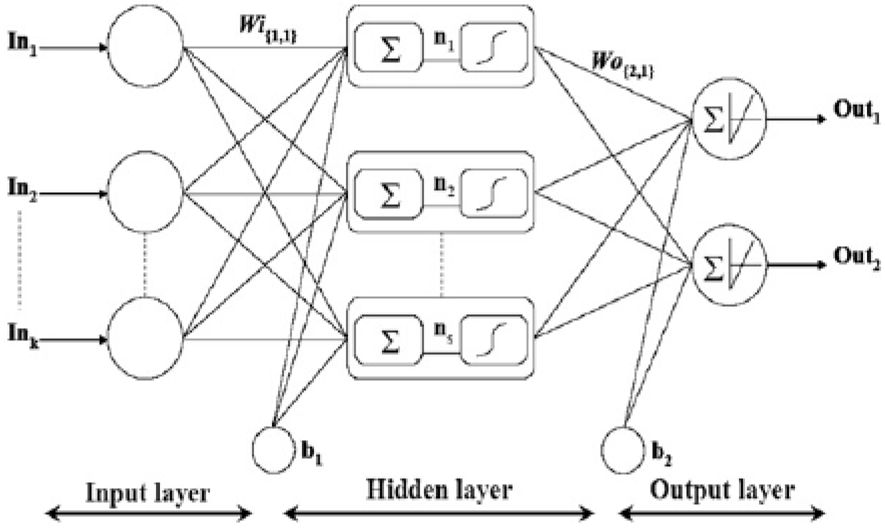


Figure 1. A diagram of CNN architecture (CNN).

2.2.2 Recurrent Neural Network (RNN)

RNN differs from CNN and MLP in its consideration of time sequence. RNN provides continuity of information flow, based on which time series analysis is performed. Figure 2 shows the RNN architecture with its unfold version *(*LeCun et al., 2015). The symbol x_t is input sequence, o_t is output sequence, s_t is hidden state vector, and W, U, V weight matrices. RNN maps an input sequence (x_t) into an output sequence (o_t) according to the recursive formulas of RNN as follows:

$$s_t = \mathit{tanh}\ (Ws_{t-1} + Ux_t) \qquad (1)$$

$$o_t = Vs_{t-1} \qquad (2)$$

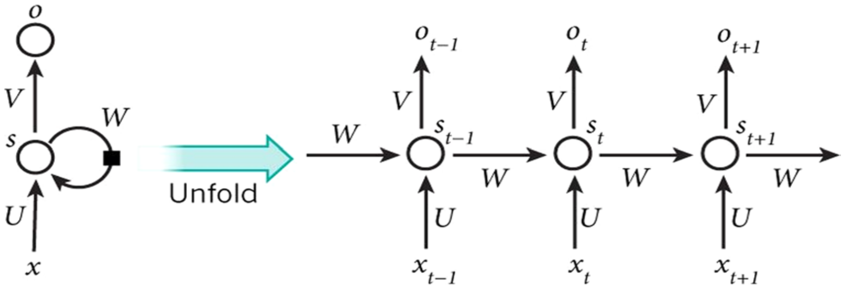
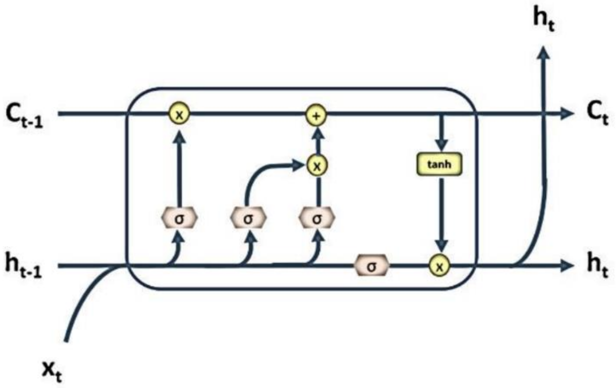


Figure 2. The architecture of recurrent neural network (RNN).

2.2.3 Long Short-Term Memory (LSTM)

LSTM is a modified version of RNN (Gers et al. 2003). It differs from traditional RNN by introducing a memory block i to reduce/vanish the gradient problem of RNN. In LSTM, each neuron in its structure performs as a memory cell. The structure of LSTM functions in such a way that the current cell uses the information from the previous memory cell during processing. In this way, data is transferred from one cell to another and temporal dependencies are stored. Moreover, LSTM can be used in modeling longer sequence data and it differs from other types of learning models with its three-gate structure (forget gate, input gate, output gate). The architecture of LSTM is presented in figure 3 below (Yang et al., 2020), which is composed of cells and intercellular data transfer. The symbols in the figure are x_t is the input of current cell, c_t is the cell memory, h_t is the output of current cell which will be used in the next cell as a hidden layer. c_{t-1} and h_{t-1} represent previous cells and ensures sequential dependency. σ is the Sigmoid activation function and *tanh* is the Hyperbolic Tangent activation function.



σ

Figure 3. The architecture of Long Short-Term Memory (LSTM).

The equations below show how LSTM cells in recurrent layers process data forward through different gates (forget gate, input gate, output gate). In the equations, i refers to the operation of input, o refer to output operation and f refers to forgetting gate operation, t is the current time, t-1 is previous time, h refers to hidden state and C refers to cell state, W and b are the weight and bias vector, σ is Sigmoid activation function and *tanh* is Hyperbolic Tangent activation functions.

2.2.3.1 Forget gate

Forget gate decides how much state data to preserve. At forget gate, the output from previous cell h_{t-1} is combined with the input of current cell x_t and this combination is introduced into Sigmoid functions as in Eq. 3. Then, it determines the extent of the existing forgotten information according to the multiplication of the output of Sigmoid function and C_{t-1} as in eq. 4. The Sigmoid function output is between 0 and 1, where 0 denotes complete forgetting and 1 denotes complete remembering.

$$s_t = 1/1 + e\text{-}t \qquad (3)$$

$$f_t = \sigma\,(W_f\,.\,[h_{t-1}, x_t] + b_f) \qquad (4)$$

2.2.3.2 Input gate

The input gate layer is composed of the *Sigmoid* layer and *tanh* layer. The *Sigmoid* layer decides which value will be updated as in eq. 5 whereas, *tanh* layer generates candidate value of $\hat{C}_t$ as in eq. 6. The output of these two layers is added to the function c_t after element-wise multiplication as in eq. 7.

$$i_t = \sigma\,(W_i\,.\,[h_{t-1}, x_t] + b_i) \qquad (5)$$

$$\hat{C}_t = tanh\,(W_c\,.\,[h_{t-1}, x_t] + b_c) \qquad (6)$$

$$c_t = f_t * c_{t-1} + i_t * \hat{C}_t \qquad (7)$$

2.2.3.3 Output gate

The output gate decides cell output at time t as in eq. 8 where the output of $h_{t-1}$and input of x_t is combined within a Sigmoid function. This output determines how much information will be retrieved from the cell state. The cell output is determined according to the eq. 9.

$$o_t = \sigma\ (W_o\ .\ [h_{t-1}, x_t] + b_o) \qquad (8)$$

$$h_t = o_t * tanh\ (c_t) \qquad (9)$$

2.2.4 Gated Recurrent Unit Network (GRU)

GRU is an extension of LSTM architecture (Dey & Salem, 2017). It consists of update and forget/reset gates which together balance the flow of data inside the unit (LeCun et al., 2015). The structure of a GRU is shown in figure 4 as described by Yang et al. (2020).

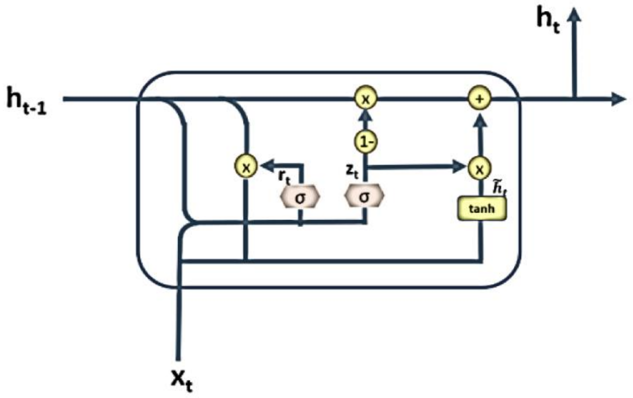


Figure 4. The architecture of gated recurrent unit (GRU).

The principal difference between LSTM and GRU lies in their gates and weights. The update gate performs functions similar to the forget and input gates of the LSTM as in eq. 10 where update gate controls new information and previous information of cell/unit. The forget gate indicates how much past information to forget as in eq. 11 that shows which forget gate is included in candidate activation. The eq. 12 and eq. 13 combine the candidate state with previous output and filter the data to obtain the output of the current state. In the formulae below x denotes the input vector, h is the output vector, z is the update gate vector, r is the reset/forget gate vector, w and b are the weight and biases respectively, and t is the time.

$$z_t = \sigma\,(W_z\,.\,[h_{t-1}, x_t] + b_z) \qquad (10)$$

$$r_t = \sigma\,(W_r\,.\,[h_{t-1}, x_t] + b_r\,) \qquad (11)$$

$$\hat{h}_t = tanh\,(W\,.\,[r_t * h_{t-1}, x_t] + b_h\,) \qquad (12)$$

$$h_t = \mathrm{z}_t\,.\,\hat{h}_t + (1 - \mathrm{z}_t) * h_{t-1} \qquad (13)$$

3 LITERATURE REVIEW

This chapter presents the related work of air quality prediction with machine learning and deep learning methods. This chapter also addresses the concept and related work of the smart city project imitative.

3.1 Background

Traditionally, the methods for modeling and forecasting air quality data can roughly be divided into deterministic and statistical methods. The first approach is theoretical and employs mostly meteorological emissions and chemical models (Jeong et al., 2011). These models are based on simulations that quantify the deterministic relationship between the pollutant sources including its remittance, exchange, diffusion and expulsion process, meteorological processes, chemical changes and pollutant concentrations, and so on (Baklanov et al., 2008). For example, chemical models e.g WRF-Chem was used in a deterministic way for forecasting urban PM_{10} and $PM_{2.5}$ concentrations (Saide et al, 2011), simulated meteorological model (Kim et al., 2010), and deterministic Lagrangian trajectory model (Schlink et al., 2003) were developed in studying the urban ozone concentration. On the other hand, the statistical approach simply uses statistical modeling techniques such as multiple linear regression (MLR) (Li et al., 2011), autoregression moving average (ARMA) (Box et al., 2015), and so on. However, these methods rely on extensive historical measurements at spatially distributed monitoring stations. In addition, the statistical approach generally is based on strong assumptions such as specifying a priori and the error distribution and cannot address the issue of multicollinearity, i.e., the high degree of correlation between two or more independent variables.

However, they yield limited accuracy in prediction quality due to for example insufficient theoretical information, incomplete knowledge on the pollutant sources and underlying complex meteorological conditions in case of deterministic method, and inability to model nonlinear pattern in data for simple statistical method. Air quality data is large and inherently complex composed of both temporal and spatial data and characterized by nonlinear patterns of data. Thus, the prediction of air pollutant concentrations using these methods is not exact (Goyal et al., 2006). A promising alternative to solve the above shortcomings are artificial neural networks (ANNs) (Sánchez et al., 2013; Gardner & Dorling, 1998).

3.2 ANNs in Air quality

In the last several years, ANNs have been increasingly applied to improve the accuracy of predictive models and forecast the air quality. ANN method can be applied successfully as tools for efficient decision making and problem-solving for better atmospheric management (Azid et al., 2013). ANNs are the specialized mathematical framework for modeling a large variety of non-linear data. They are modeled in a way to mimic the functions of the human brain where several layers containing nodes (like neurons in the human brain), usually at least three, are connected to form a network for parallel processing of data input. The nodes apply an activation function on the inputs from previous layers and pass through. Lastly, an output layer generates the probability of outputs based on the previously hidden layers.

3.3 Shallow Vs Deep ANNs

3.3.1 Shallow ANNs

The most common ANN models are multilayer perceptron (MLP), recurrent neural networks (RNN), and radial basis function networks (RBFN) (Kuremoto et al., 2014). Over the years, these techniques have been applied and incorporated into multiple approaches in predicting air quality forecasting (Wang et al., 2017; Peng et al., 2016). However, these methods concern limitations regarding the performance of such methods. For example, the MLP method which is composed of three layers-input layers, hidden layer, and output layer are regarded as universal function approximation, meaning that they can be applied to different arbitral and multi-dimensional functions. However, to approximate the statistical distribution of the data and to avoid local error minima, a back-propagation algorithm is applied by adapting weights in the hidden and output layers. This necessitates definition of a priori and leads to overfitting which demands the employment of an adaptive structure (Panchal et al., 2011). The details of MLPs and back-propagation algorithm learning are mentioned in the study by Zell (1994). Moreover, they also require the determination of optimum network structure (number of input variables, or hidden layers). Over the years, many extensions on MLP have been developed for tackling specific data-driven problems such as Seasonal Artificial Neural Network (SANN), Time Lagged Neural Networks (TLNN), Probabilistic Neural Network (PNN), Generalized Regression Neural Network (GRNN) (Oludare et al., 2018).

To overcome the problems attributed to MLP NN like local minima and overfitting, support vector machine (SVM) NN algorithms are developed. The main goal of SVM is to provide a well-generalized decision rule for selecting subgroups from the support vectors (training data) and can solve a linearly constrained quadratic problem as a training. Nevertheless, the accuracy of SVM is poor while a large training dataset is used because the computational resources increase the complexity in processing (Adhikari & Agrawal, 2013). Based on SVMs other extensions have been developed such as the Least Square Support Vector Machines (LS-SVM) algorithm and its variants, i.e. the Recurrent Least Square Support Vector Machines (RLSSVM), the Dynamic Least Square Support Vector Machines (DLS-SVM), etc. and in all these proper choice of parameters in needed for better model accuracy.

3.3.2 Deep ANNs

Because of these limitations, the aforementioned neural networks which are a ‘shallow’ type of NN, produce substandard model accuracy, and therefore, the use of ‘deep’ neural networks emerges (Bengio et al., 2007). A deep learning algorithm consists of a hierarchical architecture with many layers each of which constitutes a complex and non-linear information processing unit. The concept of deep learning originated from the study on Artificial Neural Network (ANNs) (Hinton & Salakhutdinov, 2006) as mentioned in the shallow NNs. The first deep learning technique was proposed in 2006 by Hinton (Hinton & Osindero, 2006) which was a training method of a layer-wise-greedy-learning algorithm. The idea behind the layer-wise-greedy-learning is that unsupervised learning should be performed for network pre-training before the subsequent layer-by-layer training. A detailed overlook of the principles of deep neural networks (DNN) is explained in Liu et al. (2017). Deep learning architectures have been applied using four main techniques along with their variants as explained in Liu et al. paper such as Restricted Boltzmann Machine (RBM), Deep Belief Networks (DBM), Auto-Encoders (AE), and Convolutional Neural Networks (CNN).

3.4 Shallow ANNs in air quality forecast

A previous study showed that non-linear model like ANN was more accurate than linear models in predicting air quality which because of clear nonlinear pattern present in air quality data (Gardner & Dorling, 1998; Prybutok et al., 2000). Gardner and Dorling described that MLP neural network method while using temporal variations (time of day and day of the week) as an input variable, is powerful in forecasting NO_2 concentration in London relative to other statistical modeling methods. They also found consistent performance for the neural network approach when simple meteorological input variables were used. In another study in Stockholm, the MLP NN approach yielded a more accurate regression model in forecasting NO_2 concentrations over other statistical methods (Kolehmainen et al., 2001). In a study in Santiago, Chile, multilayer NN gave the best results in forecasting the hourly concertation of PM2.5 (Perez et al., 2000). An EU-funded project named APPETISE (Air Pollution Episodes: Modelling Tools for Improved Smog Management) indicated a neural network-based model as better in predicting O_3 concentration in Germany, Italy, UK, and the Czech Republic (Schlink et al., 2003).

