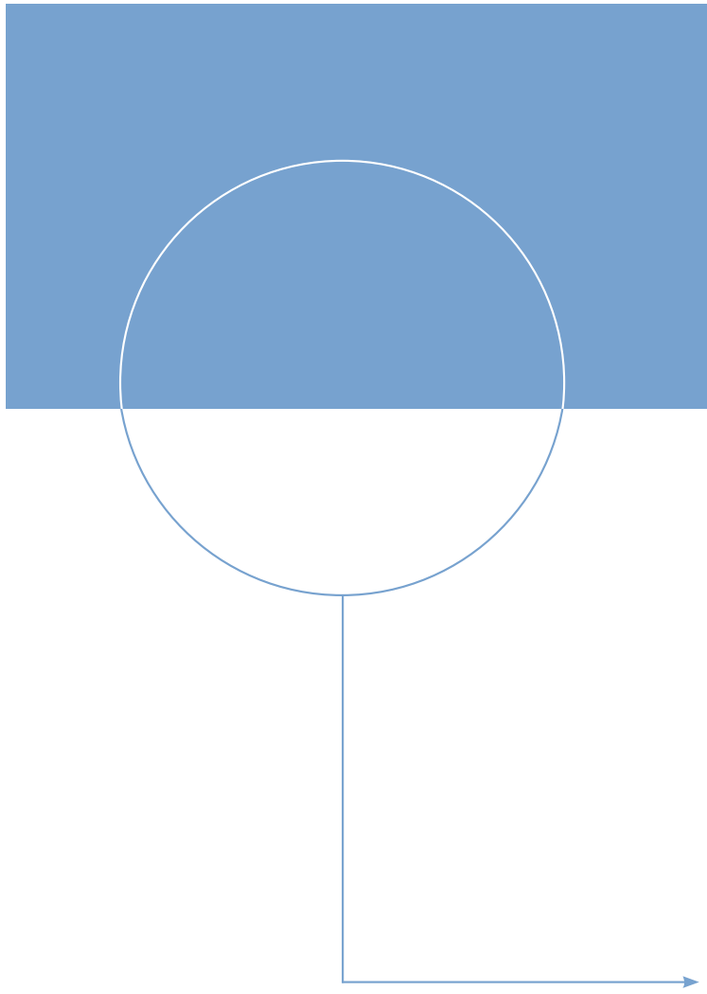


Master's thesis

2019

Andreas Jacobsen Lepperød

**NTNU**
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

Andreas Jacobsen Lepperød

# Air Quality Prediction with Machine Learning

June 2019

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

# Air Quality Prediction with Machine Learning

Andreas Jacobsen Lepperød

Master's thesis in Computer Science
Submission date: June 2019
Supervisor: Hai Thanh Nguyen, IDI
Co-supervisor: Sigmund Akselsen, Telenor
Leendert Wienhofen, Trondheim Municipality
Pinar Øzturk, IDI

Norwegian University of Science and Technology
Department of Computer Science

# Abstract

In recent years, air quality has become a significant environmental health issue due to rapid urbanization and industrialization. Because of the impact air quality has on peoples everyday life, how to predict air quality precisely, has become an urgent and essential problem. Air quality prediction is a challenging problem with several complicated factors with additional dependencies among them.

We target our air prediction study to the city of Trondheim, Norway. The air quality in Trondheim is on average at a healthy level, but has periods of high variations of severe pollution, especially in the winter months. The study demonstrates the benefits of machine learning for predicting air pollutants general pattern, and to foresee sudden spikes of a high pollution level. This paper explores a multivariate time series approach to modeling and forecasting the pollution of PM2.5, PM10, and NO2 at three air quality stations. This study is concerned with combining data of pollutants, meteorological, and traffic data with statistical temporal-spatial feature engineering, to provide multi-step-ahead air quality forecasts for 24 and 48-hours.

Extensive experiments of real-time air pollution illustrate the effectiveness of machine learning to forecast air pollutions in terms of general pattern and sudden changes. Results express that ensemble techniques could significantly improve the stability and accuracy of predicting the general trend of air quality. Among the ensemble techniques, using gradient boosting with dropouts results in prediction errors with the lowest deviation. In the case of predicting sudden changes in air pollution, using a recurrent neural network with a memory unit results in the highest accuracy of classified spikes. Lastly, the machine learning results were compared with the national air quality service, a knowledge-driven model, to evaluate real-world practice. The predictions of general pattern and anomalies of this thesis are shown to be superior for 24-hour, and more comparable results for the 48-hour forecast. The data-driven approach is thus believed to be an excellent complement for the knowledge-driven model.

# Sammendrag

I de senere år har luftkvalitet blitt et betydelig miljø- og helseproblem på grunn av rask urbanisering og industrialisering. På grunn av den påvirkning luftkvaliteten har på alles hverdag er nøyaktige observasjoner og prediksjoner av forurensing en viktig utfordring å løse. Det å forutsi luftkvaliteten er utfordrende med flere komplekse faktorer i et miljø i stadig endring.

Prosjektet er gjennomført i Trondheim, Norge og demonstrerer fordelene med maskinlæring for å forutse luftkvalitetes daglige mønster, og spesielt dens egenskap for å oppdage plutselige endringer med høyt forurensningsnivå. Dette studiet utforsker en løsning ved å bruke en tidsserie med flere variabler for å modellere forurensning av svevestøv (PM2.5 og PM10), i tillegg til nitrogendioksid (NO2) på tre målestasjoner for luftkvalitet. Forskningen fokuserer på å kombinere data over forurensende stoffer, meteorologisk data og trafikkdata sammen med en statistisk temporal-romslig teknikk for å gi luftkvalitetsprognoser for 24 og 48 timer fram i tid.

Omfattende analyse og eksperimenter av luftforurensning i sanntid illustrerer effektiviteten av maskinlæring for å forutsi luftforurensninger i form av generelt mønster og plutselige endringer. Resultatene uttrykker at Ensemble Learning kan forbedre stabiliteten og nøyaktigheten til å forutsi den generelle utviklingen i luftkvalitet betydelig. Blant flere er det Gradient Boosting som gir best resultater med lavest feilmargin. Ved forutsigelse av plutselige endringer i luftforurensning er det et Recurrent Neural Network som gir best nøyaktighet. Til slutt ble maskinlæringsresultatene sammenlignet med den nasjonale luftkvalitetstjenesten - en kunnskapsdrevet modell. Resultatene kunne da evalueres i praksis. Resultatene fra denne oppgaven viser seg å være overlegen i 24 timer, og med mer sammenlignbare resultater for 48-timers prognoser. Den data-drevne løsningen er dermed antatt å være et utmerket komplement til den kunnskapsdrevne modellen.

# Preface

This thesis was carried out with cooperation between the Department of Computer Science (IDI) at the Norwegian University of Science and Technology (NTNU), Norwegian Telecommunications Telenor, and Trondheim Municipality. The study is concerned with air quality in the city Trondheim, Norway, and intent to evaluate machine learning algorithms for accurate predictions. This thesis was inspired by the recent advances of machine learning within air quality prediction demand. Hai Thanh Nguyen (IDI, Telenor) was the supervisor of this thesis, together with Sigmund Akselsen (Telenor), Leendert Wienhofen (Trondheim Municipality), and Pinar Øzturk (IDI) as co-supervisor. I want to thank each one for their motivation, their insight, and their dedication during this project. I would also like to thank Telenor for server access and Trondheim Municipality for granting data access to city information. Lastly, a big thanks to the Norwegian Institute for Air Research (NILU) for domain knowledge and insight into their current projects.

Trondheim, June 2019
*Andreas Jacobsen Lepperød*

# Table of Contents

This page intentionally left blank.

# List of Tables

# List of Figures

# Abbreviations

| | | |
|---|---|---|
| AI | = | Artificial Intelligence |
| ANN | = | Artificial Neural Network |
| ARIMA | = | Autoregressive Integrated Moving Average |
| CART | = | Classification And Regression Trees |
| DART | = | Dropouts meet Multiple Additive Regression Trees |
| DNN | = | Deep Neural Network |
| DTR | = | Decision Tree Regression |
| FFNN | = | Feedforward neural network |
| GB | = | Gradient Boosting |
| GBDT | = | Gradient Boosting Decision Tree |
| GRU | = | Gated Recurrent Units |
| IoT | = | Internet of Things |
| LR | = | Linear Regression |
| LSTM | = | Long Short-Term Memory |
| MAE | = | Mean Average Error |
| MAPE | = | Mean Absolute Percentage Error |
| MET | = | National Air Quality Service |
| ML | = | Machine Learning |
| MLP | = | Multilayer Perceptrons |
| NN | = | Neural Network |
| NOx | = | Nitrogen Oxides |
| PM | = | Particle Matter |
| R2 | = | Coefficient of Determination |
| RAE | = | relative absolute error |
| RF | = | Random Forest |
| RMSE | = | Root Mean Square Error |
| RNN | = | Recurrent Neural Network |
| SMAPE | = | Symmetric mean absolute percentage error |
| SVM | = | Support Vector Machine |
| VOC | = | Volatile Organic Compounds |

This page intentionally left blank.

Chapter 1

# Introduction

In this thesis, we explore air quality prediction with a particular focus on applied machine learning. Promising machine learning approaches from the literature are implemented and evaluated for performance. The research presented is to improve air quality predictions and knowledge in Trondheim. The air quality in Trondheim is on average at a healthy level, but has periods of high variations of severe pollution, especially in the winter months. Along with precise predictions of air pollution levels, the public and governments can respond with appropriate decisions, such as dust cleaning and discouraging outdoor activities, to mitigate the harmful consequences of air pollution. This first chapter provides an overview of the challenges around air pollution and the motivation for this research, followed by details of the research goal and the research method. Lastly, a summary of the contributions is presented, along with the outlines of the thesis structure.

## 1.1 Background and Motivation

The necessity of healthy air has always been of great importance. As air is vital for all living beings on earth, it is our responsibility to keep the air clean. The rapid urbanization and industrialization have led the world into a new era of air pollution and is seen as a modern-day curse. Air pollution refers to the contamination of the air by excessive quantities of harmful substances. Most air pollution occurs from energy use and production, where emissions from traffic and industry are major contributors.

Air pollution is a widespread problem due to its impact on both humans and the environment. Urban cities usually have the worst air pollution due to human activities [Kampa and Castanas (2008)]. Clear links between pollution and health effects have been revealed, which includes both short- and long-term consequences [Brauer et al. (2012)]. Associations with reduced lung function and increase in heart attack [Arden et al. (2002)], direct impact on people with asthma and other types of pneumonia [Guarnieri and Balmes (2014)] and once inhaled, a fine particular matter may hardly be self-purified by the immune system [Becker (2002)]. The overall effects of ambient air pollution on premature human mortality are a falling global trend, but in a smaller geographical area, the levels do

not follow WHO's guidelines [WHO et al. (2006)]. Due to these severe problems, there are national requirements and objectives that each city must meet.

Air quality has increasingly attracted attention from environment managers and citizens all over the world. New tools continue to emerge to raise air quality awareness worldwide. Continuously improvements in air quality mapping are happening along with the advancements of smart cities and the amount of internet-of-things sensor devices. The increase in data produced contributes to further momentum in air pollution activity. A hot research topic is air pollution forecasting, the prediction of the atmospheric composition of pollutants for a given time and location. With an accurate air quality forecast, one can decide how to act due to air pollution health effects. On the national level, accurate forecasting contributes to planning and establishing procedures to reduce the severity of local pollution levels. With better knowledge at the individual level, one can choose the right choice for the cleanest routes for the commute, the best time for outdoor activity and other daily outdoor activities. Awareness like this has the potential to create a cleaner environment and a healthier population.

Accurate time series forecasting of air quality is a continuous research area, and much effort has been made by researchers to create models capable of fitting the underlying time series. Often, air quality prediction involves a noisy and limited amount of historical data. Furthermore, the prediction of a single observation usually depends on many events that rely on each other. The models are then forced to include specially adapted techniques to comply with the erroneous or lack of data. These complex problems make it hard to generalize the solution to be transferable to other locations. Besides, the air changes rapidly in short time frames, with hourly data more uncertain compared with monthly and yearly trends and seasonality. The lack of and poor quality of data, a low spatial resolution of data points, and the cost of high-quality sensors add up to the list among other obstacles.

Figure 1.1 presents the observations of particular matter smaller than 2.5 microns (PM2.5) in the period of the end of 2017 to Mai 2019. The graph shows the typical pollution trends in Trondheim, of rapid changes in pollution levels for the winter months, while the summer includes a relatively good level. These trends and patterns are typical characteristics of the air quality in cities in Norway and Scandinavia. Consequently, the challenge of these time series is then the prediction of the sudden changes in harmful pollution.

## 1.2 Goals and Research Questions

The goal is to use state-of-the-art research results with a special focus on machine learning methods about air quality prediction, to evaluate and apply ideas and algorithms from these studies in the context of time series prediction. The ultimate goal of air quality prediction is to enable national and global decision-makers, communities, and individuals to proactively take measures to reduce the health hazards caused by air pollution. This work attempts to answer the following three research questions:

**Research Question 1** Which machine learning techniques have been used within the domain of air quality prediction?

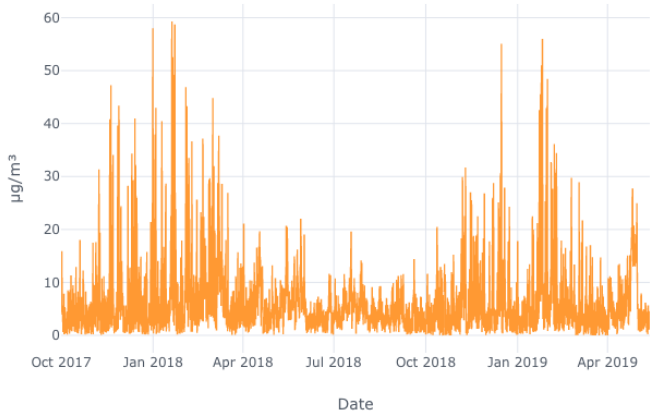



**Figure 1.1:** Average level of PM2.5, November 2017 to Mai 2019.

RQ1 relates to the structured literature review part of this thesis. Machine learning is a hot topic and frequently referred to in research within air quality prediction. State-of-the-art techniques will hold valuable knowledge which can be transferable to the experiments conducted in Trondheim.

**Research Question 2** Which features have the highest impact on the machine learning algorithm's ability to accurately perform predictions?

**Research Question 3** How accurate are machine learning methods for predicting air quality in Trondheim?

RQ2 and RQ3 concern the experimental part of this thesis, and are defined to ensure that the experiments performed are strictly objective. RQ2 focuses on how to include multiple external sources to increase the performance of the machine learning algorithm. This section will try to improve the knowledge of known and unknown external sources in Trondheim and their correlation with air quality. Multiple hypotheses are looked at to conclude an answer to the research question. RQ3 will focus on comparing the accuracy of various machine learning methods included in this thesis. The term accuracy means how close to the real observations the predictions are and is defined in section 5.1.1, with a separation of general air pattern error and the classification of sudden air pollutant changes.

## 1.3 Research Method

Firstly, we performed a state-of-the-art review of the literature to find an answer to the first research question. It is continued by an analysis of the datasets of Trondheim, to find related feature hypothesis used similarly in other research. Then an iterative process of model designing and implementation, followed by experiments with multiple feature extractions. These results were observed and analyzed to conclude the methods ability to learn from the features provided.

## 1.4 Contributions

The contributions of this thesis are primarily:

1. A state-of-the-art review of air quality prediction using machine learning models.
2. An exploratory data analysis of air pollutants, meteorological, and traffic data to discover patterns, anomalies, and to check assumptions by applying statistics and graphical representations.
3. A comparison of the performance of multiple machine learning algorithms evaluated on real air quality data in Trondheim.
4. Recommendations for machine learning methods for air quality predictions and to classify air pollutants sudden changes.

## 1.5 Thesis Structure

**Chapter 1: Introduction**

Chapter 1 presents a brief overview of the content of the project.

**Chapter 2: Background Theory**

Chapter 2 prepares the reader the knowledge necessary to understand the domain of the problem, along with the methods investigated in this study.

**Chapter 3: Literature Review**

Chapter 3 presents the reviews of air quality projects and air quality prediction.

**Chapter 4: Architecture and Models**

Chapter 4 explains the architecture, and a description of the implementation of the models used in the experiments.

**Chapter 5: Experiments**

Chapter 5 presents the experiment details of with the plan, setup and results.

**Chapter 6: Conclusion**

Chapter 6 concludes the work of this paper with an evaluation, discussion of the results and limitations discovered. Lastly, we present a description of the contributions, with constraints and proposed further work.

This page intentionally left blank.

Chapter 2

# Background Theory

This chapter introduces relevant background theory for the reader to pick up on key terminology used throughout this paper. Section 2.1 defines air pollution and its main causes. Section 2.2 describes the basics of the time series with some important features. Section 2.3 presents two common statistical approaches for time series prediction. Section 2.4 presents a brief introduction to machine learning techniques considered for prediction. Section 2.5 dives deeper into the machine learning with deep neural networks. Section 2.6 introduce the concept of ensemble learning and the algorithms which use this technique.

## 2.1 Air Pollution

Air pollution is one such form that refers to the contamination of the air, irrespective of indoors or outdoors. It occurs when pollutants enter the atmosphere and make it difficult for plants, animals, and humans to survive as the air becomes dirty [Seinfeld and Pandis (2012), Akimoto (2003)]. The sustainment of all living things is due to a combination of gases that collectively form the atmosphere. The imbalance caused by the changes in these gases can be harmful to survival.

### 2.1.1 Air Quality

Air quality refers to the condition of the air within our surroundings. Good air quality pertains to the degree to which the air is clean, clear, and free from pollutants such as smoke and dust among other gaseous impurities in the air. Table 2.1 lists the main types of pollutants, along with a short description of each. Good air quality is a requirement for preserving the delicate balance of life on earth for humans, plants, animals, and natural resources, and are at risk when pollution in the air reach critical concentrations.

| Name | Information |
|---|---|
| **CO** | Carbon monoxide primarily from combustion of natural gas, coal or wood. |
| **CO2** | Carbon dioxide is natural and essential at a steady level, but its increases have been accelerating. |
| **SOx** | Sulfur oxides from volcanoes and industry are one of the causes for concern over the environmental impact of the use of these fuels as power sources. |
| **NOx** | Nitrogen oxides are a by-product from Combustion Engines used in traffic and industry. |
| **PM** | PM2.5 denotes the diameter of the particulate matter is less than 2.5 microns, and for PM10 it is 10 microns. It is highly dependent on local conditions, such as climate, traffic, and pollution. In Norway, it is dominated by long-range transport, road dust, and wood burning. |
| **O3** | Ozone is a greenhouse gas formed by the reaction of sunlight on air. Hydrocarbons and nitrogen oxides in the air react to form Ozone directly at the source of the pollution or many kilometers downwind. |
| **VOC** | Volatile organic compounds like methane is an extremely efficient greenhouse gas which contributes to enhanced global warming. |

**Table 2.1:** The main types of pollutants. NOx, PM is the pollutants of interest in this thesis.

### 2.1.2 What Causes Reduced Air Quality

Emissions from various sources continuously reduce air quality. These are either natural or human-made sources. Natural sources include among other volcanic eruption, windstorms, biological decay, and forest fires. A human-made source may be pollution from moving vehicles, manufacturing facilities, power plants, smelters, and burning wood or coal. The pollutants from these sources are released into the air and can lead to severe health problems for humans, animals, and the environment. Air quality depends on three factors: the number of pollutants, the rate at which they are released in the atmosphere, and how long they are trapped in an area. If air pollutants are in an area with good airflow, they will mix with the air and quickly disperse. Air pollutants tend to remain in the air when there are certain conditions like light winds or obstacles that restrict the transport of these contaminants away from an area. Consequently, air pollution concentration increase rapidly.

## 2.2 Time Series

Time series is essentially a sequence of data points measured over time. The measurements in a time series are arranged in chronological order, and consist of either a single variable termed univariate or of more than one-time dependant variable called multivariate. A continuous time series are observations measured in all cases of time, while discrete include measurements on distinct points of time. Usually, a discrete set consists of succeeding

observations recorded at equal time intervals like an hour, daily or yearly separations.

A time series is reckoned to be influenced by four main components: Trend, Seasonal, Cyclic, and Irregular. The trend is a time series general tendency of changes, with a long-term movement of increase, decrease or stagnation. Seasonal variations are associated with changes during the seasons of the year, where climate and weather are important factors. The cyclic part describes differences as medium-term changes in a time series, caused by circumstances with a cyclic nature. The last component, irregular, are random variations, or so-called noise, which are not typical and is not able to be described by the previous parts.

With these four components, there are two different types of models used for time series: Multiplicative and Additive, described in Equation 2.1 and 2.2 respectively. The assumption for the multiplicative time series model is that all components are not necessarily independent and might affect each other where the additive assumes component independence.

$$Y(t) = T(t) \times S(t) \times C(t) \times I(t) \tag{2.1}$$

$$Y(t) = T(t) + S(t) + C(t) + I(t) \tag{2.2}$$

Stationary is another concept for time series. A time series is stationary if its properties such as mean and variance do not depend on time. Stationarity is thus a useful assumption because of the less mathematical complexity of the model. A time series can be homoscedastic, which relates to samples of the time series have similarity variance as any other samples in the dataset. Heteroscedastic means the opposite with different variance throughout the time series. To summarize: stationary and homoscedastic time series are well behaved and more straightforward to predict, and non-stationary and heteroscedastic series are much more complicated. In the latter case, by applying mathematical transformations to make the series stationary and homoscedastic to reduce the precise problem.

## 2.3 Statistical Methods

This section introduces two common statistical methods for prediction. The popular autoregressive integrated moving average model and linear regression with regularization called Ridge Regression.

### 2.3.1 Auto Regressive Integrated Moving Average

Auto Regressive Integrated Moving Average (ARIMA) is a well known statistical time series prediction model introduced by Box and Jenkins (1970). With its simple design, it used in many practical forecast cases. ARIMA is using the intuition that points from the past will affect the outcome of the future, and is by design smoothing the forecast horizon. The ARIMA model is defined with three parameters, p is the number of autoregressive terms, d is the number of non-seasonal differences, and q is the number of lagged forecast

errors. ARIMA is suitable for stationary problems, and will not be able to capture more complex trends and seasonality. Seasonal-ARIMA is an extended model of the ARIMA model, which can capture timing patterns from a statistical perspective. It extends ARIMA with a seasonal part that doubles up the parameters.

### 2.3.2 Ridge Regression

Ridge Regression is a technique for analyzing data that suffer from multicollinearity [Hoerl and Kennard (2000)]. Multicollinearity is the existence of near-linear correlations among the independent variables. When multicollinearity is found, the least squares estimates are unbiased, but their variances are large so that they may be far from the real value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors, and that the net effect will be to give more reliable estimates. Ridge regression is a good choice for noisy data due to its nature of reducing the variance at the cost of some bias.

## 2.4 Machine Learning

Machine learning is a branch which is set out of artificial intelligence. Its goal is to enable the computer to learn by itself without being explicitly programmed the rules. A machine learning algorithm can identify and learn underlying patterns in observed data to model and predict the world.

There are three kinds of machine learning techniques: reinforcement learning, unsupervised learning, and supervised learning. In reinforcement learning, the algorithm receives feedback based on performance as it navigates its problem space. Tasks such as playing a game or driving a car are examples where reinforcement learning is suitable. Unsupervised learning is an approach that learns from data that is unlabeled or classified. Instead of responding to feedback as in reinforcement learning, unsupervised learning identifies shared attributes and characteristics from the data. Unsupervised learning algorithms include association problems, which tries to describe parts of the data, and clustering problems, that seek to identify natural groupings. In supervised learning, the algorithm attempts to learn from informative examples of labeled data. Such algorithms can be described as a data-driven approach, where historical data is used for predictions of the future. Air quality prediction is often solved with supervised methods, as time series can convert to labeled pairs of input and output, where the output target is the ground truth of the next value in the data sequence. The machine learning models presented in this work are of the kind supervised learning.