3.5 Deep ANNs in air quality forecast

With the rapid development of computation techniques, machine learning techniques have hugely shifted towards deep learning DNN algorithms. Their complexity and accuracy have improved the solutions for existing problems particularly in air quality prediction where the data are characterized with high complexity. Ong et al. (2016) introduced a novel method called Dynamic pre-training (DynPT) for time series prediction of air quality. Their research might be one of the first DNN techniques to apply in forecasting $PM_{2.5}$ concentration levels. Another novel technique based on deep learning (DL) to forecast air quality levels in Beijing is proposed by Wang and Song (Wang & Song, 2018). They applied LSTM using both historical and meteorological data and the model increased the prediction accuracy over other machine learning methods.

Given the complexity of air pollutant data which could be explained with a progressive connection with meteorological conditions, several DL methods such as RNN, LSTM, and GRU were applied by Athira et al. (2018). They found the GRU model superior to other models in air quality forecasting considering the changeability of air quality prediction. In another study proposed by Wen et al. (2019), high accuracy for air quality prediction of different Spatio-temporal scales was established using both novel LSTM and Extended Convolutional Long-short Term Memory Neural Network (C-LSTME) models.

3.6 Deep hybrid ANN in air quality forecast

Evidence also showed that a combination of different non-linear models which is called a hybrid model demonstrates satisfying predictive performance than the single model used (Sánchez et al., 2013; Chen et al., 2013). In a recent study, a hybrid deep learning model called Deep flexible sequential (DFS) was also proposed by Kaya & Öguducu (2020). They utilized different deep learning algorithms e.g CNN, LSTM, and Dropout layers together in forecasting the PM10 in the atmosphere using multivariate time series data.

Qi et al. (2019) proposed a hybrid model to improve the forecasting of $PM_{2.5}$ concentration levels. In their approach, they applied a combination of Graph Convolutional Network (GCN) to extract spatial dependencies between different stations and an LSTM to capture temporal dependencies among observations at different times. They also stated that their hybrid method outperforms an MLP and a naïve LSTM for the same dataset.

Zhou et al. (2019) developed a Deep Learning based Multi-output LSTM Neural Network (DM-LSTM) using 2 hidden layers and three combined DL algorithms in predicting multiple air quality outputs. Their hybrid version improved the prediction quality compared to other versions of the LSTM network.

3.7 Deep learning in Smart City project

Smart City initiatives aim at creating a smarter environment to improve the quality of citizens lives. The idea behind the smart city is the utilization of smart technologies and equipping the city with the functions of such technologies. For example, a smart urban sensing system architecture using the Internet of Things (IoT) to monitor PM2.5, temperature, and noise (Liu et al., 2015). They equipped IoTs with sensing and monitoring systems to efficiently collecting data for analysis and strategy evaluation, particularly in forecasting PM2.5 in the city. Europe is not far behind the smart city initiatives. There are many smart city projects to monitor the weather and surrounding environment using IoTs that are equipped with smart sensing systems. For example, a smart city project in Uppsala, Sweden along with the collaboration of IBM, Ericsson, and Uppsala University. A pollution detection sensor was deployed all over Uppsala to monitor pollution concentrations and developing a machine learning model in forecasting the pollution concentrations (Subramanian, 2016). Apart from these projects, the EKOBUS project in Serbia where sensors being placed on bus rooftops to give real-time pollution data (Libelium World, 2015), RESCATAME project in Salamanca, Spain to monitor pollution sources (Libelium World, 2015), just to name a few. A similar smart city initiative, which is a part of the Horizon 2020 mySMARTLife project was paved in Helsinki city, where sensor systems were installed on trams to monitor the air pollutant sources. This master thesis project is developed based on the mySMARTLife project data.

Smart city projects are capable to generate heterogeneous IoT data with the advancement of sensor technology. Luckily, such various, heterogenous, high-volume and real-time data obtained from smart cities are not challenging, rather the Big Data Platform makes the process successful. A deep neural network e.g deep autoencoder was applied in detecting pollutant sources in a smart city project conducted by Zhang et al. (2016).

4 MATERIALS AND METHODS

This chapter describes the dataset, data preprocessing, and the individual model implemented in the thesis. This chapter also presents the methods and libraries used during the project.

4.1 Dataset

4.1.1 Dataset description

The dataset is a part of the Horizon 2020 mySMARTLife project and about air quality data collected using Aeromon BH-12 measurement devices installed on trams in Helsinki during 2019 from February through August. The source of the data is Forum Virium Helsinki Oy and publicly available. Three sensor sets were installed on the trams that have been moving on the streets of Helsinki and one has been constantly measuring the data at Helsinki Environmental Services (HSY) measuring station located at Mäkelänkatu. A total of 60,52,856 records were collected with a sampling measurement rate of one second. The data include meteorological information such as air pressure (in pascal), humidity (in %), and temperature (in degree Celsius). The important attributes of the dataset are timestamp divided into month, day, hour, geocoordinate (latitude and longitude), NO, NO_2, O_3, and CO. The dependent variables are NO, NO_2, CO, and O_3 that contain concentration level information in ppm and the rest of the attributes are used as independent variables which also portrays an important role in predicting the air quality.

4.1.2 Primitive background correction of the data

The data quality was hampered by the uncertainty due to calibration or gas tolerance of the sensor. There was no automated background compensation technique in the Aeromon device used for collecting the data. Thus, the device read noise (data beyond the detection range) to the data. The unstable power supply to and from tram also caused reading of data beyond the detection range. Thus, this thesis performed a primitive background correction to the data. For this, the median value of the signal for 5 minutes window/slice was calculated and then, subtracted that median value from all the data points within that time window, and the process was applied to the whole dataset.

4.1.3 Exploratory data analysis (EDA)

4.1.3.1 Visualization of trend in the dataset

The trends in the air pollutant concentration and meteorological data are shown in Figure 5 and Figure 6 respectively. This first visualization of time-series data serves us as a descriptive tool to show both trend and seasonality, potential outliers, detecting range outside of actual concentration or discontinuities in the data, and also allows us to choose appropriate techniques for forecasting the time series model. For example, figure 5 below shows that there exist clear abnormal peaks for CO pollutants which were caused due to the abruption of power to the devices.

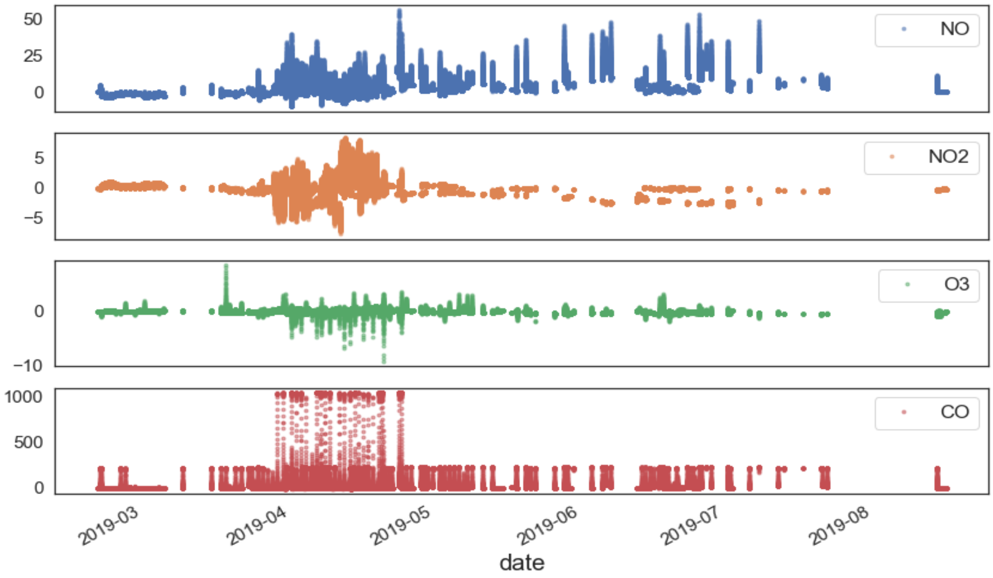


Figure 5. Hourly NO, NO$_2$, CO, and O$_3$ concentration of data.

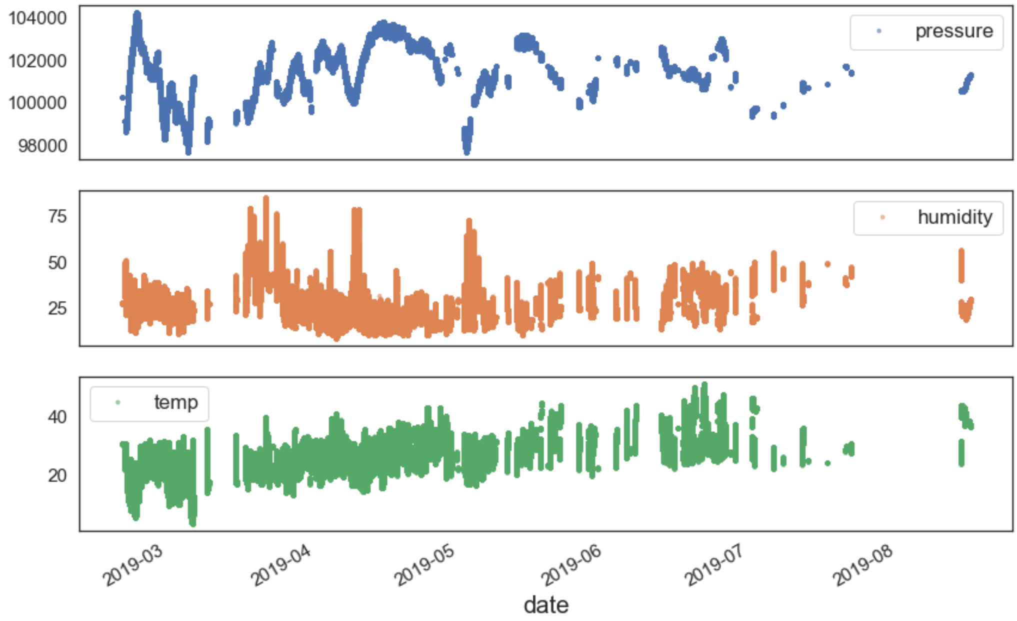


Figure 6. Hourly meteorological data for pressure, humidity, and temp (temperature).

4.1.3.2 Statistical descriptions

A short descriptive statistic of the different pollutants and meteorological variables such as minimum, maximum, mean, standard deviation, quantiles, kurtosis, and skewness is provided in table 3. This table shows a high value of skewness for most of the concentration values which indicate the presence of a sharp increase in the data. The high value of kurtosis indicates the existence of the data discontinuities which is mainly due to disruption of power supply. The high standard deviation for different variables indicates high sensitivity to uncertainties.

Table 3. Statistical descriptions of the dataset

	NO	NO2	O3	CO	Pressure	Humidity	Temperature
count	5095036	5095028	5067860	4986279	6052856	6052856	6052856
mean	1.12	0.08	0.04	0.08	101577.02	22.68	25.75
std	3.42	1.39	0.21	17.78	1327.14	8.52	8.08
min	-9.62	-7.74	-9.38	-13.78	97668.00	8.51	3.02
25%	-1.26	-0.54	-0.08	-3.42	100679.00	16.95	19.73
50%	0.14	-0.01	0.03	-0.41	101687.00	20.56	25.39
75%	3.09	0.49	0.14	0.35	102645.00	25.86	31.12
max	56.63	8.49	8.71	1036.75	104266.00	85.14	51.46
skew	1.72	1.48	0.49	36.73	-0.47	1.58	0.22
kurt	5.36	5.72	9.50	1856.62	-0.38	3.23	-0.12

4.1.4 Map of the study site

A map was created showing the tram routes, based on the information received from the route coordinators in Helsinki city using folium bubble map in Python. Figure 7 shows three tram lines or routes indicated by different colors that are run in the sampling site for collecting the data.

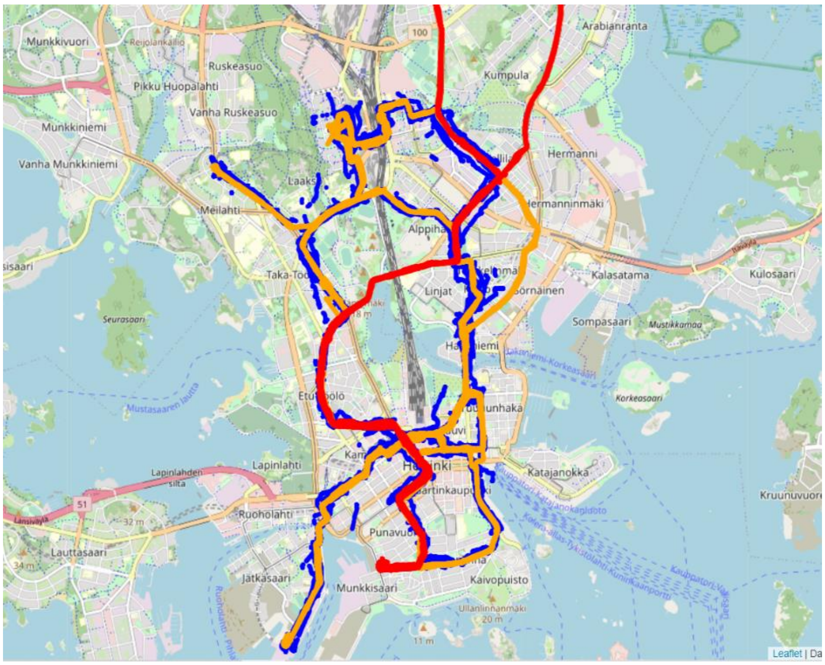


Figure 7. Routes of trams for collecting the data using the sensor.

4.1.5 Check for data stationarity

As the dataset is a timeseries dataset, the study first checks whether the dataset is stationary or not. Although for the neural network model, the data is not necessary to be stationary like the ARIMA model. However, the time series forecasting models perform well with stationary data during the process of training and fitting the model. A dataset is stationary if its statistical properties like mean, variance, and autocorrelation do not change over time. Although from the plots e.g pollutant trends (figure 5), it is visible that the pollutants variables are non-linearly distributed indicating the stationarity of the data. However, both visualizations, as well as Augmented Dickey Fuller (ADF) statistical test, were applied for checking the stationarity of the data. In the test, if the p-value is very small ($p < 0.05$), then the data set is stationary.

4.2 Data-preprocessing

Data-preprocessing is a vital step in any machine learning and deep learning process as it impacts the generalization ability of the learning algorithm. As the study utilizes neural networks, there will be many different layers in the execution including hidden layers. The objective here is to make the process simple by decreasing the processing time and also the lessen the number of attributes. This study performs different data preprocessing techniques such as handling data outside of detecting range (negative value), missing data imputation, outlier detection, data transformation, and feature engineering. The first two techniques will help to have more accurate and complete sets of data, while the third technique will provide more uniformly distributed data and minimize the data variability. Finally, the fifth step will be used to obtain a new dataset which will be more informative. The last step of data preprocessing is typically composed of feature extraction and feature selection. In the following, the author describes these steps in detail.

4.2.1 Negative value processing

First, this thesis handled the presence of negative values in the dataset that was countered due to the environmental factors altering the sensor. Table 4 shows the percentage of data below the detection limit (negative value) for different pollutant sources. The data below the detection limit (negative value) was replaced by half of the average detection limit values, following the method described by Polissar & Hopke (1998).

Table 4. The percentage of negative value and null values in the dataset

Pollutants	Negative value %	Missing value %
NO	0.41	0.16
NO_2	0.43	0.16
CO	0.54	0.18
O_3	0.35	0.16

4.2.2 Missing data imputation

In the dataset, most missing data is present in the air pollutant concentration such as NO, NO_2, CO, and O_3 (Table 4). Given that, the missing data percentage across different variables is less, it was decided to replace the missing data rather than discard. Backfilling technique was applied to handle the missing data for different concentrations. This method was adopted because it outperformed the other imputation method such as series mean and forward filling.

4.2.3 Outlier detection and replacement

Outliers if present in the dataset, can lead to many errors which consequently can bring down the accuracy of the implemented models. This thesis applied both quantitative and visual techniques for detecting outliers. An irregular behavior was observed in the CO data, as shown in Figure 5, where before the sensor start reading normal data, the starting level was unexpectedly higher approximately 1000 ppm for one device and 300 ppm for other devices. These were observed during turning on the power cycle at the tram for that sensor. As these observations are outliers, we discarded those data for CO concentrations. Additionally, the process of finding outliers was carried out from the summary statistics on the variables as presented in table 3. The table illustrates the irregular pattern for CO indicated by minimum and maximum values of -13 and 1036 respectively, which are incorrect. As most machine learning algorithms assume that data follow a normal (or Gaussian) distribution, the author also applied the skewness value of the features in detecting the presence of extreme value. Based on the skewness of the variables which should be between -1 and +1, the author determined the presence of extreme values by observing any major deviation from the range. The study has also detected the outliers visually by using a boxplot and skewed distribution pattern for each variable (Figure 8). The boxplot method revealed that most of our features have outliers and so, it was decided not to use the mean value for replacing the outliers. However, the variables/features are characterized as having a skewed distribution pattern as in figure 8, the study thus opted to use the median value instead of the mean value for a good replacement of the outliers. All the values above the 95th percentile in the data were replaced by the median value as outlier treatment.

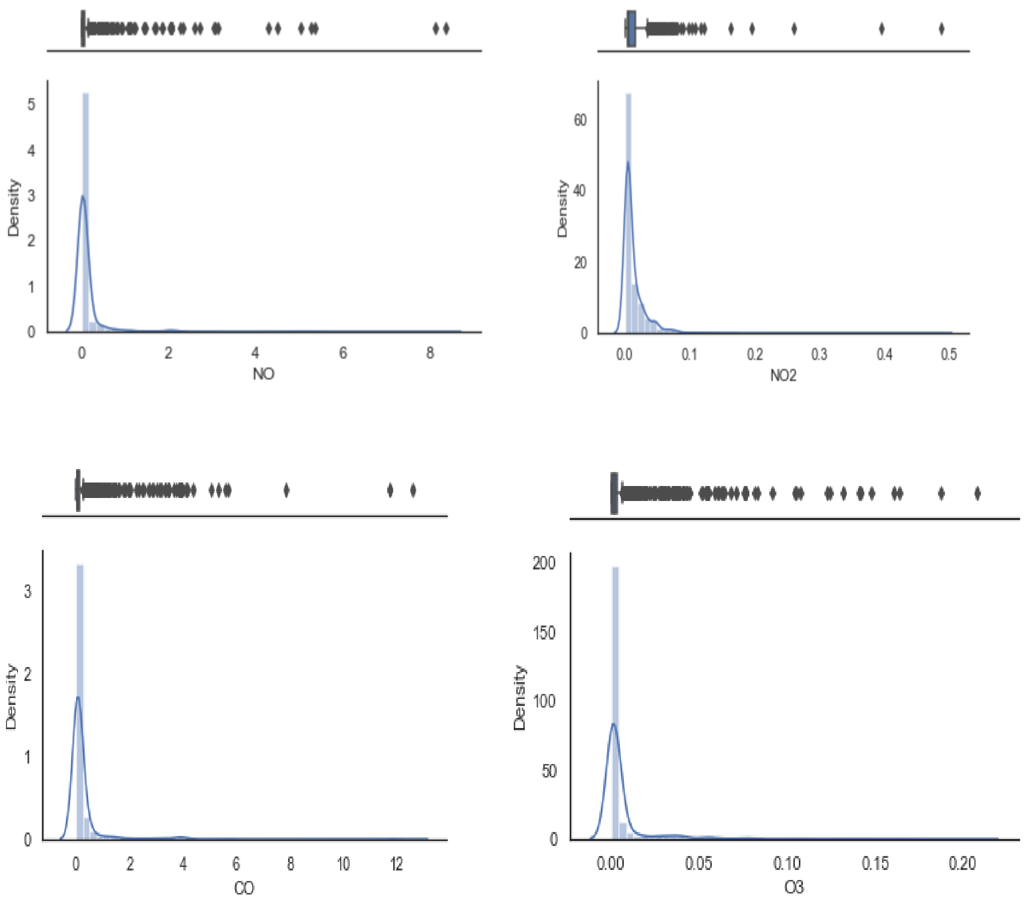


Figure 8. The boxplot and distribution plot of pollutants for outlier detection.

4.2.4 Data transformation

In neural networks, feature scaling can be done by different methods. This thesis has used the MinMaxScaler method of the Sklearn package in Python to normalize the data within a particular range. Because the neural network method utilizes the activation function such as ReLU and feature scaling or data transformation will help to reduce the training time by assisting the activation functions of the network. The activation function works better if the values are above 0 to avoid vanishing gradients, and below 1 to avoid exploding gradients. In general, the min-max normalization function scales the feature values within the range 0 and 1. The scaling was performed on the training dataset after splitting, and the same scaling was reused to scale the testing set for validation.

4.2.5 Feature Engineering

4.2.5.1 Temporal feature engineering

As the dataset contain the DateTime component, the author used this component to obtain new temporal features. This can help us in providing seasonality or a particular period of information. Using the advantage of DateTime type of variable, the author extracted different features namely hour of the day (0-23), day of the week (1-7), day of the month (1-30), month number (1-12) and weekend added as a Boolean feature. This study also added another variable named daypart to see how different parts of the day correlate with the dependent variable. Four daypart types were created such as morning (4-9), noon (10-15), evening (16-21), and night (22-3). Finally, the author visualizes the data after feature engineering to see how well the new features separate the data for inference (Figure 9).

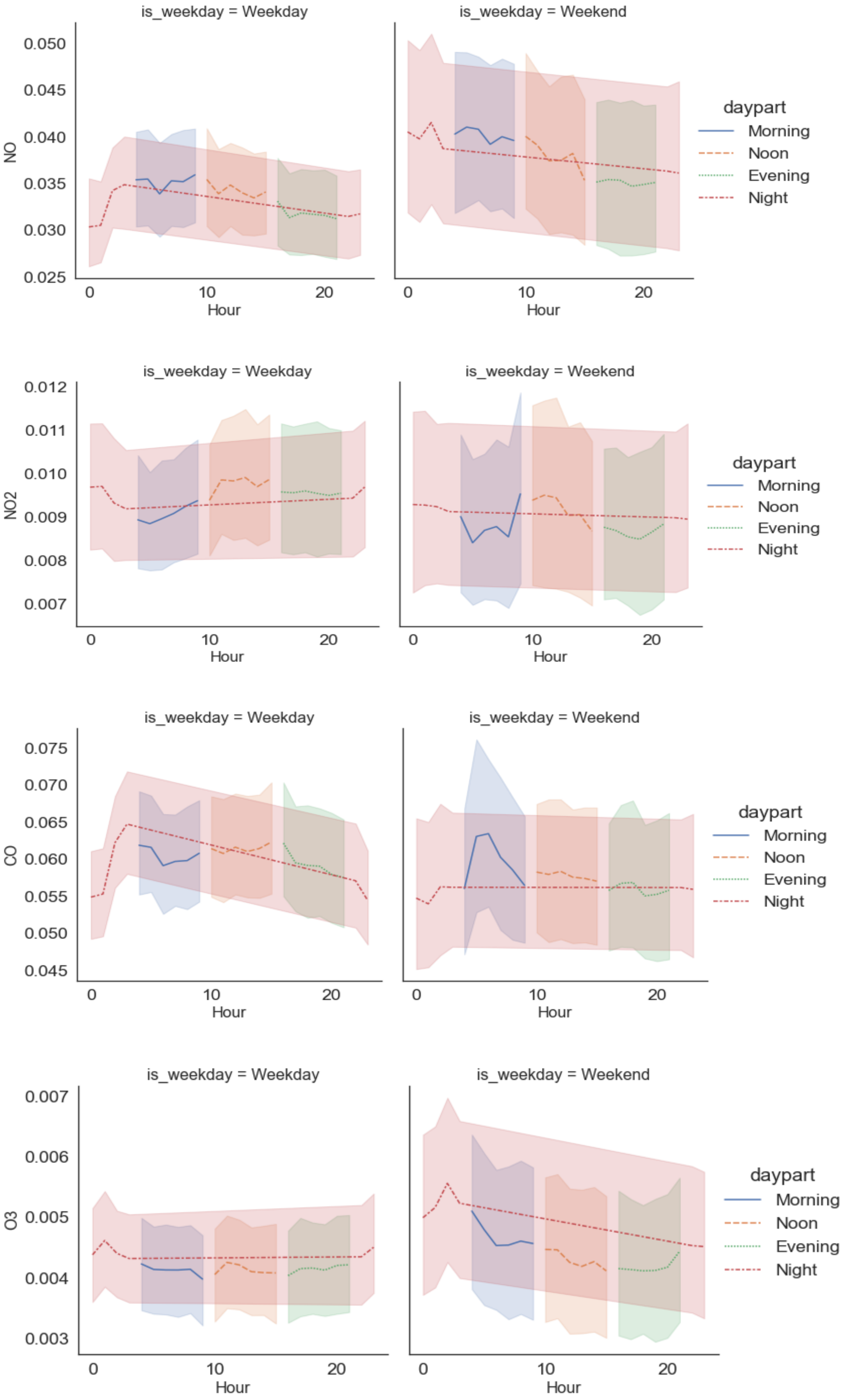
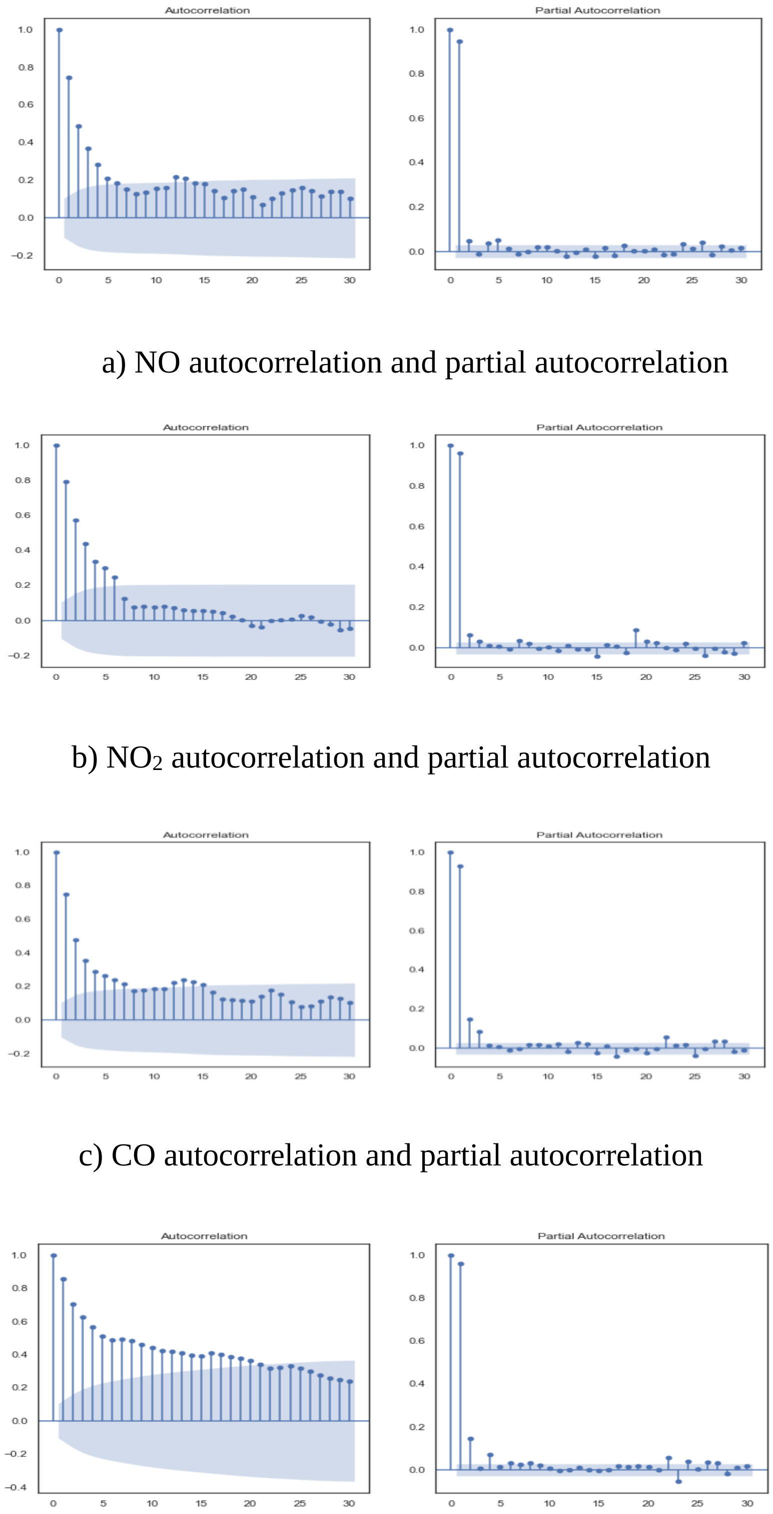


Figure 9. Temporal features of the pollutants grouped by day of the week and duration of the day.

4.2.5.2 Statistical feature engineering

This study created more features by applying a set of mathematical functions to the time series data. For example, in time-series data, lags are considered a backshift in the series and are used to measure the important phenomena, called series autocorrelation. This study defined the appropriate time lags or the number of lag variables per pollutant based on autocorrelation function (ACF) and partial autocorrelation function (PACF). ACF is the correlation between observations that are *n* time periods apart. While PACF is the correlation between series values that are *n* intervals apart, accounting for the values of the intervals between. ACF function was used to determine the lag number for moving average, moving minimum, moving maximum, and moving standard deviation features. Since the air concentration data is highly variable which is due to the underlying complex process and dependency on the mereological parameters (Demuzere et al., 2009), the study also utilized rolling minimum, rolling maximum, and rolling deviation functions which will help to reflect the fluctuation in the data in the model training phase. While PACF function was used to determine the lag difference features. The author performed such exploratory analysis in determining lag numbers for the elimination of redundant features which is an important step in forecasting. In general, this procedure helps to improve the overall model accuracy and gives a better understanding of the underlying process (Ribeiro et al., 2011). In figure 10, it is observed that in the PACF plot after 3 or 4 observation there is no correlation (assuming the confidence interval of 80 %) and only the first three or four lags were considered as relevant for all the pollutant series. While in the ACF plot it is observed that after 5 observations there is no correlation and only a 5-lag time window was considered for obtaining a rolling mean for all pollutant series except O_3 where the lags were determined as 10.

Finally, after the feature extraction phase, the complete dataset contains 36 features. In particular, during the feature extraction phase, we created the following variables: 4 lag variables for the pollutant concentration, 4 rolling mean, 4 rolling max, 4 rolling min, and 4 rolling standard deviation variables for each pollutant concentration, one variable for the hour of the day, one variable for the day of the week, one Boolean variable for the weekend, one variable for season and variables for the day.



a) NO autocorrelation and partial autocorrelation

b) NO_2 autocorrelation and partial autocorrelation

c) CO autocorrelation and partial autocorrelation

d) O_3 autocorrelation and partial autocorrelation

Figure 10. ACF and PACF plots for NO, NO2, CO, and O3 in Helsinki city.