### 2.4.1 Features

Features are individual variables found in the data set that has associations with the target prediction. The quality of the model's predictive output is no better than the quality and focus of the feature. A particular data set can have several features associated with them, and other features can be derived from the original data to create new sets of informative

data. The process of creating new features from the original dataset is called feature engineering. It is essential to find and select the features that are more relevant to the problem so that the accuracy of the model improves.

Feature Selection is the concept of reducing the feature dimension of the machine learning problem. Feature selection can benefit the model's performance to reduce overfitting, improve accuracy, and reduce training time. Overfitting is a modeling error which results from the algorithms function to fit too strongly on a particular set of data, and may subsequently fail to fit on unknown future observations. The goal is to avoid overfitting while balancing the model's ability to learn enough from the data. Feature selection is a tool to help reduce the unwanted noise of irrelevant or partially relevant features. The accuracy might improve by less misleading data, and in turn, the modeling accuracy will improve. Lastly, feature selection reduces the training duration with a smaller subset of the original dataset.

### 2.4.2 Dataset Split

Before training a machine learning model, the dataset it usually split into training, validation, and a test dataset. The training dataset is for training the model and will consist of the more substantial part of the dataset. After the model has learned from the training dataset, the model runs against a separate validation data set. This validation set is a smaller subset of the training data and can be used for evaluation during training. Lastly, is the test dataset that contains unseen data for the trained model. The training dataset is to evaluate if the model has generalized on the datasets instead of memorizing the outcome.

Different strategies related to how to split the dataset most efficiently is essential for achieving better-generalized results. Cross-validation is a technique that includes multiple rounds of partitioning the datasets into subsets. The rounds, so-called K folds, are done multiple time to reduce the variability of the model's performance and give a better estimate of the models predictive performance. While this procedure is time and computational consuming with the increase of folds, the model has tested every data points which reduce the bias, and also, the variance is estimated multiple times. The goal is to generalize the performance measure from the learning on the test set to predictions from unseen data.

### 2.4.3 Gradient Descent

A common optimization technique for training machine learning algorithms is gradient descent. The goal is to adjust the weights of $w$ to minimize the loss. This loss is a measure of how well our model is doing and is represented by $J(w)$. The gradient descent algorithm starts with random model parameters and calculates the error for each learning iteration. The error is estimated to update the model parameters to achieve a minimum loss, and ultimately, an optimal model performance. How much the weights are adjusted each iteration are controlled by a scalar, known as the learning rate. The learning rate is hyper-parameter that often is controlled by leveraging the user's experience.

The process of gradient descent can be defined as Equation 2.3 and 2.4. $\Delta\theta_i$ is the step it walks along the gradient, and a learning rate, $\alpha$, to control the size of our steps.

$$\theta_i := \theta_i + \Delta\theta_i \tag{2.3}$$

$$\Delta\theta_i = -\alpha\frac{\partial J(\theta)}{\partial\theta_i} \tag{2.4}$$

Adam, which stands for Adaptive Moment Estimation, combines the two previous ideas of using a moving average gradient and adaptive learning rates [Kingma and Ba (2014)]. These two values respectively represent the estimates of the first and second moment of the gradients, hence the name of the technique.

## 2.5 Artificial Neural Networks

Artificial Neural Networks (ANN) has gained popularity in the last years in time series forecasting. ANN is a computational network based on how the biological nervous system work. The network is built upon numerous of neurons that work collectively to process the data. A neuron *i* receives several input signals $x_1$, ..., $x_n$ from other neurons, all of which is multiplied by a weight $w_i$ which is computed from the importance of the inputs, and finally summed together. A bias weight $w_0$ is added to provide the ability to shift the results for a better fit. The output $z$ is defined in Equation 2.5

$$z = w_0 + \sum_{j=1}^{N} x_j w_{ij} \tag{2.5}$$

The last step of a neuron is to feed its output through an activation function *g(z)*. See Figure 2.1 for an image description. In its simplest form, called a perception, is using a binary activation function called Heaviside step function, which outputs 1 on positive output and 0 if it is negative. This type of neuron can approximate linear functions, which does not cover most real-world data problems that can only be described with non-linear patterns. Other activation functions have occurred in research and cover various use cases with its characteristics of strengths and weaknesses. The ones used in this thesis are:

*sigmoid* takes an input $x \in \mathbb{R}$ and squashes it to a value between 0 and 1.

$$sigmoid(z) = \frac{1}{1 + e{-}z} \tag{2.6}$$

*rectified linear unit* takes an input and replaces the negative values with 0.

$$ReLU(z) = max(0, z) \tag{2.7}$$

*leaky rectified linear unit* takes an input and replaces the negative values with a fraction, $a$, of its original value. Usually, a low value of $a$ is chosen.

$$LeakyReLU(z) = max(az, z) \tag{2.8}$$

Problems that occur with the different activation functions is when their gradient will be huge, very small or go towards 0. For the sigmoid functions, if the values are either

very large or very small, the changes in the gradient are going to be low. This problem is called ”vanishing gradients” and rejects or slows down the network to learn further. The ReLU activation function is a popular choice due to decreased training time and faster convergence. It is cheap to compute since there is no complicated math. It does not have the vanishing gradient problem suffered as with sigmoid function, but ReLU is prone to the ”dying ReLU” problem as it does not have zero-slope parts. In the case of a zero value of the gradient, it will cause the weights not to get adjusted during training, which will cause the neurons to stop responding to an error in the input data. The problem of a dying neuron is a big problem for ReLU, due to its nature of squashing values of negative value to 0. Leaky ReLU is an attempt to solve this issue by letting through some of the negative values. Also, by using Leaky ReLU, it is evidence that by having the mean activation close to 0 makes training faster, by running as more balanced.

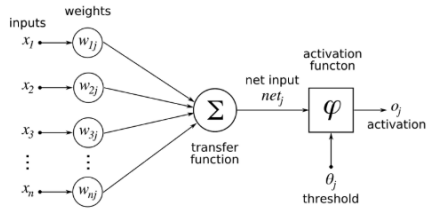



**Figure 2.1:** How a neuron works.

### 2.5.1 Multilayer Perceptron

A single perceptron can learn linear functions right. However, by building a network of these nodes arranged into more layers consisting of multiple non-linear activation functions in each, the network can now learn nontrivial problems. Such a network architecture is called multilayer perceptron (MLP). The output of each neuron in layer $j_n$ is strictly dependant on the output of the previous layers $j_{n-1}$ through weighted edges. See Figure 2.2 for an example of an MLP with a single hidden unit.

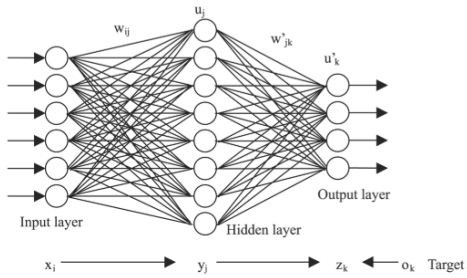



**Figure 2.2:** Diagram of a multilayer perceptron with *i* input nodes, *j* hidden nodes in a single hidden layer and *k* output nodes

### 2.5.2 Recurrent Neural Networks

Recurrent neural networks (RNN) is a type of ANN where in addition to feeding the output of each neuron to the next neuron, the output is fed back into the same neuron for the next step. RNN allows the network to keep an internal state and mimic memory [Russell and Norvig (2016)]. With this inner state, RNN is good at modeling dynamic temporal behavior for time series data, due to its dependence on historical data. A drawback with standard RNNs is that it has a vanishing gradients problem. That is because the further backward in the sequence the weight update is going, the lower the error will become and small to none changes will be done to the weights. Branches of RNNs like Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) has appeared to overcome the limits of vanishing and exploding gradients.

LSTM is a type of RNN, designed to deal with long-term dependency learning [Hochreiter and Schmidhuber (1997)]. An LSTM unit is composed of a cell, an input gate, and an output gate and a forget gate. The advantage of this architecture is the ability to let the network learn when to apply a broader context, and then determining when to rely on long-term or short-term memory, making it possible to learn more complex functions. GRU [Cho et al. (2014)] can store and filter the information using their update and reset gates, which reduces the vanishing gradient problem due to the model is keeping relevant information and passes it to the next time steps. The GRU design is different from LSTM that it does not include the memory unit. The of GRU is computationally more efficient due to fewer calculations. See Figure 2.3 for an overview of LSTM and GRU.

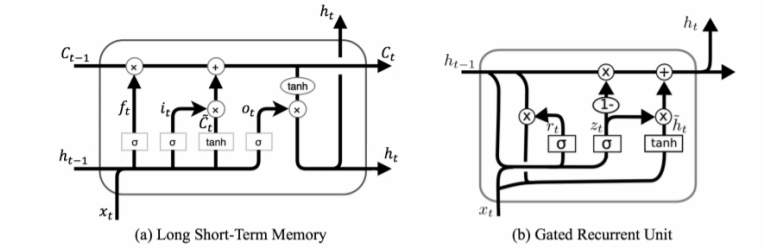



**Figure 2.3:** Overview of the LSTM and GRU-cell

## 2.6 Ensemble Learning

Regularly, as the complexity of machine learning models increases, there will be a reduction in error due to lower bias in the model. However, after a particular point, the model will start suffering from high variance, so-called overfitting. The goal of ensemble methods is to overcome this problem by combining the predictions of several estimators to improve generalizability and robustness over a single estimator. The pool of predictions of multiple models will help reduce this variance, to make partially individually errors lower. Two types of ensemble methods are usually distinguished, averaging, and boosting techniques. In averaging, the goal is to produce multiple independent estimators and then

average their predictions. This combined estimator is usually better than any of the single base estimator because of its reduced variance. In contrast, in boosting methods, the estimators are built sequentially to reduce the bias of the combined estimator. Ensemble learning is recognized as one of the most successful approaches to regression and classification tasks, with much empirical and theoretical evidence about the increase in stability and predictive accuracy [Zhang and Ma (2012)].

### 2.6.1 Bootstrap Aggregating

Bootstrap Aggregating, or Bagging methods, form a class of algorithms which build numerous instances of estimators on random subsets of the training set and then aggregate their predictions to create a final result. Bagging was introduced in Breiman (1996). The training takes place in parallel by building each model independently. The final result of the N learners is finally joined together by taking the average. The technique ultimately provides a reduced variance of the base estimator by introducing randomization of the data and averaging in its approach. In many cases, bagging constitutes a straightforward way to improve a single model without adopting the underlying base algorithm. There are two kinds of machine learning types, stable and unstable types. A machine learning approach runs in the group of stable learners if a change in the training set makes a small or no change of the model's final result. An unstable learner is a more sensitive type, where minor changes greatly impact the outcome. The choice of the bagging methods base learner is important because the training set is changing each iteration. Thus a stable learner will most likely not perform any better than the learner alone. However, by using an unstable method in the bagging ensemble will increase the probability of more variance and different bias of each model. Bagging is prone to overfitting, making them work best with strong and complex base learners.

### 2.6.2 Random Forest

Random Forest (RF) is an ensemble learning algorithm based on the bagging and decision tree learning [Liaw et al. (2002)]. The goal of the random forest algorithm is to fit different decision trees on random subsets of all the features and subsamples of the dataset. All the trees are then averaged together to improve performance and avoid overfitting. As a result, this technique reduces high variance at the cost of a slightly increased bias. This tradeoff is usually yielding a more robust model when predicting unseen input samples. Random Forest is a popular option due to its short training duration, avoid the need for normalization of the input data, and few hyperparameters to tune.

### 2.6.3 Boosting

Similar to bagging, boosting start with a pool of base learners to generate the final result. However, in the next steps, the boosting methods main difference is that it has a sequential structure [Russell and Norvig (2009)]. See Figure 2.4 for an comparison of bagging and boosting. First off, the subsampling of the dataset is weighted and therefore include some samples more often than others. After each training iteration, the weights are redistributed by emphasizing the weights for misclassified data. The next iteration will then focus the

learning on the most challenging cases. The concept is that the models are becoming more significant at classifying these cases so that the ensemble is ultimately predicting more accurately [Drucker (1997)]. During the training stage, each model receives a score of how good it is performing on the training data. This score is used in the final result when the boosting method is averaging the model's output with a weighted average of their estimates.

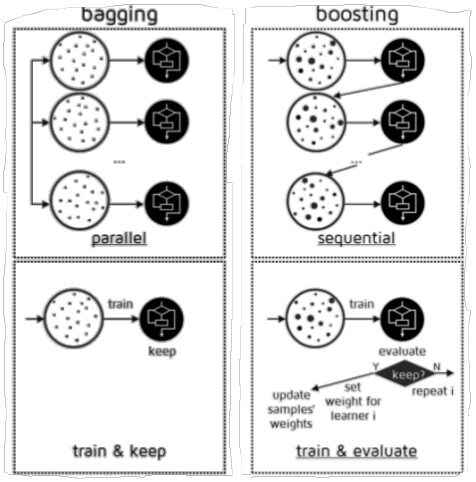



**Figure 2.4:** Comparison of Bagging and Boosting ensembles.

### 2.6.4 Gradient Boosting

From the concept of boosting ensemble, several alternatives of the original have been made. These include variations of how the weights are determined during the training phase. Gradient boosting machine (GBM) is a popular offspring [Friedman (2002)]. Gradient boosting includes a gradient descent to arbitrary differentiable the loss functions of the weights. The advantages of GBM are its natural handling of mixed data type, its predictive power, and lastly its robustness to outliers in the data. They are built up in a dependent fashion by iteratively adding trained base models to reduce the error of the current ensemble. It becomes an ensemble of multiple weak models, with subsequent model's loss lower than the previous. DART is an extension of gradient boosting with the inclusion of dropouts in the regressor buildup [Rashmi and Gilad-Bachrach (2015)]. The learner develops the next tree from the residual of a sample of previous trees. The effect on the model is similar in that individual components are forced to be more self-sufficient.

Chapter 3

# Literature Review

Air quality prediction is a hot research field, and much progress has been made to improve performance. This chapter presents the related work of air quality prediction with machine learning. Section 3.1 starts with the protocol used when gathering research for review, and section 3.2 shows an overview and analysis of the final set of studied articles.

## 3.1 Systematic Literature Review

Due to the lack of a standard framework within the research of air quality prediction, with a variety of problem descriptions, dataset, and location in the studies, a systematic literature review is conducted to achieve an overview of the literature. The research varies in methods and techniques, but also, the dataset distribution is often wholly different due to the climate and environment of the location together with the chosen pollutants to predict. For some urban cities, the poor air quality might be mainly due to PM-related causes, while in others the problem might primarily come from SOx or COx. Due to these limitations ads up, the systematic literature review is an attempt to reduce the scope and find relevant research that targets the same problems as in this thesis. This section defines a series of steps that produced a set of the most relevant studies for this thesis.

### 3.1.1 The identification phase

This phase prepared a set of literature research questions (LRQ) to find research with similar problem definitions as this thesis.

- **LRQ1** How do different machine learning methods compare to each other?
- **LRQ2** What features are included to increase the quality of the predictions of air quality?
- **LRQ3** What problems are they trying to solve with the predictions?

| Concerns | Search term |
|---|---|
| **Domain** | Air Quality, Pollution |
| **Problem** | Prediction, Forecast, Estimation |
| **Techniques** | Deep Learning, Ensemble |

**Table 3.1:** Search terms for the SLR

### 3.1.2 The search phase

The second phase is a process to find all studies that match the literature research questions above.

- Find the sources that should be used for the search by locating relevant online digital libraries and search engines.
- Then proceed to create a search string combined of terms and logical operations to limit the results.

The search engines used was Engineering Village[1] and IEEE Xplore[2], with the search terms shown in Table 3.1. Subsequently, the search query string was: (air quality OR pollution) AND (prediction OR forecast OR estimation) AND (deep learning OR ensemble).

### 3.1.3 The filtering phase

The goal of the filtering phase is to reduce the list produced from the search phase and create a final selection of articles. This phase includes a set of inclusion and quality criteria to reduce the literature list objectively.

- **IC1** The study focuses on improving air quality prediction accuracy utilizing deep learning or ensemble techniques.
- **IC2** The study clearly states which other methods the study has been compared with.
- **IC3** The dataset distribution or location of research is detailed explained.
- **IC4** The study must contain multiple combined datasets for larger feature space.

### 3.1.4 The analysis phase

The last step is to analyze the final set of research articles up against the literature research questions. This analysis covers a comparison of the studies with the main focus on feature extraction and method performance. This analysis is presented in section 3.2.

[1] https://www.engineeringvillage.com/search/quick.url
[2] https://ieeexplore.ieee.org/search/advsearch.jsp

## 3.2 State of the Art Review

This section discusses the information from the set of papers presented in Table 3.2. The results from this review are to be considered as answers to RQ1: Which machine learning based techniques are applied within the domain of air quality prediction?

### 3.2.1 Introduction

Air quality prediction methods can be split into two main categories: classical deterministic models and data-driven models (Zhang et al. (2012a), Zhang et al. (2012b)). The traditional dispersion models consist of heavy domain knowledge of air quality behavior with expertise from multiple areas among chemical, emissions, and climatological. These factors help create complex numerical models to predict the future. However, these dispersion models are computationally heavy and expensive in maintenance. The second category refers to data-driven models and machine learning. Various machine learning methods have been applied to predict air pollution with great results. In this thesis, a particular focus has been on deep neural networks and ensemble learning techniques, such as ANN, RNN, RF, GB, and combinations of these, due to their popularity. Table 3.3 presents an overview of the literature listed with methods and the features adopted.

The underlying goal of all literature is to achieve better accuracy in the predictions. The problem definition varies with the size of the horizon, the target pollutants, and the spatial resolution of the forecast. The forecast horizon varies by the range of predictions, from the next hour's values to predicting several days. The air quality models are often predicting values for a single pollutant, but the solutions use the same type of models to include a range of pollutants. Lastly, the research can be split up in station-wise prediction, which targets specific air quality locations and can compare the model's performance easily against the observed values. The second group can be defined as a fine-grained prediction that can generate predictions for a larger area and create a map of the air quality. The latter approach is more data hungry than station-wise predictions, and require a large amount of air quality sensors throughout the city. Station-wise predictions have been widely researched in the literature; however, in the latest years, an increasing trend and a demand for a fine-grained prediction throughout the city. Drivers behind this change are the increased amount of low-cost sensors that easily can cover larger urban cities. Variations of these problem definitions vary by the location, data sets available, and the goal of the researchers. Below is a summary of the literature with a focus on the techniques for exploiting the data sets and the different models used for air quality prediction.

### 3.2.2 Influential Variables

Due to the complexity of air quality behavior, it is crucial to include multiple influential variables. Among the research, the most applied variables are several pollutants and meteorological variables. The different pollutants are often PM, NOx, SOx, Cox, Ozone, and VOC. Meteorological variables are those which describes the weather and the atmospheric composition. In the literature, the meteorological variables differ in the studies by the number of included measurements. The most common meteorological variables are

| ID | Author | Title |
|---|---|---|
| STR001 | Zheng et al. (2015) | Forecasting fine-grained air quality based on big data |
| STR002 | Chen et al. (2016) | Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning |
| STR003 | Bougoudis et al. (2016) | HISYCOL a hybrid computational intelligence system for combined machine learning: The case of air pollution modeling in Athens |
| STR004 | Tamas et al. (2016) | Hybridization of air quality forecasting models using machine learning and clustering: An original approach to detect pollutant peaks |
| STR005 | Zhang et al. (2017) | Early Air Pollution Forecasting as a Service: An Ensemble Learning Approach |
| STR006 | Kök et al. (2017) | A deep learning model for air quality prediction in smart cities |
| STR007 | Fan et al. (2017) | A Spatiotemporal Prediction Framework for Air Pollution Based on Deep RNN |
| STR008 | Li et al. (2017) | Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation |
| STR009 | Ghoneim et al. (2017) | Forecasting of ozone concentration in smart city using deep learning |
| STR010 | Yi et al. (2018) | Deep Distributed Fusion Network for Air Quality Prediction |
| STR011 | Zheng et al. (2018) | A Multiple Kernel Learning Approach for Air Quality Prediction |
| STR012 | Wang and Song (2018) | A Deep Spatial-Temporal Ensemble Model for Air Quality Prediction |
| STR013 | Qi et al. (2018) | Deep Air Learning: Interpolation, Prediction, and Feature Analysis of Fine-Grained Air Quality |
| STR014 | Sinnott and Guan (2018) | Prediction of Air Pollution through Machine Learning Approaches on the Cloud |
| STR015 | Lin et al. (2018) | Exploiting Spatiotemporal Patterns for Accurate Air Quality Forecasting using Deep Learning |
| STR016 | Ghaemi et al. (2018) | LaSVM-based big data learning system for dynamic prediction of air pollution in Tehran |
| STR017 | Soh et al. (2018) | Adaptive Deep Learning-Based Air Quality Prediction Model Using the Most Relevant Spatial-Temporal Relations |
| STR018 | Athira et al. (2018) | DeepAirNet: Applying Recurrent Networks for Air Quality Prediction |
| STR019 | Zhan et al. (2018) | Satellite-Based Estimates of Daily NO2 Exposure in China Using Hybrid Random Forest and Spatiotemporal Kriging Model |
| STR020 | Chen et al. (2018) | A machine learning method to estimate PM2.5 concentrations across China with remote sensing, meteorological and land use information |

**Table 3.2:** The final set of research papers with author and title

| ID | Author | Features | Method |
|---|---|---|---|
| STR001 | Zheng et al. (2015) | O, M, WF | Ensemble of LR and NN |
| STR002 | Chen et al. (2016) | O, T | Semi Ensemble |
| STR003 | Bougoudis et al. (2016) | O, M | Fuzzy Ensemble of RN and FFA |
| STR004 | Tamas et al. (2016) | O, M, WF | MLP |
| STR005 | Zhang et al. (2017) | O, M | Weighted Ensemble of various base learners |
| STR006 | Kök et al. (2017) | O | LSTM |
| STR007 | Fan et al. (2017) | O, M | RNN |
| STR008 | Li et al. (2017) | O, M | RNN |
| STR009 | Ghoneim et al. (2017) | O, M | DNN |
| STR010 | Yi et al. (2018) | O, M, WF | Weighted Ensemble of DNN |
| STR011 | Zheng et al. (2018) | O, M | Multiple Kernel Learning |
| STR012 | Wang and Song (2018) | O, M, WF | LSTM |
| STR013 | Qi et al. (2018) | O, M | Novel NN |
| STR014 | Sinnott and Guan (2018) | O, M, T | LSTM |
| STR015 | Lin et al. (2018) | O, M | RNN and Diffution Convolution |
| STR016 | Ghaemi et al. (2018) | O, M, T | SVM |
| STR017 | Soh et al. (2018) | O, M | Ensemble of ANN, LSTM and CNN |
| STR018 | Athira et al. (2018) | O, M | RNN |
| STR019 | Zhan et al. (2018) | O | RF |
| STR020 | Chen et al. (2018) | O, M | RF |

**Table 3.3:** The final set of research papers with their authors, features and method. The feature name codes are: *O* - Other air pollutants, *M* - Meteorological, *WF* - Weather Forecast, *T* - Traffic

temperature, pressure, humidity, and surface wind with speed and direction. The meteorological variables vary from location to location and affect the air pollutants differently. Various air pollutants and meteorological variables have been extensively studied in the literature (Yi et al. (2018); Zhang et al. (2017); Zheng et al. (2018); Wang and Song (2018); Qi et al. (2018); Sinnott and Guan (2018); Lin et al. (2018); Bougoudis et al. (2016); Tamas et al. (2016); Ghaemi et al. (2018); Soh et al. (2018); Athira et al. (2018); Fan et al. (2017); Li et al. (2017); Chen et al. (2018)). However, other variables such as traffic (Chen et al. (2016); Sinnott and Guan (2018); Ghaemi et al. (2018)), weather forecast (Yi et al. (2018); Wang and Song (2018); Tamas et al. (2016); Zheng et al. (2015)) and other features have been investigated to find relations with air quality changes. These are discussed in the following subsections.

### Traffic

Chen et al. (2016) studied the combination of pollutants with data from traffic and point of interest (e.g., parks, industry, and schools). They show how to create a fine-grained grid to account for the spatiality of air quality. The traffic data source combined with air quality measures provides a clever input for their model. Their model is an ensemble of multiple classifiers that are first trained individually, then combined to select the most prominent

prediction. A different strategy is applied by Sinnott and Guan (2018) that includes the traffic volume recorded within a distance from the target location. They further include this with pollutants data and meteorological features for their LSTM model. Ghaemi et al. (2018) presents an approach by including weather, road, terrain, and traffic data to create a spatial-temporal model. The authors use the fact that low traffic on the weekends lead to lower air quality, and rising during the first days of the week. The spatial data is chosen by the distance to the road together with the wind direction to account for the concentration of pollutants. They use an SVM inspired model to forecast the next 24 hours.

### Weather Forecast

In Zheng et al. (2015) they implement an ANN spatial predictor and an LR temporal predictor. They argue that each of the modules influence should not be static, but rather dynamic based on current weather conditions. Because sometimes local prediction is more important, while spatial predictions should impact more on different occasions (e.g., strong winds). Yi et al. (2018) includes weather forecast as a feature together with air pollutants in a neural network to achieve better independent performance than using historical meteorological features or other pollutants. Wang and Song (2018) implements an ensemble approach to handle different weather patterns. This ensemble consists of multiple weak learners that are trained on different weather patterns. Their technique illustrates that the weather forecast has a significant impact on predicting sudden changes, and on making the predictions more reliable.

### Unique Spatial Techniques

Zheng et al. (2015) and Yi et al. (2018) implements similar types of a spatial transformation component for aggregation, interpolation, and dispersion of neighbouring data. These works by defining a layered circle that centers the station and partitioned into multiple parts which aggregate the air quality levels inside each part. These regions, together with the target station in the center, provide features that are added separately to the external features: meteorological, weather forecast, pollutants, and temporal. These embedded combinations feed a deep neural net that is merged by different methods. The findings of Yi et al. (2018) conclude that a distributed architecture is more suited for air quality than a sequential architecture because each indirect factor has an individual effect on future air quality. This method has its weakness when the data is sparse, where smaller cities might not have a large number of sources. Wang and Song (2018) applies a simpler solution by using Granger causality between stations rather than using geographical distance, due to their reasoning of the latter is too limited to discover the correlation between the stations. Lin et al. (2018) utilize the neighborhood characteristics to represent the spatial correlation, which means two locations would have similar air quality conditions if they share a similar built environment. They conclude that selecting a selection of geographical and neighboring feature that correlate the most have more impact than choosing all the adjacent sources.

#### Unique Temporal Techniques

Zheng et al. (2015) produce air quality prediction with a horizon of 48 hours, using a combined model of an LR-based temporal predictor and an NN-based spatial predictor. However, this approach does not adequately capture the long-term temporal dependencies among the predictions. Zhang et al. (2017) provide a solution with more focus on temporal changes in samples of air pollution data. They tackle the problem where different data samples over space and time have their inherent heterogeneity. Their research shows the performance of a weighted ensemble of multiple base learners, that includes both machine learning models and numerical models. Qi et al. (2018) creates a novel NN model named Deep Air Learning (DAL) that contains separate components for interpolation, prediction, and feature analysis. From the results, they claim that for air quality related data, temporal correlation is more important than the spatial relationship in their experiments. Their experiments include a spatiotemporal regression model that uses historical data of nearby neighboring air quality stations.

### 3.2.3 Air Quality Prediction Methods

Many methods have been applied in the literature, ranging from statistical approaches to more recent advances in machine learning. Of the neural network, deep feedforward and recurrent neural network are the most seen architectures in the literature. Deep learning has demonstrated high performance of learning hidden relations within complex problems, and the more specialized architecture recurrent neural network has shown to be a valuable tool for time series prediction. Additionally, ensemble learning is favorable due to it is prone to noise and variance.

#### Recurrent Neural Network

Multiple research applies variations of RNN to capture temporal dependencies. Wang and Song (2018) includes an LSTM model to learn short-term and long-term temporal dependencies by using the weather forecast. Kök et al. (2017) adopt an LSTM solution on IoT sensor data to predict short-term. Athira et al. (2018) provides a performance overview of different RNN cells and concludes that GRU cell has a slightly higher accuracy of learning PM10 concentration. Li et al. (2017) consist of an LSTM model that considers spatio-temporal relations for predicting air quality concentrations. From their results comparing an extended LSTM (MAPE=11.93%) to SVR (MAPE=28.45%), the deep learning based models exhibit better prediction performance.

#### Artificial Neural Network

Tamas et al. (2016) implements multiple specialized MLP networks for each weather class, determined by clustering. They further learn the relation between a high concentration of air pollutants and different weather classes to improve the classification of sudden spikes. In Ghoneim et al. (2017), they show how a deep learning regression model can learn patterns of pollutants and weather data collected from 449 sensors all around Aarhus city in Denmark. Their DNN model (R2=0.91) can outperform SVM (R2=0.74) to predict the next hour.

ANN inspired models provides excellent performance due to their character of learning hidden relations of both temporal and spatial form. However, ANN models contain many disadvantages, such as the high dimension of hyperparameters, which increase the difficulty of finding an optimal solution. Another con is the over-fitting problem that reduces the generalization of the ANN. Solutions to overcome these known issues is to do feature preprocessing, finding the best architecture for the challenge, implementing well-known regularization methods, and optimizing the hyperparameters for the learner.

#### Ensemble Learning

In Bougoudis et al. (2016) they use fuzzy inference of the results from an ensemble of an RF and FFNN. They combine the power of a non-linear relationship in a neural network and the averaging strategies of an ensemble approach to generalize the results. Zhan et al. (2018) predicts daily NO2 exposure and compares an RF model (R2=0.61) with an LR model (R2=0.38) at a national scale. Chen et al. (2018) also applied an RF model to predict PM2.5 with features including other pollutants and meteorological variables. With their RF model (R2=0.83) provides better performance than their implementation of a generalized additive model (R2 = 0.55).

Zhang et al. (2017) presents a solution with more focus on temporal changes in samples of air pollution data. They tackle the problem where different data samples over space and time have their inherent heterogeneity. A set of base-learners (RF, NN, KNN, SVM, and three knowledge driven models) are each weighted against each sample, to find the most fitted model for that sample. This large multi-channel ensemble outperforms other methods (Stacking, AdaBoost, Bagging, and each base-learner).

### 3.2.4 Norwegian Air Quality Service

A new service for nationwide air quality information service was launched on December 18. 2018 in Norway. It is delivered by the Norwegian Environment Agency (Miljødirektoratet) in collaboration with the Norwegian Public Roads Administration (Statens Vegvesen), the Norwegian Meteorological Institute (Meteorologisk Institutt), the Norwegian Institute of Public Health (Folkehelseinstituttet) and the Norwegian Directorate for Health (Helsedirektoratet)[Bruce Rolstad Denby (2018)]. Through the thesis, this service is referred to as MET. The service is available as a test version, and they mention that errors and shortages may occur. It provides a high-resolution map of air pollutants with hourly 1-2 day forecasts along with details of each pollutants origin (e.g., traffic, industry, shipping, wood burners, or non-local). The map shows the geographical distribution of how much pollution to be in different areas. The level of contamination is shown with four classes of pollution with color codes, green to purple, good to critical.

Their urban EMEP (uEMEP) is a downscaling model of EMEP, a knowledge-driven model which calculates the transboundary transport of air pollutants [Tørseth et al. (2012)]. uEMEP initiates with low spatial data (10km-2.5km resolution) from the EMEP model, that is downscaled down to an approximately 50m grid resolution based on proxy data from each grid. The proxy data consist of meteorological forecasts, historical data of emissions and traffic volume, and geographic variables. Each grid calculates its local contribution of emissions and with a Gaussian model to find non-local concentrations. Notable strengths

of uEMEP are its consideration of all primary sources of air quality pollution with a direct connection to weather forecasts and geographical terrain. In addition to weather forecasts being a strength, it is also a weakness if the forecasts deviate from the real values and can warn of too high or low air quality. They show the best results for modeling NO2 because traffic emissions of NO2 are the best-known of all emissions in Norway. For modeling of particle dust, they mention that the accuracy and uncertainty of these predictions are more significant than of NO2, due to lack of correct traffic data, missing emission sources and the complexity of the station location.

A thorough analysis has been done for a fair comparison the predictions of this service and the contribution from this thesis. The comparison consists of forecasts for each station with all pollutants, for horizons of 24hours and 48hours. Since the service has historical records of estimates from its launch, the machine learning model of this thesis is trained on data up till that date, and predictions till 30. April.

### 3.2.5 Gaps in the Literature

Air quality prediction has been widely researched using popular machine learning techniques. However, several gaps in the literature have been discovered. The most evident is the lack of a strict evaluation framework. The researchers use different problem definitions and evaluation methods to showcase their results. The problems include a large area of combinations: univariate or multivariate, fine-grained or single target predictions, and a window horizon of a few hours or a range of multiple days. Besides, the datasets used in the literature is tied up to the research location. The cities of interest each introduce a new set of data with different distribution and variables. The data is characterized by the cities unique geographical factors, climate, and the city magnitude. The datasets from a town might be different from another or challenging to obtain. These limitations make the promising solutions from the literature challenging to reproduce for other locations. However, this is not the focus of this thesis.

The impact of events and human mobility data on air quality patterns has not yet been studied, as far as we are aware. This kind of data might be of challenge to acquire or includes inaccurate information, and thus not applicable for experiments. The city population tends to gather around more significant events, and therefore, the air quality could have a correlation with this movement of population density. Despite this interesting direction, this was not followed through, due to that, we were not able to acquire the necessary data in time.

The literature includes comprehensive experiments of meteorological, temporal, and spatial techniques. These features are further used to highlight temporal and spatial relations by using feature engineering. These spatial relations are described with neighboring air quality measurements. The temporal relations are represented with historical data and with including information of the timestamp of the measure. Another approach is to include statistical calculations of the time series to complement the features. This approach is less seen from the research and is believed to add even more hidden relations of the complexity in air quality, and is studied in this thesis.

This page intentionally left blank.

Chapter 4

# Architecture and Models

This chapter describes the system architecture and the individual models implemented in this thesis. Section 4.1 outlines all the modules that build up the pipeline from acquiring the datasets, all the way to predictions. Section 4.2 gives a detailed walkthrough of the models in this thesis. Lastly, section 4.2.7 presents the framework and libraries used during the project.

## 4.1 Architecture

The architecture is split up into three modules: Server Module, Machine Learning Module, and a Visualization Module. Figure 4.1 shows a diagram of the structure.

### 4.1.1 Server Module

The server module is responsible for handling all tasks related to data acquisition, storage and provides the predictions through an API. The server includes a task scheduler that

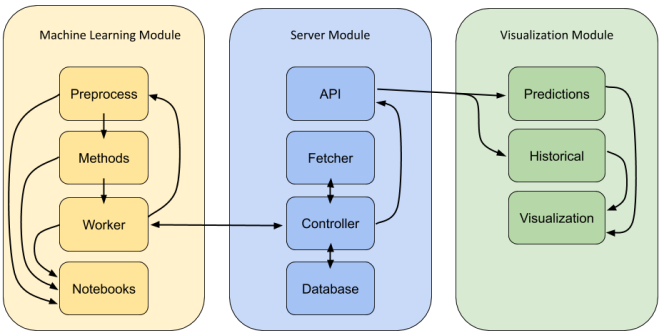



**Figure 4.1:** The architecture of the machine learning module.

updates the data and generates new predictions by calling the machine learning module. These newly generated predictions, together with the historical data, is provided through an API.

#### Data Acquisition

The server module contains a fetching service that collects data from three main sources. These three sources are of pollutants data, the meteorological data, and the traffic data. The fetching service retrieves all historical data from a given year and is updating daily with new observations. All the records are stored in a dedicated database for easier data access.

The pollutants data is obtained from Trondheim municipality. Four stations in Trondheim provides sensors for measuring PM2.5, PM10, NO, and NOx. The information is accessed through an API maintained by Norwegian Institute for Air Research (NILU)[1] Norwegian Meteorological Institute (MET) provides historical weather measurement for Trondheim at one single weather station. The weather data points collected are temperature, pressure, humidity, wind speed, wind direction, and precipitation. Archive of historical weather and climate data is provided by Frost API [2]. The traffic data is maintained by The Norwegian Public Roads Administration and includes measures of the traffic volume at selected roads. They provide an open API to access their numbers of traffic[3]. More details about the datasets are provided in 5.2.1.

### 4.1.2 Machine Learning Module

The machine learning workflow is an iterative process consisting of trial and error of problem hypotheses. In this thesis, we implemented a pipeline of components to speed up each iteration. New ideas could easily extend or replace the current implementation, and new results are analyzed. The machine learning module is responsible for machine learning-related tasks. Data preprocessing, feature engineering, hyperparameter search, training, evaluation, and visualization are all handled by this module. The different steps are shown in Figure 4.2 and described further in details below.

#### Data preprocessing

Data processing involves transforming the raw data into an understandable format. The first step is data imputation, which handles missing values by filling them with substitution values. Missing data from sensor measurements often occur due to malfunction and is no exception in the datasets collected. Table 4.1 gives an overview of the amount and percentage of missing air quality data at the stations. The technique used to fill the missing data is by using the average of neighboring stations that are within a close distance and is inspired by the work of Lin et al. (2018). The nearby stations have similar patterns like the one with missing data, and the error introduced with the dummy data is minimal.

[1] https://api.nilu.no/docs/
[2] https://frost.met.no/
[3] https://www.vegvesen.no/trafikkdata/api/

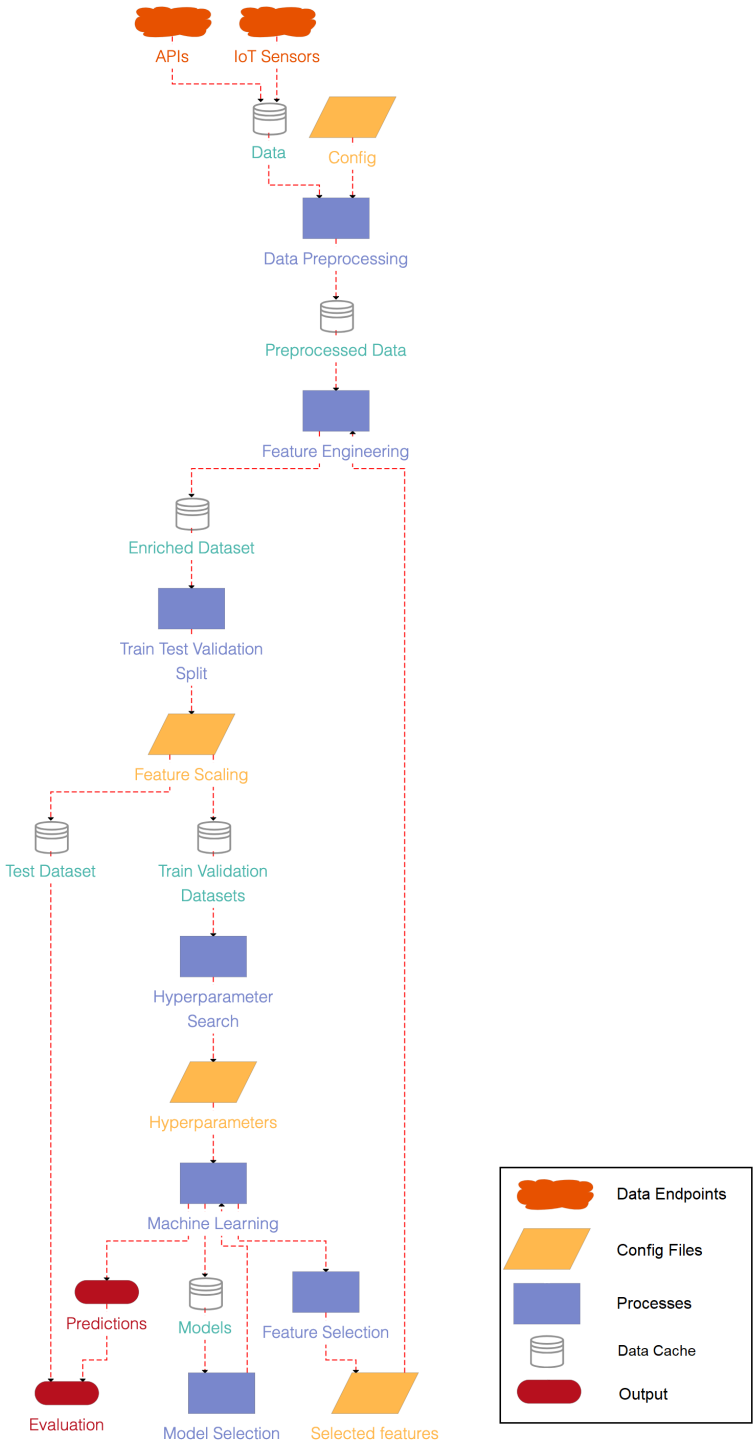





**Figure 4.2:** The architecture flow of the machine learning module.

After the data imputation, the missing values are decreased from 15% to 1%. The remainder of missing values is filled by calculating the weekly mean of all nearby stations at the hour of the missing measurement. The motivation behind filling all missing variables instead of ignoring them is to support the feature engineering strategy described below. It is necessary that the time series is a continuous sequence to include the temporal and statistical characteristics. The data imputation technique does include bias in the models but is considered less of a problem than by ignoring them. Lastly, all pollutants data are clipped for a minimum at zero to remove any negative values that might occur for sensor data. Evaluation of the models does not fill in any missing values. Instead, they ignore the predictions for those timestamps. This will give a stronger evaluation related to real-world observations.

| | Torvet | | Bakke kirke | | Elgeseter | | E6-Tiller | |
|---|---|---|---|---|---|---|---|---|
| Missing | Amount | Percent | Amount | Percent | Amount | Percent | Amount | Percent |
| PM2.5 | 5726 | 13.56% | 3062 | 6.99% | 9962 | 27.01% | 7813 | 20.02% |
| PM10 | 5593 | 13.93% | 2248 | 5.01% | 9823 | 26.53% | 7498 | 19.06% |
| NO2 | 9895 | 26.78% | 4110 | 9.61% | 11015 | 30.75% | 4930 | 11.76% |
| NO | - | - | 4110 | 9.61% | 11015 | 30.75% | 4930 | 11.76% |
| NOx | - | - | 4110 | 9.61% | 11015 | 30.75% | 4930 | 11.76% |

**Table 4.1:** Missing amount and percentages of air quality from the stations in Trondheim.

### Feature Engineering

Feature engineering is a fundamental part of machine learning to make a learning algorithm work efficiently. It requires a thorough analysis of the raw data to build new relevant features in a format that is understood by the algorithms. The goal of feature engineering is to provide strong and ideally simple relationships between new input features and the output target. The complexity in the data is moved to reason with underlying domain knowledge. This section will introduce the new features added from the original datasets. Section 5.2.2 provides further analysis and motivation of the features introduced below.

The features are divided into different types of categories. See Table 4.2 for an overview of all with their shorthand ID, type, critical parameter, and a short description. In this thesis, we deal with 3 kinds of features that measure some qualities in nature. We also identified 3 characteristics of the features which relate to the processing of the input, i.e. the 3 former types of features/measurement.

The temporal features are mainly generated by the use of the timestamp of the measurement. The timestamp includes information of the hour of the day, the day of the week, the day of the month, the month, and the season of the year. The Norwegian holiday calendar is matched against the date to see if it is a day off. The last temporal feature is created out of historical values of the parameter. The optimal number of historical values are discovered in Experiment 1.1 in Section 5.1. The spatial features contain properties from neighboring stations. These are calculated from the mean of the nearby stations of each pollutant. These features are included to help the models capture spatial relations of

the air quality.

The statistical features are produced by applying a set of mathematical functions to the time series to derive unique properties. Table 4.3 shows the formulas for the statistical functions: Lagged value difference, moving average, moving standard deviation, moving minimum, and moving maximum. The goal of statistical features is to add a more general and broader temporal dependency, then by just including historical values. The statistical functions will consist of a smarter relation of the past, that the models will easier learn. The statistical features will provide reliable and more straightforward ties between the past and the forecasts. Statistical feature engineering can help smooth the raw values of the time series to decipher the complexity. The functions minimum, maximum, and moving average can mainly support to capture trends in the series. The difference and deviation can help detect sudden changes by learning what happened just before the change.

The full feature set consist of a high-level feature vector with 655 entries. This large feature space may make it hard for the machine learning techniques to learn, due to new samples are less likely to be similar as previously learned features. With a chance of less correlation between the past and future makes it harder for the model to generalize efficiently. The variance raises along with the possibility to overfit to noise, resulting in reduced performance. However, this limitation of an increased number of features is overcome with multiple regularization methods to account for overfitting.

#### Dataset Split

For evaluating the model's performance, the dataset is split up in three portions. A part is selected for training that is of 80% of the original dataset with a small percentage of it chosen for validation, and the remaining 20% for testing. This first split is done by keeping the time series continuous, so the testing part is the latest measurements. The full dataset ranges from January 1. 2014 to April 30. 2019, which is over 5 years of data. The training data will then be approximately 4 years and 4 months of hourly data points, and the testing data of a little more than a year's measurements. The choice of evaluating the models in this fashion is to get a good idea of how the models will perform in the future when the models are trained on the full dataset before deployed in production. During the 5 years of air quality observations, there are no significant changes in the distribution throughout the years. Consequently, the data can be safely used for predicting the future, assuming the distribution remains stable. However, if the air quality data profoundly change due to some city measures taken, this will impact the predictions due to the predictions are used to avoid air pollution peaks. Also, the predictions made with the test data can be compared with the official Norwegian air quality forecasts, which commenced on December 12. 2018. All the models are trained on a shuffled version of the training dataset, supplied by 10% of the training set for validation. The validation set assists the training process to avoid overfitting. Figure 4.3 shows the distribution of all the pollutants data after the data splitting step.