4.2.6 Feature selection

After the successful feature engineering of data, the author prepared the data for feature selection from the 36 features. The feature selection process helps to reduce dataset dimensionality and eliminate the presence of collinearity. As the study aimed at predicting air pollutants with the help of several meteorological attributes and given that air pollutant concentration depends on meteorological factors and local topography (Dominick et al., 2012) and in particular, the meteorological condition can impact the air pollutant concentration through complex interactions between various processes such as emission, transportation, transformation and dispersion (Demuzere et al., 2009), this thesis kept all variables related to meteorological conditions in the dataset. The Pearson correlation-based feature selection was used as suggested in (Zhao & Magoul´es, 2011) to select all the other features. For this, a heatmap graph was created to show the existences of collinearity between the features (Figure 11). After careful observations, the less important features which show high correlation (correlation among independent features) were removed in the process of feature selection. For this, a function was created to determine the high correlation above the 0.8 threshold level.

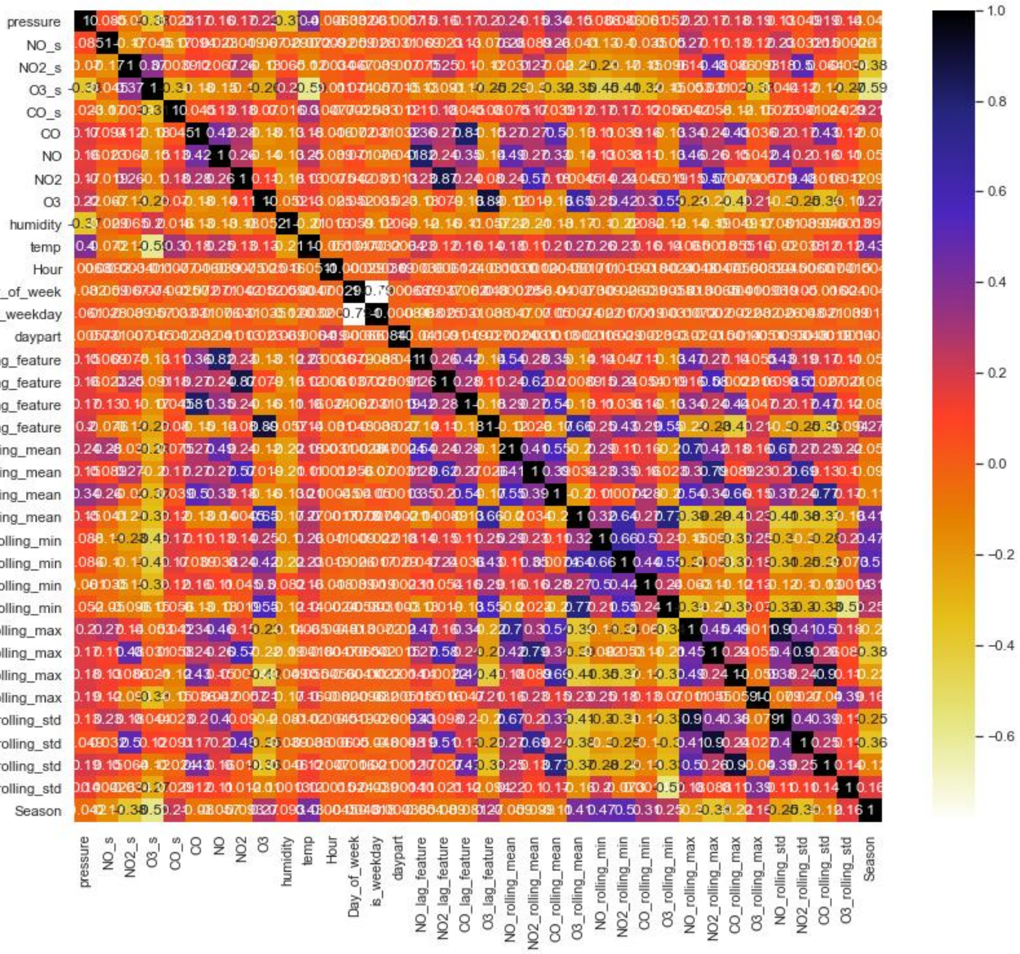


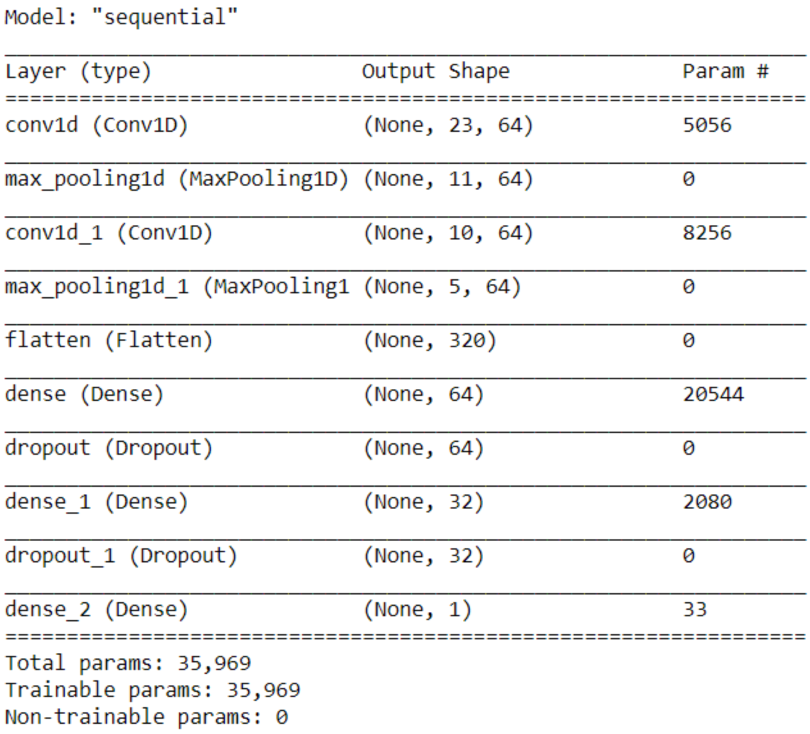
Figure 11. Pearson's correlation heatmap with correlation coefficients of the independent and dependent features.

4.3 Implementation of deep learning algorithms

4.3.1 Convolutional Neural Networks (CNN)

A one-dimensional CNN model was applied in this study. A one-dimensional CNN is a CNN model that has a convolutional hidden layer that operates over a 1D sequence and in this study the time series data is 1D sequence data. The CNN does not view the data as having time steps, instead, it is treated as a sequence over which convolutional read operations are performed, as a one-dimensional image.

To start with, this thesis first instantiated the Sequential class which is a sequential model. This study defined the convolutional layer 64 filter maps and a kernel size of 2 and then added a pooling layer whose job is to distill the output of the convolutional layer to the most salient elements. The study has added more layers into the model for learning more complex features to increase the accuracy and added a dropout layer to avoid overfitting. The convolutional and pooling layers are followed by a dense fully connected layer. A flatten layer is used between the convolutional layer and the dense layer to reduce the feature maps to a single one-dimensional vector. The model was fit using efficient Adam or RMSProp versions of stochastic gradient descent with a rectified linear unit (Relu) activation function and optimized using mean squared error or 'mse' loss function. The neural network was iterated for 20-40 epochs (vary on pollutants). Figure 12 below shows all the parameters for the implementation of the CNN algorithm.

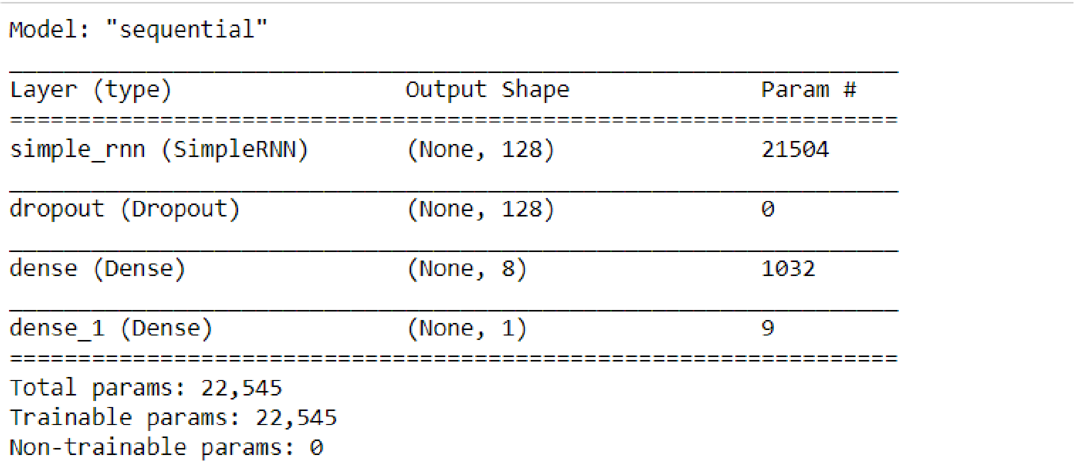


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv1d (Conv1D)              (None, 23, 64)            5056      
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 11, 64)            0         
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 10, 64)            8256      
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 5, 64)             0         
_________________________________________________________________
flatten (Flatten)            (None, 320)               0         
_________________________________________________________________
dense (Dense)                (None, 64)                20544     
_________________________________________________________________
dropout (Dropout)            (None, 64)                0         
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2080      
_________________________________________________________________
dropout_1 (Dropout)          (None, 32)                0         
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 33        
=================================================================
Total params: 35,969
Trainable params: 35,969
Non-trainable params: 0
```

Figure 12. Implementation of CNN parameters output.

4.3.2 Recurrent Neural Network (RNN)

The RNN model is also a sequential model. In the first step, this study instantiated the Sequential class which is the sequential model. Then, added the RNN layer with 128 neurons or nodes followed by adding dropout layer to the model to avoid over-fitting, which is a phenomenon where the machine learning model performs better on the training data compared to the test data. To make the model more robust a dense layer was added at the end of the model. Finally, the study complied with the RNN model before train it on training data. Mean squared error was used as a loss function and an optimizer which is Adam / RMSProp optimizer was used to reduce the loss or optimize the algorithm. Figure 13 below shows the implementation of RNN architecture.

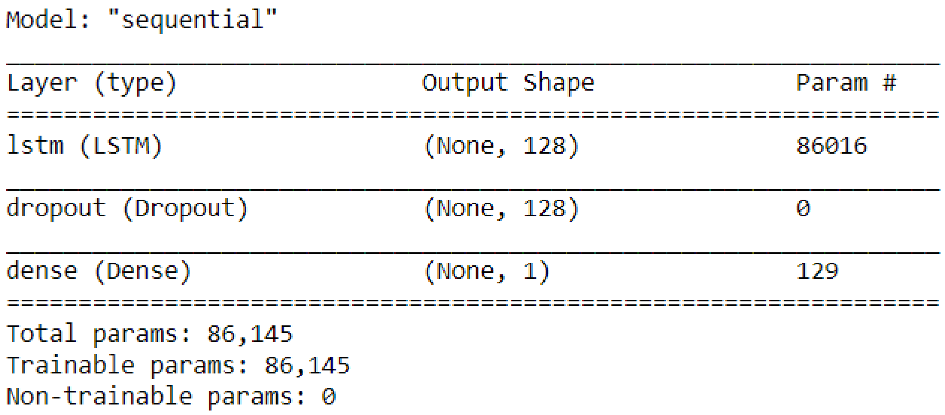


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
simple_rnn (SimpleRNN)       (None, 128)               21504     
_________________________________________________________________
dropout (Dropout)            (None, 128)               0         
_________________________________________________________________
dense (Dense)                (None, 8)                 1032      
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 9         
=================================================================
Total params: 22,545
Trainable params: 22,545
Non-trainable params: 0
```

Figure 13. Implementation of RNN parameters output.

4.3.3 Long Short-term Memory (LSTM)

The long short-term memory known as LSTM is a type of RNN architecture. The difference between RNN and LSTM is that LSTM includes a memory cell that can maintain information for long periods by using a set of gates such as input, output, and forget gate (for details see in chapter 2). As in process of implementation, the author added the LSTM layer to the model which has 128 neurons. It was also specified that the layer will have a Relu activation function which helps the network learn from non-linearities, next added a dropout layer of 0.2-0.3 (depends on the optimization) and this regularize the network by turning off 20-30 % of the neurons in the previous layer. This eventually prevents overfitting which occurs when models become so powerful that they represent the random noise in the data, in addition to the true signal. Regularization is especially important for neural networks because of the millions of parameters that they can handle. Lastly, the author added a linear output layer (dense layer). The implementation of the LSTM model architecture is shown in Figure 14.

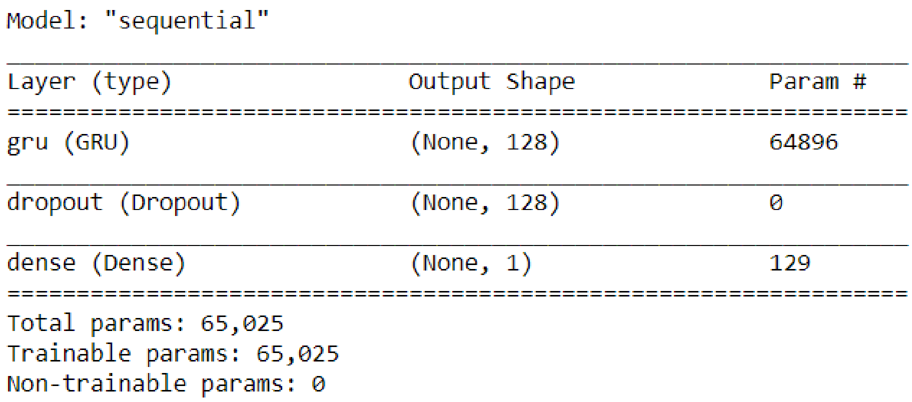


```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm (LSTM)                  (None, 128)               86016     
_________________________________________________________________
dropout (Dropout)            (None, 128)               0         
_________________________________________________________________
dense (Dense)                (None, 1)                 129       
=================================================================
Total params: 86,145
Trainable params: 86,145
Non-trainable params: 0
```

Figure 14. Implementation of LSTM parameters output.