#### Feature Scaling

For ridge, random forest and the gradient boosting methods, no scaling is done, due to the minimal effect it has on the result. With the neural networks, the features are scaled

| ID | Type | Feature | Description |
|---|---|---|---|
| M | Meteorological | humidity | Hourly average relative humidity (%) |
| | | pressure | Hourly average surface pressure ($Pa$) |
| | | temperature | Hourly average air temperature (°C) |
| | | wind direction | Hourly average wind direction (degrees) |
| | | wind speed | Hourly average wind speed ($m/s$) |
| | | precipitation | Hourly measured sum of precipitation ($mm$) |
| O | Air Quality | PM2.5 | Hourly measured particular dust below 2.5 $\mu g/m$ |
| | | PM10 | Hourly measured particular dust above 2.5 $\mu g/m$ |
| | | NO2 | Hourly measured ($\mu g/m$) |
| | | NO | Hourly measured ($\mu g/m$) |
| | | NOx | Hourly measured ($\mu g/m$) |
| V | Traffic | Traffic volume | Hourly traffic count |
| T | Temporal | hour | Hour of the timestamp (0-23) |
| | | month | Month of the timestamp (1-12) |
| | | day of week | Day of week of the timestamp (0-6) |
| | | day of month | Day of month of the timestamp (0-30) |
| | | holiday | Is the day of the timestamp a Norwegian holiday |
| | | season | The timestamp season (1-4) |
| | | N lagged of X | X parameter during the past N hours |
| C | Statistical | Moving Average | Moving average, $n = [3, 6, 12]$ |
| | | Difference | Difference between previous values, $n = [3, 6, 12]$ |
| | | Deviation | Deviation from previous values, $n = [3, 12, 24]$ |
| | | Minimum | Minimum of previous values, $n = [24, 48]$ |
| | | Maximum | Maximum of previous values, $n = [24, 48]$ |
| S | Spatial | PM2.5 | Mean value of neighbouring stations |
| | | PM10 | Mean value of neighbouring stations |
| | | NO2 | Mean value of neighbouring stations |
| | | NO | Mean value of neighbouring stations |
| | | NOx | Mean value of neighbouring stations |

**Table 4.2:** Table of the final feature set.

to reduce the training duration by assisting the activation functions of the network. The neural networks are utilizing the activation function ReLU and LeakyReLU, which works best if the values are above 0 to avoid vanishing gradients, and below 1 to avoid exploding gradients. Therefore, a min-max normalization function scales the feature values in the range [0, 1] shown in Equation 4.1. The scaling was completed on the training dataset after the split, and the same scaling parameters were reused to scale the validation, the testing set, and the output of the neural network.

$$V = \frac{V - V_{min}}{V_{max} - V_{min}} \tag{4.1}$$

| Name | Formula |
|---|---|
| Lagged Value Difference | $X_{t-n} - X_t$ |
| Moving Average | $\frac{1}{n}\sum_{i=1}^{n} X_{t-i}$ |
| Moving Standard Deviation | $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{t-i} - \bar{X})}$ |
| Moving Minimum | $MAX([X_{t-n}, ..., X_t])$ |
| Moving Maximum | $MIN([X_{t-n}, ..., X_t])$ |

**Table 4.3:** Formulas for statistical features.

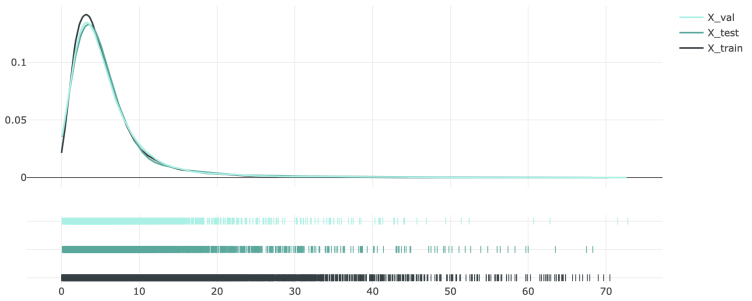



**Figure 4.3:** Distribution plot of train, validation, and test set of all the pollutants data from Trondheim 2014-2019.

#### Hyperparemeter Search

Several hyperparameters were optimized through different optimization rounds for each model. The hyperparameter search was conducted with a random search. The random parameter search helped to narrow down the large parameter space of the hyperparameters. A random hyperparameter search is shown by Bergstra and Bengio (2012) to be empirically and theoretically more efficient than a grid search. The best set of hyperparameters are chosen to train the model. During the search, there was found a strong connection of the same hyperparameters between the model and the window size. The hyperparameter search used, therefore, a window size of 24-hours. In the case of each type of pollutant, the change in results of the hyperparameter tuning for neural networks was significant enough and led to three different sets of hyperparameters for each pollutant (PM2.5, PM10, and NO2).

### 4.1.3 Multi-Output Prediction

Time series data can be transformed into a supervised learning problem with a sliding window method. This method generates pairs of input variables, $X_n$, to an target variable, $Y_n$ which is equal to the next time series $X_{n+1}$. Let us consider the converted training

data set $\mathcal{D}$ of $\mathcal{N}$ instances containing a value for each variable $X_1, ..., X_m$ and $Y_1, ..., Y_d$, i.e., $\mathcal{D} = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$. Each instance is defined by an input vector $m$ descriptive variables $\mathbf{x}^{(l)} = (x_1^{(l)}, ..., x_m^{(l)})$ and a target vector of $d$ variables $\mathbf{y}^{(l)} = (y_1^{(l)}, ..., y_d^{(l)})$, where $l \in \{1, ..., N\}$. The goal is to learn a multi-target regression model from $\mathcal{D}$ by discover a function $h$ which calculates the vector $\mathbf{y}$ given the input vector $\mathbf{x}$:

$$h : \Omega_{X_1} \times ... \times \Omega_{X_m} \rightarrow \Omega_{Y_1} \times ... \times \Omega_{Y_d}$$
$$\mathbf{x} = (x_1, ..., x_m) \mapsto \mathbf{y} = (y_1, ..., y_d),$$

where $\Omega_{X_j}$ and $\Omega_{Y_i}$ express the sample space of each variable $X_j$, for all $j \in \{1, ..., m\}$, and each target variable $Y_i$, for all $i \in \{1, ..., d\}$. The model will then after the learning process to predict all target variables for values $\{\hat{y}^{(N+1)}, ..., \{\hat{y}^{(N')}\}$ of the new input instances $\{\hat{x}(N+1), ..., \{\hat{x}(N')\}$ [Fox et al. (2018)]. See Table 4.4 for the dataset structure described.

| | $X_1$ | ... | $X_m$ | $Y_1$ | ... | $Y_d$ |
|---|---|---|---|---|---|---|
| $x_1$ | 1.5 | ... | 2.3 | 4.5 | ... | 0.5 |
| $\vdots$ | $\vdots$ | | $\vdots$ | $\vdots$ | | $\vdots$ |
| $x_m$ | 1.5 | ... | 2.3 | 3.5 | ... | $Y_d$ |
| $x$ | 2.5 | ... | 2.1 | $\hat{y_1}$ | ... | $\hat{y_d}$ |

**Table 4.4:** Training data $\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), ..., (\mathbf{x}^{(N)}, \mathbf{y}^{(N)})\}$ and new predictions $\mathbf{y} = (y_1, ..., y_d)$ for given input vector $x$.

Further, the problem is transformed to fit the methods in the thesis. For the deep learning methods, no transformation of the dataset $\mathcal{D}$ itself is done. The MLP network is using the dataset as is and no changes are made. For the RNN architecture, $\mathcal{D}$ is sampled in several sequences for the model to iterate over. The sequence length is a hyperparameter that is selected during the optimization search.

For the remainder of methods, except ARIMA, the transformation described is extended due to the limits of some of the methods that do not support multi-output out of the box. For these single-target methods (Ridge, RF, and GBM), a model is built up of $d$ single-target models. Each model are trained on a transformed set
$\mathcal{D}_i = \{(\mathbf{x}^{(1)}, \mathbf{y}_i^{(1)}), ..., (\mathbf{x}^{(N)}, \mathbf{y}_i^{(N)})\}$, where $i \in \{1, ..., d\}$, to predict the single value of $Y_i$. This transformation has a negative trait that the predictions are made independently of each other, and the relationship between them cannot be exploited.

A limitation in this setup is the overfitting problem that occurs due to the large set of training instances that is produced. To avoid overfitting, a great deal of effort has been made to implement multiple generalization techniques for all methods. These are explained in more detail for each model in their respective subsection in Section 4.2.

### 4.1.4 Visualization Module

The visualization module was included to better understand the results of machine learning. This module is the viewing parts of the machine learning with graphs and tables for analysis of the underlying processes. The includes views of the raw, pre-processed and enriched datasets, feature selection process, hyper search results, predictions, and evaluation metrics.

## 4.2 Model Implementation

This section describes the implementation of the models in this thesis.

### 4.2.1 Autoregressive Integrated Moving Average

The Autoregressive Integrated Moving Average (ARIMA) is built as a univariate model to see if it can capture patterns with a simple algorithm. It works by first splitting up a sequence of $n$ timesteps that consists of a historical window and a prediction window. The historical window is dependable on the parameters of the ARIMA model. Before inserting the data into the model, the historical window is differences to make the time series more stationary. Then the ARIMA model calculates the results based and calculates the real values from the differentiated prediction results. The parameters for the model is found in Table 4.5.

### 4.2.2 Ridge Regression

Ridge Regression is implemented with Scikit-learns classifier. This estimator has built-in support for multivariate regression, and works as is for the multi-model architecture. Table 4.5 highlights the single hyperparameter, $\alpha$, used for strength regularization.

### 4.2.3 Random Forest

The implementation of the RF model is using the Scikit-learn random forest classifier class. This meta estimator fits several decision tree classifiers on various sub-samples of the dataset. Table 4.6 shows multiple hyperparameters that have been altered to achieve optimal results. The remaining parameters are as default.

### 4.2.4 Gradient Boosting

Implemented with Microsofts version LightGBM [Ke et al. (2017)]. It is an optimized version of gradient boosting and is faster with the same accuracy than its competitors XGBoost and Scikit Learns version. LightGBM's python API supports multiple boosting variations. In this thesis, we use the implementation of the traditional Gradient Boosting Decision Tree (GBDT), and Dropouts meet Multiple Additive Regression Trees (DART). Both of them with variations of hyperparameters modified seen in Table 4.6, the remaining parameters are set as default defined from the library docs.

### 4.2.5 Multilayer Perceptron

PyTorch is used to implement the MLP model. It consists of multiple layers with the same hidden size, chosen based on the hyperparameter search. The input layer has the size of the number of features, and the output layer matches the window horizon. Every hidden node consists of a LeakyRelu activation function to avoid dying ReLU problem and speed up the training. The hidden neurons initiate with He initialization, which has shown to be a good strategy combined with LeakyRelu shown by He et al. (2015). A dropout chance of 0.3 is added after every layer to achieve better generalization by avoiding overfitting. Also, early stopping is implemented to stop training if the model does not increase its validation performance during training.

The MLP architecture described is trained with the optimization method, Adam [Kingma and Ba (2014)], which empirically has shown high training efficiency. These methods are combined with an adaptive learning rate together with the loss function mean square error that fits the regression task. Lastly, the batch size is implemented to speed up training and increase generalization. Most of said regularization methods and parameters are included in the hyperparameter search and has been carefully tuned. See Table 4.7 for a complete list of hyperparameters for MLP.

### 4.2.6 Recurrent Neural Network

PyTorch is used to implement the RNN model. The implementation can utilize either GRU or LSTM cells. Several model hyperparameters were optimized using randomized search; the RNN cell (LSTM or GRU), number of layers, number of RNN cells, learning rate, sequence length, dropout rate, and batch size. Table 4.7 presents the selected parameters. Shared settings across all were the use of Adam optimizer, LeakyReLU as activation and normal Xavier as the initiation of the hidden space. LeakyReLU is chosen as it is relatively robust to the vanishing/exploding gradient issue. Also, gradient clipping was applied to keep the value in the wanted range for the RNN model. During the search for optimal cell for the RNN, the best performance and lowest train duration were achieved with GRU.

| Parameter | ARIMA | Ridge |
|---|---|---|
| (p, d, q) | (1, 0, 2) | - |
| alpha | - | 0.6 |

**Table 4.5:** Hyper parameter fields for ARIMA and Ridge.

### 4.2.7 Implementation Environment

The research in this thesis are implemented with Python 3.7 [4] and Node 11.9 [5]. In addition, multiple libraries and frameworks have been utilized to do data collection, analysis and predictions.

[4] https://docs.python.org/3/

[5] https://nodejs.org/en/about/

| Parameter | RF | GBM | DART |
|---|---|---|---|
| num_trees | 50 | - | - |
| bootstrap | True | - | - |
| min_sample_split | 2 | - | - |
| min_samples_leaf | 4 | 31 | 31 |
| max_depth | 100 | -1 | -1 |
| objective | - | mse | mse |
| learning_rate | - | 0.05 | 0.05 |
| iterations | - | 800 | 1000 |
| skip_drop | - | - | 0.7 |

**Table 4.6:** Hyper parameter fields for ensemble methods.

| Parameter | MLP | RNN | |
|---|---|---|---|
| layer_size | 512/1024/512 | 1024/512/512 | |
| batch_size | 64/48/64 | 48/96/48 | |
| learning_rate | 1e-5/5e-5/1e-5 | 5e-5/5e-5/1e-4 | |
| num_layers | 2/2/2 | 2/2/1 | |
| activation_fn | LeakyReLU (0.02) | LeakyReLU (0.02) | |
| optimizer | $\Theta_{adam}$ | $\Theta_{adam}$ | $\Theta_{adam}$ |
| dropouts | 0.3 | 0.2 | |
| rnn_cell | - | GRU | |

**Table 4.7:** Hyper parameter fields for neural networks. Parameters for NO2 before the first slash, PM10 is the second option, and the last option is used PM2.5

### Data Handling

Numpy[6] is a Python library for high-performance scientific computing and data analysis. It is built up of operations implemented in C, making Numpy a swift tool to do numerical calculations. Numpy is used in this thesis mostly for feature calculations. Pandas[7] is the most popular Python library for data scientists and built upon Numpy arrays giving its good performance. Pandas offer plenty of features highly valuable for processing a large amount of data. Features include, among others reading, writing, filtering, row and column calculations, reshaping, combining, and selecting. In this thesis, Pandas does the handling of all the large amount of time series dataset from raw datasets to high dimensional features ready for training.

Express[8] is a web framework for Node. It includes an intuitive API with excellent documentation to create a server infrastructure. Express was used to implement the server that fetches real-time data, stored in the database, and provides the predictions with an API. The data collected from the fetching service is stored in a MongoDB database[9].

[6]https://www.numpy.org/
[7]https://pandas.pydata.org/
[8]https://expressjs.com/
[9]https://www.mongodb.com/

MongoDB is a popular storage solution with high scalability and flexibility. The data is stored as JSON-like documents, making them easy to access and analyze the data.

### Machine Learning

Scikit-learn[10] is a well known general machine learning Python library. It consists of a large set of machine learning algorithms for classification, regression, and clustering. Scikit-learn is designed to operate alongside Numpy and Pandas, which makes them a perfect trio for machine learning tasks. One downside of Scikit-learn is that it does not support GPU; therefore, it is not the most optimal solution for neural networks. On the other hand, Scikit-learn offers good support for parallelization with multiple cores that speeds up hyperparameters search, cross-validation, and the multi-model ensemble strategy. Scikit-learn also has a great range of features to streamline data processing and evaluation, used in this thesis code implementation.

Due to the efficiency limits of Scikit-learns version of ensemble algorithms, the gradient boosting methods are implemented using LightGBM [Ke et al. (2017)]. This framework is an improved version of gradient boosting based on GBDT and XGBoost [Friedman (2001)]. LightGBM has improved operational performance with greater training efficiency, lower memory usage, and support higher parallel GPU learning than its precursors.

Since Scikit-learns version of neural networks does not support GPU, PyTorch[11], a highly flexible Python library for deep learning with strong GPU support. Some key features are its simple interface that offers a range of functionality in a Pythonic nature, which makes it integrate well with other libraries and frameworks.

[10] https://scikit-learn.org/stable/
[11] https://pytorch.org/

Chapter 5

# Experiments

This chapter describes the experiments performed in this thesis. First, Section 5.1 introduces the experimental plan. Next, in Section 5.2, a description of the experimental setup with geographical information, dataset specifications, and motivation for the features. Lastly, the experiment results are presented in Section 5.3.

## 5.1 Experimental Plan

The experiments performed in this thesis are designed to answer RQ2 and RQ3 through testing of multiple models with an extensive feature set.

**Research Question 2** Which features have the highest impact on the machine learning algorithm's ability to accurately perform predictions?

Three sub-experiments were completed to determine the influence of the various feature combinations for the models. The different features are tested both individually and in combinations to analyze their impact on air quality prediction. A single method, GBM, is used for the tests in experiment 1, due to it is out of the box ability to show the weight of each feature of the outcome.

**Experiment 1: Determining influential features** involves finding the influence of the features presented in Section 4.1.2.

Experiment 1 is divided into three sub-experiments that focuses on specific sets of features. These feature sets are generated by extending the three natural features with the temporal or statistical feature engineering. A reminder of the feature abbreviations are: $M$ = Meteorological, $V$ = Traffic, $S$ = Spatial, $T$ = Temporal, and $C$ = Statistical. All three sub-experiments are using the same test framework to produce the results. The frameworks start with generating predictions for PM2.5, PM10, and NO2 at the three stations, Bakke Kirke, Elgeseter, and Torvet. The model's output consists of predictions

for 24 and 48 hours. The evaluation method used is k-fold cross-validation with five folds. The evaluation will ensure that the whole dataset is tested with a reasonable good portion of testing data in each iteration. The final set of scores is then grouped to create a summarized version of samples to analyze. Grouped data are a form of aggregating individual observations into different categories.

The first sub-experiment is about the impact of various historical time lag. The features used for the models are: *T*, *T6*, *T12*, *T24*, *T36*, and *T48*. The number is the amount of historical time lag to include in the feature. The results from these features are grouped by window size, and the number of historical data used. The results will then be the average of both station, and pollutants to present the optimal number of historical values to include.

The second sub-experiment is about the temporal influence of different features. The same framework as the previous sub-experiment is used, but a different feature set is applied. To validate the temporal difference of the features, they are extended with the temporal feature engineering: *T*, *MT*, *ST*, *VT*. The results are then grouped by pollutants and feature to analyze their differences. The difference between the first sub-experiment and the second is that the first focuses on how much historical data is optimal for accuracy and training time, while the second sub-experiment is about which features are most valuable to extend with historical values.

The third sub-experiment applies the statistical feature engineering to validate its influence on the models. The features applied are: *C*, *MC*, *SC*, *VC*. The results are then grouped in pollutants and features, to see the influence of the statistical component. These three sub-experiments will together account for experiment 1, and demonstrate the most influent features.

**Research Question 3** How accurate are machine learning methods for predicting air quality in Trondheim?

A range of different machine learning algorithms was implemented to answer RQ3. The results provide a comparison between statistical approaches, ensemble learning, and neural networks for air quality prediction. The evaluation of the different machine learning algorithms will establish an objectively answer to RQ2.

**Experiment 2: Comparison of machine learning models** show the results of the model's performance. The models differ in their design in a more extent than just by architecture, and the comparison is followed by a few guidelines to highlight these differences. Firstly, the ARIMA is a recursive univariate method and is included to compare the performance of the popular statistical solution. Secondly, the learners Ridge, RF, GBM, and DART are concerned with the performance of the multi-output strategy of the single target regressors. Lastly, the neural networks, MLP and GRU, are both designed for multi-output regression to compare all the mentioned models with deep learning. The models are trained on data from January 1. 2014 to mars 1. 2018, and tested on data until 30. April 2019. The test data is 20% of the dataset with 10,200 data points. The models in experiment 2 are using the full extent of the feature set (*MSVCT24*). More detailed information of the datasets are presented in 5.2.1.

Hyperparameters were optimized by a thorough random search for 24 hours predic-

tions, and the same parameters used for 48 hours prediction. The use of the same hyperparameters is not an optimal solution, but due to limited time, we did not do this analysis. The hyperparameters for the different methods is presented in Table 4.5, Table 4.6, and Table 4.7. All of the models produce predictions for each of the pollutants PM2.5, PM10, and NO2 for the stations Bakke Kirke, Elgeseter, and Torvet. The results are the mean of all stations presented with the metrics RMSE, MAE, RAE, R2, and SMAPE. The different visualizations of the results are to highlight the model's different strengths and weaknesses. It includes evaluation of the model's predictions against the different pollutants, the window horizons, and their ability to predict sudden changes in air pollution.

**Experiment 3: Comparison of the predictions versus official forecast** is the last experiment and focuses on the best model's performance from the previous experiment. The predictions are compared with the national air quality forecast on Norway. The same models trained in experiment 2 are reused, but with the test dataset ranging now from 12. December 2018, the start date of the national air quality forecast, to 30. April 2019. The results from experiment 2 and 3 will then involve different results due to the shorter test set. The results include the prediction of 24 and 48-hour for each station and pollutant. Three different tests are presented to evaluate a comparison of the performance. The first is the evaluation of the regression error. The second is concerned with the performance of anomaly prediction, with accuracy metrics including the precision, recall, false alarm ratio, and F1-score. Lastly, the skill of the forecast is presented. The skill score is the relative accuracy of the estimates over a persistence forecast. A persistence forecast says tomorrow is the same as today, based on observations.

### 5.1.1 Evaluation Metrics

In the literature of air quality, there is no single superior evaluation method. Therefore, a set of multiple performance metrics are applied to evaluate the experiments. A full table with their respective equations are given in Table 5.1, including the following metrics: Mean Absolute Error (MAE), Relative Absolute Error (RAE), Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), and R-squared ($R^2$). In the equations, $\hat{y}_i$ is the predicted value of the $ith$ sample, and $y_i$ is the corresponding true value. $\bar{y}$ is the mean of the observed true data. For all regression metrics, except $R^2$, a lower score is better.

| Metric | Definition | Equation |
|---|---|---|
| **MAE** | The mean absolute error of N forecasting | $\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$ |
| **RAE** | Relative Absolute Error | $\frac{\sum_{i=1}^{n}\lvert y_i-\hat{y}_i\rvert}{\sum_{i=1}^{n}\lvert y_i-\bar{y}\rvert}$ |
| **RMSE** | The square root of the mean square error | $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$ |
| **SMAPE** | The mean absolute percent error of N forecasting | $\frac{2}{n}\sum_{i=1}^{n}\frac{\lvert y_i-\hat{y}_i\rvert}{\lvert y_i\rvert+\lvert\hat{y}_i\rvert}$ |
| **R2** | Coefficient of determination | $1-\frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$ |

**Table 5.1:** Evaluation metrics.

MAE is the sum of the absolute differences between predictions and actual values. It gives an idea of how wrong, or the magnitude of the error in the predictions. RMSE, in contrast to MAE, punishes large errors by considering the impact of extreme values. If the predicted value deviates too much from the true target, it could ultimately lead to a completely false sense of the result, which makes punishing significant errors desirable. Unlike MAE and RMSE, the RAE can be compared between models of different units. RAE is comparable due to being defined as the absolute error with a fraction of the actual residual error. SMAPE is commonly used in statistics for evaluating the accuracy, and is easy to interpret by being a measure of "percentage error." Its limitation is that it puts a more substantial penalty on negative errors than on positive errors. The method is then biased that it will select a forecast that has lower predictions than the actual values. The $R^2$ metric indicates the fit of forecasts to the real measures. This measure is called the coefficient of determination and is a value up to 1, which indicates a perfect fit. However, the score can be negative because the model can be arbitrarily worse.

### 5.1.2 Evaluation Metrics for Anomaly Prediction

In addition to normal air quality patterns, it frequently occurs sudden changes in the pollution concentration. These sudden spikes or anomalies are important to detect for real-time monitoring as they can have more impact on the daily life of most people. While the evaluation metrics defined in the previous section to covers the total error and how good the model fit the actual values, it is not a suitable metric for anomaly prediction. The regression problem is therefore transformed into a classification problem by labeling sudden changes above a threshold as an anomaly. Wording for anomaly prediction in this thesis will also be known as air quality spikes and sudden changes.

| **Pollution level** | **Health Risk** | **PM2.5** ($\mu g/m$) | **PM10** ($\mu g/m$) | **NO2** ($\mu g/m$) |
|---|---|---|---|---|
| **Low** | Minimal | <25 | <50 | <100 |
| **Moderate** | Minor | 25-40 | 50-80 | 100-200 |
| **High** | Significant | 40-150 | 80-400 | 200-400 |
| **Very high** | Severe | >150 | >400 | >540 |

**Table 5.2:** Warning classes for hourly pollution levels

The problem transformation makes the time series data into an array of local maximums above a threshold. The predictions and the actual observed values are first resampled bi-hourly to smooth the time series to reduce outliers. Further, all maximums are identified that are above the moderate pollution level for the target pollution. See Table 5.2 for an overview of the thresholds for the different pollutions and the warning levels. "Moderate" pollution level is chosen due to the air quality in Trondheim has very few to none occurrences of a high pollution level. Furthermore, the anomalies of the predictions are matched against the real observed time series and are counted as a hit if the anomaly point is within 1 hour in the future and 1 hour in the past. The smoothing and interval

calculation will then account for a range of 4 hours that needs to overlap. The interval of 4 hours is fine since a typical sudden change lasts for about 4-6 hours, and there are few partial overlaps of lengthy anomalies in the time series dataset. This straightforward approach for anomaly prediction ignores the residuals of the predicted spikes, but it related well of classifying the specific warning levels. These warning levels (good, OK, or bad) are a simple indicator for the cities population to grasp the air quality at their location. See Figure 5.17 for an example of the predictions along with anomaly hits.

The comparison of maximums from the real observation set versus the predictions, we have extracted out the true positives (*tp*) that are correct hits, false positives (*fp*) that are false alarms, and false negatives (*fn*) which implies a classification miss. Finally, these variables are used to calculate the F1-score, False Alarm Ratio, Recall, and Precision, which is presented in Table 5.3. Precision is the fraction of all detected anomalies that are real anomalies. Precision is a good evaluation when the costs of a miss are high. The recall is the fraction of all real anomalies that are correctly classified. The recall is a metric of importance if the cost of false predicted anomalies is high. With the problem of air pollution, it is not desirable to forecast too many wrong sudden changes. On the other hand, one does not simply want to miss any high spikes as well. A commonly used metric which takes into account both problems is F1-Score, which is the harmonic mean of precision and recall. For all classification scores, except the false alarm ratio, a higher score is better.

| Metric | Definition | Equation |
|---|---|---|
| **False Alarm Ratio** | Fraction of false predictions | $\frac{fp}{fp+tp}$ |
| **Recall** | Probability of detection | $\frac{tp}{tp+fn}$ |
| **Precision** | Precision of the detection | $\frac{tp}{tp+fp}$ |
| **F1-score** | Harmonic mean of precision and recall | $2 * \frac{Precision*Recall}{Precision+Recall}$ |

**Table 5.3:** Evaluation metrics for sudden changes

## 5.2 Experimental Setup

This section describes how the experiments are set up, including the datasets and feature extraction for the machine learning approaches.

The datasets used in this thesis has a unique climate, weather, and terrain based on the geolocation of Trondheim. Trondheim (63°26' N Latitude and 10°25' E Longitude) is situated in the middle of Norway and is the fourth largest urban area in the country, with a population of around 200 000. Trondheim has an oceanic climate with typical mild summers and mild winters, with a narrow annual temperature. The study region is mostly protected from the strong south and southwest winds that can appear along the outer coast. The average precipitation is 873 mm yearly with moderate snowfall from November to March, although this is often mixed with mild weather and rainfall. The

temperature ranges from an average low of -4.9°C in January to an average high of 18.9 in July [Norwegian Meteorological Institute (2019)]. What distinguishes Trondheim from other urban cities is the sudden changes in weather in a matter of hours or days. The weather can go from the sun and 18 degrees a day, to zero degrees with cold sleet the next day, and then back again with sun and warm weather after that. The city is known for its four seasons days.

### 5.2.1 Datasets

This research utilized three different datasets of Trondheim city: Air pollutants, historical weather observations, traffic volume count, and wood burner dataset. These are described in the following section, and a statistical description of the datasets can be found in Table 5.5, Table 5.6, and Table 5.7 respectively. Figure 5.1 gives an overview of the city along with the position of the monitoring stations.

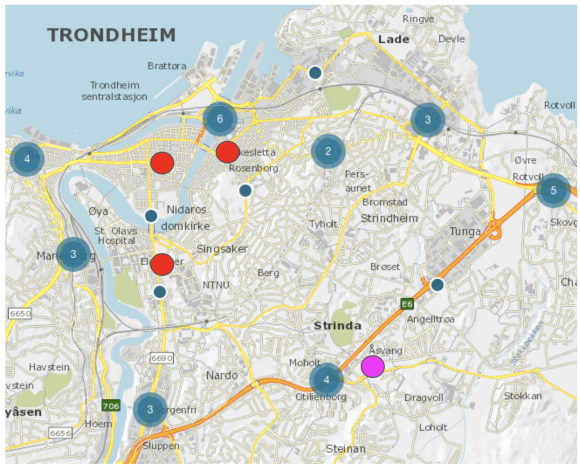



**Figure 5.1:** Map of the location of data stations in Trondheim, where red marks air quality stations, pink is a weather station, and blue (small and large) is traffic stations. The numbers within the circles are an indication of the total number of stations in that area.

#### Air Quality Monitoring Stations

There are four air quality monitoring stations measuring pollutants data in Trondheim. These are expensive sensors with reliable data of high quality. The Norwegian government is maintaining the stations, and the data is provided by the Norwegian Institute for Air Research (NILU) with an open API. Three of the stations: Bakke Kirke, Elgeseter, and Torvet are in close range of the city center, while E6-Tiller is 8km south. E6-Tiller is excluded in the predictive models, due to this distance. This exclusion is to strengthen the results of the neighboring effects used in the thesis. Table 5.4 describes the stations along with the date of the first measure, sensor owner, and the different pollutants measured. The data from Elgeseter ranges back to January 2000, while E6-Tiller was established

in late 2013. Air quality has improved during the years with a positive difference, due to initiatives done by the municipality. These are actions, among others, including road cleaning and dust suppression [The Norwegian Public Roads Administration (2019b)]. Thus the analysis and machine learning model's are utilizing data from January 1. 2014 to 30. April 2019 to avoid learning on too old data with unrelated distributions.

| Name | First Measure | Owner | Components |
|---|---|---|---|
| E6-Tiller | 2013-12-20 | Trondheim Kommune | PM2.5, PM10, NOx, NO2, NO |
| Bakke kirke | 2004-04-19 | Statens Vegvesen | PM2.5, PM10, NOx, NO2, NO |
| Elgeseter | 2000-01-05 | Statens Vegvesen | PM2.5, PM10, NOx, NO2, NO |
| Torvet | 2009-01-15 | Trondheim Kommune | PM2.5, PM10, NO2 |

**Table 5.4:** Table of air quality stations of high quality in Trondheim.

### Weather Dataset

The weather dataset contains hourly data recorded at a station at Voll in Trondheim. The weather station sensors include reading of temperature, precipitation, humidity, pressure, wind speed, and wind direction. The dataset consists of historical measures with the same range as the air quality stations, January 1, 2014, to April 30, 2019. A limitation of the weather dataset is that it consists of measures from one weather observation station. A single station implies that all weather observations is the same all over Trondheim, which is not the optimal input for the analysis.

### Traffic Dataset

The traffic data consists of traffic information on the road network in Trondheim. Figure 5.1 gives an overview of the multiple traffic counting stations in Trondheim. These stations detect vehicles utilizing inductive loops in the roadway and are further aggregated, and quality assured [The Norwegian Public Roads Administration (2019a)]. The recorded variables are hourly vehicle count in both driving directions. This thesis is using the sum of passing vehicles of both driving directions and assumes that this sum of recordings is sufficient for analyzing the traffic relations to air pollutants. At each air quality station, the data from the closest traffic stations with a road connection is combined to a single traffic series. Consequently, some of the air quality stations will use the same traffic station data for aggregating the total amount. The data from many of the traffic stations are sparse and is missing quite some periods. Therefore a combination of multiple stations will ensure a complete series. Five of the traffic stations is used to generate the features for the three air quality stations in Trondheim centrum. This technique is inspired by the work of Sinnott and Guan (2018), where they present a similar feature of volume extraction based on distance.

### 5.2.2 Dataset Analysis

An analysis of the datasets is conducted to extract features based on the hypothesis of air quality behavior in Trondheim. The results from this analysis will, in turn, strengthen the research questions. The following hypothesis is the drivers behind the features included in further experiments. The last hypothesis, marked as "Hx", do not play a large role in the experiments, but are included for short inspiration for future work.

| | PM2.5 | PM10 | NO2 | NO | NOx |
|---|---|---|---|---|---|
| **count** | 46792 | 46792 | 46838 | 46838 | 46813 |
| **mean** | 5.95 | 13.63 | 25.78 | 38.02 | 66.60 |
| **std** | 5.47 | 13.14 | 19.04 | 37.56 | 68.55 |
| **min** | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| **25%** | 2.77 | 5.57 | 11.50 | 12.74 | 21.04 |
| **50%** | 4.53 | 9.89 | 20.90 | 26.38 | 45.08 |
| **75%** | 7.25 | 17.06 | 34.92 | 49.82 | 87.15 |
| **max** | 126.97 | 297.72 | 140.40 | 387.14 | 839.61 |

**Table 5.5:** Statistical description of the air quality dataset.

| | humidity | pressure | precipitation | temperature | wind angle | wind speed |
|---|---|---|---|---|---|---|
| **count** | 46839 | 46398 | 42907 | 46839 | 46350 | 46839 |
| **mean** | 73.60 | 994.36 | 0.09 | 6.09 | 180.47 | 2.59 |
| **std** | 16.13 | 12.67 | 0.35 | 7.10 | 77.78 | 1.62 |
| **min** | 15.0 | 944.0 | 0.0 | -17.10 | 0.0 | 0.0 |
| **25%** | 61.0 | 987.0 | 0.0 | 1.10 | 138.21 | 1.45 |
| **50%** | 76.0 | 995.50 | 0.0 | 5.80 | 197.0 | 2.27 |
| **75%** | 87.0 | 1002.90 | 0.0 | 11.10 | 223.83 | 3.33 |
| **max** | 100.0 | 1034.60 | 10.50 | 31.50 | 360.0 | 15.30 |

**Table 5.6:** Statistical description of the weather dataset.

| | Torvet | Elgeseter | Bakke kirke |
|---|---|---|---|
| **count** | 23659 | 20253 | 9552 |
| **mean** | 391.74 | 462.61 | 185.01 |
| **std** | 274.85 | 311.75 | 124.45 |
| **min** | 0 | 0 | 0 |
| **25%** | 141.50 | 165.25 | 71.00 |
| **50%** | 372.00 | 453.00 | 169.20 |
| **75%** | 593.50 | 729.56 | 284.00 |
| **max** | 1350 | 1350 | 834 |

**Table 5.7:** Statistical description of traffic dataset

### H1: Air Quality Features

The air quality features consist of ground truth values of PM2.5, PM10, NO2, NO, and NOx. Based on the investigation of the correlation heatmap presented in Figure 5.2, there is a strong correlation between the pollutants. Most notable is the strong relationship between the nitrogen oxide family, within and between each monitoring station of 0.75 and higher. The correlation between particular matter and nitrogen is somewhat lower at approximate 0.4. However, this correlation still indicates that both pollutants may have the same root source, also with the same patterns through time. The relationship between PM10 and PM2.5 have a strong station wise correlation of above 0.75 and with a lower degree of relations between the stations. Based on this conclusion, all of the pollutants are highly essential to predict future air quality concentrations. The dataset distribution of air quality pollutants PM10, PM2.5, and NO2 can be seen in Figure 5.3. PM2.5 has the highest density of lower values between 0 and 15, and with a sporadic distribution of higher values up to 100. PM10 consists of a lower distribution top with a small right skew, besides with even more sporadic distribution of higher values up to around 200. Lastly, NO2 with the lowest top, and with a far-right skew with high values scattered up to mainly 150.

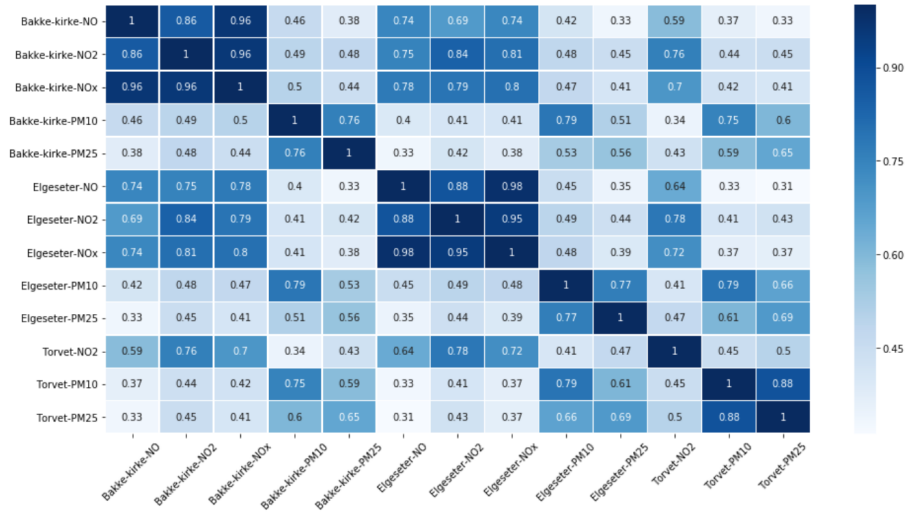



**Figure 5.2:** Spearman correlation heatmap with correlation coefficients of air pollutants levels from the stations Bakke kirke, Elgeseter, and Torvet.

### H2: Meteorological Features

A common term in the literature of air quality prediction is the inclusion of historical meteorological features due to the strong relationship with air pollutants. Occasionally the weather situations change swiftly and are directly reflected on the measured concentration of pollutants. For an understanding of the root cause of the pollutants and its movement,

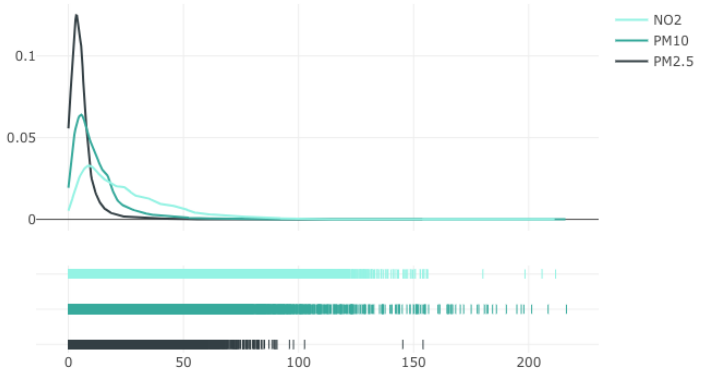



**Figure 5.3:** Distribution plot of PM2.5, PM10, and NO2 of the sum of the pollutants at all stations in Trondheim.

it is necessary to combine the study of the weather during the observation, the flow of air, and the source of release. The meteorological variables have different influence on the various air pollutants and can have both negative or positive effects. Rain washes out water-soluble pollutants like particulate matter, and the wet conditions might reduce the dispersion. In the winter months, the precipitation typically occurs in snow form, and instead of washing the pollutants, it acts as a blanket. Weeks of accumulated dust in layers of snow is later released when the temperature is rising. Wind causes the dispersal and dilution of the pollutants. High wind speed could decrease pollutants concentration of one location and rapidly transport pollutants hundreds of kilometers. Geographic area, city structure, and wind direction are all factors of the air flow. Reduced wind speed could gather air pollutants in one area and decrease air quality. Besides, there are indirect effects of weather on air quality. For example, when it is colder fireplaces are used more and people tend to travel more regular in motorized forms of transportation.

Figure 5.4 shows a pair plot to see the distribution of the relationship between temperature, precipitation, and wind speed against PM2.5, PM10, and NO2. It shows a low correlation of the temperature, but a slightly larger relation with PMx, than NO2. High values of precipitation tend to lower the concentration of PM2.5 and PM10. NO2 have a much lower affection of the precipitation compared with the particular matter. High winds ensure a low level of PM2.5. While PM10 and NO2 are less affected, but it has a declining trend when the winds are increasing.

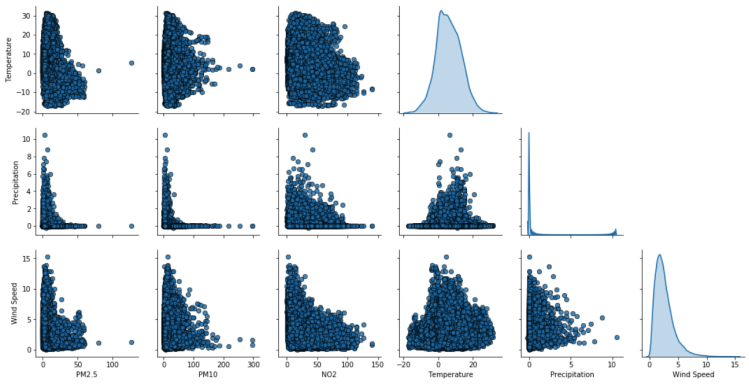



**Figure 5.4:** Pair plot of selected weather measures (temperature, precipitation, and wind speed) and the pollutants (PM2.5, PM10 and NO2).

### H3: Traffics correlation with air pollutants

The hypothesis is that wear, tear, and emissions from traffic are profoundly affecting the PM10 and NO levels. Figure 5.5 shows a Spearman Correlation between traffic in Trondheim center and the nearby stations Elgeseter, Torvet, and Bakke Kirke. Notably, we find that the traffic data correlates the strongest between the nitrogen oxides at the stations with the highest traffic. The average traffic at Elgeseter(460 hourly) is more than double than for Bakke Kirke (185 hourly) and is illustrated through the correlation with NOx at 0.75 and 0.6 respectably. The distance a station has from traffic also has a significant impact on influence, as seen from the correlation from Torvet with a value of below 0.50.

For particulate matter, the traffic does not have a high correlation as the nitrogen oxides. PM10 has a score between 0.30 and 0.35 and do not show a relation between the traffic volume. This last notice may be due to the dust to be more spread around in the city by air flow, and is more resistant to external factors and thus does not disappear so quickly. For PM2.5, the correlation is half of PM10 and is not considered further. This thesis applies this feature by taking the mean of the closest traffic stations in a radius of 1km of the target.

### H4: Temporal features

The construction of temporal features is approached by extracting predictive patterns of sequential data. Temporal features provide valuable information about previous time steps and are essential for including long-term and short-term memory to the model. Temporal inclusion can be as simple as including values from previous time steps. Where recent events have a stronger influence on current status, while past events have a weaker influence.

As seen in Figure 5.6b, Figure 5.6d, and Figure 5.6f there is a clear visual correlation between the hour of the weekdays and the pollution level. From approx 05:00 to 7:00, there is a double of PM2.5 and even larger rise of PM10 and NO2 in air pollution in the

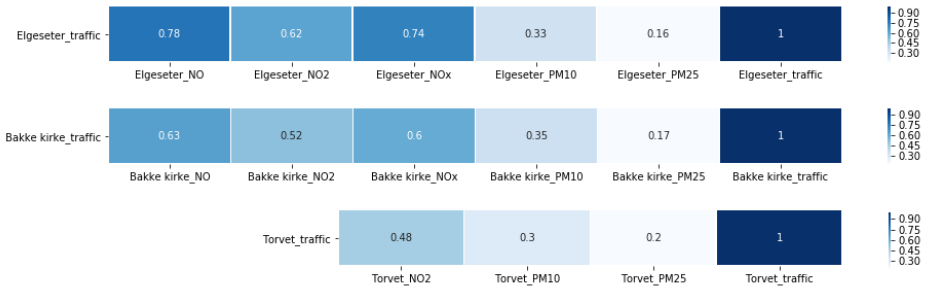
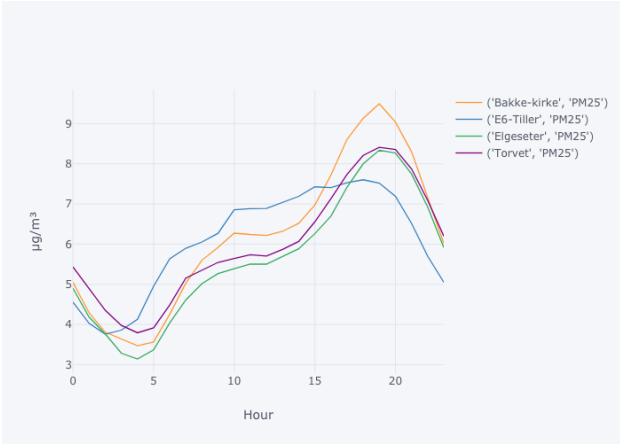



**Figure 5.5:** Spearman correlation between traffic and pollutants at three of the stations

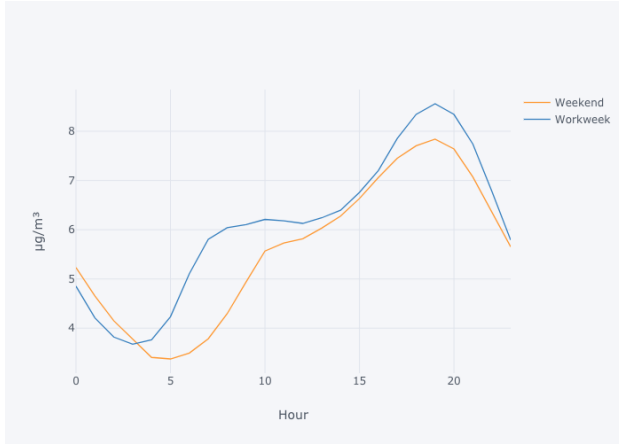
morning. This rapid increase is mainly due to traffic-related emissions. This graph reflects a regular workday from eight to five, with a slow decrease in activity in the late evening. An interesting part to notice is the decrease in NO2 start 4-5 hours earlier with a lower descending rate than the PMs. In the weekend the pollutants have a slight increase during the morning, and peaks between 16:00 and 19:00. PM2.5 differs from PM10 and NO2 by having the most similar pattern between weekdays and weekend. The main difference is by overall lower values, while the patterns of PM10 and NO2 also differ to the hour of occurrence and number of local maximum and minimum. The same trends for PM10 and NO2 strengthen the assumption of traffic-related emissions, which is a major source of PM10 and NO2 in Trondheim. Also it correlations well with the traffic graph of workweek and weekend, shown in Figure 5.7b

Figure 5.6a, Figure 5.6c, and Figure 5.6e presents the difference of air quality measured between the four stations in Trondheim. Most notable are values of PM10, and NO2 varying by more than a double. The explained difference might be due to the station Torvet is located further from traffic, while the other three are in close approximation. Also, E6-Tiller is located close to a busy motorway, explaining the highest values. This effect is not present in Figure 5.6a of PM2.5 at the stations, and Bakke Kirke has the largest average peak in the evening. By taking a look at a heatmap of all the wood burners in Trondheim, presented in Figure 5.8, the density is higher around Bakke Kirke than the other stations. This evening peak of PM2.5 at Bakke Kirke is highly due to the relation of the high amount of wood burners. Besides, the trend of PM2.5 during the winter is more than double of the other seasons, which goes together with the citizens wanting to warm up their residents.

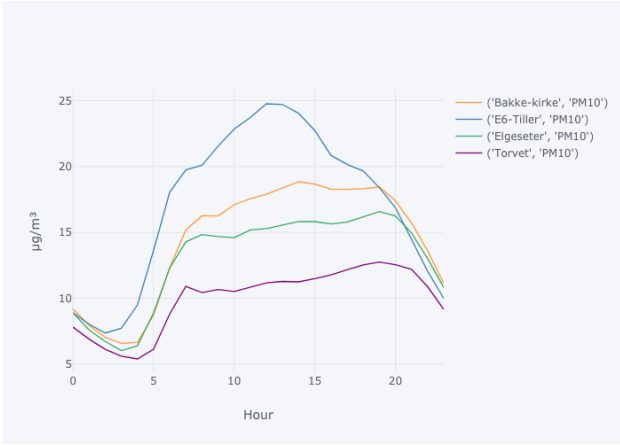
The discussion above concludes that human-made causes play an essential role in the air pollution level. In this thesis, the temporal extracted from the timestamp is the hour, month, day of the week, day of the month, and the season. Also, the Norwegian holidays are added as a feature, due to these days typically has the same effects on people as the weekends. Lastly, to account for historical patterns, a lagged version of the variable is calculated by a shift operation of $N$ multiple steps. This temporal feature extraction technique has shown high performance in several of previous work.