4.3.4 Gated Recurrent Unit (GRU)

The GRU, known as the Gated Recurrent Unit is an RNN architecture, which is similar to LSTM units. The GRU comprises the reset and update gates instead of input, output, and forget gate of the LSTM. The implementation plan of the GRU model is similar to the LSTM model as mentioned above, the GRU layer was just added instead of the LSTM layer. Figure 15 shows the architecture of GRU algorithms.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
gru (GRU)                   (None, 128)               64896

_________________________________________________________________
dropout (Dropout)           (None, 128)               0

_________________________________________________________________
dense (Dense)               (None, 1)                 129
=================================================================
Total params: 65,025
Trainable params: 65,025
Non-trainable params: 0
```

Figure 15. Implementation of GRU parameters output.

Before model training, the dataset was split into two sets, the first set is to train the data (80% of data) and the second set is to validate the data (20% of data). This study applied a thorough optimization/hyperparameter tuning using GridsearchCV to ensure that every model's performance is optimal in case of 24-hour predictions and the same parameters were used for 48 hours prediction. Different parameters were used such as increasing the number of epochs in case of no flatlined loss function, adjusting the batch size, adding a dropout layer to avoid overfitting, different optimizers for yielding minimum error with fewer epoch values. The specification of the hyperparameter is presented in Table 5. All the models were compiled and fit the training set, using the test set as validation. After running the model, we look for the similarity between the training and loss functions which shows that the model will generalize well to a new dataset.

Table 5. Hyperparameter optimization for the neural networks. Optimizer A indicates Adam and R indicate RMSProp optimizer

Parameters	RNN	LSTM	GRU	CNN
	NO/NO$_2$/CO/O$_3$	NO/NO$_2$/CO/O$_3$	NO/NO$_2$/CO/O$_3$	NO/NO$_2$/CO/O$_3$
Batches	16/16/16/16	32/16/16/16	16/16/32/32	16/64/32/16
Epochs	30/20/30/30	20/30/30/30	20/20/30/30	40/40/30/40
Dropout rate	0.2/0.3/0.3/0.2	0.2/0.2/0.2/0.2	0.2/0.2/0.3/0.2	0.3/0.2/0.3/0.2
Optimizer	R/A/A/R	R/R/A/R	A/R/A/R	A/R/R/R

4.4 Index of performance

To evaluate the model prediction error rates and model performance (goodness of the results) of the proposed methods, four measures including coefficient of determination (r^2), the mean square error (MSE), root mean square error (RMSE), and mean absolute error (MAE) were used in this study. These indicators can be formulated as follows:

$$r^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

where n is the number of sampled data in the test set, $\hat{y}_i$ is the observed value of i sample, and y_i is the predicted value of i sample and $\bar{y}$ is the mean of observed true data.

The coefficient of determination known as r^2 value was chosen to see the goodness of fit of regression models. The coefficient of determination value does not consider the over-fitting issue as it shows to what extent the variance of one variable explains the variance of the other variable and a value near to 1 indicates a better model. The author chose MSE as it is one of the most commonly used metrics and is most suitable for the dataset that contains a lot of noise such as outliers or unexpected values. A lower MSE value indicates a better model. RMSE was chosen as the evaluation metric due to its ability to penalize large errors in the dataset by assigning a higher weight to larger errors. Here, the errors are squared before they are averaged. RMSE also gives a higher error metric for large outliers, which are desirable in the case of air quality concentration forecasting, and as such it is well suited for the comparison of different model's performance. The lower the RMSE value the better the model performance is. The author also used another popular error metric MAE which does not penalize large errors and reflect possible outliers in the dataset in the same way as RMSE does (Chai et al., 2014). The lower the MAE value, the better is the model's performance.

4.5 Software and Libraries

For the successful execution of this research project Python programming language has been used by launching Spyder (Python 3.7) Notebook in Anaconda Navigator (Anaconda3). Several Python (3.7) libraries had been used in this project such as Pandas, Numpy, Sklearn, Seaborn, Matplotlib, os, and DateTime. Python folium library was also used for producing the Helsinki city map indicating the sampling routes. This thesis research has applied Tensor Flow and Keras of python libraries for the development of the prediction algorithms. They are largely accepted in the research community as tools to implement robust algorithms for data analysis using ANN and the deep learning method.

5 RESULTS AND DISCUSSION

This chapter describes the experimental setup with dataset specification according to the objectives. Then the experimental results are presented both visually and textually. Following the results are discussed and validated with dataset specification and objectives.

5.1 Results

5.1.1 Feature correlation experiment and pollutant's hotspot

To find the answer to research question 2 that was discovering the correlation between dependent variables particularly meteorological components and concentration level of pollutants (NO, NO_2, O_3, CO) that affect the air quality, a pair plot analysis was performed. Figure 16 shows the distribution of the single variable and the relationship between the variables. The figure describes that the pollutant variables are positively correlated, for example, higher NO tends to produce higher NO2, although it does not prove that one causes the other. The correlation between the independent variable e.g pollutants and the dependent variables e.g meteorological variables are also evident from the figure. For example, high temperature ensures higher pollutants concentration for NO, NO_2, and CO, while high humidity gives lower pollutant concentration for NO, NO_2, CO, and O3. Given that air pollutant concentrations are highly dependent on the meteorological variable particularly local temperature, this study further explored the relationship between pollutants and temperature using day of the week and duration of day variables created from DateTime index feature engineering (Figure 17).

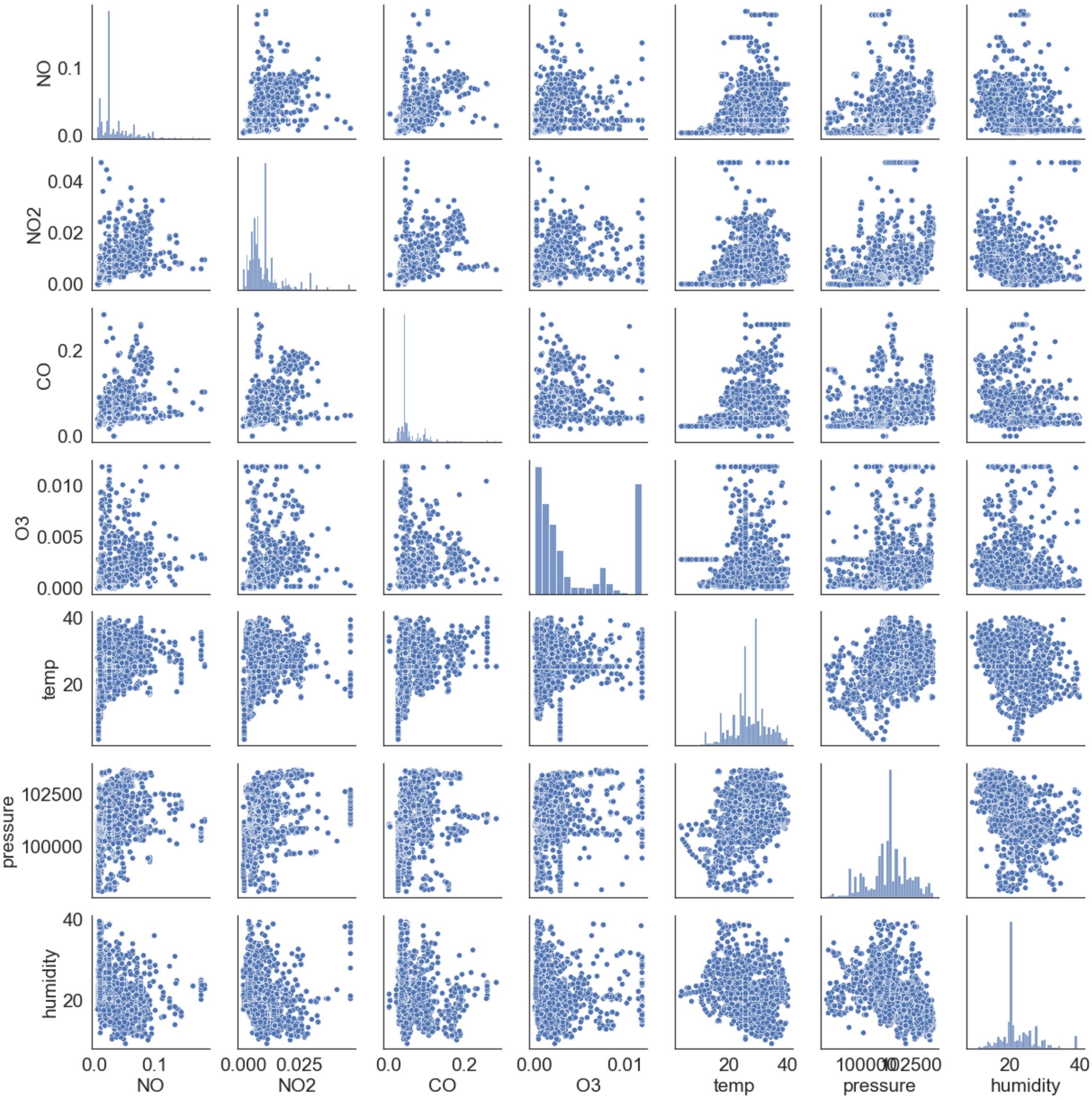


Figure 16. Pair plot of dependent variables (temperature, humidity, and air pressure) and the pollutants (NO, NO2, CO, and O3).

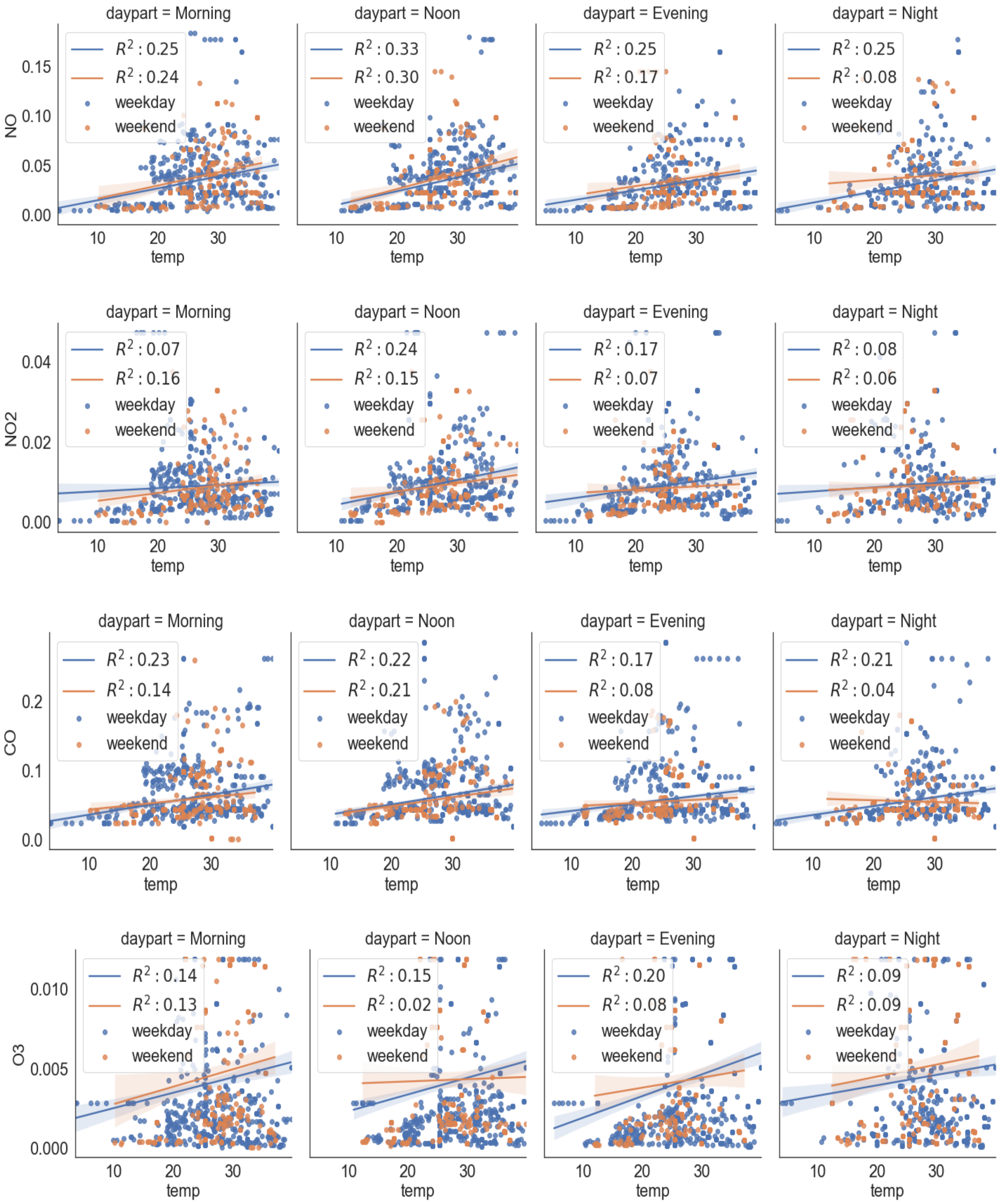
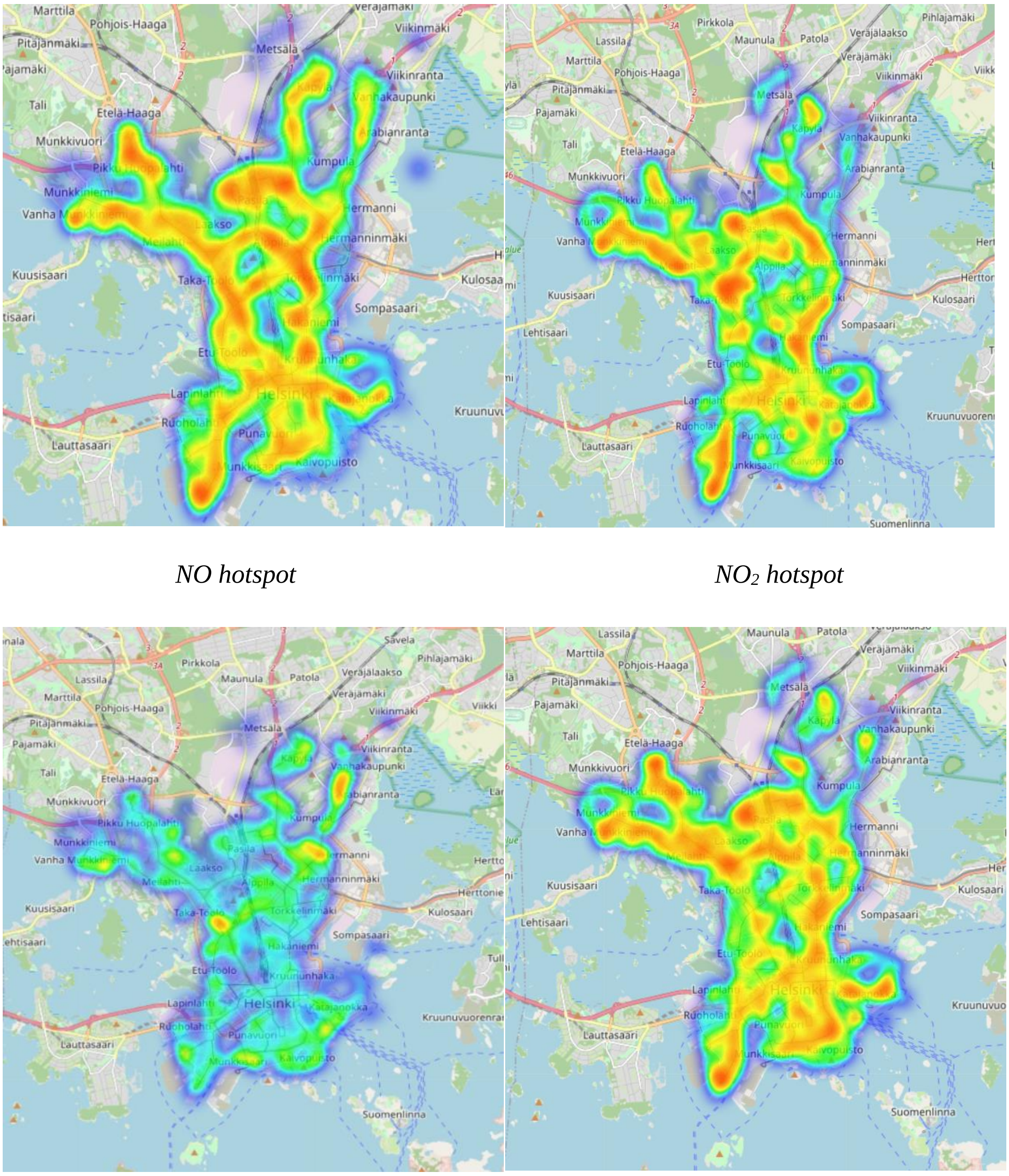


Figure 17. The correlation between temperature and pollutants by different dayparts and days of the week.

Figure 17 observes how temperature can be related to the pollutant concentrations by different parts of the day as well as days of the week. High correlation, r^2 value ranging from 0.25 to 0.33 was observed during the weekday for almost all the pollutants. The results for daypart indicate that noon daypart has a higher correlation for all the pollutants having r^2 value ranging between 0.22-0.33 except for O_3 (r^2 value of 0.15). The analysis also indicates that during evening and night along with the weekend, there seems no correlation between temperature and different pollutants except NO concentration. This might be because during night or evening or even weekend the frequency of public transport and other modes of transport is less. Among the pollutants, NO concentration shows higher a correlation respective to daypart and weekday compared to other pollutants.

The author created a heatmap with the data collected by the sensor on the trams using a folium map. The heatmap does not show the air pollutant concentration but indicates which locations have the most data points. The red areas have relatively more data points, in other words, have high density of the pollutant concentrations and the lighter colors have less. It is possible to have multiple data points on a single location for example when two trams cross each other. In that situation, the data points may overlap with each other and give the location a higher value. Locations that have more data points appear red on the map and lighter color has fewer data points. By using a heatmap, it is easy to understand which locations of Helsinki city are mostly covered by the sensor on the tram and thus indicate the hotspot for pollutant sources (Figure 18). From the map, it is clear that the presence of NO and O_3 is stronger in the Helsinki city location. The CO concentration was less significant in the location of Helsinki city. It is also evident that the concentration of pollutants is prevalent in the inner-city part compared to the city periphery.



NO hotspot

NO_2 hotspot

CO hotspot

O_3 hotspot

Figure 18. Heatmap of pollutant concentrations (NO, NO2, CO, and O3) in the study area (Helsinki city).

5.1.2 Feature influence experiment in model's performance

To answer research question 3 that was to investigate the highest impact of features/variables on the machine learning algorithm's ability to accurately perform prediction, two sub-experiments were conducted. The experiments use the LSTM model with different feature combinations such as temporal (T) and statistical (C) feature engineering with the two base features Spatial (S) and Meteorological (M). Both the sub-experiments use the same test framework to produce the prediction results for different pollutants at 24-hour timesteps. The first sub-experiment is about validating the temporal difference of the features. The feature set was generated by extending the two natural features (M, S) with the temporal feature engineering (T) such as T, TM, TS. The second sub-experiment applies statistical feature engineering to find out the feature influence on the models. The same framework was applied as the previous sub-experiment except for the features. The features applied in this experiment are C, CM, CS.

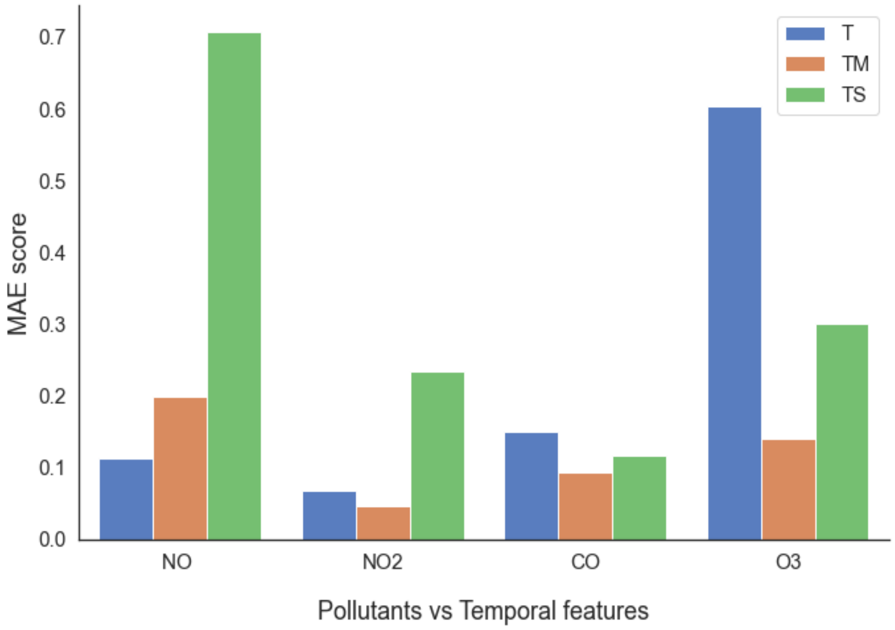


Figure 19. Influence of temporal (T) features with window size 24.

Figure 19 shows the impact of the temporal feature engineering on the meteorological (TM) and spatial (TS) features. The single feature T consists of all temporal feature engineering with the pollutant or independent variables and is used as a reference for the performance gains for the meteorological and spatial features.

For NO, the historical values of meteorological and temporal features gave no performance gain. However, for other pollutants, the model showed a good relationship between the historical and future observations of the meteorological and temporal features. For example, NO_2 has a performance gain of 30 %, CO an increase of 38 %, and O_3 an increase of 78 %.