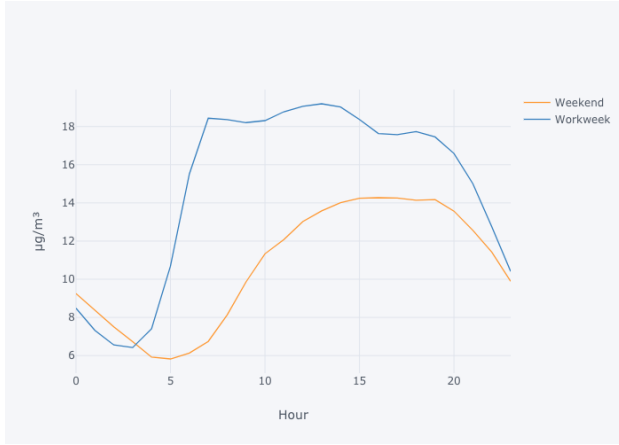

**(a)** Mean PM2.5 grouped by hour of day and station.



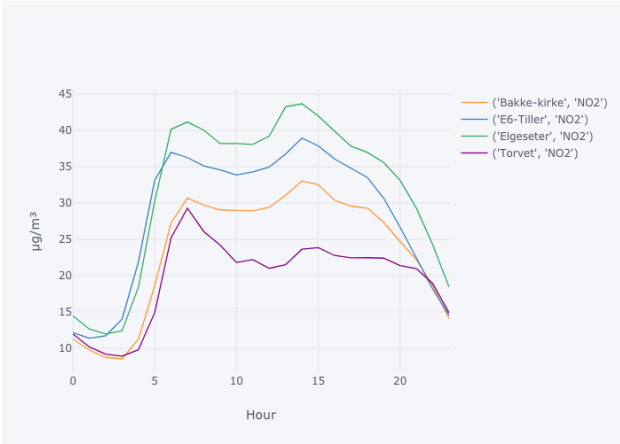

**(b)** Mean PM2.5 grouped by workweek or weekend.



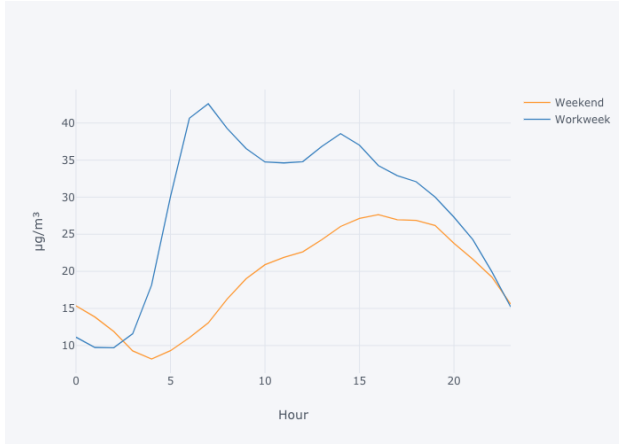

**(c)** Mean PM10 grouped by hour of day and station.




**(d)** Mean PM10 grouped by workweek or weekend.



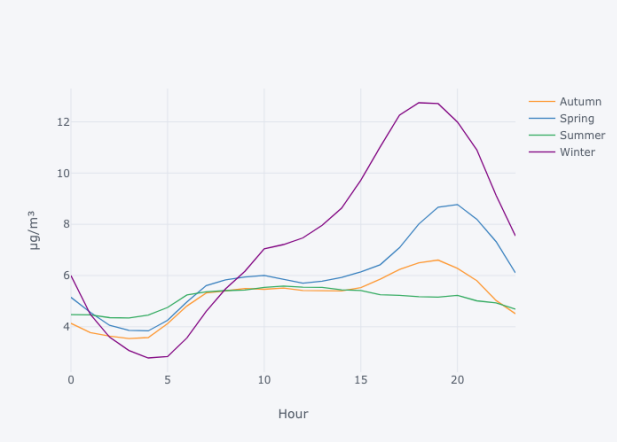

**(e)** Mean NO2 grouped by hour of day and station.



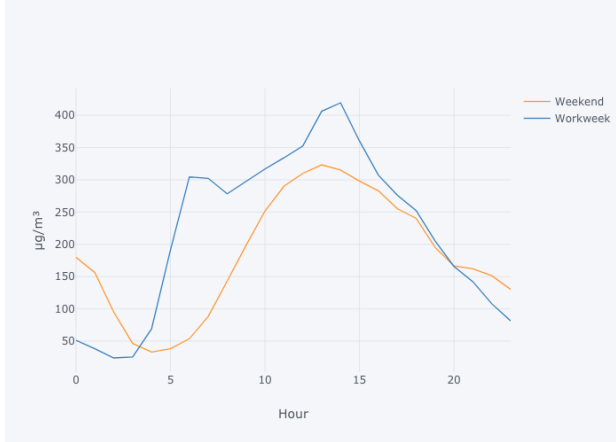

**(f)** Mean NO2 grouped by workweek or weekend.



**Figure 5.6:** Temporal features of the pollutants PM2.5, PM10, and NO2 in Trondheim grouped by hour of the day.




**(a)** Mean PM2.5 grouped by hour and season




**(b)** Mean Traffic grouped by workweek or weekend.

**Figure 5.7:** Temporal features of seasonal PM2.5 and traffic grouped by workweek and weekend.

#### H5: Spatial Features

Figure 5.2 presents a heatmap of the correlation coefficients between the three stations in Trondheim that are located close to each other. It is shown a notable relation within the pollutants of the different stations. Also, there is a slightly larger correlation between the station Bakke Kirke and Torvet, than Bakke Kirke and Elgeseter. The further distance between the latter can explain this correlation. A mean of neighboring air quality stations is calculated based on the distance to the target, to consider the spatial properties of air quality in Trondheim with the sparse amount of observation locations. The similarity of neighboring sequences has been shown to improve the accuracy in Lin et al. (2018) and is included in this thesis as well.

#### Hx: Heating accounts for most of the PM2.5 levels

Due to the impact heating and burning of wood has on the air quality, the models are dependant on learning this relation. By combining the idea of cold temperature readings and fireplace locations (clusters) to predict areas of high emissions. In combination with rain and wind features to account for the movement of the air pollutants released from the households. See Figure 5.8 for an overview of wood burners in Trondheim.

## 5.3 Experimental Results

A series of experiments were conducted to validate the research questions RQ2 and RQ3. First, a set of sub-experiments were carried out to find the most influentially features to predict air quality in Trondheim. Next, a comparison of the different model's ability to produce accurate predictions. Lastly, a comparison of the best methods of the thesis, with the official air quality forecast service in Norway.

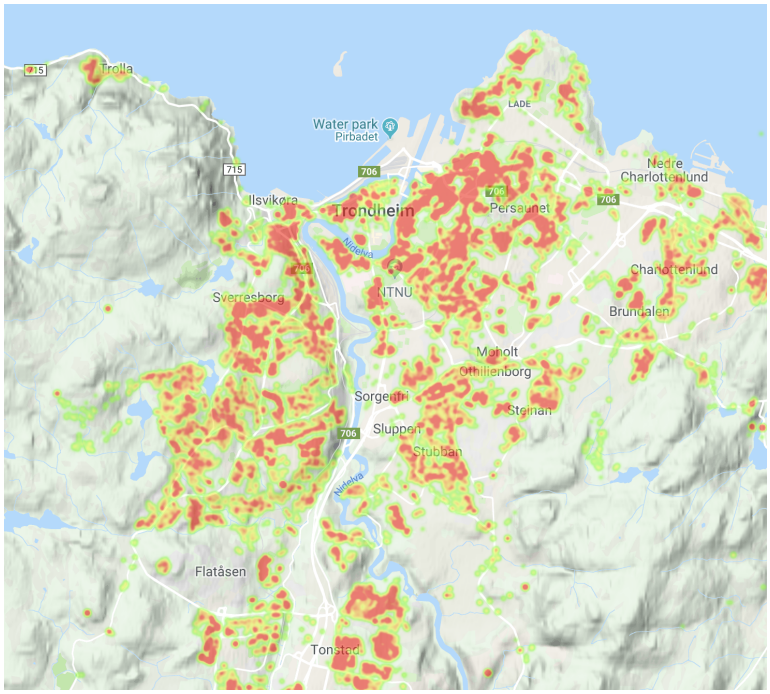



**Figure 5.8:** Heatmap of wood burners in Trondheim.

### 5.3.1 Experiment 1: Determining influential features

Three sub-experiments were completed in order to find the different feature impact. These experiments use the multi-output strategy with the GBM with different feature configurations. Figure 5.9 shows the increase in performance when including more historical lag to answer for the historical feature influence. Figure 5.10 and Figure 5.11 shows the results of the feature influence when combining the temporal (*T24*) and statistical (*C*) feature engineering with the three base features (*S*, *V*, and *M*).

### 5.3.2 Experiment 2: Comparison of machine learning models

This section presents the results of all models trained with the full feature set (*MSVCT24*). The models are trained on data from 1. January 2014, to 30. November 2018, and tested on data until 30. April 2019. The results are split up into two evaluations with the first concerned with the model's regression error for general air quality pattern and the second for its classification accuracy toward anomaly prediction of sudden changes and spikes. All results are shown in the tables below, while the figures are a summarized version of the results to highlight the results. Note that the graphs y range is modified in some of the graphs to highlight the differences better. The reader is advised to look up the results table, referred below, for exact results.

The results for the regression error in Table 5.8 showing the MAE, R2, and RMSE of

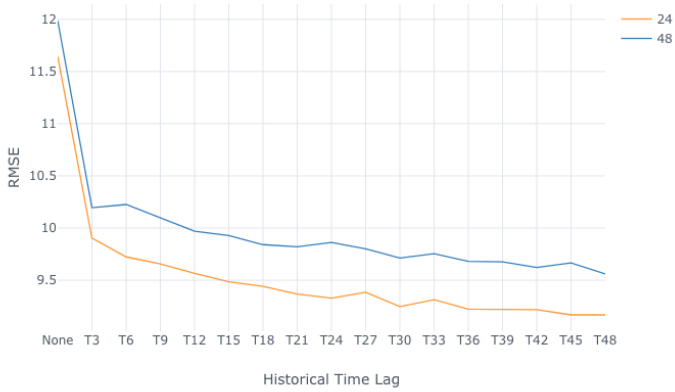



**Figure 5.9:** *Experiment 1:* Influences of increasing historical time lag.

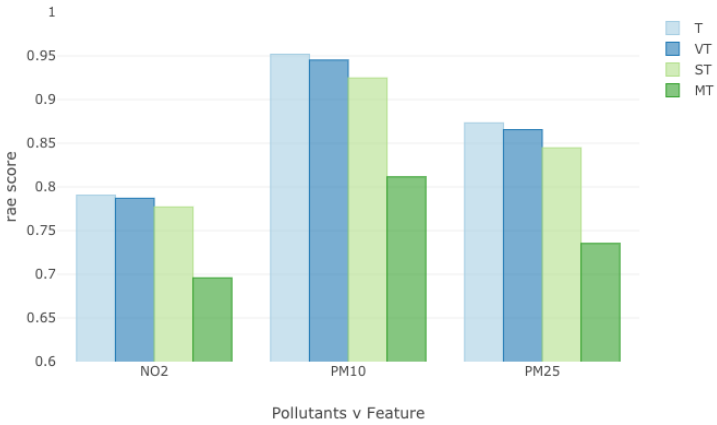



**Figure 5.10:** *Experiment 1:* Influence of temporal (*T24*) features with window size 24.

every pollutant on all test data. Then, in Figure 5.12 and Figure 5.13 a summarized version of the results are presented to highlight strengths of the model's towards the pollutants, and for various length of the window horizon.

The classification scores, presented in Table 5.8, are the results from the model's ability to predict sudden changes above the air pollutants warning level. The table includes the metrics recall, precision, false alarm ratio, and F1-score. The F1-score is a measure of

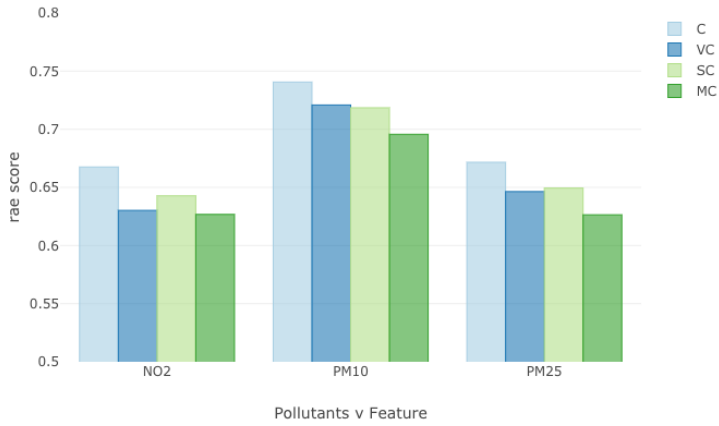



**Figure 5.11:** *Experiment 1:* Influence of statistical (*C*) features with window size 24.

the accuracy, and considers both precision and recall to compute the outcome. Then, in Figure 5.14, a graph of the F1 score is presented with the results combined for each pollutant. Next, the F1 scores are combined by window horizon to show the difference of the model's performance on the prediction window in Figure 5.15. Lastly, Figure 5.17 presents a graph of a sample of the actual predictions of this thesis and MET versus the real values.

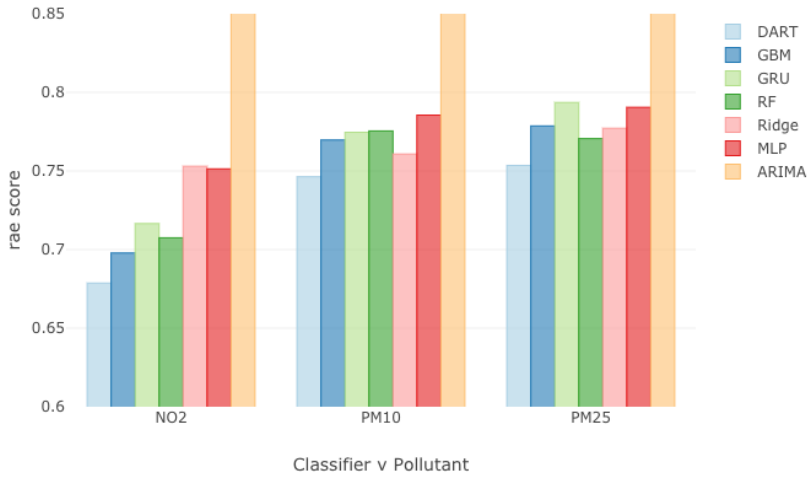



**Figure 5.12:** *Experiment 2:* Models performance with different pollutants. Note that the graphs y range are set to a limit.

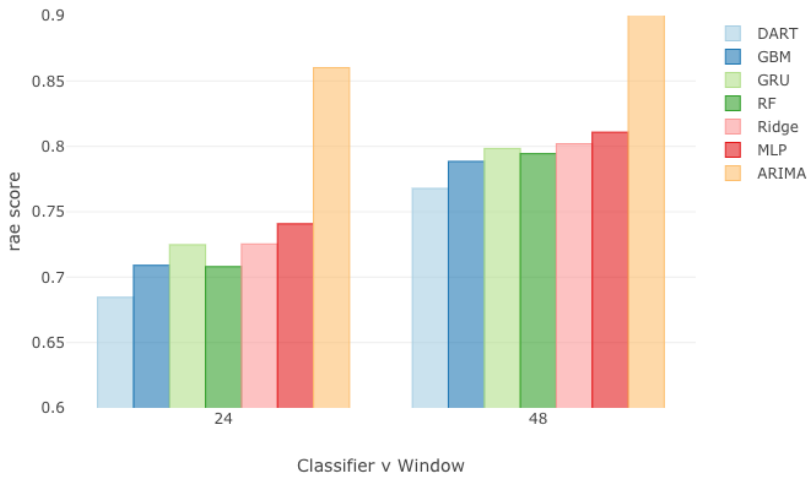



**Figure 5.13:** *Experiment 2:* Models performance with different window horizons. Note that the graphs y range are set to a limit.

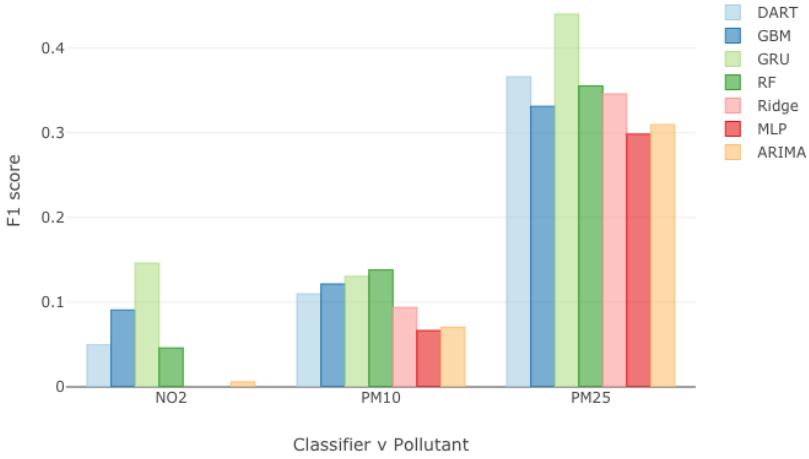



**Figure 5.14:** *Experiment 2:* Anomaly prediction with different pollutants.

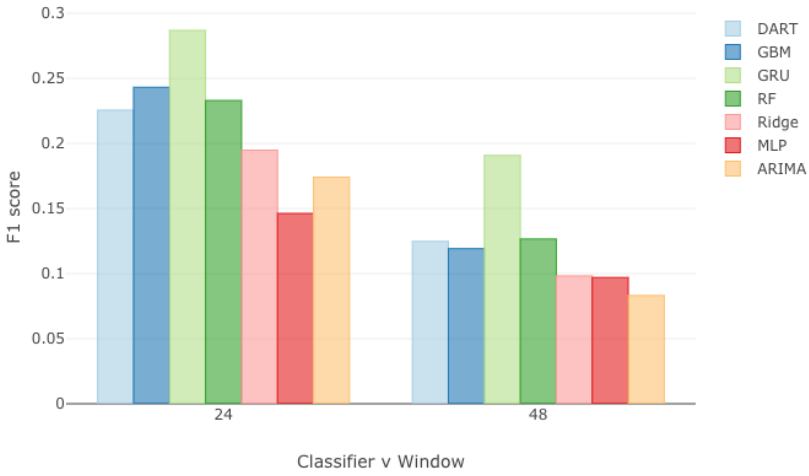



**Figure 5.15:** *Experiment 2:* Anomaly prediction with different window horizons.

| | NO2 | | | | | PM10 | | | | | PM2.5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mae | r2 | rae | rmse | smape | mae | r2 | rae | rmse | smape | mae | r2 | rae | rmse | smape |
| ARIMA | 16.727 | -0.026 | 0.868 | 24.214 | 0.813 | 6.612 | 0.230 | 0.868 | 12.315 | 0.780 | 5.356 | 0.276 | 0.844 | 8.223 | 0.885 |
| MLP | 10.669 | 0.453 | 0.718 | 14.387 | 0.509 | 6.644 | 0.348 | 0.758 | 11.221 | 0.587 | 3.095 | 0.475 | 0.746 | 4.740 | 0.625 |
| Ridge | 10.635 | 0.469 | 0.715 | 14.175 | 0.511 | 6.353 | **0.439** | 0.723 | **10.377** | 0.577 | 3.077 | 0.506 | 0.737 | 4.609 | 0.631 |
| RF | 9.979 | 0.496 | 0.671 | 13.834 | 0.477 | 6.398 | 0.361 | 0.729 | 11.100 | 0.563 | 3.026 | 0.508 | 0.724 | 4.610 | 0.605 |
| GRU | 10.131 | 0.491 | 0.682 | 13.877 | 0.501 | 6.535 | 0.403 | 0.746 | 10.707 | 0.610 | 3.094 | 0.494 | 0.746 | 4.654 | 0.640 |
| GBM | 9.845 | 0.511 | 0.663 | 13.609 | 0.473 | 6.364 | 0.379 | 0.724 | 10.965 | 0.565 | 3.097 | 0.497 | 0.740 | 4.663 | 0.617 |
| DART | **9.570** | **0.526** | **0.643** | **13.418** | **0.460** | **6.179** | 0.398 | **0.704** | 10.778 | **0.553** | **2.956** | **0.519** | **0.707** | **4.562** | **0.599** |

**(a)** Prediction scores with window size of 24.

| | NO2 | | | | | PM10 | | | | | PM2.5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | mae | r2 | rae | rmse | smape | mae | r2 | rae | rmse | smape | mae | r2 | rae | rmse | smape |
| ARIMA | 19.507 | -0.272 | 0.965 | 24.667 | 0.999 | 9.519 | -0.629 | 1.246 | 13.786 | 0.927 | 5.993 | 0.113 | 0.944 | 9.137 | 0.889 |
| MLP | 11.636 | 0.354 | 0.784 | 15.617 | 0.542 | 7.117 | 0.293 | 0.813 | 11.614 | 0.621 | 3.463 | 0.329 | 0.835 | 5.340 | 0.658 |
| Ridge | 11.702 | 0.367 | 0.791 | 15.424 | 0.546 | 7.002 | 0.335 | 0.798 | 11.247 | 0.616 | 3.411 | 0.376 | 0.817 | 5.183 | 0.664 |
| RF | 11.022 | 0.398 | 0.744 | 15.065 | 0.511 | 7.211 | 0.258 | 0.822 | 11.958 | 0.614 | 3.407 | 0.363 | 0.818 | 5.225 | 0.645 |
| GRU | 11.074 | 0.385 | 0.751 | 15.154 | 0.526 | 7.020 | **0.337** | 0.803 | **11.20**3 | 0.632 | 3.491 | 0.351 | 0.841 | 5.274 | 0.675 |
| GBM | 10.823 | 0.417 | 0.733 | 14.759 | 0.505 | 7.121 | 0.266 | 0.815 | 11.840 | 0.613 | 3.379 | 0.354 | 0.817 | 5.199 | 0.647 |
| DART | **10.552** | **0.439** | **0.715** | **14.475** | **0.495** | **6.901** | 0.292 | **0.789** | 11.653 | **0.600** | **3.320** | **0.378** | **0.800** | **5.127** | **0.640** |

**(b)** Prediction scores with window size of 48.

| | NO2 *(Total=139)* | | | | PM10 *(Total=104)* | | | | PM2.5 *(Total=33)* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F1 | FA | P | R | F1 | FA | P | R | F1 | FA | P | R |
| ARIMA | 0.012 | 0.659 | 0.008 | 0.026 | 0.111 | 0.500 | 0.167 | 0.083 | 0.399 | 0.380 | 0.620 | 0.297 |
| MLP | - | - | - | - | 0.060 | 0.733 | 0.267 | 0.034 | 0.379 | 0.407 | 0.593 | 0.280 |
| Ridge | - | - | - | - | 0.134 | 0.217 | 0.783 | 0.073 | 0.451 | **0.153** | **0.847** | 0.308 |
| RF | 0.092 | 0.000 | **0.333** | 0.053 | **0.195** | 0.536 | 0.464 | **0.127** | 0.412 | 0.318 | 0.682 | 0.306 |
| GRU | **0.215** | 0.333 | **0.333** | **0.181** | 0.149 | **0.200** | **0.800** | 0.085 | **0.496** | 0.251 | 0.749 | **0.374** |
| GBM | 0.108 | 0.056 | 0.278 | 0.067 | 0.162 | 0.590 | 0.410 | 0.106 | 0.460 | 0.281 | 0.719 | 0.354 |
| DART | 0.071 | 0.000 | **0.333** | 0.040 | 0.145 | 0.609 | 0.391 | 0.091 | 0.461 | 0.267 | 0.733 | 0.345 |

**(c)** Classification scores for anomaly prediction for 24 hour predictions

| | NO2 *(Total=33)* | | | | PM10 *(Total=104)* | | | | PM2.5 *(Total=139)* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F1 | FA | P | R | F1 | FA | P | R | F1 | FA | P | R |
| ARIMA | 0.000 | 0.667 | 0.000 | 0.000 | 0.030 | 0.315 | 0.018 | 0.044 | 0.220 | 0.683 | 0.317 | 0.192 |
| MLP | 0.000 | 0.333 | 0.000 | 0.000 | 0.073 | 0.352 | 0.315 | 0.042 | 0.218 | 0.378 | 0.622 | 0.137 |
| Ridge | - | - | - | - | 0.054 | **0.167** | **0.833** | 0.028 | 0.241 | 0.276 | 0.724 | 0.144 |
| RF | - | - | - | - | 0.081 | 0.683 | 0.317 | 0.047 | 0.298 | 0.288 | 0.712 | 0.192 |
| GRU | **0.077** | 0.333 | **0.333** | **0.043** | **0.111** | 0.454 | 0.546 | **0.063** | **0.384** | **0.172** | **0.828** | **0.250** |
| GBM | 0.074 | 0.083 | 0.250 | **0.043** | 0.081 | 0.641 | 0.359 | 0.049 | 0.203 | 0.570 | 0.430 | 0.140 |
| DART | 0.028 | **0.000** | **0.333** | 0.014 | 0.075 | 0.376 | 0.291 | 0.043 | 0.272 | 0.333 | 0.667 | 0.177 |

**(d)** Classification scores for anomaly prediction for 48 hour predictions

**Table 5.8:** *Experiment 2:* Model's results with regression error (a, b) and classification error (c, d). Note that the lines (-) implies that none spikes where detected, and the total of the anomalies of each pollutant is inside the parentheses. (FA=false alarms, P=precision, R=recall)

### 5.3.3 Experiment 3: Comparison of predictions versus official forecast

This section compares the results of machine learning predictions with the Norwegian national air quality service, a knowledge-driven model described in Section 3.2.4. The models are trained on the same train and test data as Experiment 5.3.2. The evaluation of the results are presented in three parts: The first includes the regression error of 24 and 48-hour predictions in Table 5.9. The error includes MAE, RAE, R2, RMSE, and SMAPE for each pollutant. The second evaluation is showing the results of the accuracy from classifying anomalies, found in Table 5.10. Along with the first evaluations tables, four sub-figures highlights the differences from the results, seen in Figure 5.16. A visualization of an example of the predictions is presented in Figure 5.17. Lastly, the results of the skill score are presented with persistence forecast as a reference, shown in Table 5.11.

| | | NO2 | | | | | PM10 | | | | | PM2.5 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | mae | r2 | rae | rmse | smape | mae r2 | rae | rmse | smape | mae | r2 | rae | rmse | smape | |
| 24 | MET | 14.163 | 0.204 | 0.789 | 20.323 | 0.670 | 10.716 | -0.927 | 1.134 | 18.825 | 0.752 | 5.277 | 0.069 | 0.859 | 8.873 | 0.730 |
| | DART | 11.383 | 0.566 | 0.635 | **14.985** | 0.499 | 6.252 | 0.524 | 0.659 | **9.651** | 0.606 | 3.802 | 0.609 | 0.618 | **5.753** | 0.625 |
| 48 | MET | 14.180 | 0.206 | 0.791 | 20.292 | 0.671 | 10.778 | -0.876 | 1.140 | 18.676 | 0.753 | 5.376 | 0.045 | 0.875 | 8.989 | 0.738 |
| | DART | 12.967 | 0.453 | 0.725 | **16.814** | 0.547 | 6.934 | 0.418 | 0.730 | **10.696** | 0.647 | 4.384 | 0.469 | 0.713 | **6.706** | 0.672 |

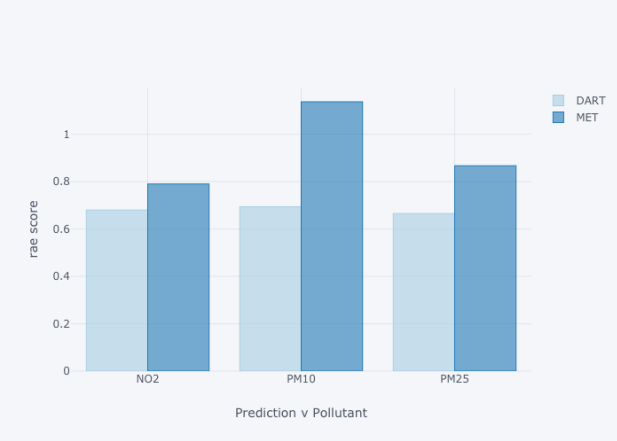
**Table 5.9:** *Experiment 3:* Predictions results of this thesis and MET.

| | | NO2 *(Total=20)* | | | | PM10 *(Total=36)* | | | | PM2.5 *(Total=66)* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | F1 | FA | P | R | F1 | FA | P | R | F1 | FA | P | R |
| 24 | MET | 0.116 | 0.500 | 0.167 | 0.089 | 0.027 | 0.981 | 0.019 | 0.051 | 0.278 | 0.682 | 0.318 | 0.282 |
| | GRU | **0.167** | 0.500 | 0.167 | 0.167 | **0.248** | 0.000 | 1.000 | 0.144 | **0.499** | 0.294 | 0.706 | 0.393 |
| 48 | MET | **0.116** | 0.500 | 0.167 | 0.089 | 0.045 | 0.969 | 0.031 | 0.085 | 0.300 | 0.670 | 0.330 | 0.307 |
| | GRU | 0.042 | 0.333 | 0.333 | 0.022 | **0.056** | 0.500 | 0.167 | 0.033 | **0.455** | 0.185 | 0.815 | 0.318 |

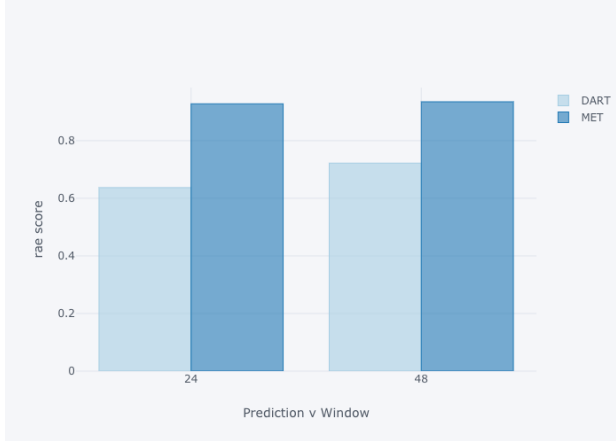
**Table 5.10:** *Experiment 3:* Comparison with observations of all stations. Total is the total number of anomalies. (FA=false alarms, P=precision, R=recall)

| | | NO2 | PM10 | PM2.5 |
|---|---|---|---|---|
| 24 | MET | 0.138 | -0.401 | 0.033 |
| | DART | **0.361** | **0.297** | **0.374** |
| 48 | MET | 0.265 | -0.154 | 0.175 |
| | DART | **0.388** | **0.365** | **0.382** |

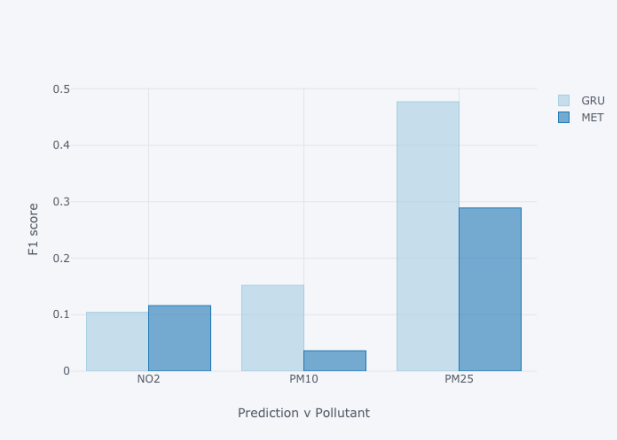
**Table 5.11:** *Experiment 3:* Comparison of skill score. A measure of how much better, or worse, a forecast is compared to a persistence forecast



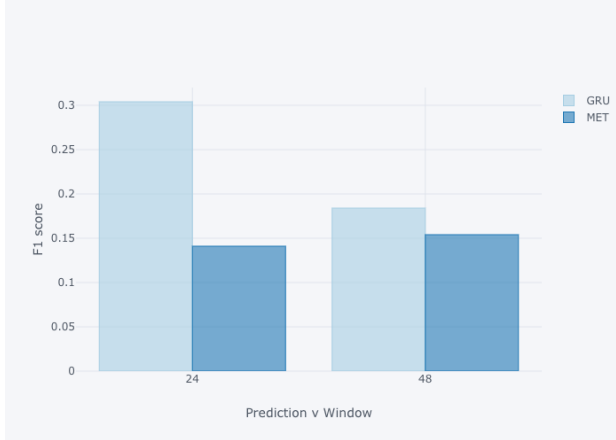

**(a)** Results of regression error grouped by pollutant type.




**(b)** Results of regression error grouped by window size.




**(c)** Results of anomaly prediction grouped by pollutant type.




**(d)** Results of anomaly prediction grouped by window size.

**Figure 5.16:** *Experiment 3:* The results showing the performance of the forecasts. The results are grouped by pollutant type and window size.

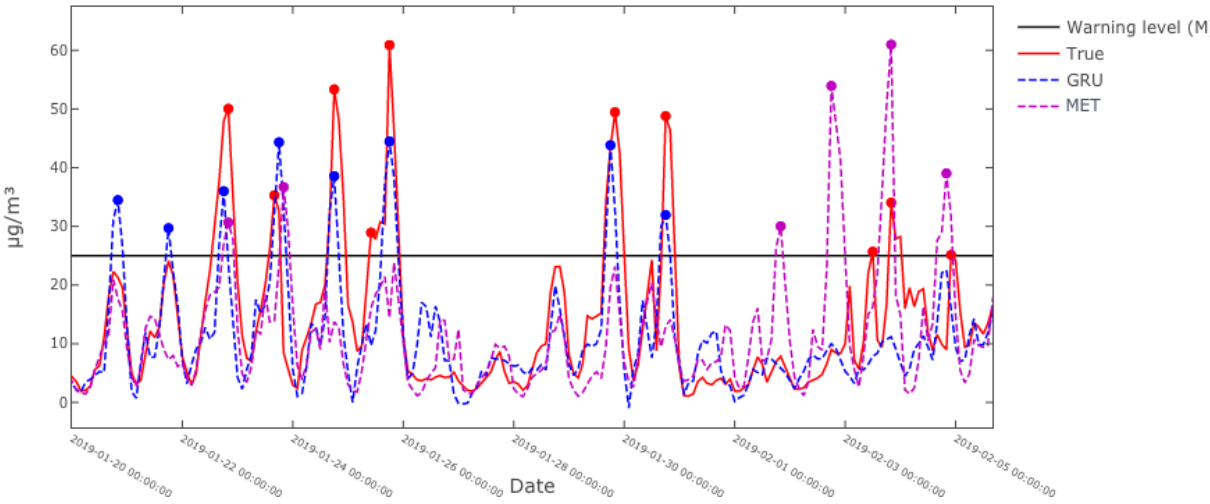



**Figure 5.17:** *Experiment 3:* Sample of the anomaly 1-day predictions for PM2.5 at Torvet.

This page intentionally left blank.

Chapter 6

# Conclusion

This research utilized three different datasets of Trondheim city: Air pollutants, historical weather observations, traffic volume count, and wood burner dataset. The goal of this study is to evaluate the performance of machine learning methods for air quality prediction in Trondheim. We started with an analysis of datasets of Trondheim, including air pollutants, weather observations, traffic volume count, and wood burners. Further, we created more features with statistical feature engineering and tested multiple state-of-the-art machine learning techniques. Several machine learning models were implemented, optimized, trained, and tested to determine the strengths and weaknesses of air quality prediction. The architecture and implementation details of the specific models are presented in Chapter 4.

We showed in the experiments in Chapter 5 that DART has the best performance of predicting the overall air quality for all the pollutants studied (PM2.5, PM10, NO2). Further, we found that GRU can classify sudden changes better than the other methods. The evaluation and proof of the conclusion are further elaborated in this chapter. This chapter starts of with an evaluation of the experiment results from Chapter 5 in Section 6.1. Section 6.2 provides a discussion of the results along with its limitations. Section 6.3 outlines the contributions from this thesis. Lastly, Section 6.4 presents extensions and future work of the limitations discovered.

## 6.1 Evaluation

This section presents an evaluation of the experiments performed. The first experiment was designed to find the different impact of the features to determine the most influential. A variety of feature engineering techniques was completed to generate feature sets in three categories: Historical lagged influence, temporal influence, and statistical influence. The results from each sub-experiment of each category are evaluated in Section 6.1.1. The second experiment is about the accuracy of the machine learning models to predict air quality in Trondheim. The results consist of a comparison of the model's ability to predict air quality patterns, and their effectiveness to predict sudden pollution changes. The methods

of experiment 2 are presented in Section 6.1.2. The third and final experiment is a comparison of the best results from this thesis versus the official air quality forecast in Norway. The results of the third experiment are evaluated in Section 6.1.3.

### 6.1.1 Experiment 1: Determining Influential Features

This experiment was mainly intended to provide answers to RQ2.

**Research Question 2** Which features have the highest impact on the machine learning algorithm's ability to accurately perform predictions?

Feature engineering is an attempt to enhance the feature set to increase the performance of the machine learning model. The temporal and statistical features are part of describing the time aspect of the time series. The spatial features define the air pollution flow and information about the physical location. These additions, together with the natural values from the datasets; air pollutants, meteorological, and traffic measurements, will, in turn, generates a huge feature space for the experiments. Three sub-experiments were completed to discover the impact of each feature groups, both individually and in combinations. The percentage gains presented below is the increase of performance related to the scores without the extensions.

#### Sub-experiment 1.1: Historical Lagged Influence

Figure 5.9 shows a steady decrease in error by extending the number of historical values. The slopes have a slow decline, which means a small positive effect on performance. The difference in development between the window size of 24 and 48-hour is minimal. Both horizons give the same influence of additional historical lag. The tradeoff between a little better performance and the increase of features is positive since the error in the graph is decreasing.

*The experiment shows a slightly better performance of including historical values, at the costs of more data and more extended training.*

#### Sub-experiment 1.2: Temporal Influence

Figure 5.6 shows the impact of the temporal feature engineering has on the features traffic (*VT*), meteorological (*MT*), and spatial (*ST*). The single feature *T* consist of temporal feature engineering with just the pollutants at the target station.

For NO2, the historical values of traffic and spatial have a small performance gain of 1% and 3% respectively. The model is not able to find a good relationship between the historical and the future observations of the traffic and spatial features. The traffic data might be too constant periodic, and the past is not able to give away any clues to the future forecasts. The same effect is also seen for PM2.5 and PM10 of the temporal traffic (*VT*) impact. The influence of the temporal extension of the spatial feature (*ST*) does neither give any significant performance increases. A notable difference is that the gain of *ST* for PM2.5 and PM10 are both approx 6%, which is the double of the impact on NO2. This

difference might indicate that NO2 has a different transport pattern than of particular dust, and is not explained based on a spatial feature.

The meteorological variables have the best performance gain when incorporating temporal features (*MT*). NO2 has a performance increase of 13%, PM2.5 an increase of 24%, and PM10 an increase of 21%. Looking back at the feature analysis of meteorological variables in Section 5.2.2, and the findings of the lower correlation between NO2 and the weather compared with PM2.5 and PM10, this relation is seen again in the results of the temporal influence experiment where NO2 achieves a considerably lower score. The observations of NO2 are not as dependable of previous weather than the particular matter has.

*The historical meteorological values have the best performance gain across predictions for all pollutants with an average of 20% increase of RAE, with PM2.5 have the best increase of 24%.*

**Sub-experiment 1.3: Statistical Influence**

Figure 5.11 shows the impact of the statistical feature engineering on the features traffic (*VC*), meteorological (*MC*), and spatial (*SC*). The feature *C* is made from the statistical feature engineering of the pollutants at the target station, and are used as a reference for the performance gains presented below. The goal of statistical features is to add a more general and wider temporal dependency, then by just including historical values. The statistical functions will include a smarter relation of the past, that the models will easier learn.

For the NO2 forecasts, the most important observation is that the statistical traffic features have a similar performance gain as the meteorology of 5%. Compared with the temporal results in sub-experiment 1.2, the statistical feature engineering has a much stronger effect on NO2. The model can incorporate the statistical properties of the traffic time series, rather than the historical values. In the case of PM10 and PM2.5, the gains from traffic and spatial are similar, while statistical for meteorological is the strongest with 6% increase.

Interestingly, the overall scores for the experiments using the statistical feature engineering are higher than the temporal technique in sub-experiment 2. The statistical has a total average rae of 0.15 performance increase of rae score than the temporal and is showing that the model can learn from the statistical properties easier than the historical values. This improvement is seen for all the pollutants.

*Statistical feature engineering manages to achieve a stronger performance than the extension of historical features. Most notable is the gain of statistical traffic features of NO2 predictions.*

### 6.1.2 Experiment 2: Comparison of machine learning models

This experiment is aimed to give answers to RQ3.

**Research Question 3** How accurate are machine learning methods for predicting air quality in Trondheim?

Here, we introduce a variety of machine learning models to compare their performance of air quality predictions. A comparison of the model's ability to forecast regular air pollutants patterns is first presented, and secondly the model's accuracy of predicting anomalies in terms of sudden changes. It was decided to implement seven forecasting techniques, each with its unique trait, and identified as potentially advantageous approaches for air quality prediction. ARIMA and Ridge had been applied to time series problems with reliable results in the past. Deep Neural Networks and Random Forest had been used in the recent literature with strong results within air quality prediction problems. A version of RNN with GRU cells was included due to its predicting powers of time series problems. Finally, because of the ability of gradient boosting to minimize error in complex problem domains, and because it is less used in the literature, two unique variations of gradient boosting were implemented.

A thorough optimization search was first completed to assure that every model's performance is optimal. Specification around the hyper search is given in 4.1.2. The neural networks, MLP and GRU, took the longest time to find the optimal parameters while the statistical and ensemble techniques included fewer hyperparameters and were not as sensitive about the tuning as the neural networks.

**General Pattern**

The results from the general pattern are given in Table 5.8, with a separation of 24 and 48-hour prediction scores. The result shows a dominant DART with the overall best performance. For PM10 with 24-hour prediction, Ridge regression exceeds the rest with best RMSE and R2 score. However, Ridges results for MAE, RAE, and SMAPE falls shortly behind those of DART. PM10 prediction differs from PM2.5 and NO2 with being more difficult to predict, with a higher variance and weaker correlation of time and space. That might be why Ridge regression is doing so well, due to its advantage of reducing this variance when minimizing error. This also explains the different score for RMSE and MAE, because the latter is more punishing of significant errors. Also notable for PM10 with 48-hour predictions, were GRU slightly outperform Ridge and the other model's regarding RMSE and R2. Surprisingly, the MLP models do not perform well in fitting the general pattern and might be because the model is limited by overfitting to the large input dimension. However, it is observed that the MLP is slightly better at predicting PM2.5 than the other pollutants.

Figure 5.12 show a better performance of DART for all pollutants. GRU and MLP have similar scores for PM2.5 and PM10, but for NO2, GRU is distinctly better. The similarity might relate to the fact that NO2 includes a stronger temporal pattern that GRU can learn. Besides, Ridge and MLP are predicting worse for NO2 than the other pollutants. This poor performance might be due to the large feature space, where Ridge and MLP weakly emphasize low correlated features which are of importance. Interestingly, Ridge and RF achieve a slightly better score for PM2.5 than of GBM.

In the case of the model's performance grouped by window horizon of 24 and 48-hour in Figure 5.13, the results show fewer differences than when they are grouped by pollutants. DART is achieving the best scores for both window horizons presented. As expected GRU does perform better than MLP, for long-term predictions. The difference is not that

big and is believed to be because of the multi-output strategy. Interestingly, GBM and RF have similar average results for both window sizes.

*DART outperforms the other models by achieving the best performance for PM2.5 and NO2. In the case of PM10, GRU and Ridge are the achieve the best performance.*

#### Anomaly Prediction

The results of range anomaly prediction are certainly different from the general pattern evaluation. The results in Table 5.8c and Table 5.8d shows another side of the models. When minimizing the forecast error of the difference between the actual and the predicted, it might come at the cost of the model's ability to predict sudden changes. GRU has the best accuracy for air quality for the pollutants NO2 and PM2.5. In the case of PM10, RF has the highest scores of F1 and recall. However, for 48-hour predictions of PM10, GRU has better accuracy again, due to RF causes most false alarms with a ratio of 0.68. MLP and Ridge do not perform well when predicting sudden changes of NO2. This is mostly due to the occurrences of NO2 anomalies are too low, and the results are not a good representation. ARIMA and MLP have the highest number of false alarm ratio for 24-hour predictions.

Figure 5.14 shows the F1 score of the classifiers against each pollutant. A notable difference is the overall better performance at predicting PM2.5, than PM10 and NO2. GRU has the best scores for anomaly prediction, with 17% better at PM2.5 than the next best by DART. In the case of PM10, RF can achieve a slightly better score than GRU. For the poorest performances, ARIMA is slightly better than MLP, where both are not able to forecast correctly. Another interesting observation is the models DART, GBM, RF, and Ridge, have all very similar results of all the forecasts.

In the case of anomaly prediction for 24-hour, highlighted in Figure 5.15, show a distinct GRU with the highest performance. GRU has a 15% better score than GBM with the second best score. Most notable is the 34% gain of GRU from second best at 48-hour predictions. While ARIMA has shown poor performance in the results for the general pattern, it does achieve better classification scores for 24hour predictions than MLP.

*GRU is considered the top performer with the lowest ratio of false alarms and the highest scores of recall and precision.*

### 6.1.3 Experiment 3: Comparison of predictions versus official forecast

This experiment was designed to answer RQ3 by comparing the machine learning results against a knowledge-driven approach, and also the national air quality forecasts in Norway (MET). Short reminder that METs results are still in a test phase. However, experiment 3 will give some suggestions for the use of machine learning for air quality prediction.

### General Pattern

As seen in Figure 5.9, all results are showing a performance increase by using machine learning to predict PM2.5, PM10, and NO2. The most significant improvement is for predicting PM10 with RMSE halved. The poor results of METs PM10 forecasts are mainly because their model is expecting multiple spikes of bad air pollution that never occurs. This error is believed to have its roots from weather forecasts that are included in the model's calculations, causing the model to create mispredictions. The model's sensitivity against other forecasts indicates that the knowledge-driven and rule-based approach is not able to capture all necessary relations to create good enough estimations. The same misprediction of sudden changes is seen at all stations for all pollutants. For PM2.5 and NO2, the sudden events are less distinct than of PM10 and are usually of consecutive days for PM2.5 and NO2, thus easier to predict. Interestingly, MET's predictions for 24 and 48 hours do not have very different results. This small difference is an advantage MET has compared to our results, which show a clear distinction between the different horizons.

The skill score is the relative accuracy of the estimates over a persistence forecast. A persistence forecast says tomorrow is the same as today, based on the observations. Results from this test presented in Table 5.11 show a dominant DART model of all pollution forecasts. For DART, the score indicates a value of around 0.375 better performance than a persistence score at all tests. However, for 24h predictions of PM10, the results of DART is lower than the other. This observation is not found in the previous results from the other experiments, and the persistence score of daily PM10 might cause the difference to correlate better than for the forecasts of PM2.5 and NO2. In the case of METs results, they show significant improvements for 48-hour forecasts compared with 24-hour. Also, METs predictions for NO2 achieves a better skill score than PM2.5 and PM10. A notable result is METs PM10 predictions with a negative score that indicates a worse prediction than the persistence.