In the case of spatial feature extension to the temporal feature engineering, NO gave no performance gain at all, like meteorological feature. The NO_2 also followed the same pattern as NO, giving no performance increase. However, CO and O_3 gave 23 % and 50 % performance gain respectively, indicating a different spatial pattern of CO and O_3 compared to the others.

Overall, the historical meteorological value has the best performance gain across predictions for all pollutants with a total average of 24 % increase of MAE, with O_3 has the best increase of 34 %.

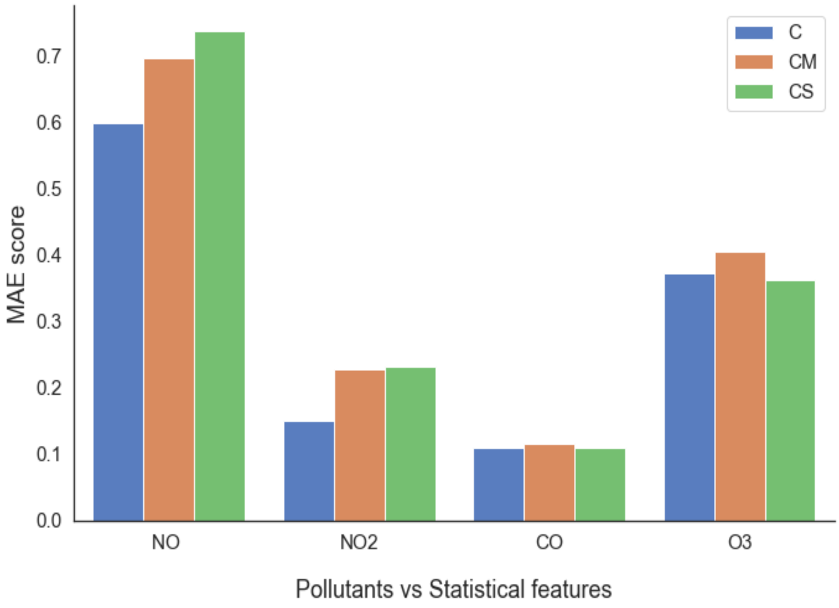


Figure 20. Influence of statistical (C) features with window size 24.

Figure 20 shows the impact of statistical feature engineering on the features meteorological (CM) and spatial (CS). The feature C is consisted of all statistical feature engineering with the pollutants and is used as a reference for the performance gains or loss for the meteorological and spatial features.

For all the pollutant sources, the historical values of meteorology and statistical features gave no performance gain or increase. The model is not able to find a good relationship between the past and future observations of the meteorological and spatial features. The results showed a more than 50% loss in performance for NO_2. The influence of statistical extension to the spatial feature (CS) does neither give any performance increase except for O_3 pollutants, where a little performance gain of 3% was observed.

The overall model performance for the experiments using temporal feature engineering is higher than the extension of the statistical technique. The temporal feature engineering has a total average MAE of 0.11 performance increase than the statistical MAE score. It indicates that the model can learn from the temporal features efficiently than the statistical feature values.

5.1.3 Model performance experiment

To evaluate the final research question, a range of different deep neural network algorithms, each with its unique trait, was applied to find the accurate deep learning method for predicting air quality in Helsinki. In this experiment, the comparison of different deep learning models shows the results of the model’s performance. The full extent of the feature set (MSTC) was applied for this experiment.

The general pattern of the results is presented in Table 6, with a separation of 24 and 48-hour prediction scores. The results show a dominant LSTM model with the overall best performance. For NO_2 and NO with 24-hour prediction, GRU outperforms in terms of r^2 score. However, GRU results for MSE, RMSE, and MAE falls shortly behind those of LSTM. Few models result in negative r^2 values and indicate that the models do not follow the trend of the data. With 48-hour predictions, RNN outperforms LSTM for CO and NO regarding MSE, RMSE, and MAE scores. Surprisingly, irrespective of pollutants the

GRU models do not perform well in fitting the general pattern and this might be because the model is limited to overfitting of large input dimension.

Table 6. Model's result with prediction error from the coefficient of determination (r^2), Mean Square Error (MSE), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) evaluation metrics

Model	NO				NO_2				CO				O_3			
	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE
CNN	0.593	0.032	0.180	0.124	-1.019	0.009	0.097	0.075	-0.001	0.054	0.232	0.147	0.310	0.069	0.263	0.199
RNN	-0.005	0.080	0.283	0.181	-0.641	0.008	0.088	0.073	-0.027	0.055	0.235	0.137	-0.011	0.101	0.318	0.240
LSTM	0.703	**0.024**	**0.154**	**0.090**	0.067	**0.005**	**0.071**	**0.056**	**0.575**	**0.023**	**0.151**	**0.096**	**0.652**	**0.035**	**0.187**	**0.114**
GRU	**0.728**	0.022	0.147	0.109	**0.209**	0.006	0.075	0.056	0.262	0.040	0.199	0.101	0.581	0.042	0.205	0.115

a) Prediction scores with window (timestep) size 24

Model	NO				NO_2				CO				O_3			
	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE
CNN	-0.025	0.087	0.294	0.208	-0.011	0.101	0.318	0.240	-0.138	0.064	0.252	0.136	-0.090	0.113	0.336	0.240
RNN	**0.007**	**0.084**	**0.290**	**0.206**	-0.957	0.009	0.097	0.084	-0.074	**0.056**	**0.237**	**0.138**	0.487	0.053	0.230	0.188
LSTM	-0.131	0.096	0.309	0.210	-0.493	**0.007**	**0.085**	**0.064**	-0.490	0.083	0.288	0.192	**0.495**	**0.052**	**0.229**	**0.145**
GRU	-0.973	0.167	0.408	0.310	-1.564	0.012	0.111	0.074	-1.471	0.138	0.372	0.277	0.404	0.062	0.248	0.172

b) Prediction scores with window (timestep) size 48

The study shows a better performance of LSTM for all pollutants in terms of MAE score followed by GRU (figure 21). LSTM and GRU have a similar score for NO_2 and O_3, but for NO_2 the performance is better for both models. RNN is predicting worse for NO and O_3 than the other pollutants, whereas for NO_2 and CO the performance quality is poor with the CNN model. Interestingly, CNN outperforms RNN for NO_2 and CO.

Overall, LSTM showed a performance gain of just 7 % compared to GRU. However, the performance gain of LSTM is prominent compared to CNN which is 53%, and RNN which is 77%.

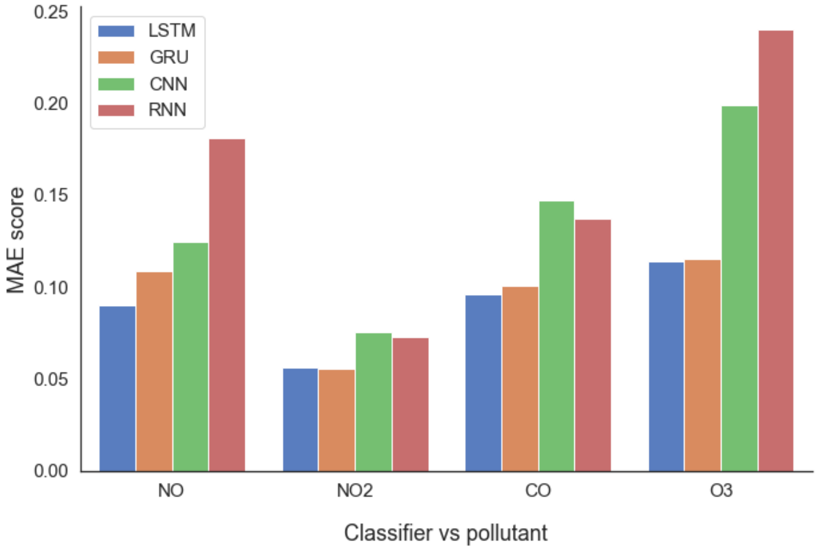


Figure 21. The performance of models in terms of MAE with different pollutants for a 24-hour window.

The study also observed different model performance when grouped by window horizon of 24 and 48-hour (Figure 22). LSTM achieved the best scores for both window horizons presented. In addition, for the 48-hour horizon RNN performs the same way as LSTM, while the GRU indicates poor performance for long-term predictions. Figure 23 displays the comparative prediction performance on the test dataset of each deep neural algorithm.

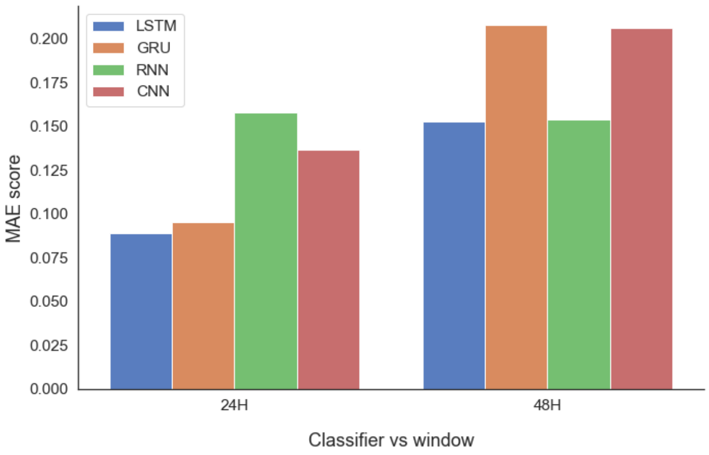
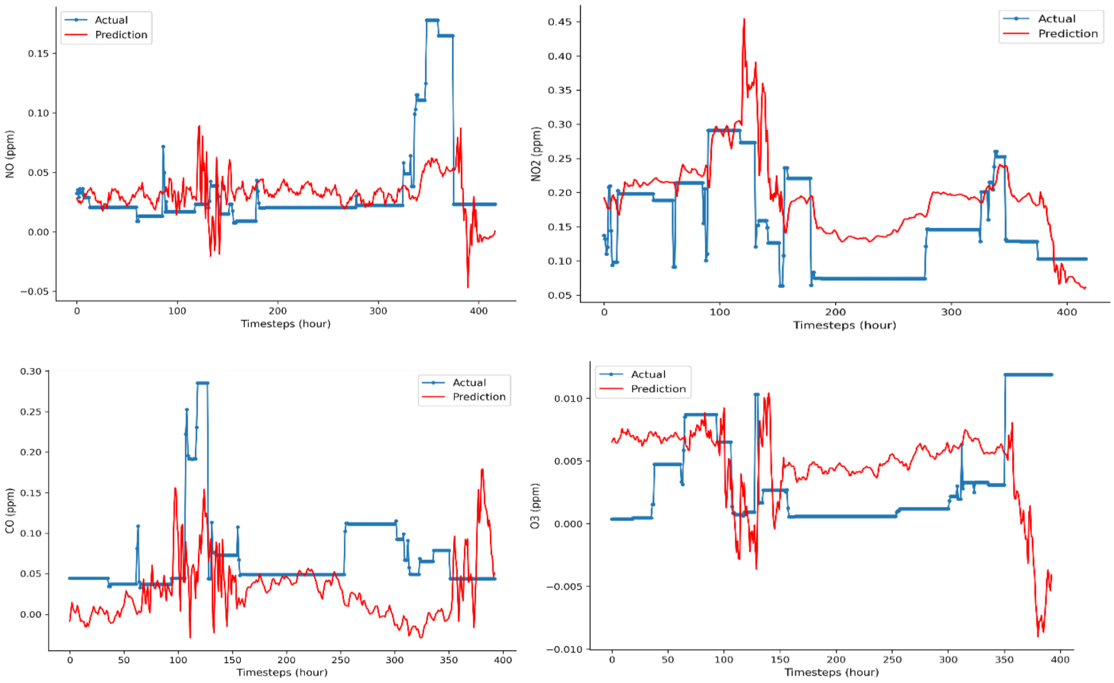
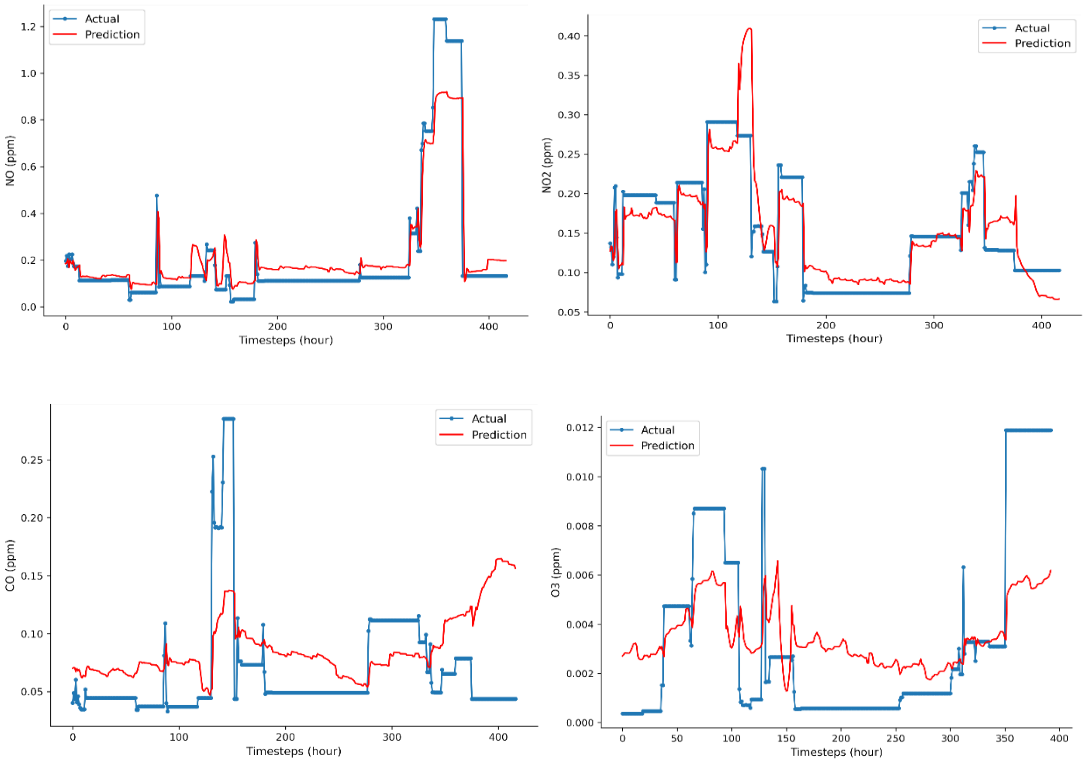


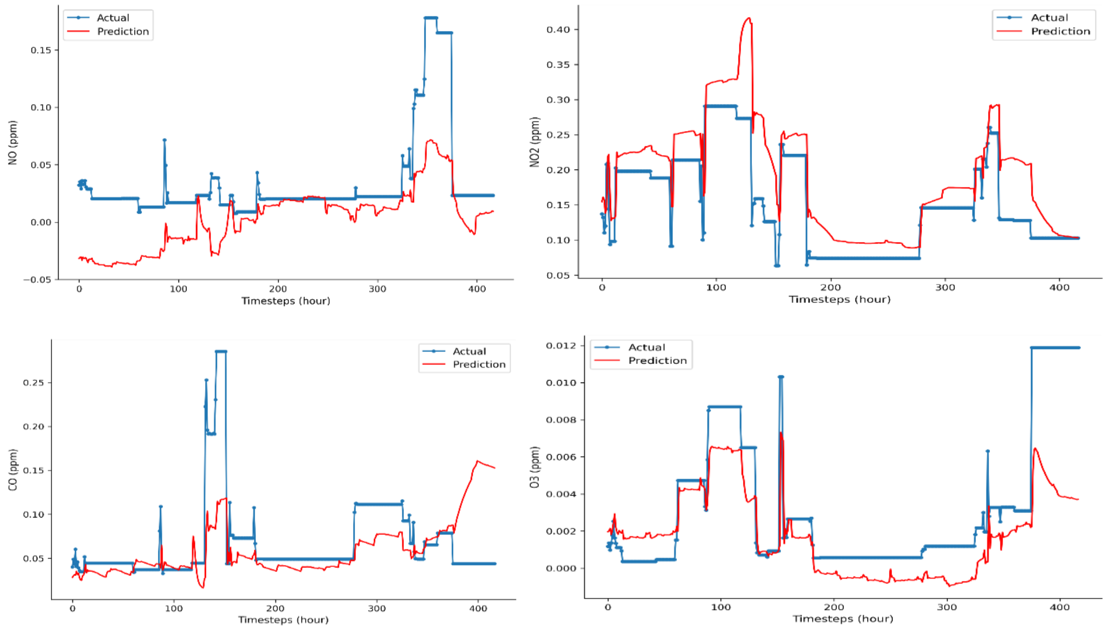
Figure 22. The performance of models in terms of MAE with 24 and 48-hour window horizons.



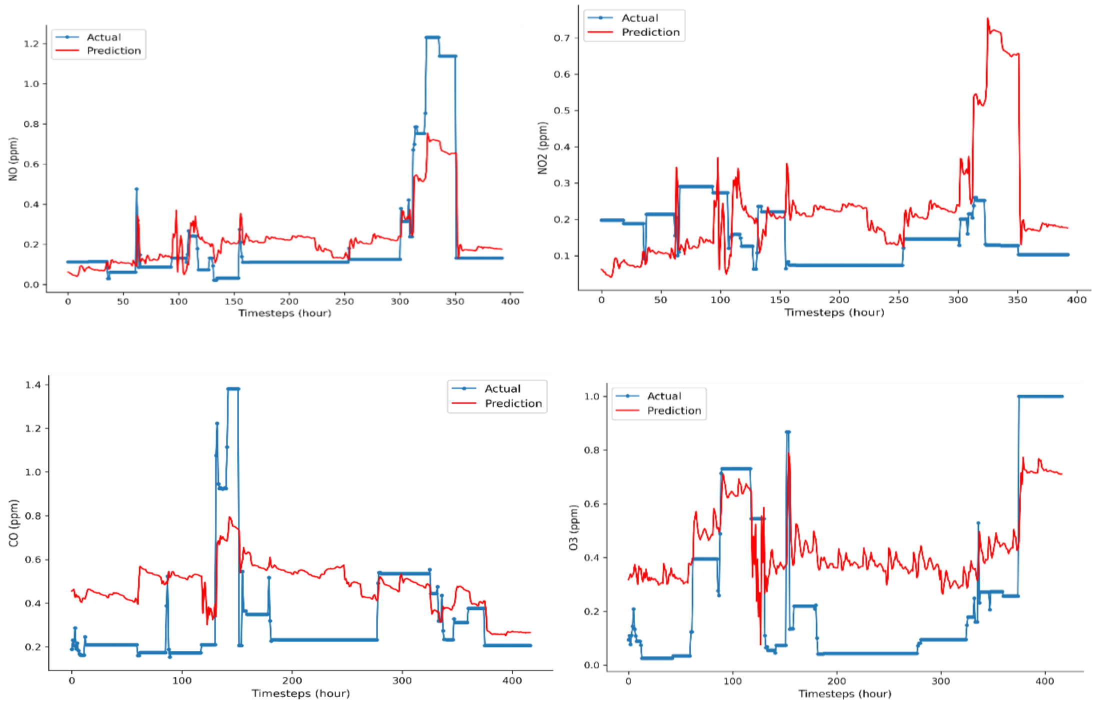
a) The forecasting results of CNN for different pollutants



b) Forecasting results of LSTM for different pollutants



c) Forecasting results of GRU for different pollutants



d) Forecasting results of RNN for different pollutants

Figure 23. The prediction results of the test data on NO, NO2, CO and O3 concentration for the proposed neural networks (based on the 24-hour window).

5.2 Discussion and Validation

Precise prediction of air quality in an urban area is a crucial problem because of the impact of air quality on people's everyday life. The study demonstrates the benefit of deep learning for short-term predictions of different air pollutants. Urban air quality prediction is a challenging task as it is influenced by many parameters. Meteorological information such as temperature, humidity, and air pressure are highly crucial to influence the air prediction compared to other dependent variables that have been studied, such as traffic information.

The dataset utilized in this thesis project has been constrained with limited data in terms of duration. This was because data collected under Horizon2020 MySmartLife project was a pilot project aiming to collect data using IoT of pollution sensors. There was no automation of the sensor for calibration of the gas tolerance in the sensor during the data collection period of the project. And as a substantial budget was involved in the calibration process, the project was constrained to avail short duration data. However, with the background compensation of data, that was done in this thesis project (details in method section), the data quality has been upgraded for the machine learning process.

As the data was limited to few months of a year, the seasonal variability was not possible to include in the model. However, in this study, it was observed that the model learned well with such a short duration of training data to predict the future. The study found tiny errors in the learning process of the model (see the results section) indicating the accuracy of the model.

No monitoring record using pollution sensors is ever complete. There will inevitably be periods of missing data which might be because of unable of auto-calibration, equipment failure, power cut, bias and drift. The data validity is never thus 100 percent valid in the case of environmental monitoring data. In addition, several environmental parameters with their underlying complexity make the data highly variable (Demuzere et al., 2009) even with the presence of outliers. In the process of quality control of the dataset, neither the outliers were removed nor dropped the missing value, rather the author dealt with this with appropriate techniques (see the method section) for data certainty and integrity.

Instrumental error or fault may occasionally produce large negative spikes in the data. Similarly, large positive spikes might be occurred due to the inadequate calibration process of the sensor, as seen in our dataset. As the process of quality data control procedure, this thesis carefully reviewed such abnormal phenomenon of the data to evaluate whether they are real or spurious. Unless there is a shred of good evidence to remove a value, it was left it in the dataset. The author did not remove all negative data from the dataset or replace the negative value with zero, as it will artificially increase the ambient concentrations. The author dealt with such data by replacing the negative value with the average positive value under a short period of window size (details in the method section). Thus, data integrity was ensured by several data pre-processing steps.

Moreover, the data was well represented as there have been 3 separate sensors in 3 separate tram routes in collecting the data. As Helsinki city is considered to be environment healthy, the coverage by tram network for data collection was proper in comparison to dense network cities which demand extensive coverage. The project produced one spatial dataset of air pollutants using Mäkelänkatu meteorological station and this thesis utilized that data in the machine learning process.

Numerous machine learning techniques have been applied in predicting air quality ranging from statistical approaches to recent advances in machine learning. Deep neural networks had been used in the recent literature with the strong prediction for air quality prediction problems as it involves several networks to address the hidden relation of data complexity. Of the neural network, deep feedforward and deep back forward which is recurrent neural network is the most common architecture in the literature. The specialized recurrent neural network such as LSTM and GRU has been shown to perform better in time series analysis. On the other hand, CNN with a one-dimensional layer has been shown to perform better in air quality prediction, although CNN is more accurate in image data analysis.