### Anomaly Prediction

In the case of anomaly prediction for 24-hour predictions, it shows a distinct higher score of the predictions by this thesis. The greatest difference is again of PM10 predictions, where MET has an inferior false alarm ratio of PM10 with over 95%. These overshooting is the reason for the poor regression error mentioned above. A notable observation is in the case of 48-hours predictions for NO2, where MET achieve a better score than our approach. A surprising finding is that METs 48-hour has similar accuracy as 24-hours.

Figure 5.17 shows the prediction graphs with GRU and MET plotted with the actual values. It shows both methods ability to predict PM2.5 within the next 24 hours efficiently. Typically as shown is the overshooting for MET predictions at the end of the sequence. Also, at the end of the series, GRU is unable to foresee the irregularities of the air quality. These events might be caused by some changes in the weather that MET has found, due to their attempts to predict a high level of pollution. These weather patterns are missing in the GRU model and could be improved upon with utilizing the weather forecasts. Due to the setup of the anomaly prediction metrics with classification hits above the warning level, a limitation of the results is found. The limitation is that the classification scores might be poorly affected if the pollution levels are close to the threshold. This issue can be found in

the two first spikes of the figure, where GRU predicts just above the threshold, while the actual values are just below, resulting in miss-classification where the predictions are not that far off. A solution to this problem would be to use regression scores of the forecast in the area where a sudden change occurs.

## 6.2 Discussion

The necessity of healthy air has always been of great importance. As air is vital for all living beings on earth, it is our responsibility to keep the air clean. The rapid urbanization and industrialization have led the world into a new era of air pollution and seen as a modern-day curse. Because of the impact air quality has on peoples everyday life, how to predict air quality precisely, has become an urgent and essential problem. We target our air prediction study to the city of Trondheim, Norway. The study demonstrates the benefits of machine learning for short and long-term predictions of air pollutants, and foresee sudden spikes of high pollution level.

The literature agrees that urban air quality prediction is a challenge that needs a multivariate temporal-spatial approach. The air pollutants occur in a space profoundly influenced by the weather with frequent changes in temperature, rain, among other weather conditions. Meteorological information is thus highly crucial to include among other dependent variables that have been studied, such as traffic and geolocation information. Problem solutions often take advantage of time series tendencies due to air quality is in constant change and is reliant on its historical trace. However, the literature search found no exploration of the impact of events. Events and urban human mobility data are believed to influence air pollution. Unfortunately, because of both technical details and limited time, this was not pursued in this study either.

A range of core elements related to air quality prediction was studied, including analysis of the spatiotemporal patterns, exploration of essential variables, preferable machine learning methods for time series prediction, and similar challenges within the air quality domain. Numerous techniques have been applied, ranging from statistical approaches to more recent advances in machine learning. Of the neural network, deep feedforward and recurrent neural network are the most seen architectures in the literature. Deep learning has demonstrated the high performance of learning hidden relations of complex problems, and the more specialized architecture recurrent neural network, LSTM, and GRU, has shown to be a valuable tool for time series prediction. Additionally, ensemble learning is favorable due to it is prone to noise and variance. Random forest has been studied multiple times and is common to include as a baseline method, due to its good performance with a minimal amount of feature preprocessing and parameter optimization. Another ensemble techniques found in the literature is a larger ensemble of multiple base learners, of the type averaging, weighting, or stacking. These approaches are motivated by the mean of combining the strengths and diversity of all the different methods to reduce the total error.

Observed during this literature review that there is no unified framework for testing models within the research community. The lack of a proper framework is problematic because results cannot be sufficiently validated. The motivation behind each research consists of a problem description specified for their city and location. Therefore, each study

will be highly representatives of the dataset utilized, forecast horizon, the pollutants in focus, and the validation methods used. Surprisingly, most studies only include the generalization error for the predictions against the measured and is ignoring the problem of sudden changes in air pollution. In this case, the solutions skill to predict sudden changes is somewhat hidden in the overall error calculation. In many real-life events, the ability to foresee these sudden changes in air pollution is helpful if there is a need to take preventative measures.

Based on the results from experiment 1 in Section 5.3.1, an extended feature engineering approach provides greatly improved results of air quality prediction. The feature vector with statistical and historical characteristics of pollutants, meteorologic, and traffic data includes patterns and relationships of the data. The gain of the statistical features, including moving average, minimum, maximum, and difference calculations, is significantly better than of the features, including historical values.

To evaluate the models, we used multiple metrics to calculate both the overall error and the accuracy of the classification of sudden changes. For calculating the erroneous of the model, the RMSE is believed to be superior due to its high weight to significant errors, then of MAE. This means the RMSE should be more useful when large errors are particularly undesirable, which we believe is in favor of air quality validation. In the case of predicting sudden changes, a high false alarm ratio is unfavorable, but on the other hand, one does not want the model to be too afraid to make the hard decisions of high pollution levels. The other measures for accuracy with recall, precision, and F1-score it desirable to get an as high score as possible.

Consistently, from experiment 2 presented in Section 5.3.2, it was found that DART can achieve the best performance. Only in the case of PM10 is DART beaten by Ridge for 24-hour predictions and GRU for 48hour predictions with both achieving a better RMSE. Overall the ensemble techniques can generalize better on the large feature space, than the neural networks. However, for the anomaly prediction, GRU can achieve the highest recall scores for classifying sudden changes. The GRU is modified only when the weights in the gates deem it appropriate. Such a mechanism is not found in the MLP model. MLP is prone to overfitting, and even with the regularization methods, it is not able to provide proper results.

However, a significant limitation of the NO2 anomaly prediction is that the total number of occurrences is too low to give a good enough result. A solution might be to lower the threshold level of the spikes, but this is not done in this work due to the idea is to predict changes above the national regularities. Furthermore, the hyperparameter search for the neural networks was a difficult task due to the variation between each pollutant for each model. The ensemble methods did not undergo this challenge at the same extent and were more stable when tuning the hyperparameters. They can reduce the variance with their nature of averaging the outcome. To overcome the sensitivity of the neural networks, an ensemble design could be utilized with them as base learners. An approach regarding the design with a multi-model neural network is to separate by different window horizons, to ultimately build up all the window steps wanted, is believed to strengthen the stability and performance of the neural networks further.

The results from experiment 3, in section 5.3.3, proves that machine learning to achieve better performance than the knowledge-driven approach of air quality prediction in Trondheim. The knowledge-driven models are not able to capture all the complexity of all the dependencies of air quality prediction. The machine learning models can reduce the error and create more accurate predictions. The most improvements are seen for 24 window horizon, with an RMSE reduction of 26%, 49%, and 35% for NO2, PM10, and PM2.5 respectively. For 48 hour predictions, the RMSE reduction of 17%, 43%, and 24% for NO2, PM10, and PM2.5 respectively. Based on the results, the prediction of a particular matter by ensemble learning is able to fit the actual measures better.

In the case of predicting sudden changes, the results show a better accuracy for GRU for 24-hour predictions. The improvements show a higher hit-rate and a lower false alarm ratio than of METs. However, for a larger window size of 48 hours, the results are showing a much lower difference in accuracy. The knowledge-driven approach can capture the long-term dependencies leading to sudden changes. This may be due to the knowledge-driven model has a much lower threshold for trying high spikes, which can be seen from the high number of false alarms. The machine learning approach might generalize too well on the training data that consist of a higher amount of nonspike air pollutant data. While the total amount of anomalies for NO2 can be regarded as too low, it does indicate the performance. The threshold for anomalies can be reduced, or include more data points when they are available in the future to produce a stronger comparison,

However, even though the machine learning approach achieves significantly better results, we want to remind the reader that the results from MET include more than just station-wise predictions. First, they provide a fine-grained resolution of the city, and the values presented in the thesis is the predictions at the target station. Besides, the MET forecasts have a unique feature that the machine learning approach has a hard time to achieve. This feature is that their model estimates the causes of each pollutant. The causes are given as percentages and are of high interest if one were to take measures to reduce emissions.

The results of the MLP model is not as good as shown in the literature. There are several reasons for the poor results in this thesis, but the leading cause is believed to be overfitting. Consequently, the extensive feature set generated will include much noise that is bad for machine learning. During training, there is too much variation in the input weights for the deep neural network to generalize on. Due to time constraints, we were not able to perform feature selection to the degree that was necessary to achieve better results.

The methods studied in this research are data-driven approaches, which are naturally determined by the data they operate. Therefore, data preprocessing and optimization steps will affect the outcome of the trained models and performance metrics. The air quality in Trondheim is considered on average healthy, but with unpredictable winters with numerous of sudden changes of higher pollutant levels. Consequently, the architecture presented in this thesis might not perform at the same level on other datasets without being trained in the right circumstances. Besides, the air quality in Trondheim is represented by four high-quality sensors, which variates from larger cities with more substantial coverage. Furthermore, the meteorological measurements are acquired from a single location. Hav-

ing additional weather stations will further support the meteorological influence on the models with more spatial recognition.

All machine learning models are highly dependent on their hyperparameters. A small change could affect the results in either direction of performance. In this research, moderately amounts of effort are put into parameter tuning of each particular model. Additionally, the hyperparameter search utilized is a straightforward approach and a more suited method for the large hyperspace with sensitive models could be used, such as Bayesian optimization or evolutionary optimization. The neural network models are more vulnerable to parameter changes than others, and it is desirable to introduce more robust architectures to avoid this pitfall. Unstable training is, however, a problem for most deep learning techniques, and is an ongoing field of research.

In conclusion, this thesis found that there are multiple worthwhile machine learning algorithms to predict air quality. Through combining data of pollutants, weather, and traffic with a statistical temporal-spatial feature engineering technique, we provide a higher level of information of the data used for the machine learning models. The multi-step-ahead prediction approach is operating well together with the extensive feature set. The selected methods show high performance to predict separate pollutants with various forecast horizons. We present results showing that the gradient boosting method DART, outperformed every model in most cases. In the case of anomaly prediction, GRU demonstrated better classification scores. The data-driven approach can exceed the national air quality forecast service for short-term predictions of 24-hours, while for long-term forecasts, the knowledge-driven method provides quite similar results as the machine learning models. The data-driven approach is thus believed to be an excellent complement for the more complex model.

## 6.3 Scientific Contributions

The research contributions are summarized in the following way:

1. A state-of-the-art review of air quality prediction with machine learning with a focus on ensemble learning and neural networks. The review includes insight and discussion of the solution to overcome the complexity of air pollution in urban cities. Chapter 3 presents the structured literature review.

2. An in-depth exploratory data analysis of air pollutants, meteorological, and traffic data in Trondheim, Norway. The motivation behind is to discover patterns, to spot anomalies, and to check assumptions by applying statistics and graphical representations. Chapter 5 introduces the dataset with the analysis in Section 5.2.

3. A comparison of the performance of multiple machine learning algorithms for multi-step-ahead predictions. The results are evaluated on real air pollution, meteorological, and traffic data. Also, by applying statistical and temporal feature engineering to produce an extensive representation of the data patterns.

4. Lastly, we present advice for machine learning methods to provide accurate air quality predictions of general pattern and sudden changes. The models are optimized and designed to serve real-time data and predictions of air quality in Trondheim city.

## 6.4 Future Work

This section presents possible extensions to the system, improvements that can increase prediction accuracy, and solutions to limitations that were revealed in this thesis.

### 6.4.1 Common Benchmark Datasets

In the future, it might be necessary to develop a framework with benchmark datasets that novelties can be tested. This direction with a specific problem description that covers the most general air quality challenges may focus the research for more significant improvements. Most research today is aimed at locations all over the world, motivated by city management. Besides, in 2018, there was arranged a machine learning challenge to forecast the air quality of Beijing, China, and London, UK [ACM (2018)]. The results of this competition show impressive results. It shows the need for benchmark datasets for a stricter comparison by evaluating the methods on the same data.

### 6.4.2 Fine-Grained Map With External Sources

This thesis focuses on predicting the target pollutant for each station separately. This station-wise approach makes it easier to validate the method with the true measurements at the location. However, this is not sufficient for a modern solution where the demand is high for a fine-grained air quality map of the city [Andreas Lepperod (2018)]. Further extension of the dataset and features is needed, to achieve a city coverage of fine-grained air prediction. Referring back to the dataset analysis in Section 5.2.2 about the feature hypothesis about wood heating, low-cost sensors, and events data. These are examples of additions that would be interesting to explore. The models are extendable to split up in a grid fashion with geolocation information to predictions all over the city, as they have done in Hu et al. (2017). This extension would increase the value of the results by offering a fine-grained spatial map of the air quality in Trondheim, with high accuracy of real-time values and the forecasts.

In the case of events data and urban human mobility data, it would be interesting to see a possible connection with air quality. The human mobility data is aggregated numbers for the population's movement patterns. The data may include what kind of transportation is used as well, based on their travel speed. These patterns could correlate with air pollution to give more accurate local forecasts.

### 6.4.3 Weather Forecast

In this thesis, we have studied feature engineering on multiple measures of pollutants, meteorological, and traffic. As seen in the previous literature, this addition is providing additional information for the models. Bougoudis et al. (2016) used weather forecasts to

improve the predictions even further. An interesting extension would be to include weather forecasts to see if the models can improve the predictions. Especially interesting would be to see if it improves accuracy for predicting long term sudden changes.

### 6.4.4 Feature Selection

Finally, we believe that better results could be achieved by reducing the feature space. Some methods are having problems with overfitting, even with the regularization methods implemented in this thesis. To improve upon this, we want to find an optimal feature dimensionality reduction technique in order to improve the model's predictions further. Ideas for further explorations are to use the results of the feature importance by the gradient boosting algorithm, which may lead to interesting feature insight. Besides, principal component analysis is another widely used techniques for dealing with large feature space. It divides the data into a set of components which try to explain as much variance as possible.

# Bibliography

ACM, K., 2018. Kdd cup 2018.
URL https://www.kdd.org/kdd2018/kdd-cup

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

Andreas Lepperod, Hai Thanh Nguyen, S. A. L. W. P. ., 2018. Machine learning for air quality prediction: A review.

Arden, P. I., RT, B., MJ, T., et al, 2002. Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution. JAMA 287 (9), 1132–1141.
URL +http://dx.doi.org/10.1001/jama.287.9.1132

Athira, V., Geetha, P., Vinayakumar, R., Soman, K., 2018. Deepairnet: Applying recurrent networks for air quality prediction. Procedia computer science 132, 1394–1403.

Becker, S., F. M. S. J., 2002. Involvement of microbial components and toll-like receptors 2 and 4 in cytokine responses to air pollution particles. American journal of respiratory cell and molecular biology 27 (5), 611–618.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of Machine Learning Research 13 (Feb), 281–305.

Bougoudis, I., Demertzis, K., Iliadis, L., 2016. Hisycol a hybrid computational intelligence system for combined machine learning: the case of air pollution modeling in athens. Neural Computing and Applications 27 (5), 1191–1206.

Box, G. E. P., Jenkins, G. M., 1970. Time series analysis: forecasting and control. san francisco, holden-day. Holden-Day, San Francisco.

Brauer, M., Amann, M., Burnett, R. T., Cohen, A., Dentener, F., Ezzati, M., Henderson, S. B., Krzyzanowski, M., Martin, R. V., Van Dingenen, R., van Donkelaar, A., Thurston, G. D., Jan 2012. Exposure assessment for estimation of the global burden of disease attributable to outdoor air pollution. Environmental Science & Technology 46 (2), 652–660.
URL https://doi.org/10.1021/es2025752

Breiman, L., 1996. Bagging predictors. Machine learning 24 (2), 123–140.

Bruce Rolstad Denby, Heiko Klein, P. W. M. G. M. P. H. F. A. V., 2018. The norwegian air quality service: Model forecasting.
URL https://www.met.no/om-oss/luftkvalitet-seminar/_/attachment/download/d039e17d-df94-4caf-b69c-a4fd57d21801:0eac9a53a33951c34ffc78e82e3324bdcbd1430b/Denby_17Sep2018.pdf

Chen, G., Li, S., Knibbs, L. D., Hamm, N., Cao, W., Li, T., Guo, J., Ren, H., Abramson, M. J., Guo, Y., 2018. A machine learning method to estimate pm 2.5 concentrations across china with remote sensing, meteorological and land use information. Science of the Total Environment 636, 52–60.

Chen, L., Cai, Y., Ding, Y., Lv, M., Yuan, C., Chen, G., 2016. Spatially fine-grained urban air quality estimation using ensemble semi-supervised learning and pruning. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. ACM, pp. 1076–1087.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using rnn encoderdecoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).
URL http://dx.doi.org/10.3115/v1/D14-1179

Drucker, H., 1997. Improving regressors using boosting techniques. In: ICML. Vol. 97. pp. 107–115.

Fan, J., Li, Q., Hou, J., Feng, X., Karimian, H., Lin, S., 2017. A spatiotemporal prediction framework for air pollution based on deep rnn. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4, 15.

Fox, I., Ang, L., Jaiswal, M., Pop-Busui, R., Wiens, J., 2018. Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp. 1387–1395.

Friedman, J. H., 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189–1232.

Friedman, J. H., 2002. Stochastic gradient boosting. Computational statistics & data analysis 38 (4), 367–378.

Ghaemi, Z., Alimohammadi, A., Farnaghi, M., 2018. Lasvm-based big data learning system for dynamic prediction of air pollution in tehran. Environmental monitoring and assessment 190 (5), 300.

Ghoneim, O. A., Manjunatha, B., et al., 2017. Forecasting of ozone concentration in smart city using deep learning. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, pp. 1320–1326.

Guarnieri, M., Balmes, J. R., 2014. Outdoor air pollution and asthma. The Lancet 383 (9928), 1581 – 1592.
URL http://www.sciencedirect.com/science/article/pii/S0140673614606176

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1735–1780.

Hoerl, A. E., Kennard, R. W., 2000. Ridge regression: biased estimation for nonorthogonal problems. Technometrics 42 (1), 80–86.

Hu, K., Rahman, A., Bhrugubanda, H., Sivaraman, V., 2017. Hazeest: Machine learning based metropolitan air pollution estimation from fixed and mobile sensors. IEEE Sensors Journal 17 (11), 3517–3525.

Kampa, M., Castanas, E., 2008. Human health effects of air pollution. Environmental Pollution 151 (2), 362 – 367, proceedings of the 4th International Workshop on Biomonitoring of Atmospheric Pollution (With Emphasis on Trace Elements).
URL http://www.sciencedirect.com/science/article/pii/S0269749107002849

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 3146–3154.
URL http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradi
pdf

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kök, İ., Şimşek, M. U., Özdemir, S., 2017. A deep learning model for air quality prediction in smart cities. In: 2017 IEEE International Conference on Big Data (Big Data). IEEE, pp. 1983–1990.

Li, X., Peng, L., Yao, X., Cui, S., Hu, Y., You, C., Chi, T., 2017. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. Environmental pollution 231, 997–1004.

Liaw, A., Wiener, M., et al., 2002. Classification and regression by randomforest. R news 2 (3), 18–22.

Lin, Y., Mago, N., Gao, Y., Li, Y., Chiang, Y.-Y., Shahabi, C., Ambite, J. L., 2018. Exploiting spatiotemporal patterns for accurate air quality forecasting using deep learning. In: Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 359–368.

Norwegian Meteorological Institute, M., 2019. Free access to weather- and climate data from norwegian meteorological institute from historical data to real time observations. URL eklima.met.no

Qi, Z., Wang, T., Song, G., Hu, W., Li, X., Zhang, Z., 2018. Deep air learning: Interpolation, prediction, and feature analysis of fine-grained air quality. IEEE Transactions on Knowledge and Data Engineering 30 (12), 2285–2297.

Rashmi, K. V., Gilad-Bachrach, R., 2015. Dart: Dropouts meet multiple additive regression trees. In: AISTATS. pp. 489–497.

Russell, S., Norvig, P., 2009. Artificial Intelligence: A Modern Approach, 3rd Edition. Prentice Hall Press, Upper Saddle River, NJ, USA.

Russell, S. J., Norvig, P., 2016. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,.

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

Sinnott, R. O., Guan, Z., 2018. Prediction of air pollution through machine learning approaches on the cloud. In: 2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT). IEEE, pp. 51–60.

Soh, P.-W., Chang, J.-W., Huang, J.-W., 2018. Adaptive deep learning-based air quality prediction model using the most relevant spatial-temporal relations. IEEE Access 6, 38186–38199.

Tamas, W., Notton, G., Paoli, C., Nivet, M.-L., Voyant, C., 2016. Hybridization of air quality forecasting models using machine learning and clustering: An original approach to detect pollutant peaks. Aerosol Air Qual. Res 16, 405–416.

The Norwegian Public Roads Administration, V., 2019a. About the norwegian public roads administration (npra) collection of data on roads and traffic.
URL https://www.vegvesen.no/trafikkdata/start/om-trafikkdata

The Norwegian Public Roads Administration, V., 2019b. Driftstiltak mot svevestv i trondheim kommune.
URL https://www.vegvesen.no/fag/publikasjoner/publikasjoner/Statens+vegvesens+rapporter/_attachment/2162111?_ts=1617aa295b8&download=true&fast_title=Driftstiltak+mot+svevest%C3%B8v+i+Trondheim+kommune%3A+Erfaringsrapport+for+tiltak+f%C3%B8r+og+etter+2013

Tørseth, K., Aas, W., Breivik, K., Fjæraa, A. M., Fiebig, M., Hjellbrekke, A.-G., Lund Myhre, C., Solberg, S., Yttri, K. E., 2012. Introduction to the european monitoring and evaluation programme (emep) and observed atmospheric composition change during 1972–2009. Atmospheric Chemistry and Physics 12 (12), 5447–5481.

Wang, J., Song, G., 2018. A deep spatial-temporal ensemble model for air quality prediction. Neurocomputing 314, 198–206.

WHO, UNAIDS, et al., 2006. Air quality guidelines: global update 2005. World Health Organization.

Yi, X., Zhang, J., Wang, Z., Li, T., Zheng, Y., 2018. Deep distributed fusion network for air quality prediction. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, pp. 965–973.

Zhan, Y., Luo, Y., Deng, X., Zhang, K., Zhang, M., Grieneisen, M. L., Di, B., 2018. Satellite-based estimates of daily no2 exposure in china using hybrid random forest and spatiotemporal kriging model. Environmental science & technology 52 (7), 4180–4189.

Zhang, C., Ma, Y., 2012. Ensemble machine learning: methods and applications. Springer.

Zhang, C., Yan, J., Li, Y., Sun, F., Yan, J., Zhang, D., Rui, X., Bie, R., 2017. Early air pollution forecasting as a service: An ensemble learning approach. In: 2017 IEEE International Conference on Web Services (ICWS). IEEE, pp. 636–643.

Zhang, Y., Bocquet, M., Mallet, V., Seigneur, C., Baklanov, A., 2012a. Real-time air quality forecasting, part i: History, techniques, and current status. Atmospheric Environment 60, 632–655.

Zhang, Y., Bocquet, M., Mallet, V., Seigneur, C., Baklanov, A., 2012b. Real-time air quality forecasting, part ii: State of the science, current research needs, and future prospects. Atmospheric Environment 60, 656–676.

Zheng, H., Li, H., Lu, X., Ruan, T., 2018. A multiple kernel learning approach for air quality prediction. Advances in Meteorology 2018.

Zheng, Y., Yi, X., Li, M., Li, R., Shan, Z., Chang, E., Li, T., 2015. Forecasting fine-grained air quality based on big data. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 2267–2276.

This page intentionally left blank.