In the literature review study, the author observed that there is no standard or unified framework for testing the model for air quality prediction. Rather, the variation of model accuracy is highly representative of the dataset utilized in the research. Besides that, different pollutants to focus on and validation methods are also responsible for the lack of a unified framework in air quality prediction.

Based on the experiment of determining the feature influence in machine learning model accuracy, the study observed that an extended feature engineering approach greatly improved the results of air quality prediction. Particularly, temporal feature engineering has shown great improvement compared to statistical feature engineering. The author hypothesized that the inclusion of statistical features will include a relation of the past and thus it will be easier to learn in the model. But the model did not learn well with statistical features which might be due to too constant periodic data and the past was not able to give any clues to the future data. On the other hand, the wider temporal dependency of the dataset is logical to the time series pattern of the data, where the past was able to give information to the future data.

This thesis used multiple metrics to calculate the overall error to evaluate the model as in the literature there is no universal metrics of such prediction accuracy. For calculating the erroneous of the model, according to Chai et al. (2014), the MAE is believed to be superior as it does not penalize large errors and reflect possible outliers in the dataset in the same way as RMSE does, which the author believe is in favor of air quality validation.

The model performance experiment shows that LSTM is the best of all models for all the pollutants. In addition, the performance score of GRU was close to LSTM for few pollutants. These results are in accordance with the earlier studies (Athira et al. 2018; Wen et al. 2019). In general, the models show no definitive convergence in terms of evaluation measures. Some model shows good performance for one particular dependent variable, while few models show poor performance (e.g negative r^2) for that variable. Thus, it was hard to find the convergence of the models. The better accuracy for LSTM and GRU is likely because these models have a special unit called memory cell in addition to a standard unit. Unlike, standard RNN architecture which suffers from vanishing and exploding gradient problem (a problem that requires learning long-term temporal dependencies because of exponential decay of loss over time), LSTM deal with these problems by introducing new gates, such as input, output and forget gates, which allow for better control over the gradient flow and enable better preservation of long-term dependencies. Similarly, GRU uses a similar structure but simplified structure compared to LSTM such as reset and update gates. In nutshell, LSTM and GRU models use gating mechanisms to control the flow of both short- and long-term dependencies. LSTM has also the ability to update the data based on specific requirements which also provides the option to remove the outliers and thus improving the model performance.

Interestingly, this study observed that CNN outperformed the standard RNN in prediction quality for NO and O_3 pollutants. Although CNN is very popular with image and video data where data is correlated in space and time. Unlike RNN where a deep feed-backward technique is applied, in CNN deep feed-forward neural networks work in capturing future observations. In this study, the neighboring information of NO and O_3 pollutants was more relevant for CNN model prediction compared to past observations in RNN structure. In other words, the neighborhood information was well captured in the CNN structure for predicting NO and O_3 pollutants.

The research methods applied in this thesis are data-driven approaches, therefore data preprocessing and optimization steps might affect the outcome of the trained models and performance metrics. The air quality of the Helsinki region is considered on average healthy, numerous unpredictable functions such as extreme winter may deteriorate the air pollutant concentrations. Consequently, the outcome of this thesis might not give the same result if applied to a new dataset without being trained in the right circumstances. However, the dataset was collected using three sensors, where larger cities might require more sensors for substantial coverage and thus the right preprocessing and optimization steps as well.

6 CONCLUSIONS AND FUTURE WORKS

The overall aim of the project was to develop deep learning algorithms for predicting the air pollutant concentration on an hourly basis. This chapter draws final findings based on the experiments and results from the previous chapter. This chapter also presents the possibility of future works based on the limitations discovered.

6.1 Conclusions

In this research work, several deep neural network methods were proposed for the prediction of pollutants in Helsinki city. The neural networks can predict NO, NO_2, CO, and O_3 concentration of the next hour according to pollutant concentrations, temperature, humidity, and air pressure over the last 24 hours and 28 hours. In the experiments, training data was used for training the model and testing data that was unused in the training phase was used for the computation of r^2, MSE, RMSE, and MAE performance evaluations.

The study found that neural network algorithms are worthwhile to predict air quality. Through combining pollutant data with spatial data and with statistical-temporal feature engineering techniques, the study provides valuable information on the data used for deep learning models. Temporal feature engineering showed a good relationship between the historical and future observations in improving the model accuracy. The results showed that model performance is more influenced by meteorological features compared to the spatial features of the data. The selected methods show high performance to predict each pollutant such as NO, NO_2, CO, and O_3 separately with various forecasting horizons. The experimental results showed that the LSTM model outperformed every model and for all the pollutants separately, having lower MAE values of 0.09, 0.056, 0.096, and 0.114 for NO, NO_2, CO, and O_3 respectively. In addition, GRU model performance was very close to that of LSTM for all pollutants having MAE value ranged 0.056 – 0.115 across the pollutants. The study also compared the total average MAE of prediction for all the pollutants, the LSTM model was more accurate compared to all other models. The comparison showed that the proposed algorithms can predict with optimal accuracy between LSTM and GRU models. However, for a longer time horizon (48-hour), the best accuracy can be achieved between LSTM and RNN methods in predicting air pollutant concentrations. Of all the pollutant sources, the NO_2 concentration level was best predicted by the deep neural network models.

In this paper, the comparative performance of all the proposed neural network methods is quite good. Particularly, the CNN architecture showed that it can predict hourly concentrations with sufficient accuracy by modeling the relationship between pollutants and meteorological variables. Even for the longer timesteps, the performance of CNN levels with the performance of GRU which is the memory of time-sequence based. It indicates that CNNs can be trained to approximate highly non-linear functions to predict non-linear processes related to air quality data. Concerning the training time, CNN can generate results within less than a minute of initiating the model compared to LSTM or GRU model. Thus, given the computational efficiency of the CNN algorithm, the CNN model can supplement back feedforward networks such as RNN and GRU models to more rapidly and accurately prediction of air pollution concentrations.

6.2 Future works

This section presents the possible extension of the current research to improve the prediction accuracy and the solutions to the limitations that were revealed in this thesis.

6.2.1 Extensive dataset

Pollution sensors have been widely used in pollution-related projects to detect each pollutant separately. The data collected using Aeromon pollution sensor was a pilot project of Horizon 2020 mySMARTLife and thus, during the collection period, the sensor was unable to automated background compensation. This eventually resulted in data range out of actual detection concentration or below the detection range of air pollutants. This was one of the reasons for having a short period of data under Horizon 2020 mySMARTLife project. Extensive dataset or full data set for example one full year or several years of data would benefit the process of learning the model. The complete dataset would also reveal the seasonal phenomenon in the neural network learning step and reflect the actual scenario.

6.2.2 External sources

This thesis focuses on predicting the target pollutants in an urban area using meteorological and spatial features as dependable variables. However, as the urban area is characterized with the dominance of traffic all the time, it would be practical to include the traffic data. There is evidence that vehicle emissions are considered one of the primary sources of air pollution in cities and contribute largely to high NO_2 and CO levels (EEA, 2017), thus utilization of traffic volumes data would make the model training phase much stronger in prediction by drawing the complexity of the data.

6.2.3 Feature selection

Feature selection is an important step in any machine learning algorithm. However, increased feature space might overfit the model during the training process. To improve this, an optimal feature dimensionality reduction technique can be utilized for the model's predictions further. To mention, principal component analysis is a widely used technique for dealing with large feature space. It reduced the high space data into few sets of components which try to explain as much variance of the feature as possible.

6.2.4 Deployment of the model as part of Smart City Initiative

Although the deployment of the best machine learning model for Android software is out of the scope of this thesis, however, the work can be further extended in developing the user interface for the Android application. Creation of Android application is one of the central outcomes of any Smart City Initiative to provide the users with real-time pollution concentration for any location and hourly forecasted pollution concentration in order to navigate the less polluted route.

REFERENCES

Akimoto, H., 2003, Global air quality and pollution, *Science* 302 (5651), 1716–1719.

Adhikari, R., and Agrawal, R., 2013, An Introductory Study on Time Series Modeling and Forecasting. *LAP LAMBERT* Academic Publishing. ISBN 3659335088, 76 p.

Athira, V., Geetha, P., Vinayakumar, R., & Soman, K.P., 2018, DeepAirNet: Applying Recurrent Networks for Air Quality Prediction, *Procedia Computer Science* 132, pp. 1394–1403 url: https://doi.org/10.1016/j. procs.2018.05.068.

Azid, A., Juahir, H., Latif, T.M., Zain, M.S., & Osman R.M., 2013, Feed-Forward Artificial Neural Network Model for Air Pollutant Index Prediction in the Southern Region of Peninsular Malaysia, *Energy Journal of Environmental Protection* 7(4): 1–10.

Bassam A., Santoyo E., Andaverde J., Hernandez J.A., Espinoza-Ojeda O.M., 2010, Estimation of static formation temperatures in geothermal wells by using an artificial neural network approach. *Computer & Geosciences* 36: 1191-1199.

Bengio Y., Lamblin P., Popovici D., Larochelle H., 2007, Greedy layer-wise training of deep networks, *Adv Neural Inform Process Syst* 19:153-160.

Battiti, R., 1992, First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method. *Neural Computation* 4.2: 141-66.

Baklanov A., Mestayer P., Clappier A., Zilitinkevich S., Joffre S., Mahura A., Nielsen N., 2008, Towards improving the simulation of meteorological fields in urban areas through updated/advanced surface fluxes description, *Atmospheric Chemistry and Physics* 8: 523-543.

Box G.E., Jenkins G.M., Reinsel G.C., Ljung G.M., 2015, Time series analysis: forecasting and control, John Wiley &Sons.

Chai, T., & Draxler, R., 2014, Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature, *Geoscientific Model Development* 7(3), pp.1247-1250.

Chen Y., Shi R., Shu S., Gao W., 2013, Ensemble and enhanced PM10 concentration forecast model based on stepwise regression and wavelet analysis, *Atmos Environ* 74:346–359. doi:10.1016/j.atmosenv.2013.04.002.

Cai, M., Yafeng Y., Min X., 2009, Prediction of Hourly Air Pollutant Concentrations near Urban Arterials Using Artificial Neural Network Approach, Transportation Research Part D: *Transport and Environment* 14.1: 32-41.

Cohen A.J., Brauer M., Burnett R., Anderson H.R., Frostad J. et al., 2017, Estimates and 25-year trends of the global burden of disease attributable to ambient air pollution: an analysis of data from the Global Burden of Diseases Study 2015, *The Lancet*, 389(10082), pp. 1907–1918.

Dey R., & Salem, F. M., 2017, Gate-variants of Gated Recurrent Unit (GRU) neural networks, *Midwest Symposium on Circuits and Systems*, 1597–1600.

Demuzere M., Trigo R.M., Vila-Guerau de Arellano, J., & Van Lipzig, N.P.M., 2009, Impact of weather and atmospheric circulation on O3 and PM10 levels at a rural mid-latitude site, *Atmospheric Chemistry and Physics* 9 (8) pp. 2695–2714.

Dominick, D., Latif, M.T., Juahir, H., Aris,A.Z., & Zain S.M., 2012, An assessment of influence of meteorological factors on PM10 and NO2 at selected stations in Malaysia, *Sustainable Environment Research* vol. 22, no. 5, pp. 305–315.

EC Directive, 2008, 2008/50/EC of the European Parliament and of the Council of 21 May 2008 on ambient air quality and cleaner air for Europe (OJ L 152, 11.6.2008, p.1).

EEA, 2017, European Environment Agency. “European Air Quality Index (EAQI)”. Published on 16 Nov 2017. https://www.eea.europa.eu/themes/air/air-quality-index

Frisk, L. & Partiklar, I.L., 2015, <!http://www.miljomal.se/Miljomalen/Allaindikatorer/Indikatorsida/?iid=105&pl=1> N.p., n.d. Web. 11 Sept. 2015.

Gers F.A., Schraudolph N.N. & Schmidhuber J., 2003, Learning precise timing with lstm recurrent networks, *J. Mach. Learn. Res.* 3, 115–143.

Guocai Z., 2004, Progress of weather research and forecast (WRF) model and application in the United States, *Meteorol Mon* 12:5.

Goyal P., Chan A., Jaiswal N., 2006, Statistical models for the prediction of respirable suspended particulate matter in urban cities, *Atmospheric Environment* 40: 2068-2077.

Gardner M.W., & Dorling S.R., 1998, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, *Atmos Environ* 32:2627–2636. doi:10.1016/S1352-2310(97)00447-0.

Hornik, K., Stinchcoombe, M., White, H., 1989, Multi layer feed forward networks are universal approximators, *Neural Networks* 2, 359-366.

Hinton G.E., & Salakhutdinov R.R., 2006, Reducing the dimensionality of data with neural networks, *Science*, 313 (5786), pp. 504-507.

Hinton G.E., & Osindero, S.Y.W., 2006, The A fast learning algorithm for deep belief nets, *Neural Comput*., 18 (7) , pp. 1527-1554.

IEA, 2018, International Energy Agency, Available online: https://www.iea.org/ Accessed: 22.02.2018.

Jeong J.I., Park R.J., Woo J., Han Y., Yi S., 2011, Source contributions to carbonaceous aerosol concentrations in Korea, *Atmos Environ* 45:1116–1125.

Karlik, B. & Vehbi A.O., 2015, Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks, *International Journal of Artificial Intelligence and Expert Systems (IJAE)*, n.d. Web. 14 Sept. 2015.

Kim Y., Fu J.S., Miller T.L., 2010, Improving ozone modeling in complex terrain at a fine grid resolution: part I- examination of analysis nudging and all PBL schemes associated with LSMs in meteorological model, *Atoms Environ* 44: 523-532.

Kuremoto T., Kimura S., Kobayashi K., Obayashi M., 2014, Time series forecasting using a deep belief network with restricted Boltzmann machines, *Neurocomputing* 137:47-56.

Kaya K., & Öguducu S.G., 2020, Deep flexible sequential (DFS) model for air pollution forecasting, *Nature* 10:3346 | https://doi.org/10.1038/s41598-020-60102-6 .

Kolehmainen, M., Martikainen, H., Ruuskanen, J., 2001, Neural networks and periodic components used in air quality forecasting, *Atmospheric Environment* 35, 815–825.

LeCun Y., Bengio Y., Hinton G., 2015, Deep learning, *nature*, 521: 436.

Li C., Hsu N.C., Tsay S., 2011, A study on the potential applications of satellite data in air quality monitoring and forecasting, *Atoms Environ* 45:3663-3675.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi F.E., 2017, A survey of deep neural network architectures and their application, *Neurocomputing* 234, pp. 11–26.

Liu, J., Li, Y., Chen, M., Dong, W., & Jin, D., 2015, Software-defined internet of things for smart urban sensing. IEEE *Commun. Mag*. 53, 55–63.

Libelium World., 2015, Libelium Connecting Sensors to the Cloud RSS. N.p., n.d. Web. 11Sept.2015.http://www.libelium.com/smart_city_environmental_parameters_public_transportation_waspmote/ & <http://www.libelium.com/smart_city_air_quality_urban_traffic_waspmote/>.

Masih, A., 2018a, Thar coalfield: Sustainable development and an open sesame to the energy security of Pakistan, *J. Physics*: Conf. Ser., 989(1): 012004.

Oludare, I., Aman, J., & Abiodun, E., 2018, State-of-the-art in artificial neural network applications: A survey. Heliyon, (June), p. e00938. ISSN 2405-8440, url: https://doi.org/10.1016/j.heliyon.2018.e00938.

Ong, B.T., Sugiura, K., & Zettsu, K., 2016, Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5, *Neural Computing and Applications* 27(6), pp. 1553–1566.

Peng H., Lima A.R., Teakles A., Jin J., Cannon A.J,. Hsieh W.W., 2016, Evaluating hourly air quality forecasting in Canada with nonlinear updatable machine learning methods, *Air Qual, Atmos Health* 10:195-211.

Panchal G., Ganatra A., Kosta Y.P., Panchal D., 2011, Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers, *Int J Comput Sci Eng* 3:333-337.

Prybutok V.R., Yi J., Mitchell D., 2000, Comparison of neural network models with ARIMA and regression models for prediction of Houston's daily maximum ozone concentrations, *Eur J Oper Res* 122: 31-40.

Perez, P., Trier, A., Reyes, J., 2000, Prediction of PM2.5concentrations several hours in advance using neural networks in Santiago, Chile. *Atmospheric Environment* 34, 1189–1196.

Polissar A.V., & Hopke P.K., 1998, Atmospheric aerosol over Alaska: Elemental composition and sources, *Journal of geophysical research* 103: 19045-19057.

Qi, Y., Li, Q., Karimian, H., & Liu, D., 2019, A hybrid model for spatiotemporal forecasting of PM2.5 based on graph convolutional neural network and long short-term memory, *Science of The Total Environment* 664, pp. 1–10.

Ruidavets J.B., Cournot M., Cassadou S., Giroux M., Meybeck M., & Ferrieres J., 2005, Ozone air pollution is associated with acute myocardial infarction, *Circulation*, 111(5):563–569.

Ribeiro G.H.T., De Neto P.S.G.M., Cavalcanti G.D.C., & Tsang I R., 2011, Lag selection for time series forecasting using Particle Swarm Optimization, *in Proceedings of the International Joint Conference on Neural Networks*, pp. 2437–2444, San Jose, CA, USA, July 2011.

Seinfeld, J.H., Pandis, S.N., 2012, Atmospheric chemistry and physics: from air pollution to climate change. John Wiley & Sons.

Saide P., Carmichael G., Spak S., Gallardo L., Osses A., Mena-Carrasco M., Pagowski M., 2011, Forecasting urban PM10 and PM2.5 pollution episodes in very stable nocturnal conditions and complex terrain using WRF–Chem CO tracer model, *Atmospheric Environment* 45: 2769-2780.

Schlink, U., Dorling, S., Pelikan, et al., 2003, A rigorous inter-comparison of ground-level ozone predictions, *Atmospheric Environment* 37, 3237–3253.

Sánchez A.B., Ordóñez C., Lasheras F.S., de Cos Juez F.J., Roca-Pardiñas J., 2013, Forecasting SO2 pollution incidents by means of Elman artificial neural networks and ARIMA models. *Abstr Appl Anal* 2013:1–6. Hindawi Publishing Corporation.

Subramanian V.N., 2016, Data analysis for predicting air pollutant concentration in Smart City Uppsala. Examensarbete, Uppsala University, Sweden.

World Energy Outlook Special Report, 2016. https://www.iea.org/publications/ freepublications/publication/WorldEnergyOutlookSpecialReport2016EnergyandAirPollution pdf Accessed: 22.02.2018.

Wang D., Wei S., Luo H., Yue C., Grunder O., 2017, A novel hybrid model for air quality index forecasting based on two-phase decomposition technique and modified extreme learning machine, *Sci Total Environ* 580:719-733.

Wang, J. & Song, G., 2018, A Deep Spatial-Temporal Ensemble Model for Air Quality Prediction, *Neurocomputing* 314, pp. 198–206.

Wen, C., Liu, S., Yao, X., Peng, L., Li, X., Hu, Y., & Chi, T., 2019, A novel spatiotemporal convolutional long short-term neural network for air pollution prediction, *Science of the Total Environment* 654, pp. 1091–1099. ISSN18791026, url: https://doi.org/10.1016/j. scitotenv.2018.11.086.

Yu Z., Liciac C., Ouric W., Hai Y., 2014, Urban Computing: Concepts, Methodologies, and Applications. ACM Trans. *Intell. Syst. Technol*. 5:1–55.

Yang G., Lee H.M. & Lee G., 2020, A hybrid deep learning model to forecast particulate matter concentration levels in Seoul, Sout Korea, *Atmosphere*. 11, 348.

Zell, A., 1994, Simulation neuronaler Netze. R. Oldenbourg Verlag München Wien.

Zhou, Y., Chang, F.J., Chang, L.C., Kao, I.F., & Wang, Y.S., 2019, Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts, *Journal of Cleaner Production* 209, pp. 134–145. ISSN 09596526, url: https://doi.org/10.1016/ j.jclepro.2018.10.243.

Zhao H.Z, & Magoul´es, F., 2011, Feature selection for support vector regression in the application of building energy prediction, in *Proceedings of the 9th IEEE International Symposium on Applied Machine Intelligence and Informatics*, pp. 219–223, Smolenice, Slovakia, January 2011.

Zhang, N., Chen, H., Chen, X., & Chen, J., 2016, Semantic framework of internet of things for smart cities: Case studies, *Sensors* 16, 1501.